

The VR Book

*Human-Centered Design
for Virtual Reality*



Jason Jerald, Ph.D.



ACM Books

Editor in Chief

M. Tamer Özsu, *University of Waterloo*

ACM Books is a new series of high-quality books for the computer science community, published by ACM in collaboration with Morgan & Claypool Publishers. ACM Books publications are widely distributed in both print and digital formats through booksellers and to libraries (and library consortia) and individual ACM members via the ACM Digital Library platform.

The VR Book: Human-Centered Design for Virtual Reality

Jason Jerald, *NextGen Interactions*

2016

Ada's Legacy

Robin Hammerman, *Stevens Institute of Technology*; Andrew L. Russell, *Stevens Institute of Technology*

2016

Edmund Berkeley and the Social Responsibility of Computer Professionals

Bernadette Longo, *New Jersey Institute of Technology*

2015

Candidate Multilinear Maps

Sanjam Garg, *University of California, Berkeley*

2015

Smarter than Their Machines: Oral Histories of Pioneers in Interactive Computing

John Cullinane, *Northeastern University*; Mossavar-Rahmani Center for Business and Government, John F. Kennedy School of Government, Harvard University

2015

A Framework for Scientific Discovery through Video Games

Seth Cooper, *University of Washington*

2014

Trust Extension as a Mechanism for Secure Code Execution on Commodity Computers

Bryan Jeffrey Parno, *Microsoft Research*

2014

Embracing Interference in Wireless Systems

Shyamnath Gollakota, *University of Washington*

2014

The VR Book

*Human-Centered Design
for Virtual Reality*

Jason Jerald

NextGen Interactions

ACM Books #8



Copyright © 2016 by the Association for Computing Machinery and Morgan & Claypool Publishers

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews—without the prior permission of the publisher.

This book is presented solely for educational and entertainment purposes. The author and publisher are not offering it as legal, accounting, or other professional services advice. While best efforts have been used in preparing this book, the author and publisher make no representations or warranties of any kind and assume no liabilities of any kind with respect to the accuracy or completeness of the contents and specifically disclaim any implied warranties of merchantability or fitness of use for a particular purpose. Neither the author nor the publisher shall be held liable or responsible to any person or entity with respect to any loss or incidental or consequential damages caused, or alleged to have been caused, directly or indirectly, by the information or programs contained herein. No warranty may be created or extended by sales representatives or written sales materials. Every company is different and the advice and strategies contained herein may not be suitable for your situation.

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which Morgan & Claypool is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

The VR Book: Human-Centered Design for Virtual Reality

Jason Jerald

books.acm.org

www.morganclaypool.com

ISBN: 978-1-97000-112-9 paperback

ISBN: 978-1-97000-113-6 ebook

ISBN: 978-1-62705-114-3 ePub

ISBN: 978-1-97000-115-0 hardcover

Series ISSN: 2374-6769 print 2374-6777 electronic

A publication in the ACM Books series, #8

Editor in Chief: M. Tamer Özsu, *University of Waterloo*

Area Editor: John C. Hart, University of Illinois

First Edition

10 9 8 7 6 5 4 3 2 1

DOIs

[10.1145/2792790](https://doi.org/10.1145/2792790) Book

[10.1145/2792790.2792791](https://doi.org/10.1145/2792790.2792791) Preface /Intro

[10.1145/2792790.2792792](https://doi.org/10.1145/2792790.2792792) Part I

[10.1145/2792790.2792793](https://doi.org/10.1145/2792790.2792793) Chap. 1

[10.1145/2792790.2792794](https://doi.org/10.1145/2792790.2792794) Chap. 2

[10.1145/2792790.2792795](https://doi.org/10.1145/2792790.2792795) Chap. 3

[10.1145/2792790.2792796](https://doi.org/10.1145/2792790.2792796) Chap. 4

[10.1145/2792790.2792797](https://doi.org/10.1145/2792790.2792797) Chap. 5

[10.1145/2792790.2792798](https://doi.org/10.1145/2792790.2792798) Part II

[10.1145/2792790.2792799](https://doi.org/10.1145/2792790.2792799) Chap. 6

[10.1145/2792790.2792800](https://doi.org/10.1145/2792790.2792800) Chap. 7

[10.1145/2792790.2792801](https://doi.org/10.1145/2792790.2792801) Chap. 8

[10.1145/2792790.2792802](https://doi.org/10.1145/2792790.2792802) Chap. 9

[10.1145/2792790.2792803](https://doi.org/10.1145/2792790.2792803) Chap. 10

[10.1145/2792790.2792804](https://doi.org/10.1145/2792790.2792804) Chap. 11

[10.1145/2792790.2792805 Part III](#)

[10.1145/2792790.2792806 Chap. 12](#)

[10.1145/2792790.2792807 Chap. 13](#)

[10.1145/2792790.2792808 Chap. 14](#)

[10.1145/2792790.2792809 Chap. 15](#)

[10.1145/2792790.2792810 Chap. 16](#)

[10.1145/2792790.2792811 Chap. 17](#)

[10.1145/2792790.2792812 Chap. 18](#)

[10.1145/2792790.2792813 Chap. 19](#)

[10.1145/2792790.2792814 Part IV](#)

[10.1145/2792790.2792815 Chap. 20](#)

[10.1145/2792790.2792816 Chap. 21](#)

[10.1145/2792790.2792817 Chap. 22](#)

[10.1145/2792790.2792818 Chap. 23](#)

[10.1145/2792790.2792819 Chap. 24](#)

[10.1145/2792790.2792821 Chap. 25](#)

[10.1145/2792790.2792820 Part V](#)

[10.1145/2792790.2792822 Chap. 26](#)

[10.1145/2792790.2792823 Chap. 27](#)

[10.1145/2792790.2792824 Chap. 28](#)

[10.1145/2792790.2792825 Chap. 29](#)

[10.1145/2792790.2792826 Part VI](#)

[10.1145/2792790.2792827 Chap. 30](#)

[10.1145/2792790.2792828 Chap. 31](#)

[10.1145/2792790.2792829 Chap. 32](#)

[10.1145/2792790.2792830 Chap. 33](#)

[10.1145/2792790.2792831 Chap. 34](#)

[10.1145/2792790.2792832 Part VII](#)

[10.1145/2792790.2792833 Chap. 35](#)

[10.1145/2792790.2792834 Chap. 36](#)

[10.1145/2792790.2792835 Appendix A](#)

[10.1145/2792790.2792836 Appendix B](#)

[10.1145/2792790.2792837 Glossary /Refs](#)

This book is dedicated to the entire community of VR researchers, developers, designers, entrepreneurs, managers, marketers, and users. It is their passion for, and contributions to, VR that makes this all possible. Without this community, working in isolation would make VR an interesting niche research project that could neither be shared nor improved upon by others. If you choose to join this community, your pursuit of VR experiences may very well be the most intense years of your life, but you will find the rewards well worth the effort. Perhaps the greatest rewards will come from the users of your experiences—for if you do VR well then your users will tell you how you have changed their lives—and that is how we change the world.

There are many facets to VR creation, ranging from getting the technology right, sometimes during exhausting overnight sessions, to the fascinating and abundant collaboration with others in the VR community. At times, what we are embarking on can feel overwhelming. When that happens, I look to a quote by George Bernard Shaw posted on my wall and am reminded about the joy of being a part of the VR revolution.

This is the true joy in life, the being used for a purpose recognized by yourself as a mighty one; the being a force of nature . . . I am of the opinion that my life belongs to the whole community and as long as I live it is my privilege to do for it whatever I can. I want to be thoroughly used up when I die, for the harder I work, the more I live. I rejoice in life for its own sake. Life is no “brief candle” to me. It is sort of a splendid torch which I have a hold of for the moment, and I want to make it burn as brightly as possible before handing it over to future generations.

This book is thus dedicated to the VR community and the future generations that will create many virtual worlds as well as change the real world. My purpose in writing this book is to welcome others into this VR community, to help fuel a VR revolution that changes the world and the way we interact with it and each other, in ways that have never before been possible—until now.

Contents

[Preface](#)

[Figure Credits](#)

[Overview](#)

PART I

INTRODUCTION AND BACKGROUND

Chapter 1

What Is Virtual Reality?

[1.1 The Definition of Virtual Reality](#)

[1.2 VR Is Communication](#)

[1.3 What Is VR Good For?](#)

Chapter 2

A History of VR

[2.1 The 1800s](#)

[2.2 The 1900s](#)

[2.3 The 2000s](#)

Chapter 3

An Overview of Various Realities

[3.1 Forms of Reality](#)

[3.2 Reality Systems](#)

Chapter 4

Immersion, Presence, and Reality Trade-Offs

[4.1 Immersion](#)

[4.2 Presence](#)

[4.3 Illusions of Presence](#)

[4.4 Reality Trade-Offs](#)

*** Chapter 5**

The Basics: Design Guidelines

[5.1 Introduction and Background](#)

[5.2 VR Is Communication](#)

[5.3 An Overview of Various Realities](#)

5.4 Immersion, Presence, and Reality Trade-Offs

PART II

PERCEPTION

Chapter 6

Objective and Subjective Reality

6.1 Reality Is Subjective

6.2 Perceptual Illusions

Chapter 7

Perceptual Models and Processes

7.1 Distal and Proximal Stimuli

7.2 Sensation vs. Perception

7.3 Bottom-Up and Top-Down Processing

7.4 Afference and Efference

7.5 Iterative Perceptual Processing

7.6 The Subconscious and Conscious

7.7 Visceral, Behavioral, Reflective, and Emotional Processes

7.8 Mental Models

7.9 Neuro-Linguistic Programming

Chapter 8

Perceptual Modalities

8.1 Sight

8.2 Hearing

8.3 Touch

8.4 Proprioception

8.5 Balance and Physical Motion

8.6 Smell and Taste

8.7 Multimodal Perceptions

Chapter 9

Perception of Space and Time

9.1 Space Perception

[9.2 Time Perception](#)

[9.3 Motion Perception](#)

[**Chapter 10 Perceptual Stability, Attention, and Action**](#)

[10.1 Perceptual Constancies](#)

[10.2 Adaptation](#)

[10.3 Attention](#)

[10.4 Action](#)

[*** Chapter 11 Perception: Design Guidelines**](#)

[11.1 Objective and Subjective Reality](#)

[11.2 Perceptual Models and Processes](#)

[11.3 Perceptual Modalities](#)

[11.4 Perception of Space and Time](#)

[11.5 Perceptual Stability, Attention, and Action](#)

[**PART III ADVERSE HEALTH EFFECTS**](#)

[**Chapter 12 Motion Sickness**](#)

[12.1 Scene Motion](#)

[12.2 Motion Sickness and Vection](#)

[12.3 Theories of Motion Sickness](#)

[12.4 A Unified Model of Motion Sickness](#)

[**Chapter 13 Eye Strain, Seizures, and Aftereffects**](#)

[13.1 Accommodation-Vergence Conflict](#)

[13.2 Binocular-Occlusion Conflict](#)

[13.3 Flicker](#)

[13.4 Aftereffects](#)

[**Chapter 14 Hardware Challenges**](#)

[14.1 Physical Fatigue](#)

[14.2 Headset Fit](#)

[14.3 Injury](#)

[14.4 Hygiene](#)

[**Chapter 15**](#)

[**Latency**](#)

[15.1 Negative Effects of Latency](#)

[15.2 Latency Thresholds](#)

[15.3 Delayed Perception as a Function of Dark Adaptation](#)

[15.4 Sources of Delay](#)

[15.5 Timing Analysis](#)

[**Chapter 16**](#)

[**Measuring Sickness**](#)

[16.1 The Kennedy Simulator Sickness Questionnaire](#)

[16.2 Postural Stability](#)

[16.3 Physiological Measures](#)

[**Chapter 17**](#)

[**Summary of Factors That Contribute to Adverse Effects**](#)

[17.1 System Factors](#)

[17.2 Individual User Factors](#)

[17.3 Application Design Factors](#)

[17.4 Presence vs. Motion Sickness](#)

[*** Chapter 18**](#)

[**Examples of Reducing Adverse Effects**](#)

[18.1 Optimize Adaptation](#)

[18.2 Real-World Stabilized Cues](#)

[18.3 Manipulate the World as an Object](#)

[18.4 Leading Indicators](#)

[18.5 Minimize Visual Accelerations and Rotations](#)

- [18.6 Ratcheting](#)
- [18.7 Delay Compensation](#)
- [18.8 Motion Platforms](#)
- [18.9 Reducing Gorilla Arm](#)
- [18.10 Warning Grids and Fade-Outs](#)
- [18.11 Medication](#)

* **[Chapter 19 Adverse Health Effects: Design Guidelines](#)**

- [19.1 Hardware](#)
- [19.2 System Calibration](#)
- [19.3 Latency Reduction](#)
- [19.4 General Design](#)
- [19.5 Motion Design](#)
- [19.6 Interaction Design](#)
- [19.7 Usage](#)
- [19.8 Measuring Sickness](#)

PART IV CONTENT CREATION

[Chapter 20 High-Level Concepts of Content Creation](#)

- [20.1 Experiencing the Story](#)
- [20.2 The Core Experience](#)
- [20.3 Conceptual Integrity](#)
- [20.4 Gestalt Perceptual Organization](#)

[Chapter 21 Environmental Design](#)

- [21.1 The Scene](#)
- [21.2 Color and Lighting](#)
- [21.3 Audio](#)

- [21.4 Sampling and Aliasing](#)
- [21.5 Environmental Wayfinding Aids](#)
- [21.6 Real-World Content](#)

Chapter 22

Affecting Behavior

- [22.1 Personal Wayfinding Aids](#)
- [22.2 Center of Action](#)
- [22.3 Field of View](#)
- [22.4 Casual vs. High-End VR](#)
- [22.5 Characters, Avatars, and Social Networking](#)

* **Chapter 23**

Transitioning to VR Content Creation

- [23.1 Paradigm Shifts from Traditional Development to VR Development](#)
- [23.2 Reusing Existing Content](#)

* **Chapter 24**

Content Creation: Design Guidelines

- [24.1 High-Level Concepts of Content Creation](#)
- [24.2 Environmental Design](#)
- [24.3 Affecting Behavior](#)
- [24.4 Transitioning to VR Content Creation](#)

PART V

INTERACTION

Chapter 25

Human-Centered Interaction

- [25.1 Intuitiveness](#)
- [25.2 Norman's Principles of Interaction Design](#)
- [25.3 Direct vs. Indirect Interaction](#)
- [25.4 The Cycle of Interaction](#)
- [25.5 The Human Hands](#)

* **Chapter 26**

VR Interaction Concepts

- [26.1 Interaction Fidelity](#)

- [26.2 Proprioceptive and Egocentric Interaction](#)
- [26.3 Reference Frames](#)
- [26.4 Speech and Gestures](#)
- [26.5 Modes and Flow](#)
- [26.6 Multimodal Interaction](#)
- [26.7 Beware of Sickness and Fatigue](#)
- [26.8 Visual-Physical Conflict and Sensory Substitution](#)

* [**Chapter 27**](#)

Input Devices

- [27.1 Input Device Characteristics](#)
- [27.2 Classes of Hand Input Devices](#)
- [27.3 Classes of Non-hand Input Devices](#)

* [**Chapter 28**](#)

Interaction Patterns and Techniques

- [28.1 Selection Patterns](#)
- [28.2 Manipulation Patterns](#)
- [28.3 Viewpoint Control Patterns](#)
- [28.4 Indirect Control Patterns](#)
- [28.5 Compound Patterns](#)

* [**Chapter 29**](#)

Interaction: Design Guidelines

- [29.1 Human-Centered Interaction](#)
- [29.2 VR Interaction Concepts](#)
- [29.3 Input Devices](#)
- [29.4 Interaction Patterns and Techniques](#)

PART VI

ITERATIVE DESIGN

[**Chapter 30**](#)

Philosophy of Iterative Design

- [30.1 VR Is Both an Art and a Science](#)

- [30.2 Human-Centered Design](#)
- [30.3 Continuous Discovery through Iteration](#)
- [30.4 There Is No One Way—Processes Are Project Dependent](#)
- [30.5 Teams](#)

* [**Chapter 31**](#) [**The Define Stage**](#)

- [31.1 The Vision](#)
- [31.2 Questions](#)
- [31.3 Assessment and Feasibility](#)
- [31.4 High-Level Design Considerations](#)
- [31.5 Objectives](#)
- [31.6 Key Players](#)
- [31.7 Time and Costs](#)
- [31.8 Risks](#)
- [31.9 Assumptions](#)
- [31.10 Project Constraints](#)
- [31.11 Personas](#)
- [31.12 User Stories](#)
- [31.13 Storyboards](#)
- [31.14 Scope](#)
- [31.15 Requirements](#)

* [**Chapter 32**](#) [**The Make Stage**](#)

- [32.1 Task Analysis](#)
- [32.2 Design Specification](#)
- [32.3 System Considerations](#)
- [32.4 Simulation](#)

[32.5 Networked Environments](#)

[32.6 Prototypes](#)

[32.7 Final Production](#)

[32.8 Delivery](#)

* [**Chapter 33**](#)

[**The Learn Stage**](#)

[33.1 Communication and Attitude](#)

[33.2 Research Concepts](#)

[33.3 Constructivist Approaches](#)

[33.4 The Scientific Method](#)

[33.5 Data Analysis](#)

* [**Chapter 34**](#)

[**Iterative Design: Design Guidelines**](#)

[34.1 Philosophy of Iterative Design](#)

[34.2 The Define Stage](#)

[34.3 The Make Stage](#)

[34.4 The Learn Stage](#)

[**PART VII**](#)

[**THE FUTURE STARTS NOW**](#)

[**Chapter 35**](#)

[**The Present and Future State of VR**](#)

[35.1 Selling VR to the Masses](#)

[35.2 Culture of the VR Community](#)

[35.3 Communication](#)

[35.4 Standards and Open Source](#)

[35.5 Hardware](#)

[35.6 The Convergence of AR and VR](#)

* [**Chapter 36**](#)

[**Getting Started**](#)

[**Appendix A**](#)

[**Example Questionnaire**](#)

[Appendix B](#)

[Example Interview Guidelines](#)

[Glossary](#)

[References](#)

[Index](#)

[Author's Biography](#)

*Practitioner chapters are marked with a star next to the chapter number. See page [5](#) for an explanation.

Preface

I've known for some time that I wanted to write a book on VR. However, I wanted to bring a unique perspective as opposed to simply writing a book for the sake of doing so. Then insight hit me during Oculus Connect in the fall of 2014. After experiencing the Oculus Crescent Bay demo, I realized the hardware is becoming really good. Not by accident, but as a result of some of the world's leading engineers diligently working on the technical challenges with great success. What the community now desperately needs is for content developers to understand human perception as it applies to VR, to design experiences that are comfortable (i.e., do not make you sick), and to create intuitive interactions within their immersive creations. That insight led me to the realization that I need to stop focusing primarily on technical implementation and start focusing on higher-level challenges in VR and its design. Focusing on these challenges offers the most value to the VR community of the present day, and is why a new VR book is necessary.

I had originally planned to self-publish instead of spending valuable time on pitching the idea of a new VR book to publishers, which I know can be a very long, arduous, and disappointing path. Then one of the most serendipitous things occurred. A few days after having the idea for a new VR book and committing to make it happen, my former undergraduate adviser, Dr. John Hart, now professor of computer science at the University of Illinois at Urbana–Champaign and, more germanely, the editor of the Computer Graphics Series of ACM Books, contacted me out of the blue. He informed me that Michael Morgan of Morgan & Claypool Publishers wanted to publish a book on VR content creation and John thought I was the person to make it happen. It didn't take much time to enthusiastically accept their proposal, as their vision for the book was the same as mine.

I've condensed approximately 20 years and 30,000 hours of personal study, application, notes, and VR sickness into this book. Those two decades of my VR career have somehow been summarized with six months of intense writing and editing from January to July of 2015. I knew, as others working hard on VR know, the time is now, and after finishing up a couple of contracts I was finally able to put other aspects of my life on hold (clients have been very understanding!), sometimes writing more than 75 hours a week. I hope this rush to get these timely concepts into one book does not sacrifice the overall quality, and I am always seeking feedback. Please contact me at book@nextgeninteractions.com to let me know of any errors, lack of clarity, or important points missed so that an improved second edition can emerge at some point.

It all started in 1980

I owe much of my pursuit of developing VR systems and applications to my parents. It started at the age of six with an Atari game system that my family couldn't afford but I wanted so badly. For Christmas 1980, I somehow received it. Then in 1986 my mother took away the games in hopes of curing my addiction, which naturally forced me to make my own games (with advanced moving 2D sprites!) on the family's Commodore 64. Here, I taught myself programming and essential software design concepts, such as simple computer graphics and collision detection, which turned out to be quite important for VR development. At that point, she thankfully gave up on trying to cure me. I also owe my father, who connected me with my first internship in the summer of 1992 after my junior year of high school at a design firm where he worked. My job was to deliver plots from the printers to the designers. This did not completely fill my time, and I somehow managed to gain access to AutoCad and 3D Studio R2 (long before today's 3ds Max!). I was soon extruding 2D architectural plans into 3D worlds and animating horrible-looking flat-shaded polygons in the evenings and weekends when nobody was the wiser.

SIGGRAPH 1995 and 1996

After regrettably missing SIGGRAPH 1994, I somehow acquired the conference course notes on creating virtual worlds and other related concepts. Soon afterwards, I discovered the SIGGRAPH Student Volunteer Program and realized that was my ticket to the conference. After realizing what I missed, I was not going to leave the opportunity to chance. I sought out a referral from Dr. John Hart, a professor very heavily involved with SIGGRAPH. He must have given me a solid referral based on my initiative and passion for computer graphics, as I had not yet taken a class from him. John would later become my undergraduate adviser and eventually editor of this book. Both his advice through the years and SIGGRAPH had an enormous influence on my career. In fact, I ended up leading the SIGGRAPH Student Volunteer Program more than a decade later, out of appreciation and respect for its ability to inspire careers in computer graphics.

I was quite fortunate to be accepted to the 1995 SIGGRAPH Student Volunteer Program in Los Angeles, which is where I first experienced VR and have been fully hooked ever since. Having never attended a conference up to that point that I was this passionate about, I was completely blown away by its people and its magnitude. I was no longer alone in this world; I had finally found "my people." Soon afterward, I found my first VR mentor, Richard May, now director of the National Visualization and Analytics Center. I enthusiastically jumped in on helping him build one of the world's first immersive VR medical applications at Battelle Pacific Northwest National Laboratories. Richard and I returned to SIGGRAPH 1996 in New Orleans, where I more intentionally sought out VR. VR was now bigger than ever and I remember two things from the conference that defined my future. One was a course on VR interactions by the legendary Dr. Frederick P. Brooks, Jr. and one of his students, Mark Miné. The other event was a VR demo that I still remember vividly to this day—and still one of the most compelling demos I've seen yet. It was a virtual Legoland where the user built an entire world around himself by snapping Legos and Lego neighborhoods together in an intuitive, almost effortless manner.

Post-1996

SIGGRAPH 1996 was a turning point for me in knowing what I wanted to do with my life. Since that time, I have been fortunate to know and work with many individuals that have inspired me. This book is largely a result of their work and their mentorship.

Since 1996, Dr. Brooks unknowingly inspired me to move on from a full-time dream VR job in the late 1990s at HRL Laboratories to pursue VR at the next level at UNC-Chapel Hill. Once there, I managed to persuade Dr. Brooks to become my PhD adviser in studying VR latency. In addition to being a major influence on my career, two of his books—*The Mythical Man Month* and more recently *The Design of Design*—are

heavily referenced throughout Part [VI](#), Iterative Design. He also had significant input for Part [II](#), Perception, especially Chapter [16](#), Latency, as some of that writing originally came from my dissertation that was largely a result of significant suggestions from him. His serving as adviser for Mark Miné’s seminal work on VR interaction also indirectly affected Part [V](#), Interaction. It is still quite difficult for me to fathom that before Dr. Brooks came along, bytes—the molecules of all digital technology—were not always 8 bits in size like they are defined to be today. That 8-bit design decision he made in the 1960s, which most all of us computer scientists now just assume to be an inherent truth, is just one of his many contributions to computers in general, along with his more specific contributions to VR research. It boggles my mind even more how a small-town kid like myself, growing up in a tiny town of 1,200 people on the opposite side of the country, somehow came to study under an ACM Turing Award recipient (the equivalent of the Nobel Prize in computer science). For that, I will forever be grateful to Dr. Brooks and the UNC-Chapel Hill Department of Computer Science for admitting me.

In 2009, I interviewed with Paul Mlyniec, president of Digital ArtForms. At some point during the interview, I came to the realization that this was the man that led the VR Lego work from SIGGRAPH that had inspired and driven me for over a decade. The interface in that Lego demo is one of the first implementations of what I refer to as the 3D Multi-Touch Pattern in Section [28.3.3](#), a pattern two decades ahead of its time. I’ve now worked closely with Paul and Digital ArtForms for six years on various projects, including ones that have improved upon 3D Multi-Touch (Sixense’s MakeVR also uses this same viewpoint control implementation). We are currently working together (along with Sixense and Wake Forest School of Medicine) on immersive games for neuroscience education funded by the National Institutes of Health. In addition to these games, several other examples of Digital ArtForms’ work (along with work from its sister company Sixense that Paul was essential in helping to form) are featured throughout the book.

VR Today

After a long VR drought after the 1990s, VR is bigger than ever at SIGGRAPH. An entire venue, the VR Village, is dedicated specifically to VR, and I have the privilege of leading the Immersive Realities Contest. After so many years of heavy involvement with my SIGGRAPH family, it feels quite fitting that this book is launched at the SIGGRAPH bookstore on the 20th anniversary of my discovery of SIGGRAPH and VR.

I joke that when I did my PhD Dissertation on latency perception for head-mounted displays, perhaps ten people in the world cared about VR latency—and five of those people were on my committee (in three different time zones!). Then in 2011, people started wanting to know more. I remember having lunch with Amir Rubin, CEO of Sixense. He was inquiring about consumer HMDs for games. I thought the idea was crazy; we could barely get VR to work well in a lab and he wanted to put it in people’s living rooms! Other inquiries led to work with companies such as Valve and Oculus. All three of these companies are doing spectacular jobs of making high-quality VR accessible, and now suddenly everyone wants to experience VR. Largely due to these companies’ efforts, VR has turned a corner, transitioning from a specialized laboratory instrument available only to the technically elite, to a mainstream mode of content consumption available to any consumer. Now, most everyone that is involved in VR technology understands at least the basics of latency and its challenges/dangers, most notably motion sickness—the greatest risk for VR. Even better, ultra-low-latency hardware technologies (e.g., low-persistence OLED displays) that I once unsuccessfully searched the world for (I ended up having to build/simulate my own; the best I did was 7.4 ms of end-to-end latency with tracking and rendering performed at ~1,500 frames per second) are being developed in mass quantities by giants such as Samsung, Sony, Valve, and Oculus. Times have certainly changed!

The last 20 years of pursuing VR have truly been a dream. During that time, I imagined and even seriously considered starting companies devoted to VR, but it was never feasible until recently. Today it is more of a real fantasy than a simple dream, now that VR technology is delivering upon its promise of the 1990s. Describing the feeling is like trying to describe a VR experience. Words cannot do justice as to what it is like to be a part of this VR community and contributing to the VR revolution. Through my VR consulting and contracting firm, NextGen Interactions, I have the privilege of working with some of the best companies in the world that are able to do things that could only previously be imagined. Virtual reality is unlike any technology devised to date and has the potential not only to change the fictional synthetic worlds we make up but to change people’s real lives. I very much look forward to seeing what this community discovers and creates over the next 20 years!

Acknowledgments

It would be untrue if I claimed that this book was completely based on my own work. The book was certainly not written in isolation and is a result of contributions from a vast number of mentors, colleagues, and friends in the VR community who have been working just as hard as I have over the years to make VR something real. I couldn’t possibly say everything there is about VR, and I encourage readers to further investigate the more than 300 references discussed throughout the book, along with other writings. I’ve learned so much from those who came before me as well as those newer to VR. Unfortunately, it is not possible to list everyone who has influenced this book, but the following gives thanks to those who have given the most support.

First of all, this book would never have been anything beyond a simple self-published work without the team at Morgan & Claypool Publishers and ACM Books. On the Morgan & Claypool Publishers side, I’d like to thank Executive Editor Diane Cerra, Editorial Assistant Samantha Draper, Copyeditor Sara Kreisman, Production Manager Paul Anagnostopoulos, Artist Laurel Muller, Proofreader Jennifer McClain, and President Michael Morgan. On the ACM Books side, I’d like to thank Computer Graphics Series Editor John Hart and Editor-in-Chief M. Tamer Özsu.

This book you are reading is very different and much better than initial drafts due to suggestions from great reviewers. Reviewers of the book include Paul Mlyniec (president of Digital ArtForms), Arun Yoganandan (formerly of Digital ArtForms and Disney, now a research engineer at Samsung), Russell Taylor (former professor of computer science at UNC–Chapel Hill, now cofounder of Rheomics and independent consultant at Relia Solve), Beau Cronin (an expert VR neuroscientist at Salesforce, who is writing his own VR book), Mike McArdle (cofounder of the Virtual Reality Learning Experience), Ryan McMahan (professor at University of Texas at Dallas), Eugene Nalivaiko (professor at New Castle University), Kyle Yamamoto (cofounder and game developer at MochiBits), Ann McNamara (professor at Texas A&M), Mark Fagiano

(professor at Emory University), Zach Wendt (computer games guru and independent software developer), Francisco Ortega (postdoc fellow at Florida International University and who is writing his own VR book), Matt Cook (emerging technologies librarian at the University of Oklahoma), Sharif Razzaque (CTO of Inneroptic), Neeta Nahta (cofounder of NextGen Interactions), Kevin Rio (UX researcher at Microsoft), Chris Puszak (project manager at Burnout Game Ventures and SymbioVR), Daniel Ackley (psychologist), my mother, Susan Jerald (behavioral psychologist), and my father, Rick Jerald (CEO of Envirocept).

Additional specific input came from Mark Bolas (professor at the University of Southern California), Neil Schneider (founder of Meant to be Seen and founder/executive director of the Immersive Technology Alliance), Eric Greenbaum (attorney and founder of Jema VR), John Baker (COO of Chosen Realities), Denny Unger (founder and president of Cloudhead Games), Andrew Robinson and Sigurdur Gunnarsson (developers at CCP Games), Max Rheiner (founder and CTO of Somniacs), Chadwick Wingrave (founder of Conquest Creations), Barry Downes (CEO of TSSG), Denise Quesnel (research associate at Emily Carr University of Art and Design, cofounder of the Canadian & Advanced Imaging Society, and cochair of the SIGGRAPH VR Village), Nick England (founder and CEO of 3rdTech), Jesse Joudrey (CEO of VRChat and CTO of Jespionage Entertainment), David Collodi (cofounder and CTO of CI Dynamics), David Beaver (cofounder of the Overview Institute), and Bill Howe (CEO of the Growth Engine Group).

I've been very fortunate to be employed by some amazing organizations and even more amazing bosses that are world-leading experts in VR. These individuals gave me the opportunity to make a career out of what I love doing, and working with them led to much of what is contained within these pages: Richard May of Battelle Pacific Northwest National Laboratories (now director of the National Visualization and Analytics Center), Michael Papka of Argonne National Laboratories, Mike Daily of HRL Laboratories, Greg Schmidt of the Naval Research Laboratory, Steve Ellis and Dov Adelstein of NASA Ames, Paul Mlyniec of Digital ArtForms, and Michael Abrash at Valve (now at Oculus).

In addition to those listed above, the following individuals have been invaluable as VR advisers/mentors throughout the years: Howard Neely, research project manager at HRL Laboratories and now CEO at Three Birds Systems; Ron Azuma, senior research staff member at HRL Laboratories and now at Intel; Fred Brooks, Henry Fuchs, Mary Whitton, and Anselmo Lastra—all professors at UNC—Chapel Hill; Jeff Bellinghausen, formerly lead engineer at Digital ArtForms and CTO at Sixense, now at Valve; and Amir Rubin, CEO of Sixense.

I also very much appreciate working with the teams at Sixense, Valve, and Oculus. I've been incredibly impressed with the people and their commitment to creating the highest-quality software and hardware, and not just resurrecting VR but raising awareness to levels beyond anything that previously existed.

Others I've had the pleasure of working with who have influenced this book include Rupert Meghnat and his team at Burnout Game Ventures, especially Chris Puszak of the SymbioVR team; Jan Goetgeluk of Virtuix; Simon Solotko, the Kickstarter guru who marketed some of the biggest VR Kickstarters such as the Virtuix Omni, the Cyberith Virtualizer, and the Sixense STEM; the UNC Effective Virtual Environments team; Karl Krantz of Silicon Valley Virtual Reality, who was the first to pioneer local meetups where his events are duplicated throughout the world; and Henry Velez, founder of the RTP Virtual Reality Enthusiasts, which I now have the privilege of leading.

I've also enjoyed working with literally thousands of volunteers through ACM SIGGRAPH, IEEE VR, and IEEE 3DUI, ranging from student volunteers to the legends of computer graphics and VR. In particular, I'd like to thank the SIGGRAPH conference committees of the last 20 years. Of course, I've also enjoyed working with those outside of conferences, ranging from those in academics to those doing startups. Then there are those who have given me invaluable feedback on VR projects, ranging from middle schoolers to VR experts.

I'd especially like to thank Neeta Nahta, my partner in business and life, who comes from a completely different world of sales, financial analysis, and corporate boardrooms. Her input enables me to see VR from a different perspective than most anyone else in the VR community. Seeing VR as a sustainable and profitable business that adds real value to the world will ensure that this time VR is here for good.

I would also like to thank the Link Foundation for supporting me through the Link Foundation Advanced Simulation and Training Fellowship, which helped to fund graduate work in studying and reducing latency. It is certainly an honor to be funded by the foundation started by Edwin A. Link, who built the world's first mechanical flight simulator in 1928 (Figure 3.5), which some consider to be one of the first VR systems, as well as to be among a list of world-leading VR experts who also received the Fellowship. I highly encourage any graduate student readers who have already selected a dissertation topic to apply for the Link Foundation Fellowship (\$28,500 to be awarded for 2016). Details are available at <http://www.linksim.org>.

Figure Credits

Figure 1.1 Adapted from Dale, E. (1969). *Audio-Visual Methods in Teaching* (3rd ed.). The Dryden Press. Based on Edward L. Counts Jr.

Figure 2.1 Based on: Ellis, S. R. (2014). Where are all the Head Mounted Displays? Retrieved April 14, 2015, from http://humansystems.arc.nasa.gov/groups/acd/projects/hmd_dev.php.

Figure 2.3 Courtesy of The National Media Museum / Science & Society Picture Library, United Kingdom

Figure 2.4 From Pratt, A. B. (1916). Weapon. US.

Figure 2.5 Courtesy of Edwin A. Link and Marion Clayton Link Collections, Binghamton University Libraries' Special Collections and University Archives, Binghamton University.

Figure 2.6 From Weinbaum, S. G. (1935, June). Pygmalion's Spectacles. *Wonder Stories* .

Figure 2.7 From Heilig, M. (1960). Stereoscopic-television apparatus for individual use.

Figure 2.8 Courtesy of © Morton Heilig Legacy.

Figure 2.9 From Comeau, C.P. & Brian, J.S. "Headsight television system provides remote surveillance," *Electronics* , November 10, 1961, pp. 86-90.

Figure 2.10 From Rochester, N., & Seibel, R. (1962). Communication Device. US.

Figure 2.11 (left) Courtesy of Tom Furness.

Figure 2.11 (right) From Sutherland, I. E. (1968). A Head-Mounted Three Dimensional Display. In *Proceedings of the 1968 Fall Joint Computer Conference AFIPS* (Vol. 33, part 1, pp. 757–764). Copyright © ACM 1968. Used with permission. DOI: [10.1145/1476589.1476686](https://doi.org/10.1145/1476589.1476686)

Figure 2.12 From Brooks, F. P., Ouh-Young, M., Batter, J. J., & Jerome Kilpatrick, P. (1990). Project GROPE Haptic displays for scientific visualization. *ACM SIGGRAPH Computer Graphics*, 24 (4), 177–185. Copyright © ACM 1990. Used with permission. DOI: [10.1145/97880.97899](https://doi.org/10.1145/97880.97899)

Figure 2.13 Courtesy of NASA/S.S. Fisher, W. Sisler, 1988

Figure 3.1 Adapted from Milgram, P., & Kishino, F. (1994). Taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, E77-D (12), 1321–1329. DOI: [10.1.1.102.4646](https://doi.org/10.1.1.102.4646)

Figure 3.2 Adapted from Jerald, J. (2009). *Scene-Motion- and Latency-Perception Thresholds for Head-Mounted Displays* . Department of Computer Science, University of North Carolina at Chapel Hill.

Figure 3.3 (top left) Courtesy of Oculus VR.

Figure 3.3 (top right) Courtesy of CastAR.

Figure 3.3 (lower left) Courtesy of Marines Magazine.

Figure 3.3 (lower right) From Jerald, J., Fuller, A. M., Lastra, A., Whitton, M., Kohli, L., & Brooks, F. (2007). Latency compensation by horizontal scanline selection for head-mounted displays. *Proceedings of SPIE*, 6490 , 64901Q–64901Q–11. Copyright © 2007 Society of Photo Optical Instrumentation Engineers. Used with permission.

Figure 3.4 (left) From Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V., & Hart, J. C. (1992). The CAVE: audio visual experience automatic virtual environment. *Communications of the ACM* . Copyright © ACM 1992. Used with permission. DOI: [10.1145/129888.129892](https://doi.org/10.1145/129888.129892)

Figure 3.4 (right) From Daily, M., Sarfaty, R., Jerald, J., McInnes, D., & Tinker, P. (1999). The CABANA: A Re-configurable Spatially Immersive Display. In *Projection Technology Workshop* (pp. 123–132). Used with permission.

Figure 3.5 From Jerald, J., Daily, M. J., Neely, H. E., & Tinker, P. (2001). Interacting with 2D Applications in Immersive Environments. In *EUROIMAGE International Conference on Augmented Virtual Environments and 3d Imaging* (pp. 267–270). Used with permission.

Figure 3.6 From Krum, D. M., Suma, E. A., & Bolas, M. (2012). Augmented reality using personal projection and retroreflection. *Personal and Ubiquitous Computing*, 16 (1), 17–26. Copyright © 2012 Springer. Used with Kind permission of Springer Science + Business Media. DOI: [10.1007/s00779-011-0374-4](https://doi.org/10.1007/s00779-011-0374-4)

Figure 3.7 Courtesy of USC Institute for Creative Technologies.

Figure 3.8 (left) Courtesy of Geomedia.

Figure 3.8 (right) Courtesy of NextGen Interactions.

Figure 3.9 Courtesy of Tactical Haptics.

Figure 3.10 Courtesy of Dexta Robotics.

Figure 3.11 Courtesy of INITION.

Figure 3.12 Courtesy of Haptic Workstation with HMD at VRLab in EPFL, Lausanne, 2005.

Figure 3.13 Courtesy of Shifz, Syntharturalist Art Association.

Figure 3.14 Courtesy of Swissnex San Francisco and Myleen Hollero.

Figure 3.15 Courtesy of Virtuix.

Figure 4.1 Courtesy of Gail [Drinnan/Shutterstock.com](#).

Figure 4.2 Courtesy of NextGen Interactions.

Figure 4.3 Based on: Ho, C. C., & MacDorman, K. F. (2010). Revisiting the uncanny valley theory: Developing and validating an alternative to the Godspeed indices. *Computers in Human Behavior*, 26 (6), 1508–1518. DOI: [10.1016/j.chb.2010.05.015](#)

Figure II.1 Courtesy of Sleeping [cat/Shutterstock.com](#).

Figure 6.2 From: Harmon, Leon D. (1973). The Recognition of Faces. *Scientific American*, 229 (5). © 2015 Scientific American, A Division of Nature America, Inc. Used with permission.

Figure 6.4 Courtesy of Fibonacci.

Figure 6.5 b From: Pietro Guardini and Luciano Gamberini, (2007) The Illusory Contoured Tilting Pyramid. Best illusion of the year contest 2007. Sarasota, Florida. Available on line at <http://illusionoftheyear.com/cat/top-10-finalists/2007/>. Used with permission.

Figure 6.5 c From: Lehar, S. (2007). The Constructive Aspect of Visual Perception: A Gestalt Field Theory Principle of Visual Reification Suggests a Phase Conjugate Mirror Principle of Perceptual Computation. Used with permission.

Figure 6.6 a, b Based on: Lehar, S. (2007). The Constructive Aspect of Visual Perception: A Gestalt Field Theory Principle of Visual Reification Suggests a Phase Conjugate Mirror Principle of Perceptual Computation.

Figure 6.9 Courtesy of PETER ENDIG/AFP/Getty Images.

Figure 6.12 Image © 2015 The Association for Research in Vision and Ophthalmology.

Figure 6.13 Courtesy of [MarcoCapra/Shutterstock.com](#).

Figure 7.1 Based on: Razzaque, S. (2005). *Redirected Walking*. Department of Computer Science, University of North Carolina at Chapel Hill. Used with permission.

Adapted from: Gregory, R. L. (1973). *Eye and Brain: The Psychology of Seeing* (2nd ed.). London: Weidenfeld and Nicolson. Copyright © 1973 The Orion Publishing Group. Used with permission.

Figure 7.2 Adapted from: Goldstein, E. B. (2007). *Sensation and Perception* (7th ed.). Belmont, CA: Wadsworth Publishing.

Figure 7.3 Adapted from: James, T., & Woodsmall, W. (1988). *Time Line Therapy and the Basis of Personality*. Meta Pubns.

Figure 8.1 Based on: Coren, S., Ward, L. M., and Enns, J. T. (1999). *Sensation and Perception* (5th ed.). Harcourt Brace College Publishers.

Figure 8.2 Based on: Badcock, D. R., Palmisano, S., & May, J. G. (2014). Vision and Virtual Environments. In K. S. Hale & K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed.). CRC Press.

Figure 8.3 Based on: Coren, S., Ward, L. M., & Enns, J. T. (1999). *Sensation and Perception* (5th ed.). Harcourt Brace College Publishers.

Figure 8.4 Adapted from: Clulow, F. W. (1972). *Color: Its principle and their applications*. New York: Morgan & Morgan.

Figure 8.5 Based on: Coren, S., Ward, L. M., & Enns, J. T. (1999). *Sensation and Perception* (5th ed.). Harcourt Brace College Publishers.

Figure 8.6 Adapted from: Coren, S., Ward, L. M., & Enns, J. T. (1999). *Sensation and Perception* (5th ed.). Harcourt Brace College Publishers.

Figure 8.7 Adapted from: Goldstein, E. B. (2014). *Sensation and Perception* (9th ed.). Cengage Learning.

Figure 8.8 Based on: Burton, T.M.W. (2012) *Robotic Rehabilitation for the Restoration of Functional Grasping Following Stroke*. Dissertation, University of Bristol, England.

Figure 8.9 Based on: Jerald, J. (2009). *Scene-Motion- and Latency-Perception Thresholds for Head-Mounted Displays*. Department of Computer Science, University of North Carolina at Chapel Hill. Used with permission. Adapted from: Martini (1998). *Fundamentals of Anatomy and Physiology*. Upper Saddle River, Prentice Hall.

Figure 9.1 Courtesy of Paolo [Giant/Shutterstock.com](#)

Figure 9.4 Based on: Kersten, D., Mamassian, P. and Knill, D. C. (1997). Moving cast shadows induce apparent motion in depth. *Perception* 26, 171–192.

Figure 9.5 Courtesy of Galkin [Grigory/Shutterstock.com](#)

Figure 9.7 Courtesy of NextGen Interactions.

Figure 9.8 Adapted from: Mirzaie, H. (2009, March 16). *Stereoscopic Vision*.

Figure 9.9 Adapted from: Coren, S., Ward, L. M., & Enns, J. T. (1999). *Sensation and Perception* (5th ed.). Harcourt Brace College Publishers.

Figure 10.1 From: Pfeiffer, T., & Memili, C. (2015). GPU-Accelerated Attention Map Generation for Dynamic 3D Scenes. In *IIEE Virtual Reality*. Used with permission. Courtesy of “BMW 3 Series Coupe” by mikepan (Creative Commons Attribution, Share Alike 3.0 at <http://www.blendswap.com>).

Figure 12.1 Adapted from: Razzaque, S. (2005). *Redirected Walking*. Department of Computer Science, University of North Carolina at Chapel Hill.

Figure 14.1 Courtesy of Jema VR.

Figure 15.1 Based on: Gregory, R. L. (1973). *Eye and Brain: The Psychology of Seeing* (2nd ed.). London: Weidenfeld and Nicolson.

Figure 15.2 Adapted from: Jerald, J. (2009). *Scene-Motion- and Latency-Perception Thresholds for Head-Mounted Displays*. Department of Computer Science, University of North Carolina at Chapel Hill.

Figure 15.3 Adapted from: Jerald, J. (2009). *Scene-Motion-and Latency-Perception Thresholds for Head-Mounted Displays*. Department of Computer Science, University of North Carolina at Chapel Hill.

Figure 15.4 Based on: Jerald, J. (2009). *Scene-Motion-and Latency-Perception Thresholds for Head-Mounted Displays*. Department of Computer Science, University of North Carolina at Chapel Hill. Used with permission.

Figure 18.1 Courtesy of CCP Games.

Figure 18.2 Courtesy of NextGen Interactions.

Figure 18.3 Courtesy of Sixense.

Figure 20.1 From: Heider, F., & Simmel, M. (1944). An experimental study of apparent behavior. *The American Journal of Psychology*, 57, 243–259. DOI: [10.2307/1416950](https://doi.org/10.2307/1416950)

Figure 20.2 From: Lehar, S. (2007). The Constructive Aspect of Visual Perception: A Gestalt Field Theory Principle of Visual Reification Suggests a Phase Conjugate Mirror Principle of Perceptual Computation. Used with permission.

Figure 20.3 Courtesy of Sixense.

Figure 20.4 Based on: Wolfe, J. (2006). *Sensation & perception*. Sunderland, Mass.: Sinauer Associates.

Figure 20.5 From: Yoganandan, A., Jerald, J., & Mlyniec, P. (2014). Bimanual Selection and Interaction with Volumetric Regions of Interest. In *IEEE Virtual Reality Workshop on Immersive Volumetric Interaction*. Used with permission.

Figure 20.7 Courtesy of Sixense.

Figure 20.9 Courtesy of Sixense.

Figure 21.1 Courtesy of Digital ArtForms.

Figure 21.3 Courtesy of NextGen Interactions.

Figure 21.4 Courtesy of Digital ArtForms.

Figure 21.5 Courtesy of Digital ArtForms.

Figure 21.6 Image © Strangers with Patrick Watson / Félix & Paul Studios

Figure 21.7 Courtesy of 3rdTech.

Figure 21.8 Courtesy of Digital ArtForms.

Figure 21.9 Courtesy of UNC CISMM NIH Resource 5-P41-EB002025 from data collected in Alisa S. Wolberg's laboratory under NIH award HL094740.

Figure 22.1 Courtesy of Digital ArtForms.

Figure 22.2 Courtesy of NextGen Interactions.

Figure 22.3 Courtesy of Visionary VR.

Figure 22.4 From: Daily, M., Howard, M., Jerald, J., Lee, C., Martin, K., McInnes, D., & Tinker, P. (2000). Distributed design review in virtual environments. In *Proceedings of the third international conference on Collaborative virtual environments* (pp. 57–63). ACM. Copyright © ACM 2000. Used with permission. DOI: [10.1145/351006.351013](https://doi.org/10.1145/351006.351013)

Figure 22.5 Courtesy of NextGen Interactions.

Figure 22.6 Courtesy of VRChat.

Figure 25.1 Adapted from: Norman, D. A. (2013). *The Design of Everyday Things, Expanded and Revised Edition. Human Factors and Ergonomics in Manufacturing*. New York, NY: Basic Books. DOI: [10.1002/hfm.20127](https://doi.org/10.1002/hfm.20127)

Figure 26.1 Courtesy of Digital ArtForms.

Figure 26.2 Courtesy of NextGen Interactions.

Figure 26.3 Courtesy of NextGen Interactions.

Figure 26.4 Courtesy of NextGen Interactions.

Figure 26.5 Courtesy of Cloudhead Games.

Figure 27.1 From: Pausch, R., Snoddy, J., Taylor, R., Watson, S., & Haseltine, E. (1996). Disney's Aladdin: first steps toward storytelling in virtual reality. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (pp. 193–203). ACM Press. Copyright © ACM 1996. Used with permission. DOI: [10.1145/237170.237257](https://doi.org/10.1145/237170.237257)

Figure 27.2 Courtesy of Stock photo © juniorbeep.

Figure 27.3 Courtesy of Sixense and Oculus.

Figure 27.4 Courtesy of CyberGlove Systems LLC.

Figure 27.5 Courtesy of Leap Motion.

Figure 27.6 Courtesy of Dassault Systèmes, iV Lab.

Figure 28.1 (left) Courtesy of Cloudhead Games.

Figure 28.1 (center) Courtesy of NextGen Interactions.

Figure 28.1 (right) Courtesy of Digital ArtForms.

Figure 28.2 From: Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., & Miné, M. R. (1997). Image Plane Interaction Techniques in 3D Immersive Environments. In *ACM Symposium on Interactive 3D Graphics* (pp. 39–44). ACM Press. Copyright © ACM 1997. Used with permission. DOI: [10.1145/253284.253303](https://doi.org/10.1145/253284.253303)

Figure 28.3 Courtesy of Digital ArtForms.

Figure 28.4 From: Hinckley, K., Pausch, R., Goble, J. C., & Kassell, N. F. (1994). Passive Real-world Interface Props for Neurosurgical Visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94* (Vol. 30, pp. 452–458). Copyright © ACM 1994. Used with permission. DOI: [10.1145/191666.191821](https://doi.org/10.1145/191666.191821)

Figure 28.5 Courtesy of Digital ArtForms and Sixense.

Figure 28.6 From: Mlyniec, P., Jerald, J., Yoganandan, A., Seagull, J., Toledo, F., & Schultheis, U. (2011). iMedic: A two-handed immersive medical environment for distributed interactive consultation. In *Studies in Health Technology and Informatics* (Vol. 163, pp. 372–378). Copyright © 2011 IOS Press. Reprinted with permission.

Figure 28.7 From: Schultheis, U., Jerald, J., Toledo, F., Yoganandan, A., & Mlyniec, P. (2012). Comparison of a two-handed interface to a wand interface and a mouse interface for fundamental 3D tasks. In *IEEE Symposium on 3D User Interfaces 2012, 3DUI 2012 - Proceedings* (pp. 117–124). Copyright © 2012 IEEE. Used with permission. DOI: [10.1109/3DUI.2012.6184195](https://doi.org/10.1109/3DUI.2012.6184195)

Figure 28.8 Courtesy of Sixense and Digital ArtForms.

Figure 28.9 From: Bowman, D. A., & Wingrave, C. A. (2001). Design and evaluation of menu systems for immersive virtual environments. In *IEEE Virtual Reality* (pp. 149–156). Copyright © 2001 IEEE. Used with permission. DOI: [10.1109/VR.2001.913781](https://doi.org/10.1109/VR.2001.913781)

Figure 31.1 Based on: Rasmusson, J. (2010). The Agile Samurai—How Agile Masters Deliver Great Software. *Pragmatic Bookshelf*.

Figure 31.2 Based on: Rasmusson, J. (2010). The Agile Samurai—How Agile Masters Deliver Great Software. *Pragmatic Bookshelf*.

Figure 31.6 Courtesy of Geomedia.

Figure 32.1 Courtesy of Andrew Robinson of CCP Games.

Figure 32.2 Based on: Neely, H. E., Belvin, R. S., Fox, J. R., & Daily, M. J. (2004). Multimodal interaction techniques for situational awareness and command of robotic combat entities. In *IEEE Aerospace Conference Proceedings* (Vol. 5, pp. 3297–3305). DOI: [10.1109/AERO.2004.1368136](https://doi.org/10.1109/AERO.2004.1368136)

Figure 32.4 Based on: Daily, M., Howard, M., Jerald, J., Lee, C., Martin, K., McInnes, D., & Tinker, P. (2000). Distributed design review in virtual environments. In *Proceedings of the third international conference on Collaborative virtual environments* (pp. 57–63). ACM. DOI: [10.1145/351006.351013](https://doi.org/10.1145/351006.351013)

Figure 33.2 Adapted from: Gabbard, J. L. (2014). Usability Engineering of Virtual Environments. In K. S. Hale & K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 721–747). CRC Press.

Figure 33.3 Courtesy of NextGen Interactions.

Figure 35.1 Courtesy of USC Institute for Creative Technologies. Principal Investigators: Albert (Skip) Rizzo and Louis-Philippe Morency.

Figure 35.2 From: Grau, C., Ginhoux, R., Riera, A., Nguyen, T. L., Chauvat, H., Berg, M., . . . Ruffini, G. (2014). Conscious Brain-to-Brain Communication in Humans Using Non-Invasive Technologies. *PloS One*, 9 (8), e105225. DOI: [10.1371/journal.pone.0105225](https://doi.org/10.1371/journal.pone.0105225)

Figure 35.3 Courtesy of Tyndall and TSSG.

Overview

Virtual reality (VR) can provide our minds with direct access to digital media in a way that seemingly has no limits. However, creating compelling VR experiences is an incredibly complex challenge. When VR is done well, the results are brilliant and pleasurable experiences that go beyond what we can do in the real world. When VR is done badly, not only do users get frustrated, but they can get sick. There are many causes of bad VR; some failures come from the limitations of technology, but many come from a lack of understanding perception, interaction, design principles, and real users. This book discusses these issues by emphasizing the human element of VR. The fact is, if we do not get the human element correct, then no amount of technology will make VR anything more than an interesting tool confined to research laboratories. Even when VR principles are fully understood, the first implementation is rarely novel and almost never ideal due to the complex nature of VR and the countless possibilities that can be created. The VR principles discussed in this book will enable readers to intelligently experiment with the rules and iteratively design toward innovative experiences.

Historically, most VR creators have been engineers (the author included) with expertise in technology and logic but limited in their understanding of humans. This is primarily due to the fact that VR has previously been so technically challenging that it was difficult to build VR experiences without an engineering background. Unfortunately, we engineers often believe “I am human, therefore I understand other humans and what works for them.” However, the way humans perceive and interact with the world is incredibly complex and not typically based on logic, mathematics, or a user manual. If we stick solely to the expertise of knowing it all through engineering and logic, then VR will certainly be doomed. We have to accept human perception and behavior the way it is, not the way logic tells us it should be. Engineering will always be essential as it is the core of VR systems that everything else builds upon, but VR itself presents a fascinating interplay of technology and psychology, and we must understand both to do VR well.

Technically minded people tend to dislike the word “experience” as it is less logical and more subjective in nature. But when asked about their favorite tool or game they often convey emotion as they discuss how, for example, the tool responds. They talk about how it makes them feel often without them consciously realizing they are speaking in the language of emotion. Observe how they act when attempting to navigate through a poorly designed voice-response system for technical support via phone. In their frustration, they very well may give up on the product they are calling about and never purchase from that company again. The experience is important for everything, even for engineers, for it determines our quality of life on a moment-by-moment basis. For VR, the experience is even more critical. To create quality VR, we need to continuously ask ourselves and others questions about how we perceive the VR worlds that we create. Is the experience understandable and enjoyable? Or is it sometimes confusing to use while at other times just all-out frustrating and sickness inducing? If the VR experience is intuitive to understand, easily controlled, and comfortable, then just like anything in life, there will be a sense of mastery and satisfaction with “logical” reasons why it works well. Emotion and cognition are tightly coupled, with cognition more often than not justifying decisions that were actually made emotionally. We must create VR experiences with both emotion and logic.

What This Book Is

This book focuses on **human-centered design**, a design philosophy that puts human needs, capabilities, and behavior first, then designs to accommodate those needs, capabilities, and ways of behaving (Norman, 2013). More specifically, this book focuses on the human elements of VR—how users perceive and intuitively interact with various forms of reality, causes of VR sickness, creating content that is pleasing and useful, and how to design and iterate upon effective VR applications.

Good VR design starts with an understanding of both technology and perception. It requires good communication between human and machine, indicating what interactions are possible, what is currently occurring, and what is about to occur. Human-centered design also comes about through observation, for humans are often unaware of their perceptual processes and methods of interacting (at least for VR that works well). Getting the specification of a VR experience is difficult, and rarely does a VR creator do it well for his first few projects. In fact, even VR experts do not perfectly define the project from the start if they are creating a novel experience. A human-centered design principle, like lean methods, is to avoid completely defining the problem at the start and to iterate upon repeated approximations and modifications through rapid tests of ideas with real users.

Developing intuitive VR should not be driven by software/hardware engineering considerations alone (e.g., we need to do much more than figure out how to efficiently display at the highest resolution available based on the most current hardware). A good portion of this book is devoted to how the human mind works in order to help readers create higher-quality VR applications. VR design, however, goes beyond just technology and psychology; VR is intensely multidisciplinary. VR is an incredibly complex challenge and the study, design, and implementation of high-quality VR requires an understanding of various disciplines, including the behavioral and social sciences, neuroscience, information and computer science, physics, communications, art, and even philosophy. When reflecting back upon VR design and implementation, [Wingrave and LaViola \[2010 \]](#) point out: “Practitioners must be carpenters, electricians, engineers, artists, and masters of duct tape and Velcro.” This book takes a broad perspective, applying insights from various disciplines to VR design.

In summary, this book provides basic theory, an overview of various concepts useful for VR, examples that put the theory and concepts into more understandable form, useful guidelines, and a foundation for further exploration and interaction design for virtual worlds that do not yet exist.

What This Book Is Not

There are more questions about VR than there are answers and the intent of this book is not to attempt to provide all of the answers. VR covers a broad range of imaginary spaces broader than the real world; nobody could possibly have all the answers to the real world, so it is unreasonable to expect to find all the answers for virtual worlds. Instead, this book attempts to help readers build and iterate upon creative answers and compelling experiences. Although this book can’t possibly cover all aspects of VR in detail, it does provide an overview of various topics and delves more

deeply into those that are most important. References are provided throughout for those wishing to further study concepts of interest.

In some cases, the concepts presented in this book follow well-understood principles or conclusive research (although even conclusive research rarely holds 100% of the time across all conditions). In other cases, concepts are not the “truth” but have been found to be useful in the way we think about design and interaction. Studying theory can be useful, but VR development should always follow pragmatism over theory.

Although there is a brief chapter on getting started (Chapter 36), and there are tips throughout on high- and mid-level implementation concepts, this book is not a step-by-step tutorial of how to implement an example VR system. In fact, this book intentionally contains no code or equations so that all concepts can be understood by anyone from any discipline (references are provided for more rigorous detail). Although researchers have been experimenting with VR for decades (Section 2.1), VR in no way comes close to reaching its full potential. Not only are there many unknowns, but VR implementation very much depends on the project. For example, a surgical training system is very different from an immersive film.

Who This Book Is For

This book is for the entire team that works on a VR project, not just for those who define themselves as designers. It is intended to act as a foundation for anyone and everyone involved with creating VR experiences. The book also is meant to serve as a bridge between academic research and practical advice to a wide range of individuals who wish to build compelling experiences. This includes designers, managers, programmers, artists, psychologists, engineers, students, educators, and user experience professionals so the entire team can have a common understanding and language of VR. Everyone involved with a VR project should understand at least the basics of perception, VR sickness, interaction, content creation, and iterative design. VR requires specialized experts in various disciplines to each contribute in their own unique way, but we each must also know at least a little about human-centered design in order to effectively communicate with teammates and to integrate the various components together into a seamless, quality experience.

How to Read This Book

Readers may wish to read and use this book differently depending on their background, particular interests, and how they would like to apply it.

Newcomers

Those completely new to VR who want a high-level understanding will most appreciate Part I, Introduction and Background. After reading Part I, the reader may want to skip ahead to Part VII, The Future Starts Now. Once these basics are understood, most of Part IV, Content Creation, should be able to be understood. As the reader learns more about VR, the other parts will become easier to digest.

Teachers

This book will be especially relevant to interdisciplinary VR courses. Teachers will want to choose chapters that are most relevant to the course requirements and student interests. It is highly suggested that any VR course take a project-centered approach. In such a case, the teacher should consider the suggestions below for both students and practitioners. An outline for starting a first project, albeit slightly different for a class project, is outlined in Chapter 36, Getting Started.

Students

Students will gain a core understanding of VR by first gaining a high-level overview of VR through Part I, Introduction and Background. For those wishing to understand theory, Part II, Perception, and Part III, Adverse Health Effects, will be invaluable. For students working on VR projects, they should also follow the advice below for practitioners.

Practitioners

Practitioners who want to immediately get the most important points that apply to their VR creations will want to start with the practitioner chapters that have a leading star (*) in the table of contents (mostly Parts IV – VI). In particular, they may want to start with the design guidelines chapters at the end of each part, where each part contains multiple chapters on one of the primary topics. Most of the guidelines provide back references to the relevant sections for more detailed information.

VR Experts

VR experts will likely use this book more as a reference so they do not spend time reading material they are already familiar with. The references will also serve VR experts for further investigation. Those experts who primarily work with head-mounted displays may find Part I, Introduction and Background, useful to understand how head-mounted displays fit within the larger scheme of other implementations of VR and augmented reality (AR). Those interested in implementing straightforward applications that have already been built may not find Part II, Perception, useful. However, for those who want to innovate with new forms of VR and interaction, they may find this part useful to understand how we perceive the world in order to help invent novel creations.

Overview of the Seven Parts

Part I, Introduction and Background, provides a background of VR including a brief history of VR, different forms of VR and related

technologies, and a broad overview of some of the most important concepts that will be further discussed in later parts.

Part II, Perception, provides a background in perception to educate VR creators on concepts and theories of how we perceive and interact with the world around us. This part serves as an intellectual framework that will enable the reader to not only implement the ideas discussed in later chapters but more thoroughly understand why some techniques do or do not work, to extend those techniques, to intelligently experiment with new concepts that have a better chance of working without causing human factors issues, and to know when it might be appropriate to break the rules.

Part III, Adverse Health Effects, describes one of the most difficult challenges of VR and helps to reduce the greatest risk to VR succeeding at a massive scale: VR sickness. Whereas it may be impossible to remove 100% of VR sickness for the entire population, there are several ways to dramatically reduce it if we understand the theories of why it occurs. Other adverse health effects such as risk of injury, seizures, and aftereffects are also discussed.

Part IV, Content Creation, discusses high-level concepts for designing/building assets and how subtle design choices can influence user behavior. Examples include story creation, the core experience, environmental design, wayfinding aids, social networking, and porting existing content to VR.

Part V, Interaction, focuses on how to design the way users interact within the scenes they find themselves in. For many applications, we want to engage the user by creating an active experience that consists of more than simply looking around; we want to empower users by enabling them to reach out, touch, and manipulate that world in a way that makes them feel they are a part of the world instead of just a passive observer.

Part VI, Iterative Design, provides an overview of several different methods for creating, experimenting, and improving upon VR designs. Whereas each project may not utilize all methods, it is still good to understand them all to be able to apply them when appropriate. For example, you may not wish to conduct a formal and rigorous scientific user study, but you do want to understand the concepts to minimize mistaken conclusions due to confounding factors.

Part VII, The Future Starts Now, summarizes the book, discusses the current and future state of VR, and provides a brief plan to get started.

PART I

INTRODUCTION AND BACKGROUND

What is virtual reality (VR)? What does VR consist of and for what situations is it useful? What is different about VR that gets people so excited? How do developers engage users so that they feel present in a virtual environment? This part of the book answers such questions, and provides a basic background that later chapters build upon. This introduction and background serves as a simple high-level toolbox of options to intelligently choose from, such as different forms of virtual and augmented reality (AR), different hardware options, various methods of presenting information to the senses, and ways to induce presence into the minds of users.

Part 1 consists of five chapters that cover the basics of VR.

Chapter 1, What Is Virtual Reality?, begins by describing what VR is at a high level and what it is suitable/effective for. This includes descriptions of different forms of communication that are at the heart of what VR is—communication between the user and a system created by the VR designer.

Chapter 2, A History of VR, provides a history of VR starting with stereoscopes created in the 1800s. The concept and implementation of VR is not new.

Chapter 3, An Overview of Various Realities, discusses forms of reality ranging from the real world to augmented reality (AR) to VR. Whereas the focus of this book is on fully immersive VR, this chapter provides context of where VR fits into the overall picture of related technologies. The chapter also gives a high-level description of various forms of input and output hardware options that can be used as part of AR and VR systems.

Chapter 4, Immersion, Presence, and Reality Trade-Offs, discusses the often-used terms of immersion and presence. Readers may be surprised to learn that realism is not necessarily the goal of VR and there are trade-offs for attempting to perfectly simulate reality, even if reality could be perfectly simulated.

Chapter 5, The Basics: Design Guidelines, concludes this introductory part of the book and gives a small number of guidelines for those looking to create VR experiences.

What Is Virtual Reality?

1.1 The Definition of Virtual Reality

The term virtual reality (VR) is commonly used by the popular media to describe imaginary worlds that only exist in computers and our minds. However, let us more precisely define the term. [Sherman and Craig \[2003 \]](#) point out in their book *Understanding Virtual Reality* that [Webster's New Universal Unabridged Dictionary \[1989 \]](#) defines virtual as “being in essence or effect, but not in fact” and reality as “the state or quality of being real. Something that exists independently of ideas concerning it. Something that constitutes a real or actual thing as distinguished from something that is merely apparent.” Thus, virtual reality is a term that contradicts itself—an oxymoron! Fortunately, the website [merriam-webster.com \[Merriam-Webster 2015 \]](#) has more recently defined the full term virtual reality to be “an artificial environment which is experienced through sensory stimuli (as sights and sounds) provided by a computer and in which one’s actions partially determine what happens in the environment.” In this book, **virtual reality** is defined to be a computer-generated digital environment that can be experienced and interacted with as if that environment were real.

An ideal VR system enables users to physically walk around objects and touch those objects as if they were real. Ivan Sutherland, the creator of one of the world’s first VR systems in the 1960s, stated [[Sutherland 1965](#)]: “The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal.” We haven’t yet come anywhere near Ivan Sutherland’s vision (nor do we necessarily want to!) and perhaps we never will. However, there are some quite engaging virtual realities today—many of which are featured throughout this book.

1.2 VR Is Communication

Normally, communication is thought of as interaction between two or more people. This book defines **communication** more abstractly: the transfer of energy between two entities, even if just the cause and effect of one object colliding with another object. Communication can also be between human and technology—an essential component and basis of VR. VR design is concerned with the communication of how the virtual world works, how that world and its objects are controlled, and the relationship between user and content: ideally where users are focused on the experience rather than the technology.

Well-designed VR experiences can be thought of as collaboration between human and machine where both software and hardware work harmoniously together to provide intuitive communication with the human. Developers write complex software to create, if designed well, seemingly simple transfer functions to provide effective interactions and engaging experiences. Communication can be broken down into direct communication and indirect communication as discussed below.

1.2.1 Direct Communication

Direct communication is the direct transfer of energy between two entities with no intermediary and no interpretation attached. In the real world, pure direct communication between entities doesn’t represent anything as the purpose is not communication, but it is a side effect. However, in VR, developers insert an artificial intermediary (the VR system that is ideally unperceivable) between the user and carefully controlled sensory stimuli (e.g., shapes, motions, sounds). When the goal is direct communication, VR creators should focus on making the intermediary transparent so users feel like they have direct access to those entities. If that can be achieved, then users will perceive, interpret, and interact with stimuli as if they are directly communicating with the virtual world and its entities.

Direct communication consists of structural communication and visceral communication.

Structural Communication

Structural communication is the physics of the world, not the description or the mathematical representation but the *thing-in-itself* [[Kant 1781](#)]. An example of structural communication is the bouncing of a ball off of the hand. We are always in relationship to objects, which help to define our state; e.g., the shape of our hand around a controller. The world, as well as our own bodies, directly tells us what the structure is through our senses. Although thinking and feeling do not exist within structural communication, such communication does provide the starting point for perception, interpretation, thinking, and feeling.

In order for our ideas to persist through time, we must put those ideas into structural form, what [Norman \[2013 \]](#) calls knowledge in the world. Recorded information and data is the obvious example of structural form, but sometimes less obvious structural forms are the signifiers and constraints (Section [25.1](#)) of interaction. In order to induce experiences into others through VR, we present structural stimuli (e.g., pixels on a display, sound through headphones, or the rumble/vibration of a controller) so the users can sense and interact with our creations.

Visceral Communication

Visceral communication is the language of automatic emotion and primal behavior, not the rational representation of the emotions and behavior (Section [7.7](#)). Visceral communication is always present for humans and is the in-between of structural communication and indirect communication. Presence (Chapter [4](#)) is the act of being fully engaged via direct communication (albeit primarily one way). Examples of visceral

communication are the feeling of awe while sitting on a mountaintop, looking down at the earth from space, or being with someone via solid eye contact (whether in the real world or via avatars in VR). The full experience of such visceral communication cannot be put into words, although we often attempt to do so at which point the experience is transformed into indirect communication (e.g., explaining VR to someone is not the same as experiencing VR).

1.2.2 Indirect Communication

Indirect communication connects two or more entities through some intermediary. The intermediary need not be physical; in fact, the intermediary is often our mind's interpretation that sits between the world and behavior/action. Once we interpret and give something meaning, then we have transformed the direct communication into indirect communication. Indirect communication includes what we normally think of as language, such as spoken and written language, as well as sign languages and our internal thoughts (i.e., communicating with oneself). Indirect communication consists of talking, understanding, creating stories/histories, giving meaning, comparing, negating, fantasizing, lying, and romancing. These are not part of the objective real world but are what we humans describe and create with our minds. Indirect VR communication includes the user's internal mental model (Section 7.8) of how the VR world works (e.g., the interpretation of what is occurring in the VR world), and indirect interactions (Section 28.4) such as moving a slider that changes an object property, speech recognition that changes the system's state, and indirect gestures that act as a sign language to the computer.

1.3 What Is VR Good For?

The recent surge in media coverage about VR has inspired the public to become quite excited about its potential. This coverage has focused on the entertainment industry, specifically video games and immersive film. VR is a great fit for the entertainment industry and will certainly be the driving force behind VR in the short term. However, what is VR good for beyond entertainment? It turns out VR can have enormous benefit over a wide range of verticals. VR has been successfully deployed in various industries for many years now. Successful applications include oil and gas exploration, scientific visualization, architecture, flight simulation, therapy, military training, theme-park entertainment, engineering analysis, and design review. Using VR in such situations has successfully revealed costly design mistakes before manufacturing anything, reduced time to market by speeding up iterative processes, provided safe learning environments that would otherwise be dangerous, reduced PTSD by gradually increasing exposure to feared stimuli, and helped to visualize large datasets that would be difficult to comprehend with traditional systems.

Unfortunately, to date, VR has largely been limited to well-funded academic and corporate research labs to which few have access. That is all changing with consumer-priced systems now becoming widely available. The VR market is expected to first explode in the entertainment industry but will soon expand significantly in other industries. Education, telepresence, and professional training will likely be the next industries to take advantage of VR in a big way.

Whatever the industry, VR is largely about providing understanding—whether that is understanding an entertaining story, learning an abstract concept, or practicing a real skill. Actively using more of the human sensory capability and motor skills has been known to increase understanding/learning for some time [Dale 1969]. This is in part due to the increased sensory bandwidth between human and information, but there is much more to understanding. Actively participating in an action, making concepts intuitive, encouraging motivation through engaging experiences, and the thoughts inside one's head all contribute to understanding. This book focuses on how to design such concepts into VR experiences.

Figure 1.1 shows Edgar Dale's Cone of Experience [Dale 1969]. As can be seen in the figure, direct purposeful experiences provide the best basis for understanding. As Confucius stated, "I see and I forget. I hear and I remember. I do and I understand." Note this diagram does not suggest direct purposeful experiences should be the only method of learning, but instead describes the progression of learning experience. Adding other indirect information within direct purposeful VR experiences can further enhance understanding. For example, embedding abstract information such as text, symbols, and multimedia directly into the scene and onto virtual objects can lead to more efficient understanding than what can be achieved in the real world.

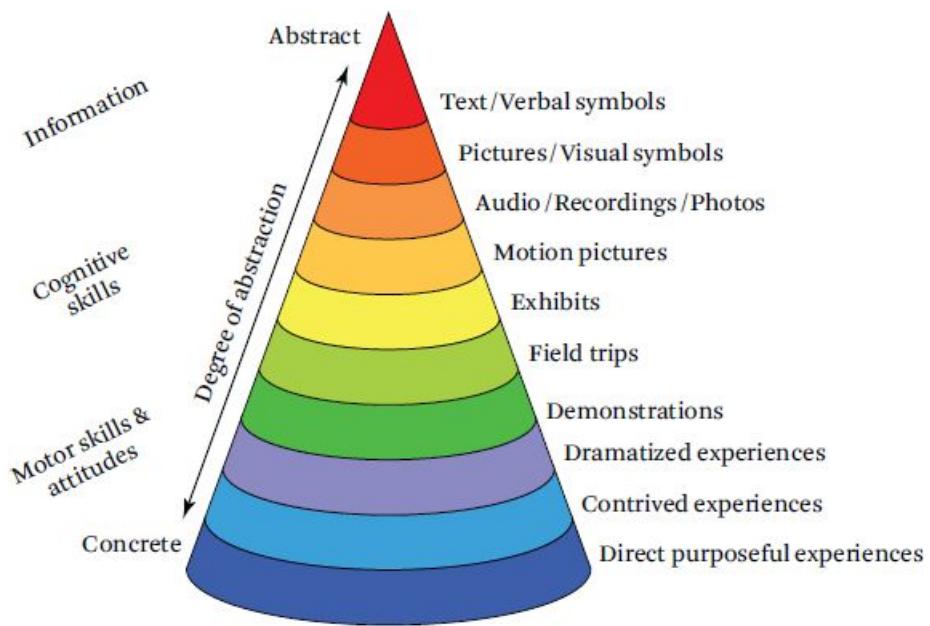


Figure 1.1 The Cone of Experience. VR uses many levels of abstraction. (Adapted from [Dale \[1969\]](#))

A History of VR

When anything new comes along, everyone, like a child discovering the world, thinks that they've invented it, but you scratch a little and you find a caveman scratching on a wall is creating virtual reality in a sense. What is new here is that more sophisticated instruments give you the power to do it more easily .

—Morton Heilig [[Habit 1993](#)]

Precursors to what we think of today as VR go back as far as humans have had imaginations and the ability to communicate through the spoken word and cave drawings (what could be called analog VR). The Egyptians, Chaldeans, Jews, Romans, and Greeks used magical illusions to entertain and control the masses. In the Middle Ages, magicians used smoke and concave mirrors to produce faint ghost and demon illusions to gull naive apprentices as well as larger audiences [[Hopkins 2013](#)]. Although the words and implementation have changed over the centuries, the core goals of creating the illusion of conveying that which is not actually present and capturing our imaginations remain the same.

2.1 The 1800s

The static version of today's stereoscopic 3D TVs is called a stereoscope and was invented before photography in 1832 by Sir Charles Wheatstone [[Gregory 1997](#)]. As shown in Figure 2.2, the device used mirrors angled at 45° to reflect images into the eye from the left and right side.

David Brewster, who earlier invented the kaleidoscope, used lenses to make a smaller consumer-friendly hand-held stereoscope (Figure 2.3.). His stereoscope was demonstrated at the 1851 Exhibition at the Crystal Palace where Queen Victoria found it quite pleasing. Later the poet Oliver Wendell Holmes stated “... is a surprise such as no painting ever produced. The mind feels its way into the very depths of the picture.” [[Zone 2007](#)]. By 1856 Brewster estimated over a half million stereoscopes had been sold [[Brewster 1856](#)]. This first 3D craze included various forms of the stereoscope, including self-assembled cardboard versions with moving images controlled by the hand in 1860 [[Zone 2007](#)]. One company alone sold a million stereoscopic views in 1862. Brewster's design is conceptually the same as the 20th century View-Master and today's Google Cardboard. In the case of Google Cardboard and similar phone-based VR systems, a cellular phone is used to display the images in place of the actual physical images themselves.

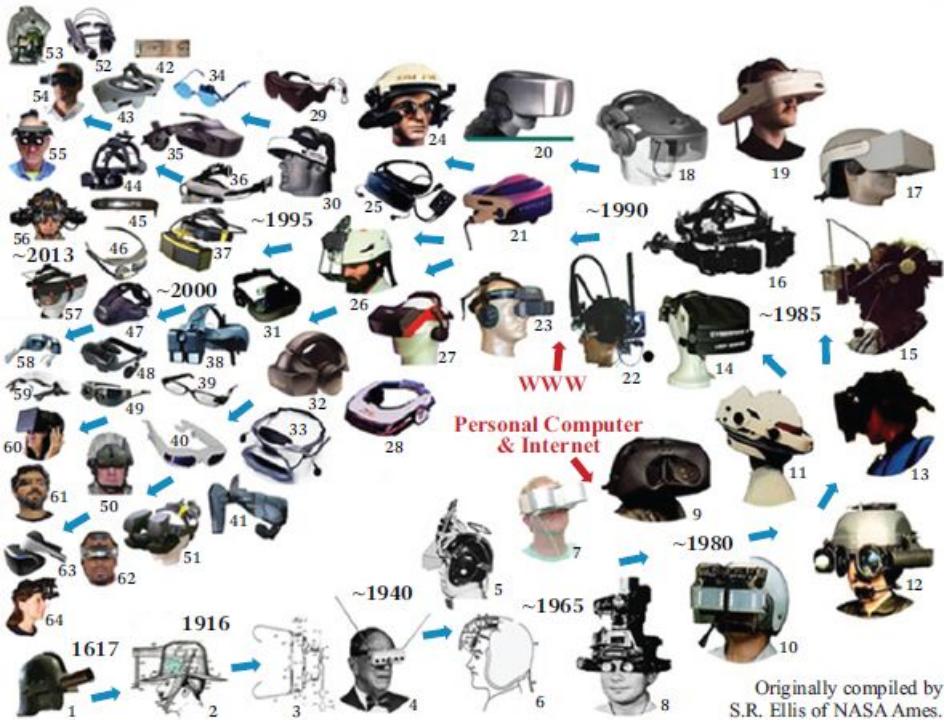


Figure 2.1 Some head-mounted displays and viewers over time. (Based on [Ellis \[2014 \]](#))

Many years later, a 360° VR-type display known as the Haunted Swing was shown at the '95 Midwinter Fair in San Francisco, and is still one of the most compelling technical demonstrations of an illusion to this day. The demo consisted of a room and large swing that held approximately 40 people. After the audience seated themselves, the swing was put in motion and as the swing oscillated, users felt motion similar to being in an elevator while they involuntarily clutched their seats. In fact, the swing hardly moved at all, but the surrounding room moved substantially, resulting in the sense of self-motion (Section 9.3.10) and motion sickness (Chapter 12). The date was not 1995, but was 1895 [[Wood 1895](#)].



Figure 2.2 Charles Wheatstone’s stereoscope.



Figure 2.3 A Brewster stereoscope from 1860. (Courtesy of The National Media Museum/Science & Society Picture Library, United Kingdom)

It was also in 1895 that film began to go mainstream; and when the audience saw a virtual train coming at them through the screen in the short film “L’Arrivée d’un train en gare de La Ciotat,” some people reportedly screamed and ran to the back of the room. Although the screaming and running away is more rumor than verified reports, there was certainly hype, excitement, and fear about the new artistic medium, perhaps similar to what is happening with VR today.

2.2 The 1900s

VR-related innovation continued in the 1900s that moved beyond simply presenting visual images. New interaction concepts started to emerge that might be considered novel for even today’s VR systems. For example, Figure 2.4 shows what is a head-worn gun pointing and firing device patented by Albert Pratt in 1916 [[Pratt 1916](#)]. No hand tracking is required to fire this device as the interface consists of a tube the user blows through.

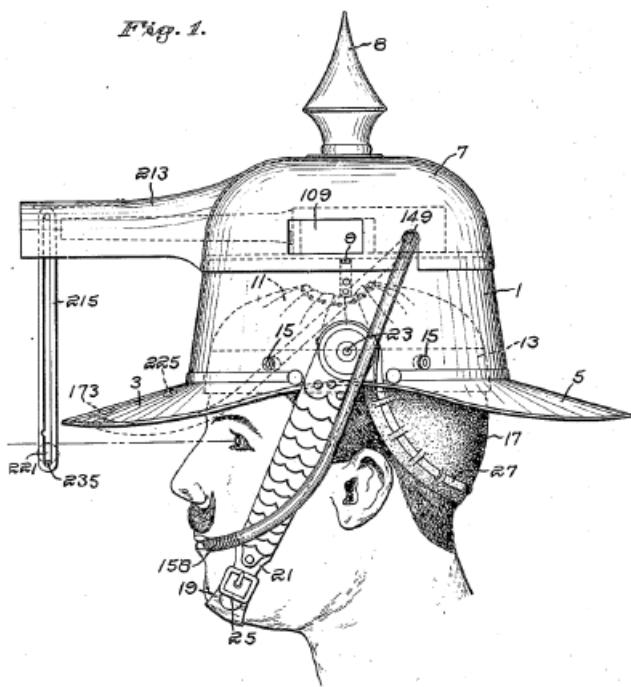


Figure 2.4 Albert Pratt's head-mounted targeting and gun-firing interface. (From [Pratt \[1916\]](#))

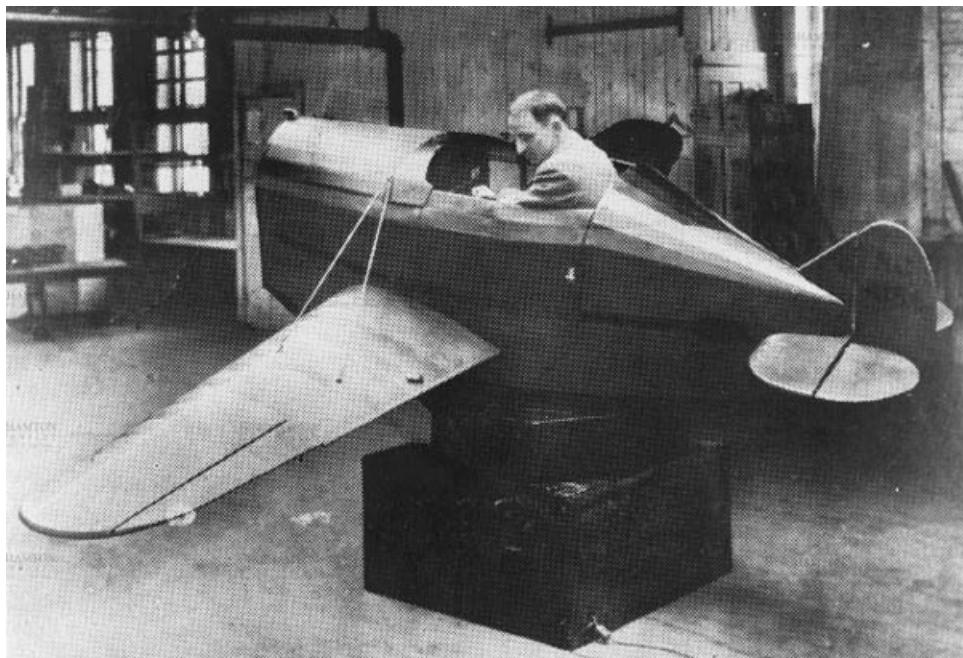


Figure 2.5 Edwin A. Link and the first flight simulator in 1928. (Courtesy of Edwin A. Link and Marion Clayton Link Collections, Binghamton University Libraries' Special Collections and University Archives, Binghamton University)

A little over a decade after Pratt received his weapon patent, Edwin Link developed the first simple mechanical flight simulator, a fuselage-like device with a cockpit and controls that produced the motions and sensations of flying (Figure 2.5). Surprisingly, his intended client—the military—was not initially interested, so he pivoted to selling to amusement parks. By 1935, the Army Air Corps ordered six systems and by the end of World War II, Link had sold 10,000 systems. Link trainers eventually evolved into astronaut training systems and advanced flight simulators complete with motion platform and real-time computer-generated imagery, and today is Link Simulation & Training, a division of L-3 Communications. Since 1991, the Link Foundation Advanced Simulation and Training Fellowship Program has funded many graduate students in their pursuits of improving upon VR systems, including work in computer graphics, latency, spatialized audio, avatars, and haptics [[Link 2015](#)].

As early 20th century technologies started to be built, science fiction and questions inquiring about what makes reality started to become popular. In 1935, for example, science fiction readers got excited about a surprisingly similar future that we now aspire to with head-mounted displays and other equipment through the book *Pygmalion's Spectacles* (Figure 2.6). The story opens with the words “But what is reality?” written by a professor friend of George Berkeley, the Father of Idealism (the philosophy that reality is mentally constructed) and for whom the University of California, Berkeley, is named. The professor then explains a set of eyeglasses along with other equipment that replaces real-world stimuli with artificial stimuli. The demo consists of a quite compelling interactive and immersive world where “The story is all about you, and you are in it” through vision sound, taste, smell, and even touch. One of the virtual characters calls the world Paracosma—Greek for “land beyond-the-world.” The demo is so good that the main character, although skeptical at first, becomes convinced that it is no longer illusion, but reality itself. Today

there are countless books ranging from philosophy to science fiction that discuss the illusion of reality.

PYGMALION'S SPECTACLES

By STANLEY G. WEINBAUM

Author of "The Black Flame," "A Martian Odyssey," etc.

© 1935 by Continental Publications, Inc.



Figure 2.6 *Pygmalion's Spectacles* is perhaps the first science fiction story written about an alternate world that is perceived through eyeglasses and other sensory equipment. (From [Weinbaum \[1935 \]](#))

Perhaps inspired by *Pygmalion's Spectacles*, McCollum patented the first stereoscopic television glasses in 1945. Unfortunately, there is no record of the device ever actually having been built.

In the 1950s Morton Heilig designed both a head-mounted display and a world-fixed display. The head-mounted display (HMD) patent [[Heilig 1960](#)] shown in Figure 2.7 claims lenses that enable a 140° horizontal and vertical field of view, stereo earphones, and air discharge nozzles that provide a sense of breezes at different temperatures as well as scent. He called his world-fixed display the Sensorama. As can be seen in Figure 2.8, the Sensorama was created for immersive film and it provided stereoscopic color views with a wide field of view, stereo sounds, seat tilting, vibrations, smell, and wind [[Heilig 1992](#)].

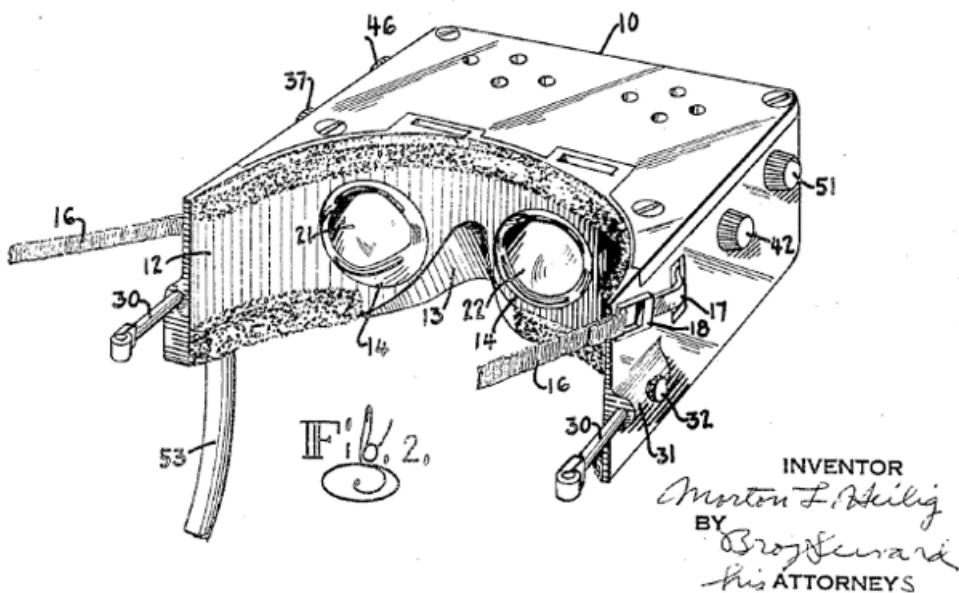


Figure 2.7 A drawing from Heilig's 1960 Stereoscopic Television Apparatus patent. (From [Heilig \[1960 \]](#))

In 1961, Philco Corporation engineers built the first actual working tracked HMD that included head tracking (Figure 2.9). As the user moved his head, a camera in a different room moved so the user could see as if he were at the other location. This was the world's first working telepresence system.

One year later IBM was awarded a patent for the first glove input device (Figure 2.10). This glove was designed as a comfortable alternative to keyboard entry, and a sensor for each finger could recognize multiple finger positions. Four possible positions for each finger with a glove in each hand resulted in 1,048,575 possible input combinations. Glove input, albeit with very different implementations, later became a common VR input device in the 1990s.

Starting in 1965, Tom Furness and others at the Wright-Patterson Air Force Base worked on visually coupled systems for pilots that consisted of head-mounted displays (Figure 2.11, left). While Furness was developing head-mounted displays at Wright-Patterson Air Force Base, Ivan Sutherland was doing similar work at Harvard and the University of Utah. Sutherland is known as the first to demonstrate a head-mounted display that used head tracking and computer-generated imagery [[Oakes 2007](#)]. The system was called the Sword of Damocles (Figure 2.11, right), named after the story of King Damocles who, with a sword hanging above his head by a single hair of a horse's tail, was in constant peril. The

story is a metaphor that can be applied to VR technology: (1) with great power comes great responsibility; (2) precarious situations give a sense of foreboding; and (3) as stated by [Shakespeare \[1598 \]](#) in *Henry IV*, “uneasy lies the head that wears a crown.” All these seem very relevant for both VR developers and VR users even today.

Introducing . . .

Sensorama

The Revolutionary Motion Picture System
that takes you into another world
with

- 3-D
- WIDE VISION
- MOTION
- COLOR
- STEREO-SOUND
- AROMAS
- WIND
- VIBRATIONS



SENSORAMA, INC., 855 GALLOWAY ST., PACIFIC PALISADES, CALIF. 90272
TEL. (213) 459-2162

Figure 2.8 Morton Heilig’s Sensorama created the experience of being fully immersed in film. (Courtesy of © Morton Heilig Legacy)



Figure 2.9 The Philco Headsight from 1961. (From [Comeau and Brian \[1961 \]](#))

Dr. Frederick P. Brooks, Jr., inspired by Ivan Sutherland’s vision of the Ultimate Display [[Sutherland 1965](#)], established a new research program in interactive graphics at the University of North Carolina at Chapel Hill, with the initial focus being on molecular graphics. This not only resulted in a visual interaction with simulated molecules but also included force feedback where the docking of simulated molecules could be felt. Figure 2.12 shows the resulting Grope-III system that Dr. Brooks and his team built. UNC has since focused on building various VR systems and applications with the intent to help practitioners solve real problems ranging from architectural visualization to surgical simulation.

In 1982, Atari Research, led by legendary computer scientist Alan Kay, was formed to explore the future of entertainment. The Atari research team, which included Scott Fisher, Jaron Lanier, Thomas Zimmerman, Scott Foster, and Beth Wenzel, brainstormed novel ways of interacting with computers and designed technologies that would soon be essential for commercializing VR systems.

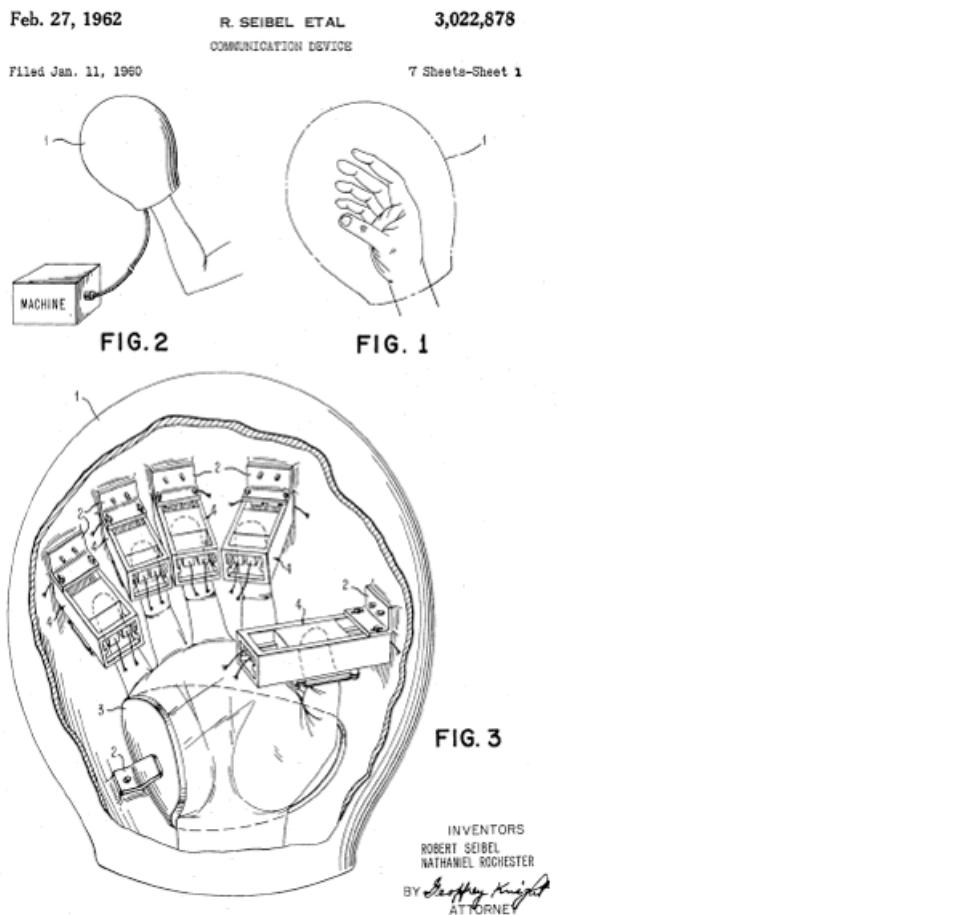


Figure 2.10 An image from IBM's 1962 glove patent. (From [Rochester and Seibel \[1962\]](#).)

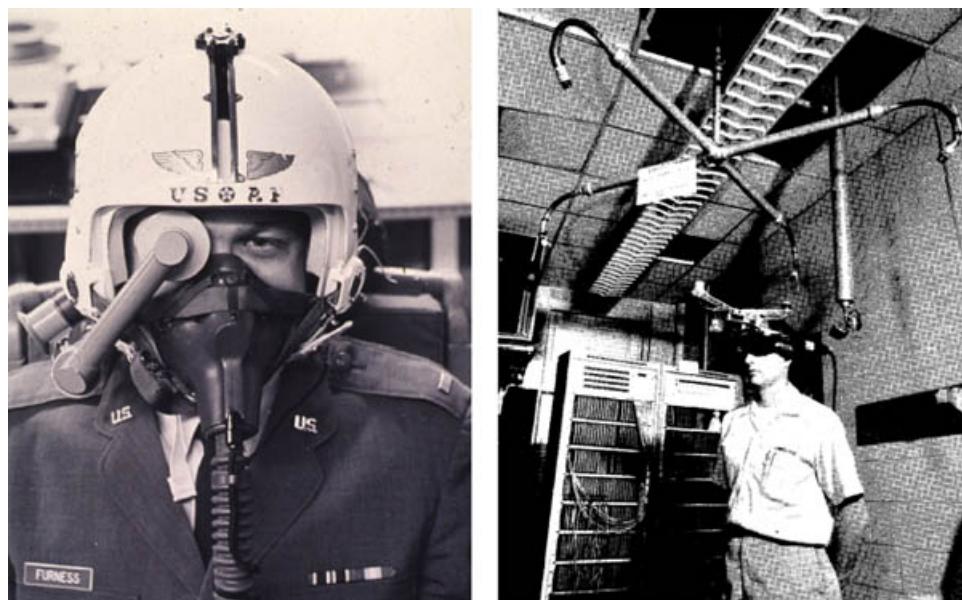


Figure 2.11 The Wright-Patterson Air Force Base head-mounted display from 1967 (courtesy of Tom Furness) and the Sword of Damocles [[Sutherland 1968](#)])

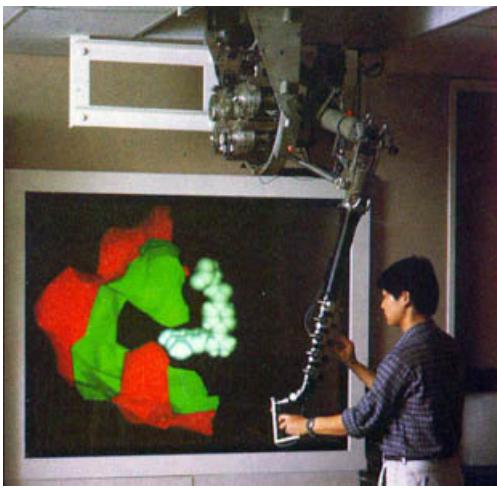


Figure 2.12 The Grope-III haptic display for molecular docking. (From [Brooks et al. \[1990 \]](#))



Figure 2.13 The NASA VIEW System. (Courtesy of NASA/S.S. Fisher, W. Sisler, 1988)

In 1985, Scott Fisher, now at NASA Ames, along with other NASA researchers developed the first commercially viable, stereoscopic head-tracked HMD with a wide field of view, called the Virtual Visual Environment Display (VIVED). It was based on a scuba diver's face mask with the displays provided by two Citizen Pocket TVs (Scott Fisher, personal communication, Aug 25, 2015). Scott Foster and Beth Wenzel built a system called the Convovoltron that provided localized 3D sounds. The VR system was unprecedented as the HMD could be produced at a relatively affordable price, and as a result the VR industry was born. Figure 2.13 shows a later system called the VIEW (Virtual Interface Environment Workstation) system.

Jaron Lanier and Thomas Zimmerman left Atari in 1985 to start VPL Research (VPL stands for Visual Programming Language) where they built commercial VR gloves, head-mounted displays, and software. During this time Jaron coined the term "virtual reality." In addition to building and selling head-mounted displays, VPL built the Dataglove specified by NASA—a VR glove with optical flex sensors to measure finger bending and tactile vibrator feedback [[Zimmerman et al. 1987](#)].

VR exploded in the 1990s with various companies focusing mostly on the professional research market and location-based entertainment. Examples of the more well-known newly formed VR companies were Virtuality, Division, and Fakespace. Existing companies such as Sega, Disney, and General Motors, as well as numerous universities and the military, also started to more extensively experiment with VR technologies. Movies were made, numerous books were written, journals emerged, and conferences formed—all focused exclusively on VR. In 1993, *Wired* magazine predicted that within five years more than one in ten people would wear HMDs while traveling in buses, trains, and planes [[Negroponte 1993](#)]. In 1995, the *New York Times* reported that Virtuality Managing Director Jonathan Waldern predicted the VR market to reach \$4 billion by 1998 [[Bailey 1995](#)]. It seemed VR was about to change the world and there was nothing that could stop it. Unfortunately, technology could not support the promises of VR. In 1996, the VR industry peaked and then started to slowly contract with most VR companies, including Virtuality, going out of business by 1998.

2.3 The 2000s

The first decade of the 21st century is known as the "VR winter." Although there was little mainstream media attention given to VR from 2000 to 2012, VR research continued in depth at corporate, government, academic, and military research laboratories around the world. The VR community started to turn toward human-centered design with an emphasis on user studies, and it became difficult to get a VR paper accepted at a conference without including some form of formal evaluation. Thousands of VR-related research papers from this era contain a wealth of knowledge that today is unfortunately largely unknown and ignored by those new to VR.

A wide field of view was a major missing component of consumer HMDs in the 1990s, and without it users were just not getting the "magic" feeling of presence (Mark Bolas, personal communication, June 13, 2015). In 2006, Mark Bolas of USC's MxR Lab and Ian McDowell of Fakespace Labs created a 150° field of view HMD called the Wide5, which the lab later used to study the effects of field of view on the user

experience and behavior. For example, users can more accurately judge distances when walking to a target when they have a larger field of view [Jones et al. 2012]. The team's research led to the low-cost Field of View To Go (FOV2GO), which was shown at the IEEE VR 2012 conference in Orange County, California, where the device won the Best Demo Award and was part of the MxR Lab's Open-source project that is the precursor to most of today's consumer HMDs. Around that time, a member of that lab named Palmer Luckey started sharing his prototype on Meant to be Seen ([mtbs3D.com](#)) where he was a forum moderator and where he first met John Carmack (now CTO of Oculus VR) and formed Oculus VR. Shortly after that he left the lab and launched the Oculus Rift Kickstarter. The hacker community and media latched onto VR once again. Companies ranging from start-ups to the Fortune 500 began to see the value of VR and started providing resources for VR development, including Facebook, which acquired Oculus VR in 2014 for \$2 billion. The new era of VR was born.

An Overview of Various Realities

This chapter aims to provide a basic high-level overview of various forms of reality, as well as different hardware options to build systems supporting those forms of reality. Whereas most of the book focuses on fully immersive VR, this chapter takes a broader view; its aim is to put fully immersive VR in the context of the larger array of options.

3.1 Forms of Reality

Reality takes many forms and can be considered to range on a virtuality continuum from the real environment to virtual environments [[Milgram and Kishino 1994](#)]. Figure 3.1 shows various forms along that continuum. These forms, which are somewhere between virtual and augmented reality, are broadly defined as “mixed reality,” which can be further broken down into “augmented reality” and “augmented virtuality.” This book focuses on the right side of the continuum from augmented virtuality to virtual environments.

The **real environment** is the real world that we live in. Although creating real-world experiences is not always the goal of VR, it is still important to understand the real world and how we perceive and interact with it in order to replicate relevant functionality into VR experiences. What is relevant depends on the goals of the application. Section [4.4](#) further discusses trade-offs of realism vs. more abstract implementations of reality. Part [II](#) discusses how we perceive real environments in order to help build better fully immersive virtual environments.

Instead of replacing reality, **augmented reality** (AR) adds cues onto the already existing real world, and ideally the human mind would not be able to distinguish between computer-generated stimuli and the real world. This can take various forms, some of which are described in Section [3.2](#).

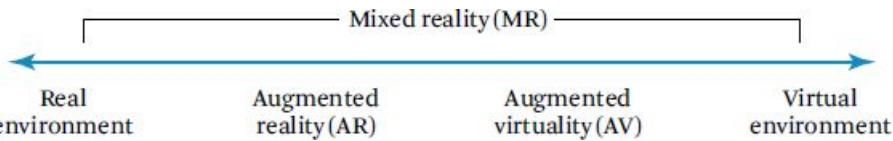


Figure 3.1 The virtuality continuum. (Adapted from [Milgram and Kishino \[1994 \]](#))

Augmented virtuality (AV) is the result of capturing real-world content and bringing that content into VR. Immersive film is an example of augmented virtuality. In the simplest case, the capture is taken from a single viewpoint, but in other cases, real-world capture can consist of light fields or geometry, where users can freely move about the environment, perceiving it from any perspective. Section [21.6](#) provides some examples of augmented virtuality.

True **virtual environments** are artificially created without capturing any content from the real world. The goal of virtual environments is to completely engage a user in an experience so that she feels as if she is present (Chapter [4](#)) in another world such that the real world is temporarily forgotten, while minimizing any adverse effects (Part [III](#)).

3.2 Reality Systems

The screen is a window through which one sees a virtual world. The challenge is to make that world look real, act real, sound real, feel real .

—[Sutherland \[1965 \]](#)

A **reality system** is the hardware and operating system that full sensory experiences are built upon. The reality system’s job is to effectively communicate the application content to and from the user in an intuitive way as if the user is interacting with the real world. Humans and computers do not speak the same language so the reality system must act as a translator or intermediary between them (note the reality system also includes the computer). It is the VR creator’s obligation to integrate content with the system so the intermediary is transparent and to ensure objects and system behaviors are consistent with the intended experience. Ideally, the technology will not be perceived so that users forget about the interface and experience the artificial reality as if it is real.

Communication between the human and system is achieved via hardware devices. These devices serve as input and/or output. A transfer function, as it relates to interaction, is a conversion from human output to digital input or from digital output to human input. What is output and what is input depends on whether it is from the point of view of the system or the human. For consistency, input is considered information traveling from the user into the system and output is feedback that goes from the system back to the user. This forms a cycle of input/output that continuously occurs for as long as the VR experience lasts. This loop can be thought of as occurring between the action and distal stimulus stages of the perceptual process (Figure [7.2](#)) where the user is the perceptual process.

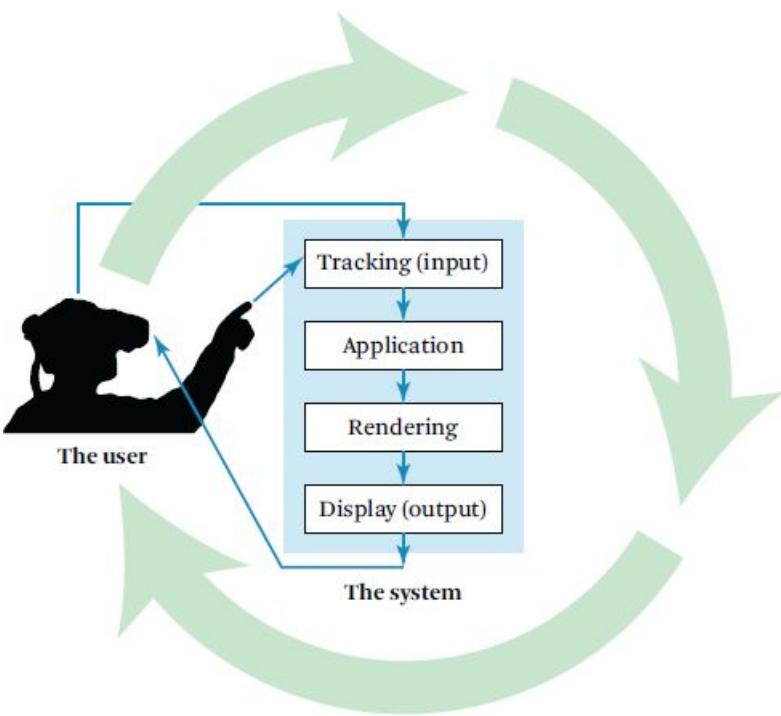


Figure 3.2 A VR system consists of input from the user, the application, rendering, and output to the user. (Adapted from [Jerald \[2009\]](#))

Figure 3.2 shows a user and a VR system divided into their primary components of input, application, rendering, and output. **Input** collects data from the user such as where the user's eyes are located, where the hands are located, button presses, etc. The **application** includes non-rendering aspects of the virtual world including updating dynamic geometry, user interaction, physics simulation, etc. **Rendering** is the transformation of a computer-friendly format to a user-friendly format that gives the illusion of some form of reality and includes visual rendering, auditory rendering (called auralization), and haptic (the sense of touch) rendering. An example of rendering is drawing a sphere. Rendering is already well defined (e.g., [Foley et al. 1995](#)) and other than high-level descriptions and elements that directly affect the user experience the technical details are not the focus of this book. **Output** is the physical representation directly perceived by the user (e.g., a display with pixels or headphones with sound waves).

The primary output devices used for VR are visual displays, speakers, haptics, and motion platforms. More exotic displays include olfactory (smell), wind, heat, and even taste displays. Input devices are only briefly mentioned in this chapter as they are described in detail in Chapter 27. Selecting appropriate hardware is an essential part of designing VR experiences. Some hardware may be more appropriate for some designs than others. For example, large screens are more appropriate than head-mounted displays for large audiences located at the same physical location. The following sections provide an overview of some commonly used VR hardware.

3.2.1 Visual Displays

Today's reality systems are implemented in one of three ways: head-mounted displays, world-fixed displays, and hand-held displays.

Head-Mounted Displays

A **head-mounted display** (HMD) is a visual display that is more or less rigidly attached to the head. Figure 3.3 shows some examples of different HMDs. Position and orientation tracking of HMDs is essential for VR because the display and earphones move with the head. For a virtual object to appear stable in space, the display must be appropriately updated as a function of the current pose of the head; for example, as the user rotates his head to the left, the computer-generated image on the display should move to the right so that the image of the virtual objects appears stable in space, just as real-world objects are stable in space as people turn their heads. Well-implemented HMDs typically provide the greatest amount of immersion. However, doing this well consists of many challenges such as accurate tracking, low latency, and careful calibration.

HMDs can be further broken down into three types: non-see-through HMDs, video-see-through HMDs, and optical-see-through HMDs. **Non-see-through HMDs** block out all cues from the real world and provide optimal full immersion conditions for VR. **Optical-see-through HMDs** enable computer-generated cues to be overlaid onto the visual field and provide the ideal augmented reality experience. Conveying the ideal augmented reality experience using optical-see-through head-mounted displays is extremely challenging due to various requirements (extremely low latency, extremely accurate tracking, optics, etc.). Due to these challenges, **video-see-through HMDs** are sometimes used. Video-see-through HMDs are often considered to be augmented virtuality (Section 3.1), and have some advantages and disadvantages of both augmented reality and virtual reality.



Figure 3.3 The Oculus Rift (upper left; courtesy of Oculus VR), CastAR (upper right; courtesy of CastAR), the Joint-Force Fighter Helmet (lower left; courtesy of Marines Magazine), and a custom built/modified HMD (lower right; from [Jerald et al. \[2007 \]](#)).

World-Fixed Displays

World-fixed displays render graphics onto surfaces and audio through speakers that do not move with the head. Displays take many forms, ranging from a standard monitor (also known as fish-tank VR) to displays that completely surround the user (e.g., CAVEs and CAVE-like displays as shown in Figures 3.4 and 3.5.). Display surfaces are typically flat surfaces, although more complex shapes can be used if those shapes are well defined or known, as shown in Figure 3.6. Head tracking is important for world-fixed displays, but accuracy and latency requirements are typically not as critical as they are for head-mounted displays because stimuli are not as dependent upon head motion. High-end world-fixed displays with multiple surfaces and projectors can be highly immersive but are more expensive in dollars and space.

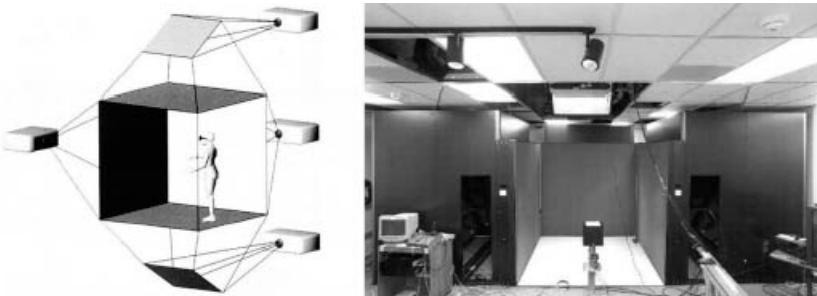


Figure 3.4 Conceptual drawing of a CAVE (left). Users are surrounded with stereoscopic perspective-correct images displayed on the floor and walls that they interact with. The CABANA (right) has movable walls so that the display can be configured into different display shapes such as a wall or L-shape. (From [Cruz et al. \[1992 \]](#) (left) and [Daily et al. \[1999 \]](#) (right))

World-fixed displays typically are considered to be part virtual reality and part augmented reality. This is because real-world objects are easily integrated into the experience, such as the physical chair shown in Figure 3.7. However, it is often the intent that the user's body is the only visible real-world cue.

Hand-Held Displays

Hand-held displays are output devices that can be held with the hand(s) and do not require precise tracking or alignment with the head/eyes (in fact the head is rarely tracked for hand-held displays). Hand-held augmented reality, also called indirect augmented reality, has recently become popular due to the ease of access and improvements in smartphones/tablets (Figure 3.8). In addition, system requirements are much less since viewing is indirect—rendering is independent of the user's head and eyes.

3.2.2 Audio

Spatialized audio provides a sense of where sounds are coming from in 3D space. Speakers can be fixed in space or move with the head. Headphones are preferred for a fully immersive system as they block out more of the real world. How the ears and brain perceive sound is

discussed in Section [8.2](#). Audio more specific to how content is created for VR is discussed in Section [21.3](#).



Figure 3.5 The author interacting with desktop applications within the CABANA. (From [Jerald et al. \[2001 \]](#))

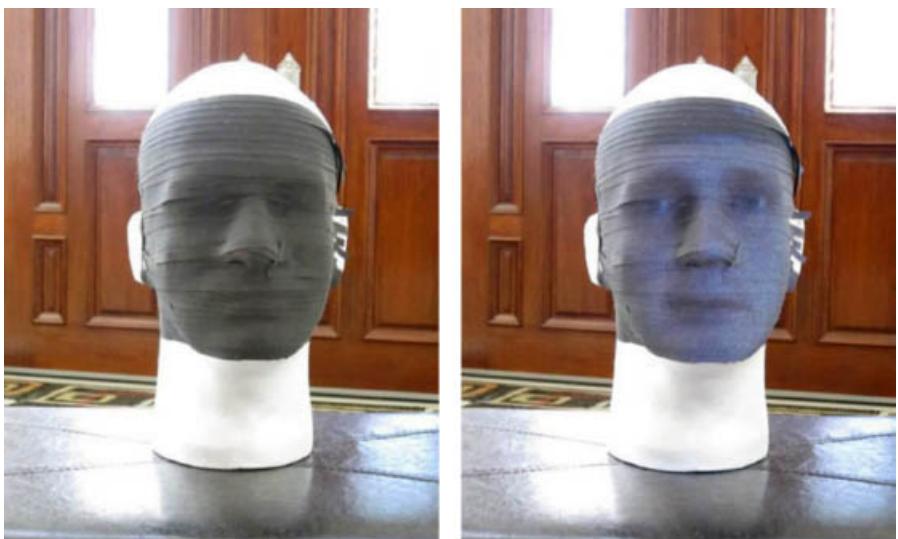


Figure 3.6 Display surfaces do not necessarily need to be planar. (From [Krum et al. \[2012 \]](#))



Figure 3.7 The University of Southern California's Gunslinger uses a mix of the real world along with world-fixed displays. (Courtesy of USC Institute for Creative Technologies)

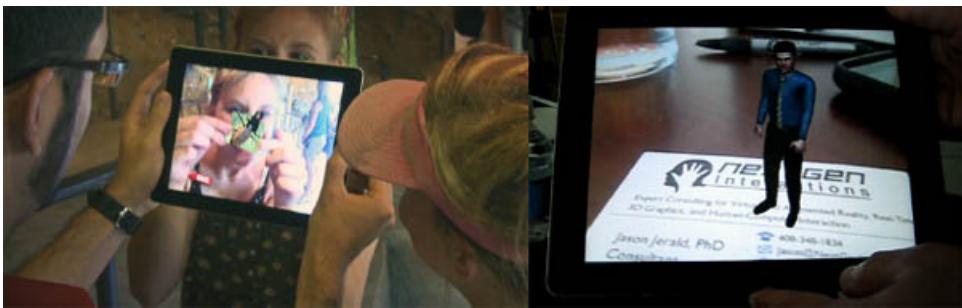


Figure 3.8 Zoo-AR from GeoMedia and a virtual assistant that appears on a business card from NextGen Interactions. (Courtesy of Geomedia (left) and NextGen Interactions (right))

3.2.3 Haptics

Haptics are artificial forces between virtual objects and the user's body. Haptics can be classified as passive (static physical objects) or active (physical feedback controlled by the computer), tactile (through skin) or proprioceptive force (through joints/muscles), and self-grounded (worn) or world-grounded (attached to real world). Many haptic systems also serve as input devices.

Passive vs. Active Haptics

Passive haptics provide a sense of touch in VR at a low cost—one simply creates a real-world physical object and matches that object to the shape of a virtual object [[Lindeman et al. 1999](#)]. These physical objects can be hand-held props or larger objects in the world that can be touched. Passive haptics increases presence, improves cognitive mapping of the environment, and improves training performance [[Insko 2001](#)].

Touching a few objects with passive haptics can make everything else seem more real. Perhaps the most compelling VR experience to this day is the legendary UNC-Chapel Hill Pit demo [[Meehan et al. 2002](#)]. Users first experience a virtual room that includes passive haptics made from Styrofoam blocks and other real-world material to match the visual VR environment. After touching different parts of the room, users walk into a second room and see a pit in the floor. The pit is quite compelling (in fact, heart rate increases) because everything else they have touched up to this point has physically felt real, thus users assume the pit is physically real as well. There is an even more startling response from many users when they put their toe over the virtual ledge and feel a real ledge. What they don't realize is the physical ledge is only a 1.5 inch drop-off compared to the visual pit that is 20 feet deep.

Active haptics are controlled by a computer and are the most common form of haptics. Active haptics have the advantage that forces can be dynamically controlled to provide a feeling of a wide range of simulated virtual objects. The remainder of this section focuses on active haptics.

Tactile vs. Proprioceptive Force Haptics

Tactile haptics provide a sense of touch through the skin. Vibrotactile stimulation evokes tactile sensations using mechanical vibration of the skin. Electrotactile stimulation evokes tactile sensation via an electrode passing current through the skin.

Figure 3.9 shows Tactical Haptics' Reactive Grip technology, which provides a sense of tactile feedback that is surprisingly compelling, especially when combined with fully immersive visual displays [[Provancher 2014](#)]. The system utilizes sliding skin-contact plates that can be added to any hand-held controller. Translational motions and forces are portrayed along the length of the grip by moving the plates in unison. Opposing motion and forces from different plates creates the feeling of a virtual object wrenching within the user's grasp.

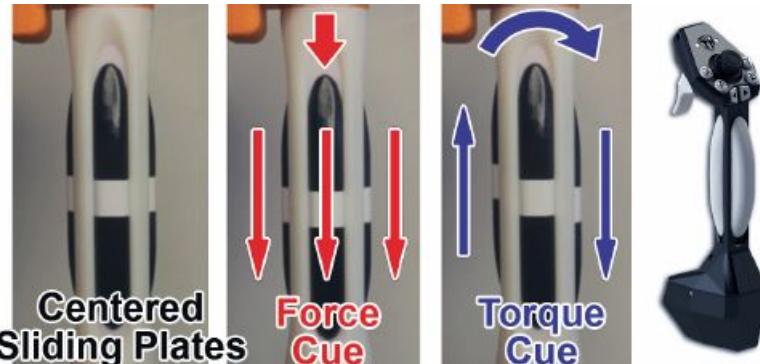


Figure 3.9 Tactical Haptics technology uses sliding plates to provide a sense of up or down force as well as rotational torque. The rightmost image shows the sliding plates integrated into the latest controller design. (Courtesy of Tactical Haptics)

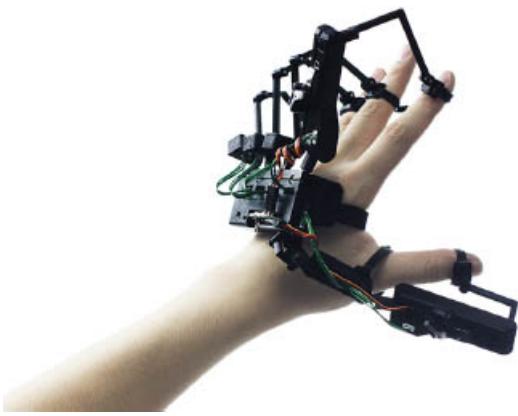


Figure 3.10 The Dexta Robotics Dexmo F2 device provides both finger tracking and force feedback. (Courtesy of Dexta Robotics)

Proprioceptive force provides a sense of limb movement and muscular resistance. Proprioceptive haptics can be self-grounded or world-grounded.

Self-Grounded vs. World-Grounded Haptics

Self-grounded haptics are worn/held by and move with the user. The forces applied are relative to the user. Gloves with exoskeletons or buzzers are examples of self-grounded haptics. Figure 3.10 shows an exoskeleton glove. Hand-held controllers are also examples of self-grounded haptics. Such controllers might be simply a passive prop that acts as a handle to virtual objects or might be rumble controllers that vibrate to provide feedback to the user (e.g., to signify the user has put his hand through a virtual object).

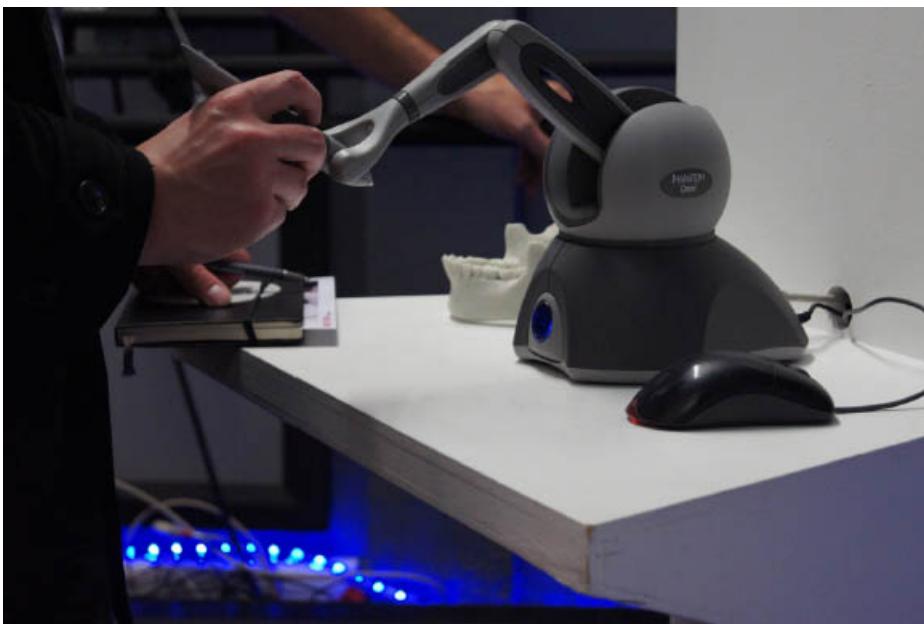


Figure 3.11 Sensable's Phantom haptics system. (Courtesy of INITION)

World-grounded haptics are physically attached to the real world and can provide a true sense of fully solid objects that don't move because the position of the virtual object providing the force can remain stable relative to the world. The ease of which the object can be moved can also be felt, providing a sense of weight and friction [Craig et al. 2009].

Figure 3.11 shows Sensable's Phantom haptic device, which provides stable force feedback for a single point in space (the tip of the stylus). Figure 3.12 shows Cyberglove's CyberForce glove that provides the sense of touching real objects with the entire hand as if the objects were stationary in the world.

3.2.4 Motion Platforms

A **motion platform** is a hardware device that moves the entire body resulting in a sense of physical motion and gravity. Such motions can help to convey a sense of orientation, vibration, acceleration, and jerking. Common uses of platforms are for racing games, flight simulation, and location-based entertainment. When integrated well with the rest of a VR application, motion sickness can be reduced by decreasing the conflict between visual motion and felt motion. Section 18.8 discusses how motion platforms can be used to reduce motion sickness.



Figure 3.12 Cyberglove's Cyberforce Immersive Workstation. (Courtesy of Haptic Workstation with HMD at VRLab in EPFL, Lausanne, 2005)

Motion platforms can be active or passive. An **active motion platform** is controlled by the computer simulation. Figure 3.13 shows an example of an active motion platform that moves a base platform via hydraulic actuators. A **passive motion platform** is controlled by the user. For example, the tilting of a passive motion platform might be achieved by leaning forward, such as used with Birdly shown in Figure 3.14. Note active and passive as described here are from the point of view of the motion platform and system. When describing motion from the point of view of the user, passive implies the user is passively along for the ride, with no way to influence the experience, and active implies the user is actively influencing the experience.

3.2.5 Treadmills

Treadmills provide a sense that one is walking or running while actually staying in one place. Variable-incline treadmills, individual foot platforms, and mechanical tethers providing restraint can convey hills by manipulating the physical effort required to travel forward.



Figure 3.13 An active motion platform that moves via hydraulic actuators. A chair can be attached to the top of the platform. (Courtesy of Shifz, Syntharturalist Art Association)



Figure 3.14 Birdly by Somniacs. In addition to providing visual, auditory, and motion cues, this VR experience provides a sense of taste and smell. (Courtesy of Swissnex San Francisco and Myleen Hollero)



Figure 3.15 The Virtuix Omni. (Courtesy of Virtuix)

Omnidirectional treadmills enable simulation of physical travel in any direction and can be active or passive.

Active omnidirectional treadmills have computer-controlled mechanically moving parts. These treadmills move the treadmill surface in order to recenter the user on the treadmill (e.g., [Darken et al. 1997](#) and [Iwata 1999](#)). Unfortunately, such recentering can cause the user to lose balance.

Passive omnidirectional treadmills contain no computer-controlled mechanically moving parts. For example, the feet might slide along a low-friction surface (e.g., the Virtuix Omni as shown in Figure 3.15). A harness and surrounding encasing keeps the user from falling. Like other forms of non-real walking, the experience of walking on a passive treadmill does not perfectly match the real world (it feels more like walking on slippery ice), but can add a significant amount of presence and reduce motion sickness.

3.2.6 Other Sensory Output

VR largely focuses on the display component, but other components such as taste, smell, and wind can more fully immerse users and add to a VR experience. Figure 3.14 shows Birdly by Somniacs, a system that adds smells and wind to a VR experience. Section 8.6 discusses the senses of taste and smell.

3.2.7 Input

A fully immersive VR experience is more than simply presenting content. The more a user physically interacts with a virtual world using his own body in intuitive ways, the more that user feels engaged and present in that virtual world. VR interactions consist of both hardware and software working closely together in complex ways, yet the best interaction techniques are simple and intuitive to use. Designers must take into account input device capabilities when designing experiences—one input device might work well for one type of interaction but be inappropriate for another. Other interactions might work across a wider range of input devices. Part V contains multiple chapters on interaction with Chapter 27 focusing exclusively on input devices.

3.2.8 Content

VR cannot exist without content. The more compelling the content, the more interesting and engaging the experience. Content includes not only the individual pieces of media and their perceptual cues, but also the conceptual arc of the story, the design/layout of the environment, and computer- or user-controlled characters. Part [IV](#) contains multiple chapters dedicated to content creation.

Immersion, Presence, and Reality Trade-Offs

VR is about psychologically being in a place different than where one is physically located, where that place may be a replica of the real world or may be an imaginary world that does not exist and never could exist. Either way there are some commonalities and essential concepts that must be understood to have users feel like they are somewhere else. This chapter discusses immersion, presence, and trade-offs of reality.

4.1 Immersion

Immersion is the objective degree to which a VR system and application projects stimuli onto the sensory receptors of users in a way that is extensive, matching, surrounding, vivid, interactive, and plot informing [[Slater and Wilbur 1997](#)].

Extensiveness is the range of sensory modalities presented to the user (e.g., visuals, audio, and physical force).

Matching is the congruence between sensory modalities (e.g., appropriate visual presentation corresponding to head motion and a virtual representation of one's own body).

Surroundness is the extent to which cues are panoramic (e.g., wide field of view, spatialized audio, 360° tracking).

Vividness is the quality of energy simulated (e.g., resolution, lighting, frame rate, audio bitrate).

Interactivity is the capability for the user to make changes to the world, the response of virtual entities to the user's actions, and the user's ability to influence future events.

Plot is the story—the consistent portrayal of a message or experience, the dynamic unfolding sequence of events, and the behavior of the world and its entities.

Immersion is the objective technology that has the potential to engage users in the experience. However, immersion is only part of the VR experience as it takes a human to perceive and interpret the presented stimuli. Immersion can lead the mind but cannot control the mind. How the user subjectively experiences the immersion is known as presence.

4.2 Presence

Presence, in short, is a sense of “being there” inside a space, even when physically located in a different location. Because presence is an internal psychological state and a form of visceral communication (Section [1.2.1](#)), it is difficult to describe in words—it is something that can only be understood when experienced. Attempting to describe presence is like attempting to describe concepts such as consciousness or the feeling of love, and can be just as controversial. Nonetheless, the VR community craves a definition of presence, as such a definition would be useful for designing VR experiences. The definition of presence is based on a discussion that took place via the presence-l listserv during the spring of 2000 among members of a community of scholars interested in the presence concept. The lengthy explication, available on the ISPR website (<http://ispr.info>), begins with

Presence is a psychological state or subjective perception in which even though part or all of an individual's current experience is generated by and/or filtered through human-made technology, part or all of the individual's perception fails to accurately acknowledge the role of the technology in the experience. (International Society for Presence Research, 2000)

Whereas immersion is about the characteristics of technology, presence is an internal psychological and physiological state of the user; an awareness in the moment of being immersed in a virtual world while having a temporary amnesia or agnosia of the real world and the technical medium of the experience. When present, the user does not attend to and perceive the technology, but instead attends to and perceives the objects, events, and characters the technology represents. Users who feel highly present consider the experience specified by VR technology to be a place visited rather than simply something perceived.

Presence is a function of both the user and immersion. Immersion is capable of producing the sense of presence but immersion does not always induce presence—users can simply shut their eyes and imagine being somewhere else. Presence is, however, limited by immersion; the greater immersion a system/application provides then the greater potential for a user to feel present in that virtual world.

A **break-in-presence** is a moment when the illusion generated by a virtual environment breaks down and the user finds himself where he truly is—in the real world wearing an HMD [[Slater and Steed 2000](#)]. These breaks-in-presence can destroy a VR experience and should be avoided as much as possible. Example causes of a break-in-presence include loss of tracking, a person speaking from the real world that is not part of the virtual environment, tripping on a wire, a real-world phone ringing, etc.

4.3 Illusions of Presence

Presence induced by technology-driven immersion can be considered a form of illusion (Section [6.2](#)) since in VR stimuli are just a form of energy projected onto our receptors; e.g., pixels on a screen or audio recorded from a different time and place. Different researchers distinguish different forms of presence in different ways. Presence is divided below into four core components that are merely illusions of a nonexistent reality.

The Illusion of Being in a Stable Spatial Place

Feeling as if one is in a physical environment is the most important part of presence. This is a subset of what Slater calls “place illusion” [[Slater 2009](#)], and it occurs due to all of a user’s sensory modalities being congruent such that stimuli presented to that user (ideally with no encumbrances such as no restrictions on field of view, no cables pulling on the head, and the freedom to move around) behave as if those stimuli originated from real-world objects in 3D space. Depth cues (Section [9.1.3](#)) are especially important for providing a sense of being at a remote location, and the more depth cues that are consistent with each other the better. This illusion can be broken when the world does not feel stable due to long latency, low frame rate, miscalibration, etc.

The Illusion of Self-Embodiment

We have a lifetime of experience perceiving our own bodies when we look down. Yet many VR experiences contain no personal body—the user is a disembodied viewpoint in space. **Self-embodiment** is the perception that the user has a body within the virtual world. When disembodied users feel quite present and then are given a virtual body that properly matches their movements, they quickly realize that there are different levels of presence. Then if a user sees a visual object touching the skin while a physical object also touches the skin, presence is greatly strengthened and experienced more deeply (called the rubber hand illusion; [Botvinick and Cohen 1998](#)).

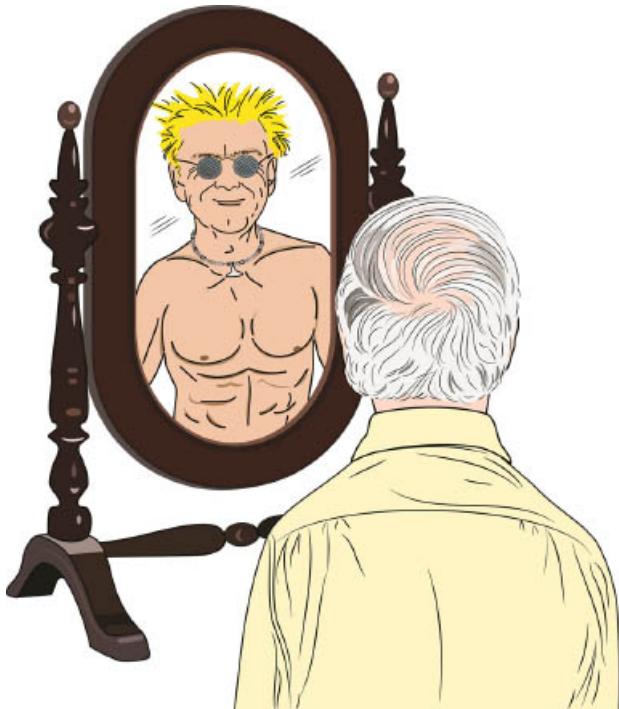


Figure 4.1 How we perceive ourselves can be quite distorted.

Surprisingly, the presence of a virtual body can be quite compelling even when that body does not look like one’s own body. In fact, we don’t necessarily perceive ourselves objectively and how we perceive ourselves, even in real life, through the lens of our subjectivity can be quite distorted (Figure [4.1](#); [Maltz 1960](#)). The mind also automatically associates the visual characteristics of what is seen at the location of the body with one’s own body. In VR, one can perceive oneself as a cartoon character or someone of a different gender or race, and the experience can still be quite compelling. Research has shown this to be quite effective for teaching empathy by “walking in someone else’s shoes” and can reduce racial bias [[Peck et al. 2013](#)]. Perhaps this is possible due to our being accustomed to changing clothing on a regular basis—just because our clothes are a different color/texture than our skin or what we were recently wearing does not mean what is underneath those clothes is not our own body. Whereas body shape and color are not so important, motion is extremely important and presence can be broken when visual body motion does not match physical motion in a reasonable manner.

The Illusion of Physical Interaction

Looking around is not enough for people to believe for more than a few seconds that they are in an alternate world. Even if not realistic, adding some form of feedback such as audio, visual highlighting, or a rumble of a controller can give the user a sense that she has in some way touched the world. Ideally, the user should feel a solid physical response that matches the visual representation (as described in Section [3.2.3](#)). As soon as one reaches out to touch something and there is no response, then a break-in-presence can occur. Unfortunately, strong physical feedback can be difficult to attain so sensory substitution (Section [26.8](#)) is often used instead.

The Illusion of Social Communication

Social presence is the perception that one is really communicating (both verbally and through body language) with other characters (whether computer- or user-controlled) in the same environment. Social realism does not require physical realism. Users have been found to exhibit anxiety responses when causing pain to a relatively low-fidelity virtual character [[Slater et al. 2006a](#)] and when users with a fear of public speaking must

talk in front of a low-fidelity virtual audience [[Slater et al. 2006b](#)].

Although social presence does increase as behavioral realism (the degree to which human representations and objects behave as they would in the physical world) increases [[Guadagno et al. 2007](#)], even tracking and rendering only a few points on human players can be quite compelling (Section [9.3.9](#)). Figure [4.2](#) shows a screenshot of the multiplayer VR game Arena by NextGen Interactions where the player's head and both hands are directly controlled (i.e., three tracked points) and lower-body turning/walking/running animations are indirectly controlled with analog sticks on Sixense Razer Hydra Controllers.

4.4 Reality Trade-Offs

Some consider real reality the gold standard of what we are trying to achieve with VR whereas others consider reality a goal to surpass—for if we can only get to the point of matching reality, then what is the point? This section discusses the trade-offs of trying to replicate reality vs. creating more abstract experiences.

4.4.1 The Uncanny Valley

Some people say robots and computer-generated characters come across as creepy. Although our sense of familiarity with simulated characters representing real characters increases as we get closer to reality, this is only up to a point. If reality is approached, but not attained, some of our reactions shift from empathy to revulsion. This descent into creepiness is known as the **Uncanny Valley**, as first proposed by Masahiro [Mori](#) [[1970](#)]. Figure [4.3](#) shows a graph of observers' comfort level with virtual humans as a function of human realism. Observer comfort increases as a character becomes more human-like up until a certain point at which observers start to feel uncomfortable with the almost, but not quite, human character.

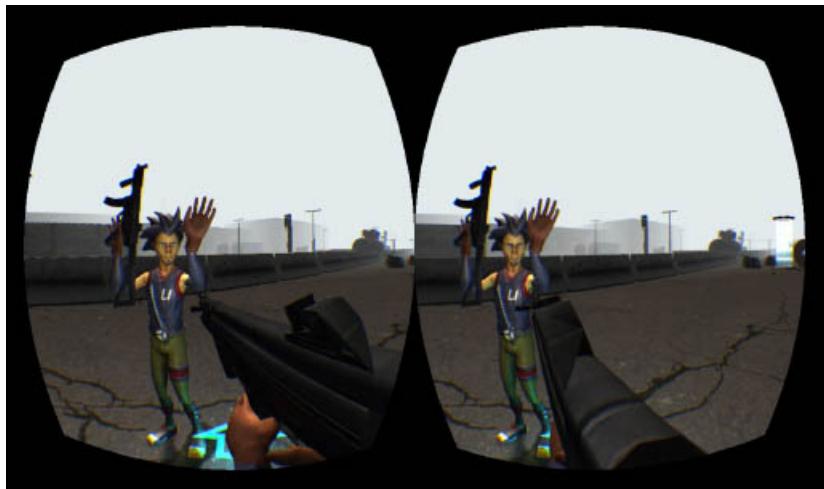


Figure 4.2 Paul Mlyniec, President of Digital ArtForms, surrenders without having to think about the interface as he is threatened by the author. Head and hand tracking alone can be extremely socially compelling in VR due to the ability to naturally convey body language. (Courtesy of NextGen Interactions)

The Uncanny Valley is a controversial topic that is more a simple explanatory theory rather than backed by scientific evidence. However, there is power in simplicity and the theory helps us to think about how to design and give character to VR entities. Creating cartoon characters often yields superior results over creating near-photorealistic humans. Section [22.5](#) discusses such issues in further detail.

4.4.2 Fidelity Continua

The Uncanny Valley of character creepiness is not the only case of things being close to reality not necessarily being better. The goal of VR is not necessarily to replicate reality. Contrary to what one might initially think, presence does not require photorealism and there are more important presence-inducing cues such as responsiveness of the system, character motion, and depth cues. Simple worlds consisting of basic structures that provide a sense of spatial stability can be extremely compelling, and making worlds more photorealistic does not necessarily increase presence [[Zimmons and Panter 2003](#)]. Being in a cartoon world can feel as real as a world captured by 3D scanners.

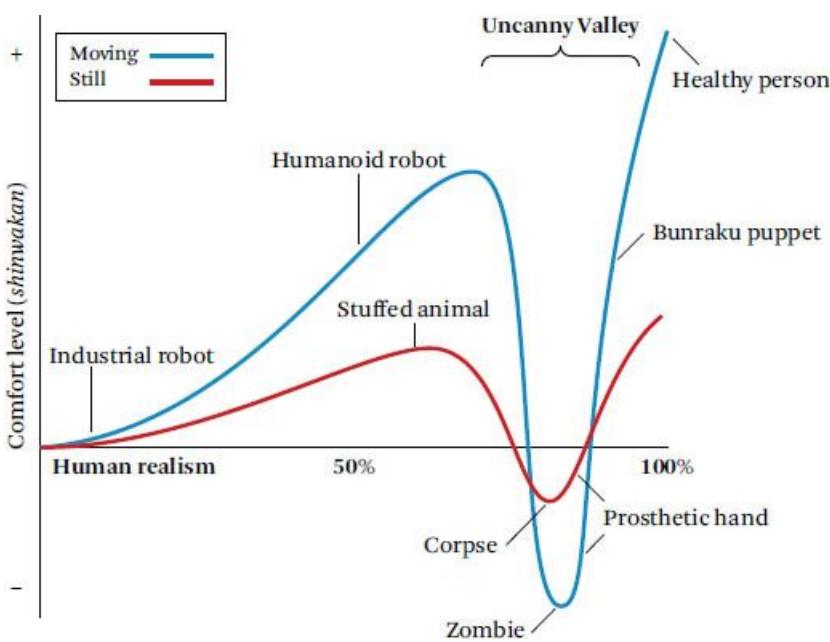


Figure 4.3 The Uncanny Valley. (Based on [Ho and MacDorman \[2010 \]](#))

Highly presence-inducing experiences can be ranked on different continua, and one extreme on each continuum is not necessarily better than the other extreme. What points on the continua VR creators choose depends upon the vision and goals of the project.

Listed below are some VR fidelity continua that VR creators should consider.

Representational fidelity is the degree to which the VR experience conveys a place that is, or could be, on Earth. At the high end of this spectrum is photorealistic immersive film in which the real world is captured with depth cameras and microphones, then re-created in VR. At the extreme low end of the spectrum are purely abstract or non-objective worlds (e.g., blobs of color and strange sounds). These may have no reference to the real world, simply conveying emotions, exploring pure visual events, or presenting other non-narrative qualities (Paul Mlyniec, personal communication, April 28, 2015). Cartoon worlds and abstract video games are somewhere in the middle depending on how closely the scene and characters represent the real world and its inhabitants.

Interaction fidelity is the degree to which physical actions for a virtual task correspond to physical actions for the equivalent real-world task (Section [26.1](#)). On one end of this spectrum are physical training tasks where low interaction fidelity risks negative training effects (Section [15.1.4](#)). At the other extreme are interaction techniques that require no physical motion beyond a button press. Magical techniques are somewhere in the middle where users are able to do things they are not able to do in the real world such as grabbing objects at a distance.

Experiential fidelity is the degree to which the user's personal experience matches the intended experience of the VR creator (Section [20.1](#)). A VR application that closely conveys what the creator intended has high experiential fidelity. A free-roaming world where endless possibilities exist and every usage results in a different experience has low experiential fidelity.

The Basics: Design Guidelines

No other technology can cause fear, running, or fleeing in the way that VR can. VR can also induce frustration and anger when things don't work, and worse—even physical sickness if not properly implemented and calibrated. Well-designed VR can produce the awe and excitement of being in a different world, improve performance and cut costs, provide new worlds to experience, improve education, and create better understanding by walking in someone else's shoes. With the foundation set in this chapter, the reader can better understand subsequent parts of this book and start creating VR experiences. As VR creators, we have the chance to change the world—let's not blow it through bad design and implementation due to not understanding the basics.

High-level guidelines for creating VR experiences are provided below.

5.1 Introduction and Background (Part 1)

- In order to have a toolbox of options to intelligently choose from, learn the basics of VR, such as the different forms of augmented and virtual reality, different hardware options, ways to present information to the senses, and how to induce presence into the minds of users.

5.2 VR Is Communication (Section 1.2)

- Focus on the user experience rather than the technology (Section 1.2).
- Simplify and harmonize the communication between user and technology (Section 1.2).
- Focus on making the technical intermediary between the user and content transparent so users feel like they have direct access to the virtual world and its entities (Section 1.2.1).
- Design for visceral communication (Section 1.2.1) in order to induce presence (Section 4.2) and inspire awe in users.

5.3 An Overview of Various Realities (Chapter 3)

- Choose what form of reality you wish to create (Section 3.1). Where does it fall on the virtuality continuum?
- Choose what type of input and output hardware to use (Section 3.2).
- Understand a VR application is more than just the hardware and technology. Create a strong conceptual story, an interesting design or layout of the environment, and engaging characters (Section 3.2.8).

5.4 Immersion, Presence, and Reality Trade-Offs (Chapter 4)

- Since presence is a function of both immersion and the user, we can't entirely control it. Maximize presence by focusing on what we do have control over—immersion (Section 4.1).
- Minimize breaks-in-presence (Section 4.2).
- For maximum presence, focus first on world stability and depth cues. Then consider adding physical user interactions, cues of one's own body, and social communication (Section 4.3).
- Avoid the Uncanny Valley by not trying to make characters appear too close to the way real humans look (Section 4.4.1).
- Choose what level of representational fidelity, interaction fidelity, and experiential fidelity you want to create (Section 4.4.2).

PART II

PERCEPTION

We see things not as they are, but as we are—that is, we see the world not as it is, but as molded by the individual peculiarities of our minds.

—G. T. W. Patrick ([1890](#))

Of all the technologies required for a VR system to be successful, there is one component that is most essential. Fortunately, it is widely available, although most of the population is only vaguely aware of it, fewer have even a moderate understanding of how it operates, and no single individual completely understands it. Like most technologies, it originally started in a very simple form, but has since evolved into a much more sophisticated and intelligent platform. In fact, it is the single most complicated system in existence today, and will remain so for the foreseeable future. It is the human brain.

The human brain contains approximately 100 billion neurons and, on average, each of these neurons is connected by thousands of synapses—structures that allow neurons to pass an electrochemical signal to another cell—resulting in hundreds of trillions of synaptic connections that can simultaneously exchange and process prodigious amounts of information over a massively parallel neural network in milliseconds (Marois and Ivanoff, 2005). In fact, every cubic millimeter of cerebral cortex contains roughly a billion synapses (Alonso-Nanclares et al., 2008). The white matter of the nervous system contains approximately 150,000–180,000 km of myelinated nerve fibers at age 20 (i.e., the connecting wires in our white matter could circle the earth ~4 times), connecting all these neuronal elements. Despite the monumental number of components in the brain, it is estimated that each neuron is able to contact any other neuron with no more than six interneuronal connections or “six degrees of separation” (Drachman, 2005).

Many of us tend to believe that we understand human behavior and the human mind—after all we are human and should therefore understand ourselves. To some extent this is true—knowing the basics is enough to function in the real world or to experience VR. In fact, this part can be skipped if the intent is to simply re-create what others have already built or to even create new simple worlds (although a theoretical background does help to fine-tune one’s creations). However, most of what we perceive and do is a result of subconscious processes. To go beyond the basics of VR to create the most innovative experiences in a way that is comfortable and intuitive to work with, a more detailed understanding of perception is helpful.

What is meant by “beyond the basics of VR”? VR is a relatively new medium and the vast space of artificial realities that could be created is largely unexplored. Thus, we get to make it all up. In fact, many believe providing direct access to digital media through VR has no limits. Whereas that may be true, that does not mean every possible arrangement of input and output will result in a quality VR experience. In fact, only a fraction of the possible input/output combinations will result in a comprehensible and satisfying experience. Understanding how we perceive reality is essential to designing truly innovative VR applications that effectively present and react to users. Some concepts discussed in this part directly apply to VR today and other concepts may apply in ways we can’t yet fathom. Understanding the core concepts of human perception might just enable us to recognize new opportunities as VR improves and the vast array of new VR experiences is explored.

Of course studying perception is valuable in designing any product; e.g., desktop or hand-held applications. For VR, it is essential. Rarely has someone become sick due to a bad traditional software design (although plenty of us have become frustrated!), but sickness is common for a bad VR design. VR designers integrate more senses into a single experience, and if sensations are not consistent, then users can become physically ill (multiple chapters are devoted to this—see Part [III](#)). Thus, understanding the human half of the hybrid interaction design is more important for VR applications than it is for traditional applications. Once we have obtained this knowledge, then we can more intelligently experiment with the rules. Simply put, the better our understanding of human perception, the better we can create and iterate upon quality VR experiences.

The material in this part will help readers become more aware and appreciative of the real world, and how we humans perceive it. It addresses less common questions such as the following.

- Why do the images on my eyeballs not seem to move when I rotate my eyes? See Section [7.4](#).
- What is the resolution of the human eye? See Section [8.1.4](#).
- Why does a character not seem to change sizes when that character walks away from the camera in a movie or game even though his image takes up a smaller portion of the screen? See Section [10.1](#).

By asking and understanding the answers to these types of questions (albeit we are far from knowing all the answers!), we can create better VR experiences in more innovative ways.

To appreciate perception, consider what you are experiencing at this very moment. You are reading the pages of this book without consciously thinking about how these abstract symbols called letters and words convey meaning in your mind. The same holds true for pictures. **Trompe-l’œil** (French for “deceive the eye”) is an art technique that uses realistic 2D imagery to create the illusion of 3D. Looking at the trompe-l’œil lobby art in Figure [II.1](#), you may perceive a lobby floor that opens up to lower levels of the building. In reality, you are seeing multiple 2D planes giving the illusion of depth (the real 3D world containing this 2D page or display showing a 2D photograph of a 3D scene containing a planar lobby surface with 2D art). Unfortunately, trompe-l’œil art only provides an illusion of 3D from a specific static viewing location. Move a few centimeters and the illusion breaks down. VR uses head tracking so illusions work from any viewpoint, in some cases even as one physically walks around.

Subconscious perception goes beyond vision. For example, perhaps you are touching this book or a table and do not consciously consider that specific feeling of touch until this prose suggested it. Conversely, even though you may not have been aware of that touch before the previous statement, you probably would have noticed if you did not feel anything. This lack of touch in most VR applications is a major difference between the real world and virtual worlds, and is one of the significant challenges of VR (see Section 26.8). Nevertheless, innovative, well-funded companies are taking on the challenge to circumvent or otherwise solve the current lack of perfect feedback mechanisms for VR. Whereas there are undoubtedly challenges, the arena is rapidly evolving, and such problems represent opportunities rather than barriers for the proliferation of VR.

Part II consists of several chapters that focus on different aspects of human perception.

Chapter 6, Objective and Subjective Reality, discusses the differences between what actually exists in the world and how we perceive what exists. The two can often be very different as demonstrated by various perceptual illusions described in the chapter.



Figure II.1 Trompe-l'œil lobby art.

Chapter 7, Perceptual Models and Processes, discusses different models and ways of thinking about how the human mind works. Because the brain is so complex, no single model explains everything and it is useful to think about how physiological processes and perception work from different perspectives.

Chapter 8, Perceptual Modalities, discusses our sensory receptors and how signals from those receptors are transformed into sight, hearing, touch, proprioception, balance and physical motion, smell, and taste.

Chapter 9, Perception of Space and Time, discusses how we perceive the layout of the world around us, how we perceive time, and how we perceive the motion of both objects and ourselves.

Chapter 10, Perceptual Stability, Attention, and Action, discusses how we perceive the world to be consistent even when signals reaching our sensory receptors change, how we focus on items of interest while excluding others, and how perception relates to taking action.

Chapter 11, Perception: Practitioner's Summary and Design Guidelines, summarizes Part II and provides perception guidelines for VR creators.

Objective and Subjective Reality

What's reality anyway? Just a collective hunch!

—Lily Tomlin [[Hamit 1993](#)]

Objective reality is the world as it exists independent of any conscious entity observing it. Unfortunately, it is impossible for someone to perceive objective reality exactly as it is. This chapter describes how we perceive objective reality through our own subjectivity. This is done through a philosophical discussion as well as with examples of perceptual illusion.

6.1 Reality Is Subjective

The things we sense with our eyes, ears, and bodies are not simply a reflection of the world around us. We create in our minds the realities that we think we live in. **Subjective reality** is the way an individual perceives and experiences the external world in his own mind. Much of what we perceive is the result of our own made-up fiction of what happened in the past, and now believe to be the “truth.” We continually make up reality as we go along even though most of us do not realize it.

Abstract philosophy with no application to VR? That is certainly a valid opinion. Take a look at Figure 6.1. What is the objective reality? An old lady or a young lady? Look again, and try to see if you can perceive both. This is a classic textbook example of an ambiguous image. However, there is neither an old lady or a young lady. What actually exists in objective reality is a black-and-white image made of ink on paper or pixels on a digital display. This is essentially what we are doing with VR—creating artificial content and presenting that to users so that they perceive our content as being real.

Figure 6.2 demonstrates how reality is what we make of it. The image is a 100-plus-year-old face with a resolution of 14×18 grayscale pixels [[Harmon 1973](#)]. If you live in the United States then you may recognize who this is a picture of, even though there is no possible way you could have met the individual. If you do not live in the United States, then you may not recognize the face. The picture is a low-resolution image of the face on the US five dollar bill—President Abraham Lincoln. Culture and our memories (even if made up by ourselves or through someone else) contribute significantly to what we see. We don't see Abraham Lincoln because he looks like Abraham Lincoln, we see Abraham Lincoln because we “know” what Abraham Lincoln looks like even though none of us alive today have ever met him.



Figure 6.1 Old lady/young lady ambiguous image.

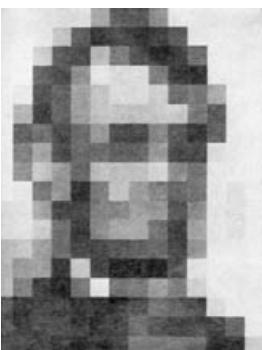


Figure 6.2 We see what or who we think we know. (From [Harmon \[1973\]](#))

Einstein and Freud, although coming from very different perspectives, came to the same conclusion that reality is malleable, that our own point of view is an irreducible part of the equation of reality. Now with VR, not only do we get to experience the realities of others but we get to *create* objective realities that are more directly communicated (vs. indirectly through a book or television) for others to experience through their own subjectivity.

Science attempts to take subjectivity out of the universe, but that can often only be done under controlled conditions. In our day-to-day lives outside of the lab, our perceptions depend upon context. VR enables us to take some of the subjectivity out of synthetically created worlds by controlling conditions as done in a lab. Although much of VR experiences occur within individuals' own minds, by understanding how different perceptual processes work we can better lead people's minds for them to more closely experience our created realities as we intend.

We go about our lives and subconsciously organize sensory inputs via our made-up "rules" and match those to our expected outcomes (Section [7.3](#)). Have you ever drunk orange juice when you were expecting milk? If so, then you may have gained a new experience of orange juice; if we don't perceive what we expect to perceive then we can perceive things very differently. When we are surprised or in need of deeper understanding, then our brains further transform, analyze, and manipulate the data via conceptual knowledge. At that time, our mental model of the world can be modified to match future expectations.

Facts, algorithms, and geometry are not enough for VR. By better understanding human perception, we can build VR applications that enable us to more directly connect the objective realities we create, by manipulating digital bits, into the minds of users. We seek to better understand how humans perceive those bits in order to more directly convey our intention to users in a way that is well received and interpreted.

The good news is that VR does not need to perfectly replicate reality. Instead we just need to present the most important stimuli well and the mind will fill in the gaps.

6.2 Perceptual Illusions

For most situations, we feel like we have direct access to reality. However, in order to create that feeling of direct access, our brains integrate various forms of sensory input along with our expectations. In many cases, our brains have been hardwired to predict certain things based on repeated exposure over a lifetime of perceiving. This hardwired subconscious knowledge influences the way we interpret and experience the world and enables our perception and thinking to take shortcuts, which provides more room for higher-level processing. Normally we don't notice these shortcuts. However, when atypical stimuli are apparent then **perceptual illusions** can result. These illusions can help us understand the shortcuts and how the brain works. The remarkable thing about some of these illusions is that even when you know things are not the way they seem, it is still difficult to perceive it in some other way. The subconscious mind is indeed powerful even when the conscious mind knows something to the contrary.

Our brains are constantly and subconsciously looking for patterns in order to make sense of the stream of information coming in from the senses. In fact, the subconscious can be thought of as a filter that only allows things out of the ordinary that are not predictable to pass into consciousness (Sections [7.9.3](#) and [10.3.1](#)). In many cases, these predictable patterns are so hardwired into the brain that we perceive these patterns even when they do not exist in reality.

Perhaps the earliest philosophical discussion of perception and the illusion of reality created by patterns can be found within Plato's *The Republic* (Plato, ~350 BC). Plato used the analogy of a person facing the back of a cave that is alive with shadows, with the shadows being the only basis for inferring what real objects are. Plato's cave was actually a motivation that revolutionized the VR industry back in the 1990s with the creation of the CAVE VR system [[Cruz et al. 1992](#)] that projects light onto walls surrounding the user, creating an immersive VR experience (Section [3.2](#)).

In addition to helping design VR systems, knowledge of perceptual illusions can help VR creators better understand human perception, identify and test assumptions, interpret experimental results, better design worlds, and solve problems as issues are discovered. The following sections demonstrate some virtual illusions that serve as building blocks for further investigation into perception.

6.2.1 2D Illusions

Even for simple 2D shapes we don't necessarily perceive the truth. The Jastrow illusion demonstrates how size can be misinterpreted—look at Figure [6.3](#) and try to determine which shape is larger. It turns out neither shape is larger—measure the height and width of the two shapes and you will find the shapes are identical, although the lower one appears to be larger.

The Hering illusion (Figure [6.4](#)) is a geometrical perceptual illusion that demonstrates straight lines do not always appear to be straight. When two straight and parallel lines are presented in front of a radial background, the lines convincingly appear as if they are bowed outward.

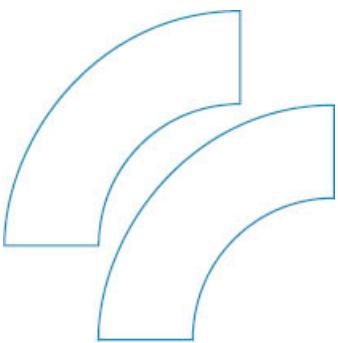


Figure 6.3 The Jastrow illusion. The two shapes are identical in size.

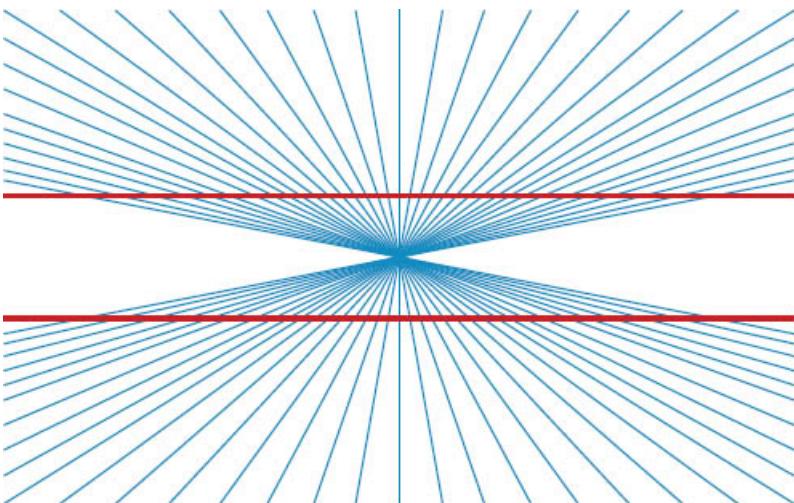


Figure 6.4 The Hering illusion. The two horizontal lines are perfectly parallel although they appear bent. (Courtesy of Fibonacci)

These simple 2D illusions show we should not necessarily trust our perceptions alone (although that is a good place to start) when calibrating VR systems, as judgments can depend upon the surrounding context.

6.2.2 Illusory Boundary Completion

Figure 6.5 shows three images, each with three “Pac Man” shapes similar to the classic video game of the 1980s. Between the three Pac Men in the leftmost figure, you might see a large white triangle overlaid on another slightly darker upside-down triangle. The perception of a triangle is known as the Kanizsa illusion, a form of **illusory contour** where boundaries are perceived even when no boundary actually exists. If you cover up these Pac Men then the supposed edges and triangles disappear. The middle figure shows how a triangle might appear in the corner of a rendered room. The rightmost figure shows how such a non-existent shape can even appear in the context of the real world.

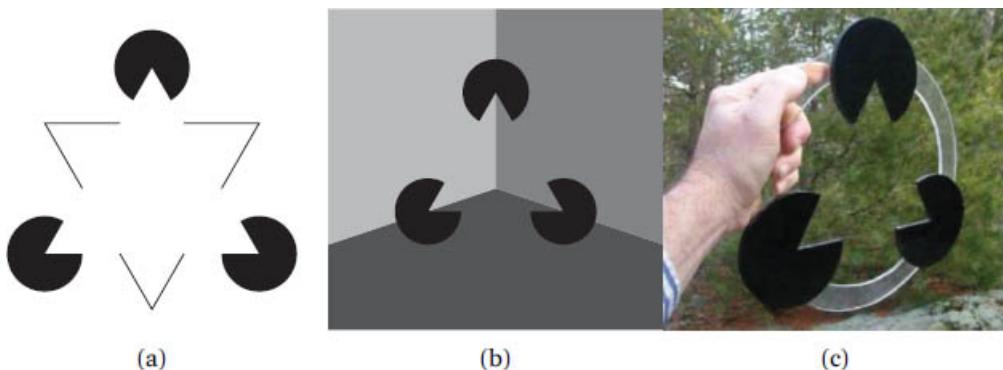
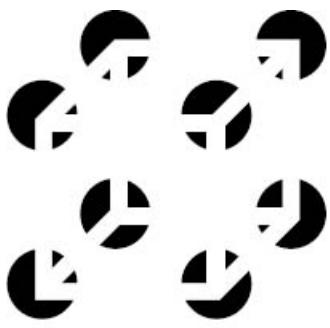
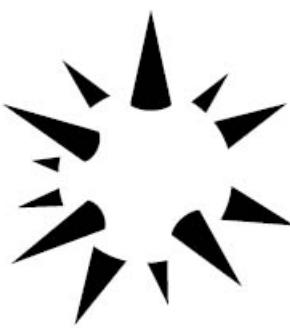


Figure 6.5 The Kanizsa illusion in (a) its traditional form, (b) 3D form (from [Guardini and Gamberini \[2007\]](#)), (c) real-world form (from [Lehar \[2007\]](#)). Triangles are perceived although they do not actually exist.



(a)



(b)

Figure 6.6 Necker cube (a) and a spiky ball illusion (b). Similar to the Kanizsa illusion, shape is seen where shape does not exist. (Based on [Lehar \[2007\]](#))

Figure 6.6 shows two more illusory contour images where our minds fill in the gaps resulting in perceived 3D shapes where those shapes don't actually exist. Gestalt theory states that the whole is other than the sum of its individual parts and is described in Section 20.4.

Illusory contours demonstrate that we do not need to perfectly replicate reality, but we do need to present the most essential stimuli to induce objects into users' minds.

6.2.3 Blind Spot

The mind can also perceive emptiness even when something is right in front of us. The **blind spot** is an area on the retina where blood vessels leave the eye and there are no light-sensitive receptors. Blind spots are so large one can make a person's head disappear when they are sitting across the room [[Gregory 1997](#)]. We can see (or not see!) the effect of the blind spot by a simple diagram as demonstrated in Figure 6.7.

6.2.4 Depth Illusions

The Ponzo Railroad Illusion

Figure 6.8 shows the Ponzo railroad illusion where the top yellow rectangle appears to be larger than the bottom yellow rectangle. In actuality the two yellow lines/rectangles are identical in shape (try measuring them for yourself). This occurs because we interpret the converging sides according to linear perspective as parallel lines receding into the distance (Section 9.1.3.). In this context, we interpret the upper yellow rectangle as though it were farther away, so we see it as larger—a farther object would have to be larger in actual physical size than a nearer one for both to produce retinal images of the same size.



Figure 6.7 A simple demonstration of the blind spot. Close your right eye. With your left eye stare at the red dot and slowly move closer to the image. Notice that the two blue bars are now one!



Figure 6.8 The Ponzo railroad illusion. The top yellow rectangle looks larger than the bottom yellow rectangle even though in actuality they are the same size.



Figure 6.9 An Ames room creates the illusion of dramatically different sized people. Even though we logically know such differences in human sizes are not possible, the scene is still convincing. (Courtesy of PETER ENDIG/AFP/Getty Images)

The Ames Room

In an Ames room such as that shown in Figure 6.9, a person standing in one corner appears to the observer to be a giant, while a person standing in the other corner appears to be impossibly small. The illusion is so convincing that a person walking back and forth from the left corner to the right corner appears to grow or shrink.

An Ames room is designed so that from a specific viewpoint, the room appears to be an ordinary room. However, this is a trick of perspective and the true shape of the room is trapezoidal: the walls are slanted and the ceiling and floor are at an incline, and the right corner is much closer to the viewer than the left corner (Figure 6.10). The resulting illusion shows our perceptions can be affected by surrounding context. The illusion also occurs because the room is viewed with one eye through a pinhole to avoid any clues from stereopsis or motion parallax. When creating VR worlds, we want to build with consistency across sensory modalities as misperceptions can occur if we are not careful. Otherwise much stranger things than the Ames room illusion can occur!

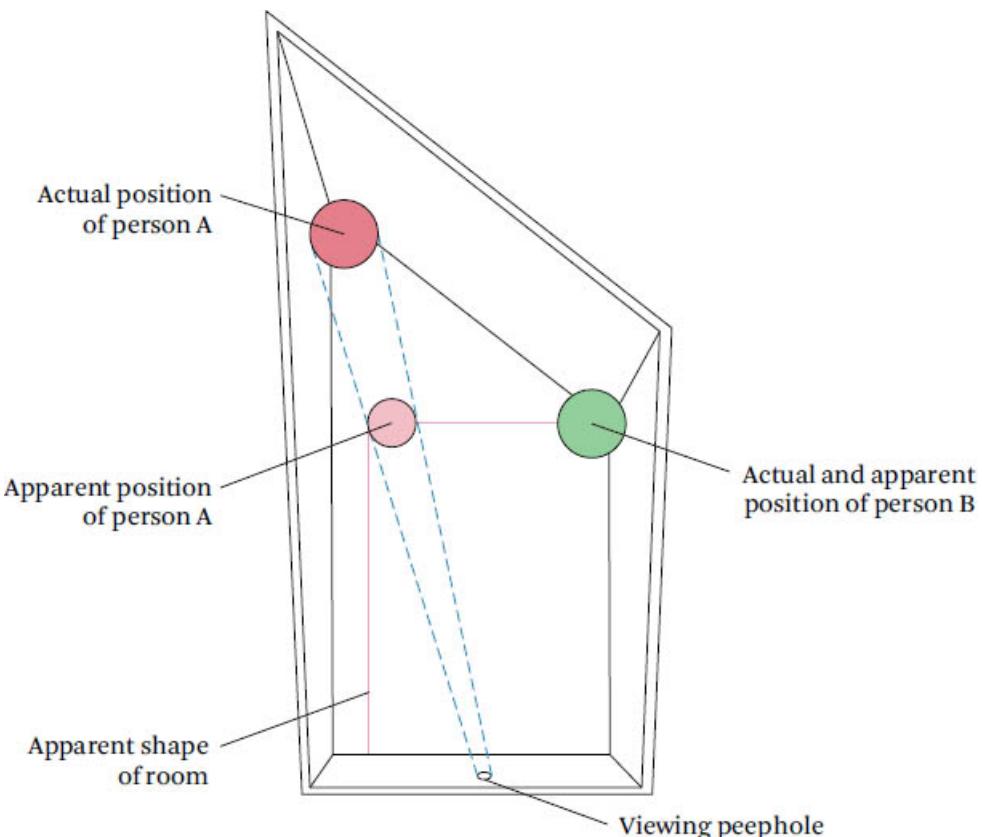


Figure 6.10 The actual shape of an Ames room.

6.2.5 The Moon Illusion

The moon illusion is the phenomenon that the moon appears larger when it is on the horizon than when it is high in the sky even though it remains the same distance from the observer and consists of the same visual angle. There are several explanations to the moon illusion (Figure 6.11), but the most likely is because when the moon is near the horizon there are other features on the terrain that can be directly compared to the moon. Because of the distance cues of the terrain and horizon we have a better sense that the moon is further away. The more distance cues, the stronger the illusion. Similar to the Ponzo railroad illusion (Section 6.2.4), the moon that is thought of as further away seems larger than the zenith moon (where the distance to the sky is perceived to be closer than the horizon) even though both moons take up the same area on the retina.



Figure 6.11 The moon illusion is the perception that the moon is larger when on the horizon than when high in the sky. The illusion can be especially strong when there are strong depth cues such as occlusion shown here.

6.2.6 Afterimages

A **negative afterimage** is an illusion where the eyes continue seeing the inverse colors of an image even after the image is no longer physically present. An afterimage can be demonstrated by staring at the small red, green, and yellow dots below the lady's eye in Figure 6.12. After staring at this dot for 30–60 seconds, look to the blank area to the right of the picture at the X. You should now see a realistically colored image of the lady. How does this occur? The photoreceptive cells in our eyes become less sensitive to a color after a period of time and when we then look at a white area (where white is a mix of all colors), the inhibited colors tend not to be seen, leaving the opposed color.

A **positive afterimage** is a shorter-term illusion that causes the same color to persist on the eye even after the original image is no longer present. This can result in the perception of multiple objects when a single object moves across a display (called strobining—see Section 9.3.6).

6.2.7 Motion Illusions

Motion illusions can certainly occur in VR, and such illusions can cause disorientation and sickness (Chapter 12). We can use motion illusions to our advantage to create a sense of self-motion calledvection (Section 9.3.10).

The Ouchi Illusion

If you look at the Ouchi illusion shown in Figure 6.13, you may see the circle move left and/or right. If you yaw your head left/right while looking at the circle, the circle may appear to be unstable in front of a stable background. There are many explanations for such motion illusions, and the effect has been found to occur in a wide range of situations. One explanation is that the two differently oriented bars are perceived to be on different depth planes, causing them to seem to move relative to each other as the eyes move (similar to what would occur if the bars were actually on different depth planes).



Figure 6.12 Afterimage effect. Stare at the small red, green, and yellow dots just below the eye for 30–60 seconds. Then look to the blank area to the right at the X. (Image © 2015 The Association for Research in Vision and Ophthalmology)

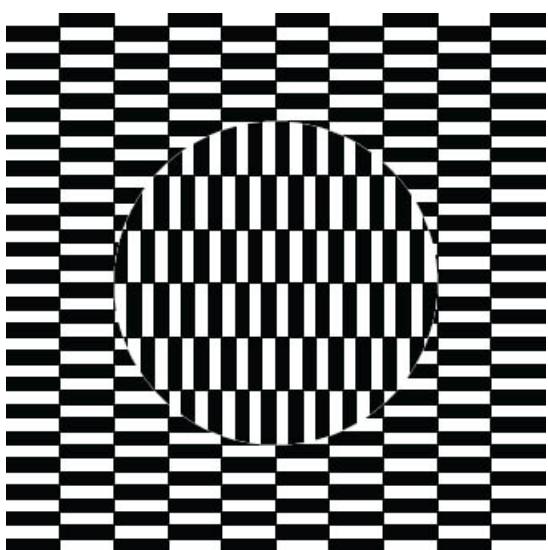


Figure 6.13 The Ouchi illusion. The rectangles appear to move although the image is static.

Motion Aftereffects

Motion aftereffects are illusions that occur after one views stimuli moving in the same direction for 30 seconds or more. After this time, one may perceive the motion to slow down or stop completely due to fatigue of motion detection neurons (a form of sensory adaptation—see Section [10.2](#)). When the motion stops or one looks away from the moving stimuli to non-moving stimuli, she may perceive motion in the opposite direction as the previously moving stimuli. We must therefore be careful when measuring perceptible motion or designing content for VR. For example, be careful of including consistently moving textures, such as moving water, in virtual worlds. Otherwise unintentional perceived motion may result after users look at such areas for long periods of time.

The Moon-Cloud Illusion

The moon-cloud illusion is an excellent example of induced motion—one may perceive clouds to be stable and the moon to be moving, when in fact the moon is stable and the clouds are moving. This illusion occurs because the mind assumes smaller objects are more likely to move than larger surrounding context (Section [9.3.5](#)).

The Autokinetic Effect

The **autokinetic effect**, also known as autokinesis, is the apparent movement of a single stable point-light when no other visual cues are present. This effect occurs even when the head is held still. This is because the brain is not directly aware of unintentional small and slow eye movements (Section [8.1.5](#)) and there is no surrounding visual context to judge motion (i.e., no object-relative motion cues are available—see Section [9.3.2](#)), so the movement is ambiguous; i.e., did the movement occur due to the eye moving or due to the point-light source moving? An observer does not know the answer, and the amount and direction of perceived motion varies. The autokinetic effect decreases as the size of the target increases, due to the brain's assumption that objects taking up a large field of view are stable.

The autokinetic effect suggests that in order to create the illusion of stable worlds, large surrounding context should be used (Section [12.3.4](#)).

Perceptual Models and Processes

The human brain is quite complex and cannot be described by a single model or process. This chapter discusses some models and processes that provide different perspectives about how the mind works. Note these concepts are not necessarily absolute “truths” but are simplified models that provide useful ways to think about perception. These concepts help explain various factors that affect human perception and behavior; enable us to create virtual environments that better communicate with a diverse set of users; help us to better program interaction into our environments to better respond to users on their terms; and help us induce more engaging and fulfilling experiences into the minds of users.

7.1 Distal and Proximal Stimuli

To understand perception, we must first understand the difference between the concepts of distal (distant) stimuli and proximal (approximate = close) stimuli.

Distal stimuli are actual objects and events out in the world (objective reality).

Proximal stimuli are the energy from distal stimuli that actually reach the senses (eyes, ears, skin, etc.).

The primary task of perception is to interpret distal stimuli in the world in order to guide effective survival-producing, goal-driven behaviors. One challenge is that a proximal stimulus is not always an ideal source of information about the distal stimulus. For example, light rays from distal objects form an image on the retina in the back of the eye. But this image continually changes as lighting varies, we move through the world, our eyes rotate, etc. Although this proximal image is what triggers the neural signals to the brain, we are often quite unaware of the overall image or pay little attention to it (Section [10.1](#)). Instead, we are aware of and respond to the distal objects that the proximal stimulus represents.

Factors that play a role in characterizing distal stimuli include context, prior knowledge, expectations, and other sensory input. Perceptual psychologists seek to understand how the brain extracts accurate stable perceptions of objects and events from limited and inadequate information.

Our job as VR creators is to create an artificial distal virtual world and then project that distal world onto the senses; the better we can do this in a way that the body and brain expects then the more users will feel present in the virtual world.

7.2 Sensation vs. Perception

Sensations are elementary processes that enable low-level recognition of distal stimuli through proximal stimuli. For example, in vision, sensation occurs as photons fall on the retina. In hearing, sensation occurs as waves of pulsating air are collected by the outer ear and are transmitted through the bones of the middle/inner ear to the cochlea.

Perception is a higher-level process that combines information from the senses and filters, organizes, and interprets those sensations to give meaning and to create subjective, conscious experiences. This gives us awareness of the world around us. Two people can receive the same sensory information yet perceive two different things. Moreover, even the same person can perceive the same stimulus differently (e.g., ambiguous figures as shown in Figure [6.1](#) or emotional responses when listening to the same song at different times).

7.2.1 Binding

Sensations of shapes, colors, motion, audio, proprioception, vestibular cues, and touch all cause independent firing of neurons in different parts of the cortex. **Binding** is the process by which stimuli are combined to create our conscious perception of a coherent object—a “red ball,” for example [[Goldstein 2014](#)]. We have integrated perceptions of objects, with all the object’s features bound together to create a single coherent perception of the object. Binding is also important to perceive multimodal events as coming from a single entity. However, sometimes we incorrectly assign features of one object to another object. An example is the ventriloquism effect that causes people to believe a sound comes from a source when it is actually coming from somewhere else (Section [8.7](#)). These illusory conjunctions most typically occur when some event happens quickly (such as errors in eyewitness testimony of a crime), or the scene is only briefly presented to the observer. Maintaining spatial and temporal compliance (Section [25.2.5](#)) across sensory modalities increases binding and is especially important for maximizing intuitive VR interactions as well as minimizing motion sickness (Section [12.3.1](#)).

7.3 Bottom-Up and Top-Down Processing

Bottom-up processing (also called data-based processing) is processing that is based on proximal stimuli [[Goldstein 2014](#)] that provides the starting point for perception. Tracking the perceptual process from sensory stimulation to transduction to neural processing is an example of bottom-up processing. VR uses bottom-up stimuli (e.g., pixels on the display) that is sufficiently convincing (i.e., to create presence) to overcome top-down data that suggests we are just wearing an HMD.

Top-down processing (also called knowledge-based or conceptually based processing) refers to processing that is based on knowledge—that is, the influence of an observer’s experiences and expectations on what he perceives. At any given time, an observer has a mental model (Section [7.8](#)) of the world. For example, we know through experience that object properties tend to be constant (Section [10.1](#)). Conversely, we modify what we expect based on situational properties or changeable aspects of the environment. The perception of a distal stimulus is heavily biased toward this internal model [[Goldstein 2007](#), [Gregory 1973](#)]. As stimuli become more complex, the role of top-down processing increases. Our

knowledge of how things are in the world plays an important role in determining what we perceive. As VR creators, we take advantage of top-down processing (e.g., the user's experience from the real or virtual world) that enables us to create interesting content and more engaging stories.

7.4 Afference and Efference

Afferent nerve impulses travel from sensory receptors inward toward the central nervous system. **Efferent nerve** impulses travel from the central nervous system outward toward effectors such as muscles. **Efference copy** sends a signal equal to the efference to an area of the brain that predicts afference. This prediction allows the central nervous system to initiate responses before sensory feedback occurs. The brain compares the efference copy with the incoming afferent signals. If the signals match, then this suggests the afference is solely due to the observer's actions. This is known as **re-afference**, since the afference reconfirms that the intended action has taken place. If the efference copy and afference do not match, then the observer perceives the stimulus to have occurred from a change in the external world (passive motion) instead of from the self (active motion).

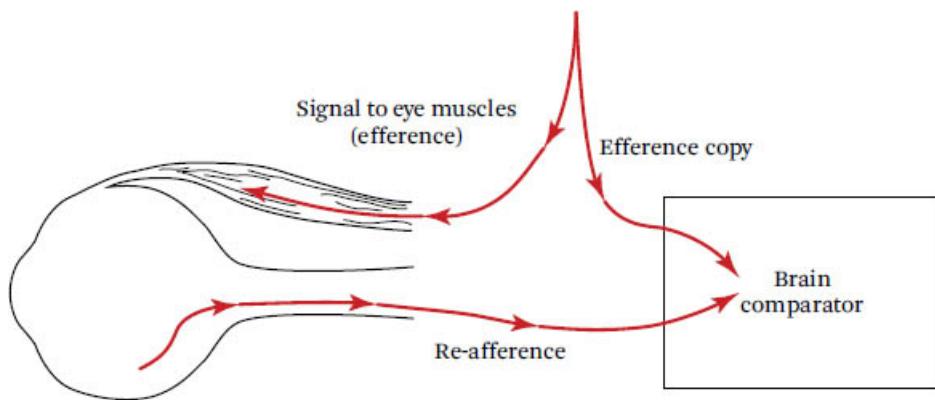


Figure 7.1 Efference copy during rotation of the eye. (Based on [Razzaque \[2005 \]](#), adapted from [Gregory \[1973 \]](#))

Afference and efference copy result in the world being perceived differently depending upon whether the stimulus is applied actively or passively [[Gregory 1973](#)]. For example, when eye muscles move the eyeball (efference), efference copy matches the expected image motion on the retina (re-afference), resulting in the perception of a visually stable world (Figure 7.1). However, if an external force moves the eyeball the world seems to visually move. This can be demonstrated by closing one eye and gently pushing the skin covering the open eye near the temple. The finger forcing the movement of the eye is considered to be passive with respect to the way the brain normally controls eye movement. The perceived visual shifting of the scene is due to the brain receiving no efference copy from the brain's command to move the eye muscles. The afference (the movement of the scene on the retina) is then greater than the zero efference copy, so the external world is perceived to move.

7.5 Iterative Perceptual Processing

In a simplified form, perception can be considered an ever-evolving sequence of processing steps, as shown in Figure 7.2.

At a high level, the three steps of Goldstein's **iterative perceptual process** [[Goldstein 2007](#)] are

1. stimulus to physiology—the external world falling onto our sensory receptors;
2. physiology to perception—our perceptual interpretation of those signals; and
3. perception to stimulus—our behavioral actions that influence the external world, resulting in an endless cycle that we call life.

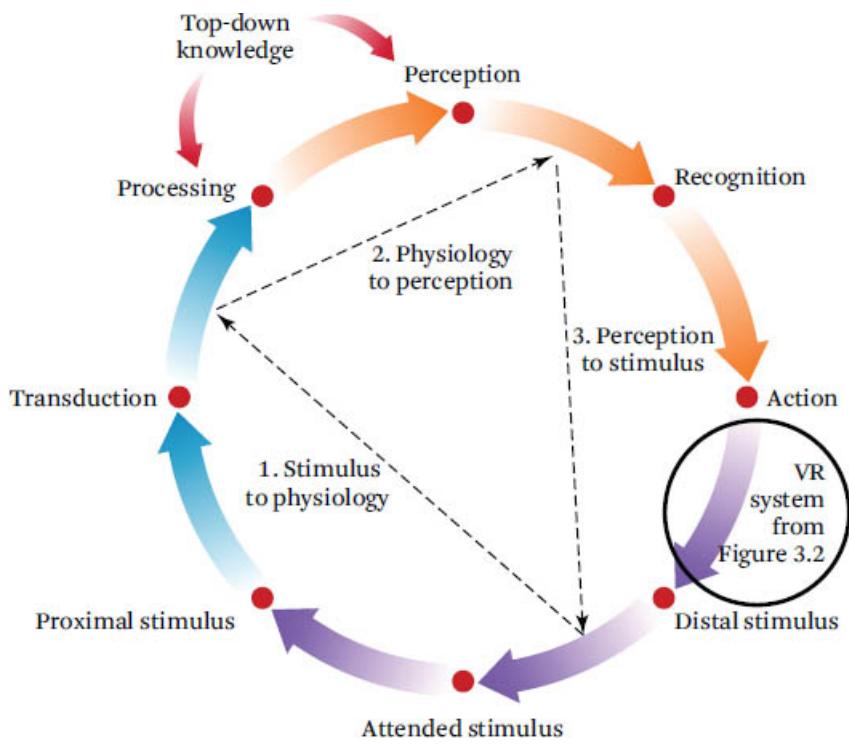


Figure 7.2 Perception is a continual process that is dynamic and continually changing. Purple arrows represent stimuli, blue arrows represent sensation, and orange arrows represent perception. (Adapted from [Goldstein \[2007\]](#))

These stages can be broken down further. As we journey through a world, we are confronted with a complex physical world (distal stimuli). Because there is far too much happening in the world, we scan the scene, looking at, listening to, or feeling things that catch our interest (attended stimuli). Photons, vibrations, and pressure falls onto our body (proximal stimuli). Our organs that sense such physical properties are transformed into electrochemical signals (transduction). These signals then go through complex neurological interactions (processing) that are influenced by our past experiences (top-down knowledge). This results in our conscious sensory experience (perception). We then are able to categorize or give meaning to those perceptions (recognition). This determines our behavior (action), which in turn affects distal stimuli, and the process continues.

For VR, we hijack real-world distal stimuli and replace those stimuli with computer-generated distal stimuli as a function of geometric models and algorithms. If we could somehow perfectly replicate the real world with our models and algorithms, then VR would be indistinguishable from reality.

7.6 The Subconscious and Conscious

The human mind can be thought as two different components—the subconscious and the conscious. Table 7.1 summarizes characteristics of the two [[Norman 2013](#)].

The **subconscious** is everything that occurs within our mind that we are not fully aware of but very much influences our emotions and behaviors. Because subconscious processing happens in parallel and without our awareness, the processing seems to occur naturally and automatically without effort. The subconscious is very much a pattern finder, generalizing incoming stimuli into the context of the past. Whereas the subconscious is powerful, it can also heavily bias us toward decisions based on false trends that cause us to behave inappropriately. The subconscious also is limited in the sense that it cannot arrange symbols in an intelligent fashion or make specific plans more than a couple of steps into the future.

When something comes into our awareness, it becomes part of our conscious. The **conscious** is the mind's totality of sensations, perceptions, ideas, attitudes, and feelings that we are aware of at any given time. The conscious is good at considering, pondering, analyzing, comparing, explaining, and rationalizing. However, the mind is slow as it primarily processes in a sequential manner and can become overwhelmed, resulting in analysis paralysis. The written word, mathematics, and programming are very much tools of the conscious.

Although the subconscious and conscious are each subject to mistakes, misunderstandings, and suboptimal solutions, the strengths of each can help balance disadvantages of the other. The integration of the two parts together often creates insight and innovation that would otherwise not be realized.

Table 7.1 Subconscious and conscious thought. (Based on [Norman \[2013\]](#))

Subconscious	Conscious
Fast	Slow
Automatic	Controlled

Multiple resources

Limited resources

Controls skilled behavior

Invoked for novel situations when learning, when in danger, when things go wrong

7.7 Visceral, Behavioral, Reflective, and Emotional Processes

A useful approximate model of human cognition and reaction can be categorized into four levels of processing: visceral, behavioral, reflective, and emotional [[Norman 2013](#)]. All four can be thought of as different forms of pattern recognition and prediction, and work together to determine a person's state. High-level cognitive reflection and emotion can trigger low-level visceral and physiological responses. Conversely, low-level visceral responses can trigger high-level reflective thinking.

7.7.1 Visceral Processing

Visceral processes, often referred to as “the lizard brain,” are tightly coupled with the motor system, serving as reflexive and protective mechanisms to help with immediate survival. Visceral communication (Section [1.2.1](#)) results in immediate reaction and judgments such as a situation or person being dangerous. Visceral reactions are immediate; there is no time to determine cause, or to blame, or even to feel emotions. Visceral responses are precursors to many emotions and can also be caused by emotions. Although high-level learning does not occur through visceral processes, sensitization and desensitization can occur through sensory adaptation (Section [10.2](#)).

Examples of visceral reactions include the startle reflex resulting from unexpected events, fight-or-flight responses, fear of heights, and the childhood anxiety produced by the lights being shut off. The visceral response to fear of heights is especially popular for demonstrating the power of VR (e.g., viewing the world from the top edge of a tall building).

Visceral processes are the initial reaction that is all about attraction or repulsion and have little to do with how usable, effective, or understandable a product or VR experience is. Many of us who consider ourselves “smart” tend to dismiss visceral responses as irrelevant and have trouble understanding the importance of an experience “feeling better.” Great designers use their aesthetic intuition to drive visceral responses that result in positive states and high affection from users.

7.7.2 Behavioral Processing

Behavioral processes are learned skills and intuitive interactions triggered by situations that match stored neural patterns, and are largely subconscious. Even though we are usually aware of our actions, we are often unaware of the details.

When we consciously act, all we have to do is think of the goal; then the body and mind handle most of the details with little awareness of those details. An example is when one picks up an object, the person first wills the action and then it happens—there is no need to consciously orient and position the hand. If a VR user wishes to grab an object in the world, she simply thinks of grabbing an object, the hand moves toward the object to intersect it, then pushes a button to pick it up (assuming she has learned how to use a tracked controller).

The desire to control the environment gives motivation to learning new behaviors. When control cannot be obtained or things do not go as planned, then frustration and anger often results. This is especially true when new useful behaviors cannot be learned due to not understanding the reasons why something doesn't work. Feedback (Section [25.2.4](#)) is critical to learning new interfaces, even if that feedback is negative, and VR applications should always provide some form of feedback.

7.7.3 Reflective Processing

Reflective processing ranges from basic conscious thinking to examining one's own thoughts and feelings. Higher-level reflection gives understanding and enables one to make logical decisions. Reflective processing is much slower than visceral and behavioral processing; reflective processing typically occurs after an event has happened. Indirect communication (Section [1.2.2](#)) such as spoken language or internal thoughts (i.e., speaking to oneself) is the tool of reflective processing. In contrast to visceral processing, reflective processing evaluates circumstances, assigns cause, and places blame.

Reflective processing is where we create stories within our minds (that might be far from what actually happened). Reflective stories and memories are often more important than what previously actually happened because the past no longer exists—what is more important is to predict and plan into the future based off the stories that we created. Reflective stories create the highest levels of emotion—highs and lows of remembered or anticipated events. Emotion and cognition are much more tightly intertwined than most people believe.

7.7.4 Emotional Processing

Emotional processing is the affective aspect of consciousness that powerfully processes data, resulting in physiological and psychological visceral and behavioral responses. This is done in a way that requires less effort and occurs faster than more reflective thinking—providing intuitive insight to assess a situation or whether a goal is desirable or not.

If a person did not have emotions, then making decisions would be difficult as emotion assigns subjective value and judgments, whereas rational

thinking assigns more objective understanding. Emotions are also more easily recalled than objective facts that are not associated with emotions. Logic is often used to backward-rationalize a decision that has already been made emotionally. Once a person has made an emotional decision, then some significant event or insight must occur for them to change that decision. People use emotion to pursue rational thought, such as being motivated to pursue the study of mathematics, art, or VR. Emotions can also be detrimental because they evolved in order to maximize immediate survival and reproduction rather than to achieve longer-term goals that empower us to move beyond just survival.

The emotional connection at the end of a VR experience is often what is remembered, so if the end experience is good the memories may be positive—but only if the user can make it that far. Conversely, one bad initial exposure can be so bad that they may never want to experience that application again, or in the worst case never want to experience any VR again.

7.8 Mental Models

A **mental model** is a simplified explanation in the mind of how the world or some specific aspect of the world works. The primary purpose of a mental model is prediction, and the best model is the simplest one that enables one to predict how a situation will turn out well enough to take appropriate action.

An analogy of a mental model is a physical map of the real world. The map is not the territory—meaning we form subjective representations (or maps) of objective reality (the territory), but the representation is not actually the reality, just as a map of a place is not actually the place itself but a more or less accurate representation [[Korzybski 1933](#)]. Most people confuse their maps of reality with reality itself. A person's interpretation/model created in the mind (the map) is not the truth (the territory), although it might be a simplified version of the truth. The value of a model is in the result that can be produced (e.g., an entertaining experience or learning new skills).

We all create mental models of ourselves, others, the environment, and objects we interact with. Different people often have different mental models of the same thing as these models are created through top-down processes that are derived from previous experience, training, and instruction that may not be consistent. In fact, an individual might have multiple models of the same thing, with each model dealing with a different aspect of its operation [[Norman 2013](#)]. These models might even be in conflict with each other.

A good mental model need not be complete or even accurate as long as it is useful. Ideally, complexity should be able to be intuitively understood with the simplest mental model possible that can achieve the desired results, but not simpler. An oversimplified model can be dangerous when implicit assumptions are not valid, in which case the model might be erroneous and therefore result in difficulties. Useful models serve as guides to help with understanding of a world, predict how objects in that world will behave, and interact with the world in order to achieve goals. Without such a mental model, we would fumble around mindlessly without the intelligence of fully appreciating why something works and what to expect from our actions. We could manage by following directions precisely, but when problems occur or an unexpected situation arises, then a good mental model is essential. VR creators should use signifying cues, feedback, and constraints (Section [25.1](#)) to help the user form quality mental models and to make assumptions explicit.

Learned helplessness is the decision that something cannot be done, at least by the person making the decision, resulting in giving up due to a perceived absence of control. It has been difficult for VR to gain wide adoption, and this can be compounded by poor interfaces. Bad interaction design for even a single task can quickly lead to learned helplessness—a couple of failures not only can erode confidence for the specific task but can generalize to the belief that interacting in VR is difficult. If users don't get feedback of at least a sign of minimal success, then they may get bored, not realizing they could do more, and mentally or physically eject themselves from the experience.

7.9 Neuro-Linguistic Programming

Neuro-linguistic programming (NLP) is a psychological approach to communication, personal development, and psychotherapy that is built on the concept of mental models (Section [7.8](#)). NLP itself is a model that explains how humans process stimuli that enter into the mind through the senses, and helps to explain how we perceive, communicate, learn, and behave. Its creators, Richard Bandler and John Grinder, claim NLP connects neurological processes ("neuro") to language ("linguistic") and our behaviors are the results of psychological "programs" built by experiences. Furthermore, they claim that skills of experts can be acquired by modeling the neuro-linguistic patterns of those experts. NLP is quite controversial and some of the claims/details are disputed. However, even if some of the specific details of NLP do not describe neurological processes accurately, some NLP high-level concepts are useful in thinking about how users interpret stimuli presented to them by a VR application.

Everyone perceives and acts in their own unique way, even when they are in the same situation. This is because humans manipulate pure sensory information depending upon their "program." Each individual's program is determined by the sum of one's entire life experiences, so each program is unique. With this being said, there are patterns to programs and we can make different people's experiences of an event more consistent if we understand the mind's programs. In the case of VR, we can control what stimuli are presented to users and influence, not control, what that person experiences. Understanding how sensory cues are integrated into the mind and precisely controlling stimuli enables us to more directly and consistently communicate and influence users' experiences. This understanding also helps us to change what is presented to users based on users' behaviors and personal preferences.

7.9.1 The NLP Communication Model

The NLP communication model describes how external stimuli affect our internal state and behavior. As shown in Figure [7.3](#), the NLP communication model [[James and Woodsmall 1988](#)] can be divided into three components: external stimuli, filters, and internal state. External stimuli enter through our senses and then are filtered in various ways before integrating with our internal state. The person then subconsciously and

consciously behaves in some way that potentially modifies external stimuli, creating a continuously repeating communication cycle.

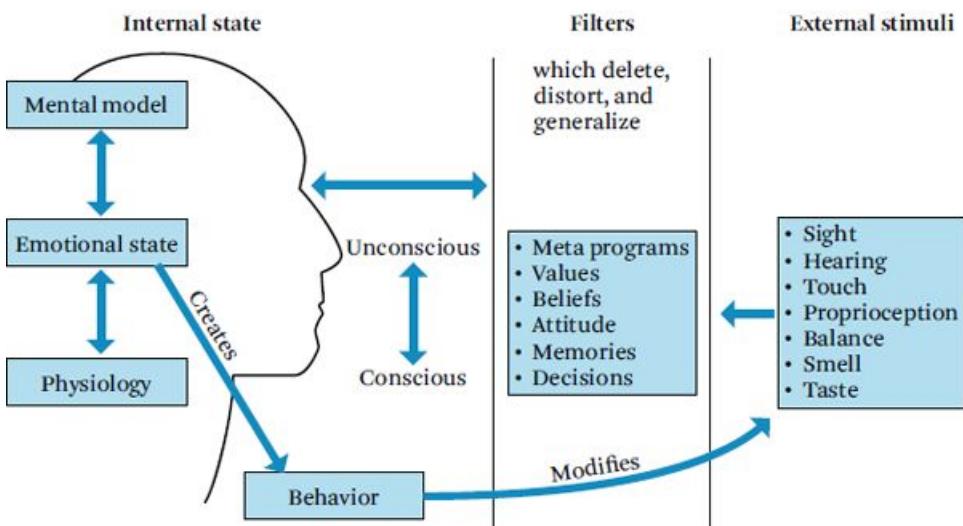


Figure 7.3 The NLP communication model states that external events are filtered as they enter into the mind, and then result in an individual's specific behavior. (Adapted from James and Woodsmall [1988])

7.9.2 External Stimuli

NLP asserts that humans perceive events and external stimuli through the sensory input channels. Furthermore, individuals have a dominant or preferred sensory modality. Not only do individuals perceive their preferred modality better but their primary method of thinking is in terms of that preferred modality. So a visual person thinks more with pictures than an auditory person, who thinks more with sounds. Even when external stimuli are in one form, they are often perceived partially in the preferred modality. For example, visually dominant individuals will react more to visual words that convey color or form. These concepts of preferred modalities are especially important for education as students learn in different ways. By providing sensory cues across all modalities, individuals are more likely to comprehend the concept being presented. For example, kinesthetic-oriented students learn best by doing rather than by watching or listening. One trap VR creators often fall into is that they focus on their preferred modality, when many users of their system may have a different preferred modality. VR creators should present sensory cues across all modalities to cover a wider range of users.

7.9.3 Filters

A vast majority of sensory information that we absorb is assimilated by our minds subconsciously. While our subconscious minds are able to absorb and store a large portion of what is happening around us, our conscious minds can only handle about 7 ± 2 "chunks" [[Csikszentmihalyi 2008](#)]. In addition, our minds interpret incoming data in different ways depending on the context and our past experiences in order to comprehend, discern, and utilize the complexity of the world. If our minds objectively accepted all information, then our lives would immediately become overwhelmed and we would not be able to sustain our individual selves or society as a whole. These processes can be thought of as perceptual filters that do three things: delete, distort, and generalize.

Deletion omits certain aspects of incoming sensory information by selectively paying attention to only parts of the world. For specific moments of time, we focus on what is most important and the rest is deleted from our conscious awareness in order to prevent becoming overwhelmed with information.

Distortion modifies incoming sensory information, enabling us to take into account the context of the environment and previous experiences. This is often beneficial but can be deceiving when something that happens in the world does not match one's mental model of the world. Being intimidated by technology, frightened of situations that we logically know are not real, or misinterpreting what occurred in a simulation are examples of how we distort incoming information. Categories of distortion include polarization (thinking of things in absolute extremes; e.g., the enemy in a game (or war) is wholly evil), mind reading (e.g., that avatar doesn't like me), and catastrophizing (e.g., I'm going to die if the goblin hits me).

Generalization is the process of drawing global conclusions based off of one or more experiences. A majority of the time generalization is beneficial. For example, a user that works through a tutorial that teaches how to open a door for the first time quickly generalizes their new ability so that all types of doors can be opened from then on. Designing VR experiences that are consistent will, over time, cause users to form filters that generalize perception of events and interactions. Generalization can also cause problems such as occurs with learned helplessness (Section [7.8](#)). For example, a user might generalize that memorizing VR controls is difficult after a couple of failed attempts.

The filters that delete, distort, and generalize vary from deeply unconscious processes to more conscious processes. Different people perceive things in different ways even when sensing the same thing. Listed below are descriptions of these filters in approximate order of increasing awareness.

Meta programs are the most unconscious filters and are content-free applying to all situations. Some consider meta programs to be one's "hardware" that individuals are born with and are difficult to change. Meta programs are similar to personality types. An example of a meta program is "optimism" where a person is generally optimistic regardless of the situation.

Values are filters that automatically judge if things are good or bad, or right or wrong. Ironically, values are context specific; therefore, what's important in one area of a person's life may not be important in other areas of his life. A player of a video game may value game characters very differently than how that person values real living people.

Beliefs are convictions about what is true or real in and about the world. Deep beliefs are assumptions that are not consciously realized. Weaker beliefs can be modified when brought to conscious thought and questioned by others. Beliefs can very much empower or hinder performance in VR (e.g., the belief "I'm not smart enough" can dramatically reduce performance).

Attitudes are values and belief systems about specific subjects. Attitudes are sometimes conscious and sometimes subconscious. An example of an attitude is "I don't like this part of the training."

Memories are past experiences that are reenacted consciously in the present moment. Individual and collective experiences influence current perceptions and behaviors. Designers can take advantage of memories by providing cues in the environment that remind users of some past event.

Decisions are conclusions or resolutions reached after reflection and consideration. Past decisions affect present decisions and people tend to stick with a decision once it is made. Past decisions are what create values, beliefs, and attitudes, and therefore influence how we make current decisions and respond to current situations. A person who decided they liked a first VR experience will certainly be more open to new VR experiences, whereas someone who had an initial bad VR experience will be more difficult to convince VR is the future, even with higher-quality subsequent experiences. Make sure users have a positive experience from the beginning of each session so that they are more likely to decide they like what they experience overall.

Although VR creators don't have control over users' filters, learning about the target user's general values, beliefs, attitudes, and memories can help in creating a VR experience. Section [31.10](#) discusses how to create personas to better target a VR application to users' wants and needs.

7.9.4 Internal State

As incoming information passes through our filters, thoughts are constructed in the form of a mental model called an internal representation. This internal representation takes the form of sensory perceptions (e.g., a visual scene with sounds, tastes, and smells) that may or may not exist as external stimuli. This model triggers and is triggered by emotional states, which in turn changes one's physiology and motivates behavior. Although internal representation and emotional state cannot be directly measured, physiological state and behavior can.

Perceptual Modalities

Mind-machine collaboration demands a two-way channel. The broadband path into the mind is via the eyes. It is not, however, the only path. The ears are especially good for situational awareness, monitoring alerts, sensing environmental changes, and speech. The haptic (feeling) and the olfactory systems seem to access deeper levels of consciousness. Our language is rich in metaphors suggesting this depth. We have “feelings” about complex cognitive situations on which we need to “get a handle” because we “smell” a rat.

—Frederick P. Brooks, Jr. [[2010](#)]

We interact with the world through sight, hearing, touch, proprioception, balance/motion, smell, and taste. This chapter discusses all of these with a focus on aspects of each that most relate to VR.

8.1 Sight

8.1.1 The Visual System

Light falls onto photoreceptors in the retina of the eye. These photoreceptors transduce photons into electrochemical signals that travel through different pathways in the brain.

The Photoreceptors: Cones and Rods

The **retina** is a multilayered network of neurons covering the inside back of the eye that processes photon input. The first retinal layer includes two types of photoreceptors—cones and rods. **Cones** are primarily responsible for vision during high levels of illumination, color vision, and detailed vision. The **fovea** is a small area in the center of the retina that contains only cones packed densely together. The fovea is located on the line of sight, so that when a person looks at something, its image falls on the fovea. **Rods** are primarily responsible for vision at low levels of illumination and are located across the retina everywhere except the fovea and blind spot. Rods are extremely sensitive in the dark but cannot resolve fine details. Figure 8.1 shows the distribution of rods and cones across the retina.

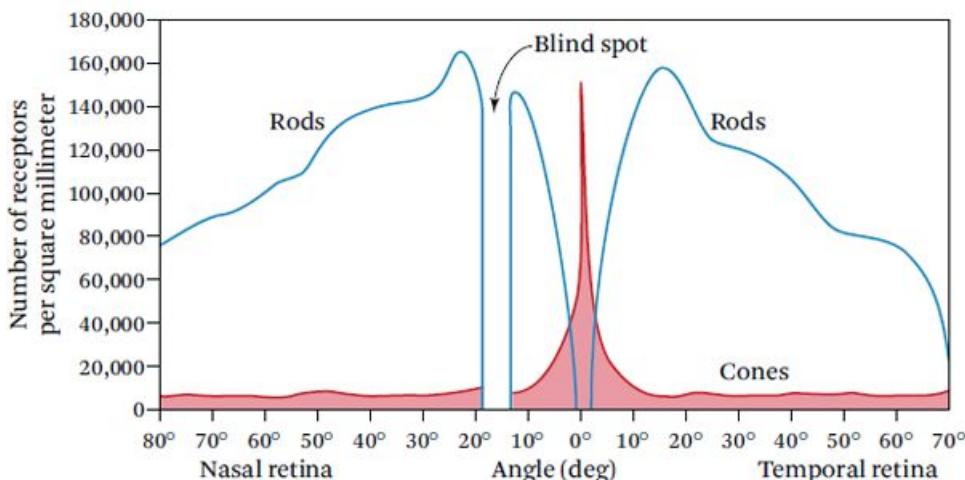


Figure 8.1 Distribution of rods and cones in the eye. (Based on [Coren et al. \[1999\]](#))

Electrochemical signals from multiple photoreceptors converge to single neurons in the retina. The number of converging signals per neuron increases toward the periphery, resulting in higher sensitivity to light but decreased visual acuity. In the fovea, some cones have a “private line” to structures deeper within the brain [[Goldstein 2007](#)].

Beyond Cones and Rods: Parvo and Magno Cells

Beyond the first layer of the retina, neurons relating to vision come in many shapes and sizes. Scientists categorize these neurons into parvo cells (those with smaller bodies) and magno cells (those with larger bodies). There are more parvo cells than magno cells with magno cells increasing in distribution toward peripheral vision whereas parvo cells decrease in distribution toward peripheral vision [[Coren et al. 1999](#)].

Parvo cells have a slower conduction rate (20 m/s) compared to magno cells, have a sustained response (continual neural activity as long as the stimulus remains), have a small receptive field (the area that influences the firing rate of the neuron), and are color sensitive. These qualities result in parvo cells being optimized for local shape, spatial analysis, color vision, and texture.

Magno cells are color blind, have a large receptive field, have a rapid conduction rate (40 m/s), and have a transient response (the neurons briefly fire when a change occurs and then stop responding). Magno cells are optimized for motion detection, time keeping operations/temporal analysis,

and depth perception. Magno neurons enable observers to quickly perceive large visuals before perceiving small details.

Multiple Visual Pathways

Signals from the retina travel along the optic nerve before diverging along different paths.

The primitive visual pathway. The **primitive visual pathway** (also known as the tectopulvinar system) starts to branch off at the end of the optic nerve (about 10% of retinal signals, mostly magno cells) toward the **superior colliculus**, a part of the brain just above the brain stem that is much older in evolutionary terms and a more primitive visual center than the cortex. The superior colliculus is highly sensitive to motion and is a primary cause of VR sickness (Part [III](#)) as it alters the response to the vestibular system, plays an important role in reflexive eye/neck movement and fixation, changes the curvature of the lens to bring objects into focus (eye accommodation), mediates postural adjustments to visual stimuli, induces nausea, and coordinates the diaphragm and abdominal muscles that evoke vomiting [[Goldstein 2014](#), [Coren et al. 1999](#), [Siegel and Sapru 2014](#), [Lackner 2014](#)].

The superior colliculus also receives input from the auditory and somatic (e.g., touch and proprioception) sensory systems as well as the visual cortex. Signals from the visual cortex are called **back projections** —a form of feedback from higher-level areas of the brain based on previous information that has already been processed (i.e., top-down processing).

The primary visual pathway. The **primary visual pathway** (also known as the geniculostriate system) goes through the **lateral geniculate nucleus** (LGN) in the thalamus. The LGN acts as a relay center to send visual signals to different parts of the visual cortex. The LGN is represented by central vision more than peripheral vision. In addition to receiving information from the eye, the LGN receives a large amount of information (as much as 80–90%) from higher visual centers in the cortex (back projections) [[Coren et al. 1999](#)]. The LGN also receives information from the reticular activating system, the part of the brain that helps to mediate increased alertness and attention (Section [10.3.1](#)). Thus, LGN processes are not solely a function of the eyes—visual processing is influenced by both bottom-up (from the retina) and top-down processing (from the reticular activating system and cortex) as described in Section [7.3](#). In fact, the LGN receives more information from the cortex than it sends out to the cortex. What we see is largely based on our experience and how we think.

Signals from the parvo and magno cells in the LGN feed into the visual cortex (and conversely back projections from the visual cortex feed into the LGN). After the visual cortex, signals diverge into two different pathways toward the temporal and parietal lobes. The path to the temporal lobe is known as the ventral or “what” pathway. The path to the parietal lobe is known as the dorsal or “where/how/action” pathway. Note the pathways are not totally independent and signals flow in both directions along each pathway.

The what pathway. The **ventral pathway** leads to the temporal lobe, which is responsible for recognizing and determining an objects’ identity. Those with damaged temporal lobes can see but have difficulty in counting dots in an array, recognizing new faces, and placing pictures in a sequence that relates a meaningful story or pattern [[Coren et al. 1999](#)]. Thus, the ventral pathway is commonly called the “what” pathway.

The where/how/action pathway. The **dorsal pathway** leads to the parietal lobe, which is responsible for determining an object’s location (as well as other responsibilities). Picking up an object requires knowing not only where the object is located but also where the hand is located and how it is moving. The parietal reach region at the end of this pathway is the area of the brain that controls both reaching and grasping [[Goldstein 2014](#)]. Thus, the dorsal pathway is commonly called the “where,” “how,” or “action” pathway. Section [10.4](#) discusses action, which is heavily influenced by the dorsal pathway.

Central vs. Peripheral Vision

Central and peripheral vision have different properties, due to not only the retina but also the different visual pathways described above. Central vision

- has high visual acuity,
- is optimized for bright daytime conditions, and
- is color-sensitive.

Peripheral vision

- is color insensitive,
- is more sensitive to light than central vision in dark conditions,
- is less sensitive to longer wavelengths (i.e., red),
- has fast response and is more sensitive to fast motion and flicker, and
- is less sensitive to slow motions.

Sensitivity to motion in central and peripheral vision is described further in Section [9.3.4](#).

Field of View and Field of Regard

Although we use central vision to see detail, peripheral vision is extremely important to function in real or virtual worlds. In fact, peripheral vision is so important that the US government defines those who cannot see more than 20° in the better eye as legally blind.

The **field of view** is the angular measure of what can be seen at a single point in time. As can be seen in Figure 8.2, a human eye has about a 160° horizontal field of view and both eyes can see the same area over an angle of about 120° when looking straight ahead [Badcock et al. 2014]. The total horizontal field of view when looking straight ahead is about 200°—thus we can see “behind” us by 10° on each side of the head! If we rotate our eyes to one side or another, we can see an additional 50° on each side—thus for an HMD to cover our entire visual potential range, we would need an HMD with 300° horizontal field of view! Of course we can also turn our bodies and head to see 360° in all directions. The measure for what can be seen by physically rotating the eyes, head, and body is known as **field of regard**. Fully immersive VR has the capability to provide a 360° horizontal and vertical field of regard.

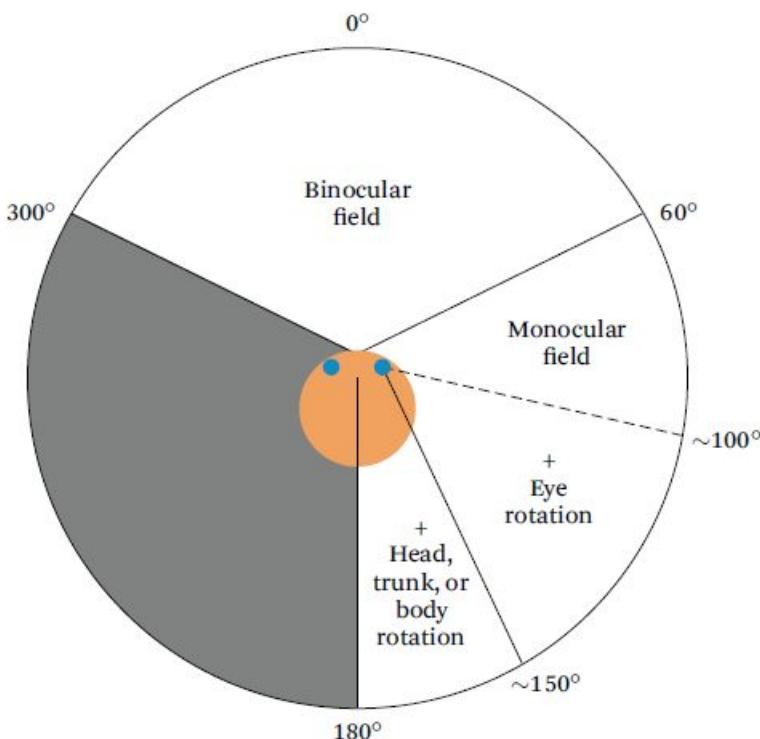


Figure 8.2 Horizontal field of view of the right eye with straight-ahead fixation (looking toward the top of diagram), maximum lateral eye rotation, and maximum lateral head rotation. (Based on Badcock et al. [2014].)

Our eyes cannot see nearly as much in the vertical direction due to our foreheads, torso, and the eyes not being vertically additive. We only see about 60° above due to the forehead getting in the way whereas we can see 75° below, for a total of 135° vertical field of view [Spector 1990].

8.1.2 Brightness and Lightness

Brightness is the apparent intensity of light that illuminates a region of the visual field. Under ideal conditions, a single photon can stimulate a rod and we can perceive as few as six photons hitting six rods [Coren et al. 1999]. Even at higher levels of illumination, fluctuations of only a few photons can affect brightness perception.

Brightness is not simply explained by the amount of light reaching the eye, but is also dependent on several other factors. Different wavelengths appear as different brightnesses (e.g., yellow light appears to be brighter than blue light). Peripheral vision is more sensitive than foveal vision (looking directly at a dim star can cause it to disappear) for all but longer wavelengths that appear as red in foveal vision. Dark-adapted vision provides six orders of magnitude more sensitivity than light-adapted vision (Section 10.2.1). Longer bursts of light (up to about 100 ms) are more easily detected than shorter bursts, and increasing stimulus sizes are more easily detected up to about 24°. Surrounding stimuli also affect brightness perception. Figure 8.3 shows how a light square can be made to appear darker by darkening the surrounding background.

Lightness (sometimes referred to as whiteness) is the apparent reflectance of a surface, with objects reflecting a small proportion of light appearing dark and objects reflecting a larger proportion of light appearing light/white. The lightness of an object is a function of much more than just the amount of light reaching the eyes. The perception that objects appear to maintain the same lightness even as the amount of light reaching the eyes changes is known as lightness constancy and is discussed in Section 10.1.1.

8.1.3 Color

Colors do not exist in the world outside of ourselves, but are created by our perceptual system. What exists in objective reality are different wavelengths of electromagnetic radiation. Although what we call colors is systematically related to light wavelengths, there is nothing intrinsically “blue” about short wavelengths or “red” about longer wavelengths.

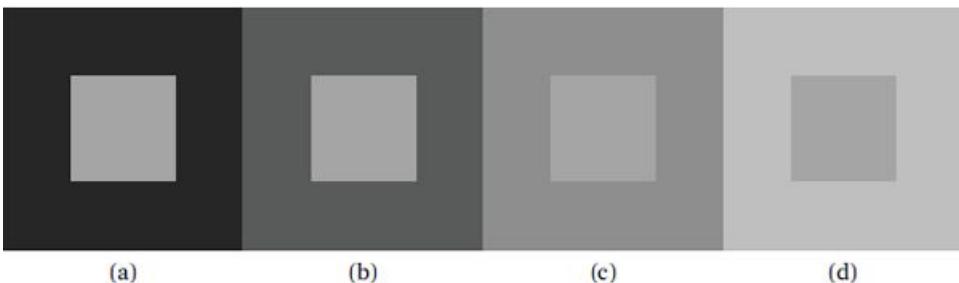


Figure 8.3 The luminance of surrounding stimuli affects our perception of brightness. All four squares have the same luminance, but (d) is perceived to be darkest. (Based on [Coren et al. \[1999 \]](#))

We perceive colors in wavelengths from about 360 nm (violet) to 830 nm (red) [[Pokorny and Smith 1986](#)], and bands of wavelengths within this range are associated with different colors. Colors of objects are largely determined by wavelengths of light that are reflected from the objects into our eyes, which contain three different cone visual pigments with different absorption spectra. Figure 8.4 shows plots of reflected light as a function of wavelength for some common objects. Black paper and white paper both reflect all wavelengths to some extent. When some objects reflect more of some wavelengths than other wavelengths, we call these chromatic colors or hues.

We can add more variation to hues by changing the intensity and by adding white to change the saturation (e.g., changing red to pink). **Color vision** is the ability to discriminate between stimuli of equal luminance on the basis of wavelength alone. Given equal intensity and saturation, people can discriminate between about 200 colors, but by changing the wavelength, intensity, and saturation, people can differentiate between about one million colors [[Goldstein 2014](#)]. Color discrimination ability decreases with luminance, especially at lower wavelengths, but remains fairly constant at higher luminance. Chromatic discrimination is poor at eccentricities beyond 8°.

Colors are more than just a direct effect of the physical variation of wavelengths—colors can subconsciously evoke our emotions and affect our decisions [[Coren et al. 1999](#)]. Colors can delight and impress. The color of detergent changes how consumers rate the strength of the detergent. The color of pills may affect whether patients take prescribed medications—black, gray, tan, or brown pills are rejected whereas blue, red, and yellow pills are preferred. Blue colors are described as cool whereas yellows tend to be described as warm. In fact, people turn a heat control to a higher setting in a blue room than in a yellow room. VR creators should be very cognizant of what colors they choose as arbitrary colors may result in unintended experiences.

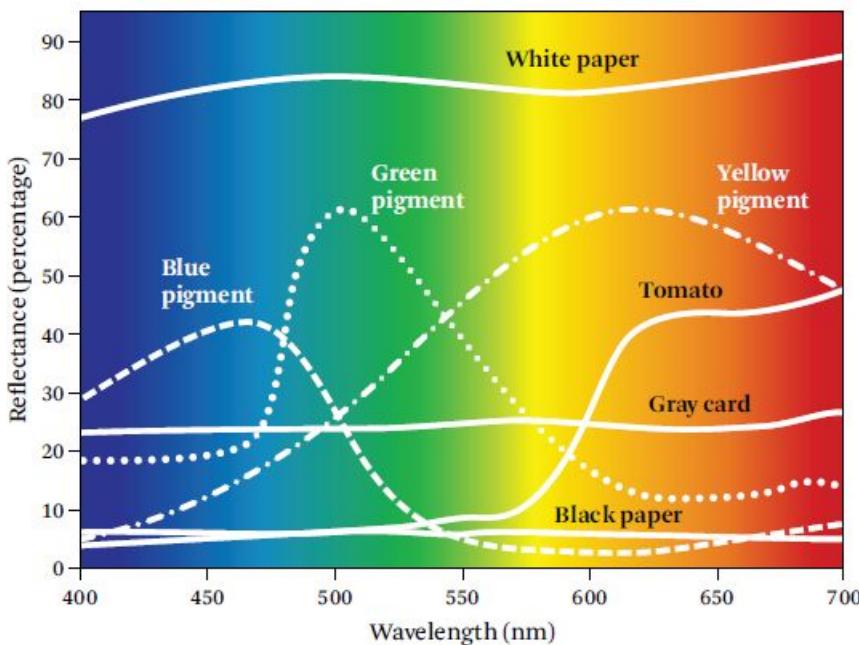


Figure 8.4 Reflectance curves for different colored surfaces. (Adapted from [Clulow \[1972 \]](#))

8.1.4 Visual Acuity

Visual acuity is the ability to resolve details and is often measured in visual angle. A “rule of thumb” is that the thumb or a quarter viewed at arm’s length subtends an angle of about 2° on the retina. A person with normal eyesight can see a quarter at 81 meters (nearly the length of a football field), which corresponds to 1 arc min (1/60th of a degree) [[Coren et al. 1999](#)]. Under ideal conditions, we can see a line as thin as 0.5 arc sec (1/7200th of a degree) [[Badcock et al. 2014](#)]!

Factors

Several factors affect visual acuity. As can be seen in Figure 8.5, acuity dramatically falls off with eye eccentricity. Note how visual acuity matches

quite well with the distribution of cones shown in Figure 8.1. We also have better visual acuity with illumination, high contrast, and long line segments. Visual acuity also depends upon the type of visual acuity that is being measured.

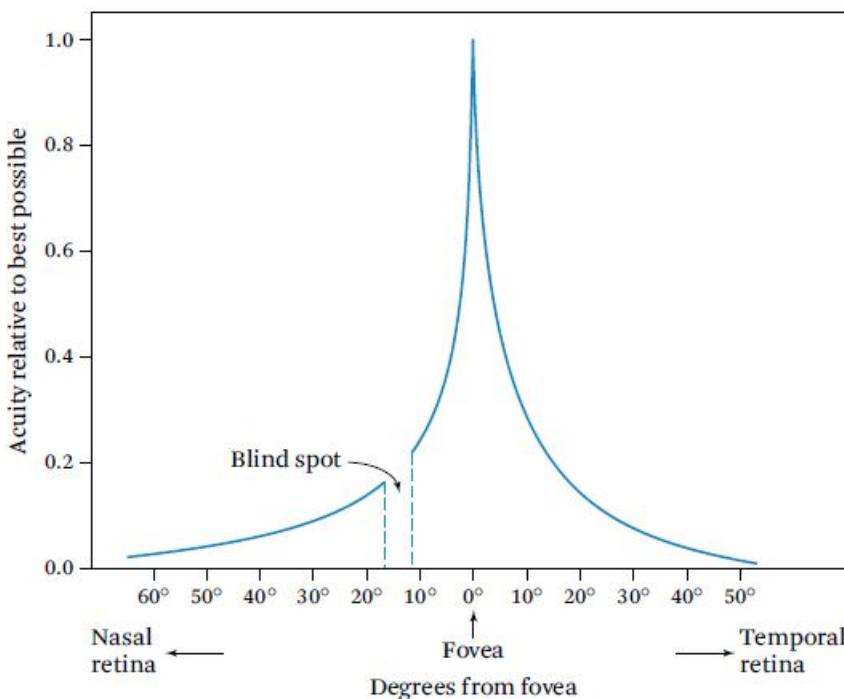


Figure 8.5 Visual acuity is greatest at the fovea. (Based on [Coren et al. \[1999 \]](#))

Types of Visual Acuity

There are different types of visual acuity: detection acuity, separation acuity, grating acuity, vernier acuity, recognition acuity, and stereoscopic acuity. Figure 8.6 shows some acuity targets used for measuring some of these visual acuities.

Detection acuity (also known as visible acuity) is the smallest stimulus that one can detect in an otherwise empty field and represents the absolute threshold of vision. Under ideal conditions a person can see a line as thin as 0.5 arc sec ($1/7200^\circ$ or 0.00014°) [[Badcock et al. 2014](#)]. Increasing the target size up to a point is equivalent to increasing its relative intensity. This is because the mechanism of detection is contrast, and for small objects, minimum visible acuity does not actually depend on the width of the line that cannot be discerned. As the line gets thinner, it appears to get fainter but not thinner.

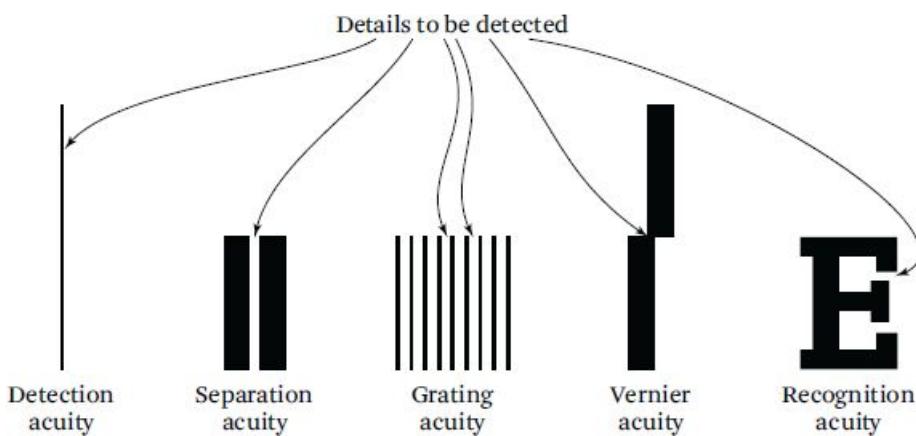


Figure 8.6 Typical acuity targets for different methods of measuring visual acuity. (Adapted from [Coren et al. 1999](#))

Separation acuity (also known as resolvable acuity or resolution acuity) is the smallest angular separation between neighboring stimuli that can be resolved, i.e., the two stimuli are perceived as two. More than 5,000 years ago, Egyptians measured separation acuity by the ability to resolve double stars [[Goldstein 2010](#)]. Today, minimum separation acuity is measured by the ability to resolve two black stripes separated by white space. Using this method, observers are able to see the separation of two lines with a cycle of 1 arc min.

Grating acuity is the ability to distinguish the elements of a fine grating composed of alternating dark and light stripes or squares. Grating acuity is similar to separation acuity, but thresholds are lower. Under ideal conditions, grating acuity is 30 arc sec [[Reichelt et al. 2010](#)].

Vernier acuity is the ability to perceive the misalignment of two line segments. Under optimal conditions vernier acuity is very good at 1–2 arc sec [[Badcock et al. 2014](#)].

Recognition acuity is the ability to recognize simple shapes or symbols such as letters. The most common method of measuring recognition acuity is to use the Snellen eye chart. The observer views the chart from a distance and is asked to identify letters on the chart, and the smallest recognizable letters determine acuity. The smallest letter on the chart is about 5 arc min at 6 m (20 ft); this is also the size of the average newsprint at a normal viewing distance. For the Snellen eye chart, acuity is measured relative to the performance of a normal observer. An acuity of 6/6 (20/20) means the observer is able to recognize letters at a distance of 6 m (20 ft) that a normal observer is able to recognize. An acuity of 6/9 (20/30) means the observer is able to recognize letters at 6 m (20 ft) that a normal observer can recognize at 9 m (30 ft), i.e., the observer cannot see as well.

Stereoscopic acuity is the ability to detect small differences in depth due to the binocular disparity between the two eyes (Section [9.1.3](#)). For complex stimuli, stereoscopic acuity is similar to monocular visual acuity. Under well-lit conditions, stereo acuity of 10 arc sec is assumed to be a reasonable value [[Reichert et al. 2010](#)]. For simpler targets such as vertical rods, stereoscopic acuity can be as good as 2 arc sec. Factors that affect stereoscopic acuity include spatial frequency, location on the retina (e.g., at an eccentricity of 2°, stereoscopic acuity decreases to 30 arc sec), and observation time (fast-changing scenes result in reduced acuity).

Stereopsis can result in better visual acuity than monocular vision. In one recent experiment, disparity-based depth perception was determined to occur beyond 40 meters when no other cues were present [[Badcock et al. 2014](#)]. Interestingly, the contribution of stereopsis to depth was considerably greater than that of motion parallax that occurred by moving the head.

8.1.5 Eye Movements

Six extraocular muscles control rotations of each eye around three axes. Eye movements can be classified in several different ways. They are categorized here as gaze-shifting, fixational, and gaze-stabilizing eye movements.

Gaze-Shifting Eye Movements

Gaze-shifting eye movements enable people to track moving objects to look at different objects.

Pursuit is the voluntary tracking with the eye of a visual target. The purpose of pursuit is to stabilize a target on the fovea, in order to maintain maximum resolution, and to prevent motion blur, as the brain is too slow to completely process all details of foveal images moving faster than a few degrees per second.

Saccades are fast voluntary or involuntary movements of the eye that allow different parts of the scene to fall on the fovea and are important for visual scanning (Section [10.3.2](#)). Saccades are the fastest moving external part of the body with speeds up to 1,000°/s and are ballistic where once started the destination cannot be changed [[Bridgeman et al. 1994](#)]. Durations are about 20–100 ms, with most being about 50 ms in duration [[Hallett 1986](#)], and amplitudes are up to 70°. Saccades typically occur at about three times per second.

Saccadic suppression greatly reduces vision during and just before saccades, effectively blinding observers. Although observers do not consciously notice this loss of vision, events can occur during these saccades and observers will not notice. A scene can rotate by 8–20% of eye rotation during these saccades without observers noticing [[Wallach 1987](#)]. If eye tracking were available in VR, then the system could perhaps (for the purposes of redirected walking; Section [28.3.1](#)) move the scene during saccades without users perceiving motion.

Vergence is the simultaneous rotation of both eyes in opposite directions in order to obtain or maintain binocular vision for objects at different depths. **Convergence** (the root con means “toward”) rotates the eyes toward each other. **Divergence** (the root di means “apart”) rotates the eyes away from each other in order to look further into the distance.

Fixational Eye Movements

Fixational eye movements enable people to maintain vision when holding the head still and looking in a single direction. These small movements keep rods and cones from becoming bleached. Humans do not consciously notice these small and involuntary eye movements, but without such eye movements the visual scene would fade into nothingness.

Small and quick movements of the eyes can be classified as microtremors (less than 1 arc min at 30–100 Hz) and microsaccades (about 5 arc min at variable rates) [[Hallett 1986](#)].

Ocular drift is slow movement of the eye, and the eye may drift as much as a degree without the observer noticing [[May and Badcock 2002](#)]. Involuntary drifts during attempts at steady fixation have a median extent of 2.5 arc min and have a speed of about 4 arc min per second [[Hallett 1986](#)]. In the dark, the drift rate is faster. Ocular drift plays an important part in the autokinetic illusion described in Section [6.2.7](#). As discussed there, this may play an important role in judgments of position constancy (Section [10.1.3](#)).

Gaze-Stabilizing Eye Movements

Gaze-stabilizing eye movements enable people to see objects clearly even as their heads move. Gaze-stabilizing eye movements play a role in position constancy adaptation (Section [10.2.2](#)), and the eye movement theory of motion sickness (Section [12.3.5](#)) states problems with gaze-stabilizing eye movements can cause motion sickness.

Retinal image slip is movement of the retina relative to a visual stimulus being viewed [[Stoffregen et al. 2002](#)]. The greatest potential source of retinal image slip is due to rotation of the head [[Robinson 1981](#)]. Two mechanisms work together to stabilize gaze direction as the head moves—the vestibulo-ocular reflex and the optokinetic reflex.

Vestibulo-ocular reflex. The **vestibulo-ocular reflex** (VOR) rotates the eyes as a function of vestibular input and occurs even in the dark with no visual stimuli. Eye rotations due to the VOR can reach smooth speeds up to 500°/s [[Hallett 1986](#)]. This fast reflex (4–14 ms from the onset of head motion) serves to keep retinal image slip in the low-frequency range to which the optokinetic reflex (described below) is sensitive; both work together to enable stable vision during head rotation [[Razzaque 2005](#)]. There are also proprioceptive eye-motion reflexes similar to VOR that use the neck, trunk, and even leg motion to help stabilize the eyes.

Optokinetic reflex. The **optokinetic reflex** (OKR) stabilizes retinal gaze direction as a function of visual input from the entire retina. If uniform motion of the visual scene occurs on the retina, then the eyes reflexively rotate to compensate. Eye rotations due to OKR can reach smooth speeds up to 80°/s [[Hallett 1986](#)].

Eye rotation gain. **Eye rotation gain** is the ratio of eye rotation velocity divided by head rotation velocity [[Hallett 1986](#), [Draper 1998](#)]. A gain of 1.0 means that as the head rotates to the right the eyes rotate by an equal amount to the left so that the eyes are looking in the same direction as they were at the start of the head turn. Gain due to the VOR alone (i.e., in the dark) is approximately 0.7. If the observer imagines a stable target in the world, gain increases to 0.95. If the observer imagines a target that turns with the head, then gain is suppressed to 0.2 or lower. Thus, VOR is not perfect, and OKR corrects for residual error. VOR is most effective at 1–7 Hz (e.g., VOR helps maintain fixation on objects while walking or running) and progressively less effective at lower frequencies, particularly below 0.1 Hz. OKR is most effective at frequencies below 0.1 Hz and has decreasing effectiveness at 0.1–1 Hz. Thus, the VOR and OKR complement each other for typical head motions—both VOR and OKR working together result in a gain close to 1 over a wide range of head motions.

Gain also depends on the distance to the visual target being viewed. For an object at an infinite distance, the eyes are looking straight ahead in parallel. In this case, gain is ideally equal to 1.0 so that the target image remains on the fovea. For a closer target, eye rotation must be greater than head rotation for the image to remain on the fovea. These differences in gains are because rotational axis for the head is different than the rotational axis of the eyes. The differences in gain can quickly be demonstrated by comparing eye movements while looking at a finger held in front of the eyes versus an object further in the distance.

Active vs. passive head and eye motion. Active head rotations by the observer result in more robust and more consistent VOR gains and phase lag than passive rotations by an external force (e.g., being automatically rotated in a chair) [[Draper 1998](#)]. Similarly, sensitivity to motion of a visual scene while the head moves depends upon whether the head is actively moved or passively moved [[Howard 1986a](#), [Draper 1998](#)]. Afference and efference help explain motion perception by taking into account active eye movements. See Section [7.4](#) for an explanation of afference and efference, and an explanation of how the world appears stable even as the eyes move.

Nystagmus. **Nystagmus** is a rhythmic and involuntary rotation of the eyes [[Howard 1986a](#)] caused by both VOR and OKR [[Razzaque 2005](#)] that can help stabilize gaze. Researchers typically discuss nystagmus caused by rotating the observer continuously at a constant angular velocity. The slow phase of nystagmus is the rotation of the eyes to keep the observer looking straight ahead in the world as the chair rotates. As the eyes reach their maximum amount of rotation in the eye sockets, a saccade snaps the eyes back to looking straight ahead relative to the head. This rotation is called the fast phase of nystagmus. This pattern repeats, resulting in a rhythmic rotation of the eyes. Nystagmus can also be demonstrated by looking at someone's eyes after they have become dizzy from spinning that stimulates their vestibular system [[Coren et al. 1999](#)].

Pendular nystagmus occurs when one rotates her head back and forth at a fixed frequency. This results in an always-changing slow phase with no fast phase. Pendular nystagmus occurs when walking and running. Those looking for latency in HMD often quickly rotate their head back and forth to try to estimate latency by seeing how the scene moves as they rotate their head. Little is known about how nystagmus works when perceiving such latency-induced scene motion in an HMD (see Chapter [15](#)). A user's eye gaze might remain stationary in space (because of the VOR) when looking for scene motion, resulting in retinal image slip, or might follow the scene (because of OKR), resulting in no retinal image slip. This likely varies by the amount of head motion, the type of task, the individual, etc. Eye tracking would allow an investigator to study this in detail. Subjects in one latency experiment [[Ellis et al. 2004](#)] claimed to concentrate on a single feature in the moving visual field. However, this was anecdotal evidence and was not verified.

8.1.6 Visual Displays

Given that each eye can see about 210° by rotating the eye and assuming vernier or stereoscopic acuity of 2 arc sec, then a display would require horizontal resolution of 378,000 pixels for each eye to match what we can see in reality! This number is a simple extreme analysis as this is for perfectly ideal situations and we can design around limitations for different circumstances. For example, since we do not perceive such high resolution outside of central vision, eye tracking could be used to only render at high resolution where the user is looking. This would require new algorithms, nonuniform display hardware, and fast eye tracking. Clearly, there is plenty of work to be done to get to the point of truly simulating visual reality. Latency will be even more of a concern in this case, as the system will be racing the eye to display at high resolution by the time the user is looking at a specific location.

8.2 Hearing

Auditory perception is quite complex and is affected by head pose, physiology, expectation, and its relationship to other sensory modality cues. We can deduce qualities of the environment from sound (e.g., large rooms sound different than small rooms), and we can determine where an

object is located by its sound alone. Below are discussed general concepts of sound, and Section 21.3 discusses audio as it applies specifically to VR.

8.2.1 Properties of Sound

Sound can be broken down into two distinct parts: the physical aspects and the perceptual aspects.

Physical Aspects

A person need not be present for a tree falling in the woods to make a sound. Physical sound is a pressure wave in a material medium (i.e., air) created when an object vibrates rapidly back and forth. **Sound frequency** is the number of cycles per second (hertz (Hz)) or vibrations that a change in pressure repeats. **Sound amplitude** is the difference in pressure between the high and low peaks of the sound wave. The decibel (dB) is a logarithmic transformation of sound amplitude where doubling the sound amplitude results in an increase of 3 dB.

Perceptual Aspects

Physical sound enters the ear, the eardrum is stimulated, and then receptor cells transduce those sound vibrations into electrical signals. The brain then processes these electrical signals into qualities of sound such as loudness, pitch, and timbre.

Loudness is the perceptual quality most closely related to the amplitude of a sound, although frequency can affect loudness. A 10 dB increase (a bit over three doublings of amplitude) results in approximately twice the subjective loudness.

Pitch is most closely related to the physical property of fundamental frequency. Low frequencies are associated with low pitches and high frequencies are associated with high pitches. Most real-world sounds are not strictly periodic, as they do not have repeated temporal patterns but have fluctuations from one cycle to the next. The perceived pitch of such imperfect sound patterns is the average period of the cyclical variations. Other sounds such as noise are non-periodic and do not have a perceived pitch.

Timbre is closely related to the harmonic structure (both strength of the harmonics and number of harmonics) of a sound and is the quality that distinguishes between two tones that have the same loudness, pitch, and duration, but sound different. For example, a guitar has more high-frequency harmonics than either the bassoon or alto saxophone [[Goldstein 2014](#)]. Timbre also depends on the attack (the buildup of sound at the beginning of the tone) and the tone's decay (the decrease in sound at the end of the tone). Playing a recording of a piano backwards sounds more like an organ because the original decay has become the attack and the attack has become the decay.

Auditory Thresholds

Humans can hear frequencies from about 20 to 22,000 Hz [[Vorlander and Shinn-Cunningham 2014](#)] with the most sensitivity occurring at about 2,000–4,000 Hz, which is the range of frequencies that is most important for understanding speech [[Goldstein 2014](#)]. We hear sound in a wide range of about a factor of a trillion (120 dB) before the onset of pain begins. Most sounds encountered in everyday experience span a dynamic intensity range of 80–90 dB. Figure 8.7 shows what amplitudes and frequencies we can hear and how perceived volume is significantly dependent upon frequency—lower- and higher-frequency sounds require greater amplitude than midrange frequencies for us to perceive equal loudness. This is largely due to the outer ear's auditory canal reinforcing/amplifying midrange frequencies. Interestingly, we only perceive musical melodies for pitches below 5,000 Hz [[Atneave and Olson 1971](#)].

The auditory channel is much more sensitive to temporal variations than both vision and proprioception. For example, amplitude fluctuations can be detected at 50 Hz [[Yost 2006](#)] and temporal fluctuations can be detected at up to 1,000 Hz. Not only can listeners detect rapid fluctuations but they can react quickly. Reaction times to auditory stimuli are faster than visual reaction times by 30–40 ms [[Welch and Warren 1986](#)].

8.2.2 Binaural Cues

Binaural cues (also known as stereophonic cues) are two different audio cues, one for each ear, that help to determine the position of sounds. Each ear hears a slightly different sound—different in time and different in level. Interaural time differences provide an effective cue for localizing low-frequency sounds. Interaural time differences as small as ~10 ms can be discerned [[Klumpp 1956](#)]. Interaural level differences (called acoustic shadows) occur due to acoustic energy being reflected and diffracted by the head, providing an effective cue for localizing sounds above 2 kHz [[Bowman et al. 2004](#)].

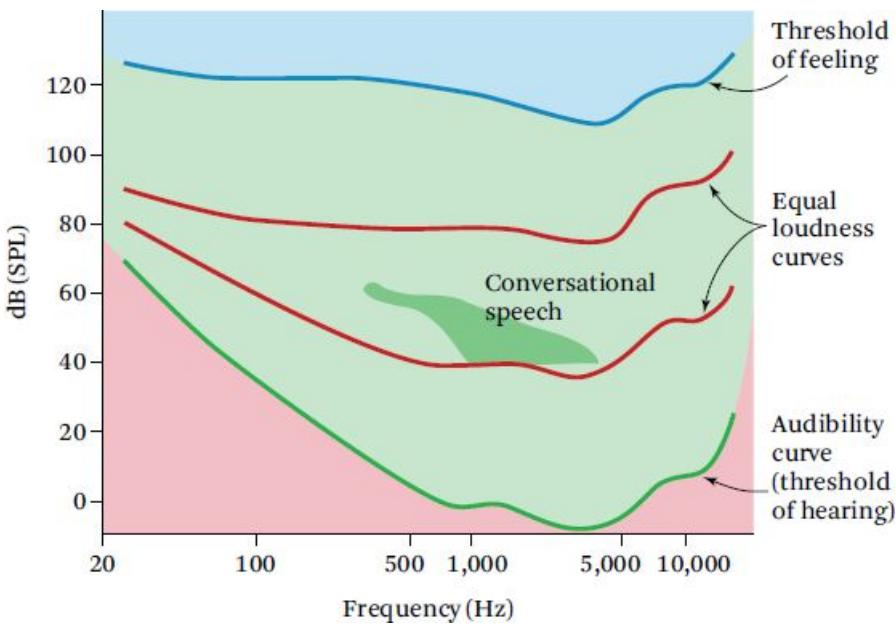


Figure 8.7 The audibility curve and auditory response area. The area above the audibility curve represents volume and frequencies that we can hear. The area above the threshold of feeling can result in pain. (Adapted from [Goldstein \[2014 \]](#))

Monaural cues use differences in the distribution (or spectrum) of frequencies entering the ear (due to the shape of the ear) to help determine the position of sounds. This monaural cue is helpful for determining the elevation (up/down) direction of a sound where binaural cues are not helpful. Head motion also helps determine where a sound is coming from due to changes in interaural time differences, interaural level differences, and monaural spectral cues. See Section [21.3](#) for a discussion of head-related transfer functions.

Spatial acuity of the auditory system is not nearly as good as vision. We can detect differences of sound at about 1° in front of or behind us, but our sensitivity decreases to 10° when the sound is to the far left/right side of us and 15° when the sound is above/below us. Vision can play a role in locating sounds—see Section [8.7](#).

8.2.3 Speech Perception

A **phoneme** is the smallest perceptually distinct unit of sound in a language that helps to distinguish between similar-sounding words. Phonemes combine to form morphemes, words, and sentences. Different languages use different sounds, so the number of phonemes varies across languages. For example, American English has 47 phonemes, Hawaiian has 11 phonemes, and some African languages have up to 60 phonemes [[Goldstein 2014](#)].

A **morpheme** is a minimal grammatical unit of a language, with each morpheme constituting a word or meaningful part of a word that cannot be divided into smaller independent grammatical parts. A morpheme can be, but is not necessarily, a subset of a word (i.e., every word is comprised of one or more morphemes). When a morpheme stands by itself, it is considered a root because it has meaning on its own. When a morpheme depends on another morpheme to express an idea, it is an affix because it has a grammatical function (e.g., adding an *s* to the end of a word to make it plural). The more ways morphemes can be combined with other morphemes, the more productive that morpheme is. Phoneme and morpheme awareness is the ability to identify speech sounds, the vocal gestures from which words are constructed, when they are found in their natural context—spoken words.

Speech segmentation is the perception of individual words in a conversation even when the acoustic signal is continuous. Our perception of words is not solely based upon energy stimulating our receptors, but is heavily influenced by our experience with those sounds and relationships between those sounds. To someone listening to an unfamiliar foreign language, words seem to speed by in a single stream of sound.

It is easier to perceive phonemes in a meaningful context. [Warren \[1970 \]](#) had test subjects listen to the sentence “The state governors met with their respective legislatures convening in the capital city.” When the first *s* in “legislatures” was replaced with a cough sound, none of the subjects were able to state where in the sentence the cough occurred or that the *s* in “legislatures” was missing. This is called the **phonemic restoration effect**.

[Samuel \[1981 \]](#) used the phonemic restoration effect to show that speech perception is determined both by the acoustic signal (bottom-up processing) and by context that produces expectations in the listener (top-down processing). Samuel also found longer words increase the likelihood of the phonemic restoration effect. A similar effect occurs for meaningfulness of spoken words in a sentence and knowledge of the rules of grammar; it is easier to perceive spoken words when heard in the context of familiar grammatical sentences [[Miller and Isard 1963](#)].

8.3 Touch

When we touch something or we are touched, receptors in the skin provide information about what is happening to our skin and about the object contacting the skin. These receptors enable us to perceive information about small details, vibration, textures, shapes, and potentially damaging stimuli.

Although those who are deaf or blind can get along surprisingly well, those with a rare condition that results in losing the sensation of touch often suffer constant bruising, burns, and broken bones due to the absence of warnings provided by touch and pain. Losing the sense of touch also makes it difficult to interact with the environment. Without the feedback of touch, actions as simple as picking objects up or typing on a keyboard can be difficult. Unfortunately, touch is extremely challenging to implement in VR but by understanding how we perceive touch, we can at least take advantage of providing some simple cues to users.

Adults have 1.3–1.7 square m (14–18 sq ft) of skin. However, the brain does not consider all skin to be created equally. Humans are very dependent on speech and manipulation of objects through the use of the hands, thus we have large amounts of our brain devoted to the hands. The right side of Figure 8.8 shows the sensory homunculus (“little man” in Latin), which represents the location and proportion of sensory cortex devoted to different body parts. The proportion of sensory cortex devoted to each body part also correlates with the density of tactile receptors on that body part. The left side shows the motor homunculus that helps to plan and execute movements. Some areas of the body are represented by a disproportionately large area of the sensory and motor cortices, notably the lips, tongue, and hands.

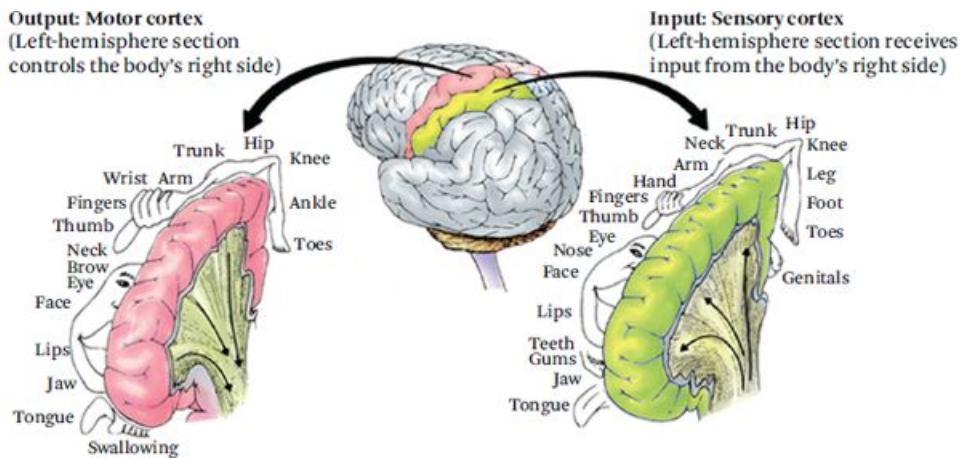


Figure 8.8 This motor and sensory homunculus represents “the body within the brain.” The size of a body part in the diagram represents the amount of cerebral cortex devoted to that body part. (Based on Burton [2012].)

8.3.1 Vibration

The skin is capable of detecting not only spatial details but vibrations as well. This is due to mechanoreceptors called Pacinian corpuscles. Nerve fibers located within corpuscles respond slowly to slow or constant pushing, but respond well to high vibrational frequencies.

8.3.2 Texture

Depending on vision for perceiving texture is not always sufficient, because seeing that texture is dependent on lighting. We can also perceive texture through touch. Spatial cues are provided by surface elements such as bumps and grooves, resulting in feelings of shape, size, and distribution of surface elements.

Temporal cues occur when the skin moves across a textured surface, and occur as a form of vibration. Fine textures are often only felt with a finger moving across the surface. Temporal cues are also important for feeling surfaces indirectly through the use of tools (e.g., dragging a stick across a rough surface); this is typically felt not as vibration of the tool but texture of the surface even though the finger is not touching the texture.

Most VR creators do not consider physical textures. Where they are considered is with passive haptics (Section 3.2.3), such as when building physical devices (e.g., hand-held controllers or a steering system) and real-world surfaces that users interact with while immersed.

8.3.3 Passive Touch vs. Active Touch

Passive touch occurs when stimuli are applied to the skin. Passive touch can be quite compelling in VR when combined with visuals. For example, rubbing a real feather on the skin of users when they see a virtual feather rubbing their physical body is used quite effectively to embody users into their avatars (Section 4.3).

Active touch occurs when a person actively explores an object, usually with the fingers and hands. Note passive and active touch is not to be confused with passive and active haptics as discussed in Section 3.2.3. Humans use three distinct systems together when using active touch.

- The sensory system, used in detecting cutaneous sensations such as touch, temperature, textures, and positions/movements of the fingers.
- The motor system, used in moving the fingers and hands.
- The cognitive system, used in thinking about the information provided by the sensory and motor systems.

These systems work together to create an experience that is quite different from passive touch. For passive touch, the sensation is typically experienced on the skin, whereas for active touch, we perceive the object being touched.

8.3.4 Pain

Pain functions to warn us of dangerous situations. Three types of pain are

- neuropathic pain caused by lesions, repetitive tasks (e.g., carpal tunnel syndrome), or damage to the nervous system (e.g., spinal cord injury or stroke);
- nociceptive pain caused by activation of receptors in the skin called nociceptors, which are specialized to respond to tissue damage or potential damage from heat, chemicals, pressure, and cold; and
- inflammatory pain caused by previous damage to tissue, inflammation of joints, or tumor cells.

The perception of pain is strongly affected by factors other than just stimulation of skin, such as expectation, attention, distracting stimuli, and hypnotic suggestion. One specific example is phantom limb pain, where individuals who have had a limb amputated continue to experience the limb, and in some cases continue to experience pain in a limb that does not exist [[Ramachandran and Hirstein 1998](#)]. An example of reducing pain using VR is Snow World, a distracting VR game set in a cold-conveying environment that doctors use when they have burn victims' bandages removed [[Hoffman 2004](#)].

8.4 Proprioception

Proprioception is the sensation of limb and whole body pose and motion derived from the receptors of muscles, tendons, and joint capsules. Proprioception enables us to touch our nose with a hand even when the eyes are closed. Proprioception includes both conscious and subconscious components, not only enabling us to sense the position and motion of our limbs, but also providing us the sensation of force generation enabling us to regulate force output.

Whereas touch seems straightforward because we are often aware of the sensations that result, proprioception is more mysterious because we are largely unaware of it and take it for granted during daily living. However, as VR creators, becoming familiar with the sense of proprioception is important for understanding how users physically move to interact with a virtual environment (at least until directly connected neural interfaces become common). This includes moving the head, eyes, limbs, and/or whole body. Without the senses of touch and proprioception, we would crush brittle objects when picking them up. Proprioception can be very useful for designing VR interactions as discussed in Section [26.2](#).

8.5 Balance and Physical Motion

The **vestibular system** consists of labyrinths in the inner ears that act as mechanical motion detectors (Figure [8.9](#)), which provide input for balance and sensing physical motion. The vestibular organs are composed of the otolith organs and the semicircular canals.

Each set (right and left) of two **otolith organs** acts as a three-axis accelerometer, measuring linear acceleration. Cessation of linear motion is sensed almost immediately by the otolith organs [[Howard 1986b](#)]. The nervous system's interpretation of signals from the otolith organs relies almost entirely on the direction and not the magnitude of acceleration [[Razzaque 2005](#)].

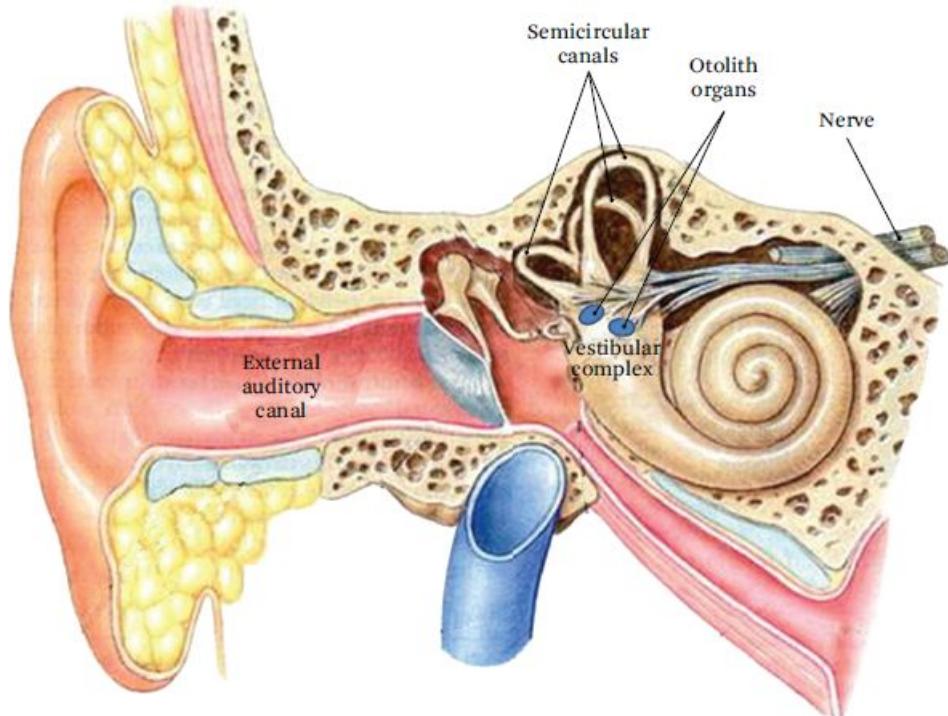


Figure 8.9 A cutaway illustration of the outer, middle, and inner ear, revealing the vestibular system. (Based on [Jerald \[2009\]](#), adapted from [Martini \[1998\]](#).)

Each set of the three nearly orthogonal **semicircular canals** (SCCs) acts as a three-axis gyroscope. The SCCs act primarily as gyroscopes measuring angular velocity, but only for a limited amount of time if velocity is constant. After 3–30 seconds the SCCs cannot disambiguate between no velocity and some constant velocity [[Razzaque 2005](#)]. In real and virtual worlds, head angular velocity is almost never constant other than zero velocity. The SCCs are most sensitive between 0.1 Hz and 5.0 Hz. Below 0.1 Hz, SCC output is roughly equal to angular acceleration, 0.1–5.0 Hz roughly equal to angular velocity, and above 5.0 Hz roughly equal to angular displacement [[Howard 1986b](#)]. The 0.1–5.0 Hz range fits well within typical head motions—head motions while walking (at least one foot always touching the ground) are in the 1–2 Hz range and 3–6 Hz while running (moments when neither foot is touching the ground) [[Draper 1998](#), [Razzaque 2005](#)].

Although we are not typically aware of the vestibular system in normal situations, we can become very aware of this sense in atypical situations when things go wrong. Understanding the vestibular system is extremely important for creating VR content as motion sickness can result when vestibular stimuli does not match stimuli from the other senses. The vestibular system and its relationship to motion sickness are discussed in more detail in Chapter [12](#).

8.6 Smell and Taste

Smell and taste both work through chemoreceptors, where the chemoreceptors detect chemical stimuli in the environment.

Smell (also known as olfactory perception) is the ability to perceive odors when odorant airborne molecules bind to specific sites on the olfactory receptors high in the nose. Very low concentrations of material can be detected by the olfactory system. The number of smells the olfactory system can distinguish between is highly debated with different researchers claiming a wide range from hundreds to trillions. Different people smell different odors due to genetic differences.

Taste (also known as gustatory perception) is the chemosensory sensation of substances on the tongue. The human tongue by itself can distinguish among five universal tastes: sweet, sour, bitter, salty, and umami.

Smell, taste, temperature, and texture all combine together to provide the perception of **flavor**. Combining these senses provides a much wider range of flavor than if taste alone determined flavor. In fact, a pseudo-gustatory VR system has been shown to fool users into thinking a plain cookie tasted differently by providing different visual and smell cues [[Miyaura et al. 2011](#)].

8.7 Multimodal Perceptions

Integration of our different senses occurs automatically and rarely does perception occur as a function of a single modality. Examining each of the senses as if it were independent of the others leads to only partial understanding of everyday perceptual experience [[Welch and Warren 1986](#)]. [Sherrington \[1920\]](#) states, “All parts of the nervous system are connected together and no part of it is probably ever capable of action without affecting and being affected by various other parts.”

Perception of a single modality can influence other modalities. A surprising example is if audio cues are not precisely synchronized with visual cues, then visual perception may be affected. For example, auditory stimulation can influence the visual flicker-fusion frequency (the frequency at which subjects start to notice the flashing of a visual stimulus; see Section [9.2.4](#)) [[Welch and Warren 1986](#)].

[Sekuler et al. \[1997 \]](#) found that when two moving shapes cross on a display, most subjects perceived these shapes as moving past each other and continuing their straight-line motion. When a “click” sounded just when the shapes appeared adjacent to each other, a majority of people perceived the shapes as colliding and bouncing off in opposite directions. This is congruent with what typically happens in the real world—when a sound occurs just as two moving objects become close to each other, a collision has likely occurred that caused that sound.

Perceiving speech is often a multisensory experience involving both audition and vision. **Lip sync** refers to the synchronization between the visual movement of the speaker’s lips and the spoken voice. Thresholds for perceiving speech synchronization vary depending on the complexity, congruency, and predictability of the audiovisual event as well as the context and applied experimental methodology [[Eg and Behne 2013](#)]. In general, we are more tolerant of visuals leading sound and thresholds increase with sound complexity (e.g., we notice asynchronies less for sentences than for syllables). One study found that when audio precedes video by 5 video fields (about 80 ms), viewers evaluated speakers more negatively (e.g., less interesting, more unpleasant, less influential, more agitated, and less successful) even when they did not consciously identify the asynchrony [[Reeves and Voelker 1993](#)].

The McGurk effect [[McGurk and MacDonald 1976](#)] illustrates how visual information can exert a strong influence on what we hear. For example, when a listener hears the sound /ba-ba/ but is viewing a person making lip movements for the sound /ga-ga/, the listener hears the sound /da-da/.

Vision tends to dominate other sensory modalities [[Posner et al. 1976](#)]. For example, vision dominates spatialized audio. **Visual capture** or the ventriloquism effect occurs when sounds coming from one location (e.g., the speakers in a movie theater) are mislocalized to seem to come from a place of visual motion (e.g., an actor’s mouth on the screen). This occurs due to the tendency to try to identify visual events or objects that could be causing the sound.

In many cases, when vision and proprioception disagree, we tend to perceive hand position to be where vision tells us it is [[Gibson 1933](#)]. Under certain conditions, VR users can be made to believe that their hand is touching a different seen shape than a felt shape [[Kohli 2013](#)]. Under other conditions, VR users are more sensitive to their virtual hand visually penetrating a virtual object than they are to the proprioceptive sense of their hand not being colocated with their visual hand in space [[Burns et al. 2006](#)]. Sensory substitution can be used to partially make up for the lack of physical touch in VR systems as discussed in Section [26.8](#).

Mismatch of visual and vestibular cues is a major problem of VR as it can cause motion sickness. Chapter [12](#) discusses such cue conflicts in detail.

8.7.1 Visual and Vestibular Cues Complement Each Other

The vestibular system provides primarily first-order approximations of angular velocity and linear acceleration, and positional drift occurs over time [[Howard 1986b](#)]. Thus, absolute position or orientation cannot be determined from vestibular cues alone. Visual and vestibular cues combine to enable people to disambiguate between moving stimuli and self-motion.

The visual system is good at capturing lower-frequency motions whereas the vestibular system is better at detecting high-frequency motions. The vestibular system is a mechanical system and has a faster response (as fast as 3–5 ms!) than the slower electrochemical visual response of the eyes. Thus, the vestibular system is initially more responsive to a sudden onset of physical motion. After some time of sustained constant velocity, vestibular cues subside and visual cues take over. Midrange frequencies use both vestibular and visual cues.

Missing or misleading vestibular cues can lead to life-threatening motion illusions in pilots [[Razzaque 2005](#)]. For example, the vestibular otolith organs cannot distinguish between linear acceleration and tilt. This ambiguity can be resolved from the semicircular canals when within the canals' sensitive range. However, when not within the sensitive range, the ambiguity is resolved from visual cues. Aircraft accidents, and loss of many lives, have occurred due to the ambiguity when visual cues are missing (e.g., low visibility conditions). This ambiguity can be used to our advantage in VR when motion platforms are available by tilting the platform to give a sense of forward acceleration that can be extremely compelling.

Perception of Space and Time

How do we perceive the layout, timing, and motion of objects in the world around us? This question is not new, as it has fascinated artists, philosophers, and psychologists for centuries. The success of painting, photography, cinema, video games, and now VR is largely contingent upon the convincing portrayal of spatial 3D relationships and their timings.

9.1 Space Perception

Trompe-l'œil art (Figure II.1) and the Ames room (Figure [6.9](#)) prove the eye can be fooled when it comes to space perception. However, these illusions are very constrained and only work when viewed from a specific viewpoint—when the observer moves and explores resulting in the perception of more information, then the illusion goes away and the truth becomes more apparent. It is relatively easy to fool observers from a single perspective, compared to fooling VR users into believing they are present in another world.

If the senses can be deceived to misinterpret space, then how can we be sure the senses do not deceive us all the time? Perhaps we cannot be 100% sure of the space around us. However, evolution tells us that our perceptions must be truthful to some degree—otherwise if our perceptions were overly deceived, we would have become extinct long ago [[Cutting and Vishton 1995](#)]. Perhaps deception of space is the goal of VR, and if we can someday completely overcome the challenges of VR, then we can defeat the argument from evolution, and hopefully not extinguish ourselves from existence in the process.

Multiple sensory modalities often work together to give us spatial location. Vision is the superior spatial modality as it is more precise, is faster for perceiving location (although audition is faster when the task is detection), is accurate for all three directions (whereas audition, for example, is reasonably good only in the lateral direction), and works well at long distances [[Welch and Warren 1986](#)].

Perceiving the space around us is not just about passive perception but is important for actively moving through the world (Section [10.4.3](#)) and interaction (Section [26.2](#)).

9.1.1 Exocentric and Egocentric Judgments

Exocentric judgments, also known as object-relative judgments, are the sense of where objects are relative to other objects, the world, or some other external reference. Exocentric judgments include the sense of gravity, geographic direction (e.g., north), and distance between two objects. Gogel's principle of adjacency states “The effectiveness of relative cues decreases as the perceived distance between objects producing the relative cues increases” [[Mack 1986](#)]. That is, objects are more difficult to compare the further they are apart.

The perception of the location of objects also includes the direction and distance relative to our bodies. **Egocentric judgments**, also known as subject-relative judgments, are the sense of where (direction and distance) cues are relative to the observer. Left/right and forward/backward judgments are egocentric. Egocentric judgments can be performed by the primary sensory modalities of vision, audition, proprioception, and touch.

Several variables affect our egocentric perception. The more stimuli available, the more stable our judgments; our judgments are much less stable in the dark as demonstrated by the autokinetic effect (Section [6.2.7](#)). A stimulus that is not centered in the visual field (e.g., an offset square) can result in the straight-ahead being perceived as biased toward that off-centered stimulus. The **dominant eye**, the eye that is used for sighting tasks, also influences direction judgments. Eye movement (Section [8.1.5](#)) also affect our judgment of the straight-ahead.

Reference frames of exocentric and egocentric space as they relate to VR are discussed in Section [26.3](#). Such reference frames are especially important for designing VR interfaces.

9.1.2 Segmenting the Space around Us

The layout of the space around each of us can be divided into three circular egocentric regions: personal space, action space, and vista space [[Cutting and Vishton 1995](#)].

Our **personal space** is generally considered to be the natural working volume within arm's reach and slightly beyond. We typically are only comfortable with others in this space in intimate situations or due to public necessity (e.g., a subway or concert). This space is within 2 meters of the person's viewpoint, which includes users' feet while standing and the space users can physically manipulate without requiring locomotion. Working within personal space offers advantages of proprioceptive cues, more direct mapping between hand motion and object motion, stronger stereopsis and head motion parallax cues, and finer angular precision of motion [[Miné et al. 1997](#)].

Beyond personal space, **action space** is the space of public action. In action space, we can move relatively quickly, speak to others, and toss objects. This space starts at about 2 meters and extends to about 20 meters from the user.

Beyond 20 meters is **vista space**, where we have little immediate control and perceptual cues are fairly consistent (e.g., binocular vision is almost non-existent). Because effective depth cues are pictorial at this range, large 2D trompe-l'œil paintings at this range are most effective in deceiving the eye that the space is real, as demonstrated in Figure [9.1](#) showing the ceiling of the Jesuit Church in Vienna painted by Andrea Pozzo.



Figure 9.1 This **trompe-l'oeil** art on the dome of the Jesuit Church in Vienna demonstrates that at a distance it can be difficult to distinguish 2D painting from 3D architecture.

9.1.3 Depth Perception

This page you are reading is likely less than a meter away and the words and lettering are less than a centimeter in height. If you look further up you might see walls, windows, and various objects further than a meter. You can intuitively estimate these distances and interact with objects without thinking much about distance even though the projection of those objects onto the retinas of your eyes are 2D images. Based on these projected scenes of the world onto the surface of the retinas alone, we somehow are able to intuitively perceive how far away objects are.

How is it that we can take individual photons that fall onto the retina's surface and re-create from them an understanding of a 3D world? If spatial perception was solely a function of a 2D image on each retina (Section [8.1.1](#)), then there would be no way to know if a particular visual stimulus was a foot away or a mile away. Clearly, there must be more going on than simply detecting points of light on the retina.

Higher-level processing occurs that results in the perception of a 3D world. Human perception relies on a large number of ways we perceive space, shapes, and distance. In this section, several cues are described that help us perceive egocentric depth or distance. The more cues available and the more consistent they are with each other, the stronger the perception of depth. If implemented well in a VR system, such depth cues can significantly add to presence.

Depth cues can be organized into four categories: pictorial cues, motion cues, binocular cues, and oculomotor cues. The context or psychological state of the user can also affect a person's judgment of depth/distance. Table [9.1](#) shows an organization of the cues and factors described in this section, and Table [9.2](#) shows the approximate order of importance for several of the specific cues.

Pictorial Depth Cues

Pictorial depth cues come from distal stimuli projecting photons onto the retina (proximal stimuli), resulting in 2D images. These pictorial cues are present even when viewed with only a single eye and when that eye or anything in the view is not moving. Described below are pictorial cues listed in approximate declining order of importance.

Occlusion. Occlusion is the strongest depth cue as it is always the case for close opaque objects to hide other more distant objects and is effective over the entire range of perceivable distances. Although occlusion is an important depth cue, it only provides relative depth information, and thus precision is dependent upon the number of objects in the scene.

Occlusion is an especially important concept when designing VR displays. Many heads-up displays found in 2D video games do not work well with VR displays due to conflicts with occlusion, motion, physiological, and binocular cues. Because of these conflicts, 2D heads-up displays should be transformed into 3D geometry that has a distance from the viewpoint so that occlusion is properly handled. Otherwise users will get confused when the text or geometry appears at a distance but is not occluded by closer objects and can lead to eye strain and headaches (Section [13.2](#)).

Table 9.1 Organization of depth cues and factors presented in this chapter.

- Pictorial Depth Cues

- Occlusion
- Linear perspective
- Relative/familiar size
- Shadows/shading
- Texture gradient
- Height relative to horizon
- Aerial perspective
- Motion Depth Cues
 - Motion parallax
 - Kinetic depth effect
- Binocular Depth Cues
- Oculomotor Depth Cues
- Vergence
- Accommodation
- Contextual Distance Factors
 - Intended action
 - Fear

Table 9.2 Approximate order of importance (with 1 being most important) of visual cues for perceiving egocentric distance in personal space, action space, and vista space.

Source of Information	Space		
	Personal Space	Action Space	Vista Space
Occlusion	1	1	1
Binocular	2	8	9
Motion	3	7	6
Relative/familiar size	4	2	2
Shadows/shading	5	5	7
Texture gradient	6	6	5
Linear perspective	7	4	4
Oculomotor	8	10	10
Height relative to horizon	9	3	3
Aerial perspective	10	9	8

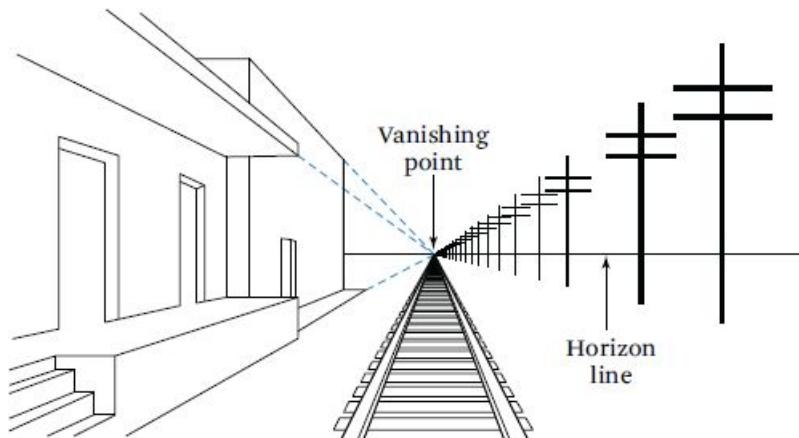


Figure 9.2 Linear perspective causes parallel lines receding into the distance to appear to converge at the vanishing point.

Linear perspective. **Linear perspective** causes parallel lines that recede into the distance to appear to converge together at a single point called the vanishing point (Figure 9.2). The Ponzo railroad illusion in Figure 6.8 demonstrates how strong of a cue linear perspective is. The sense of depth provided by linear perspective is so strong that non-realistic grid patterns may be more effective to conveying spatial presence than more common textures such as carpet or flat walls.

Relative/familiar size. When two objects are of equal size, the further object's projected image on the retina will take up less visual angle than the closer object. **Relative/familiar size** enables us to estimate distance when we know the size of an object, whether due to prior knowledge or comparison to other objects in the scene (Figure 9.3). Seeing a representation of one's own body not only adds presence (Section 4.3) but can also serve as a relative-size depth cue since users know the size of their own body.

Shadows/shading. **Shadows and shading** can provide information regarding the location of objects. Shadows can immediately tell us if an object is lying on the ground or floating above the ground (Figure 9.4). Shadows can also give us clues about the shape (and different depths of a single object).



Figure 9.3 Relative/familiar size. The visual angle of an object of known size provides a sense of distance to that object. Two identical objects having different angular sizes implies the smaller object is at a further distance.

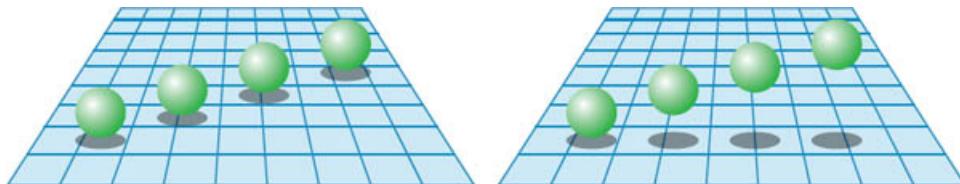


Figure 9.4 Example of how objects can appear at different depth and heights depending on where their shadows lie. (Based on [Kersten et al. \[1997\]](#))

Texture gradient. Most natural objects contain visual texture. **Texture gradient** is similar to linear perspective; texture density increases with distance between the eye and the object surface (Figure 9.5).

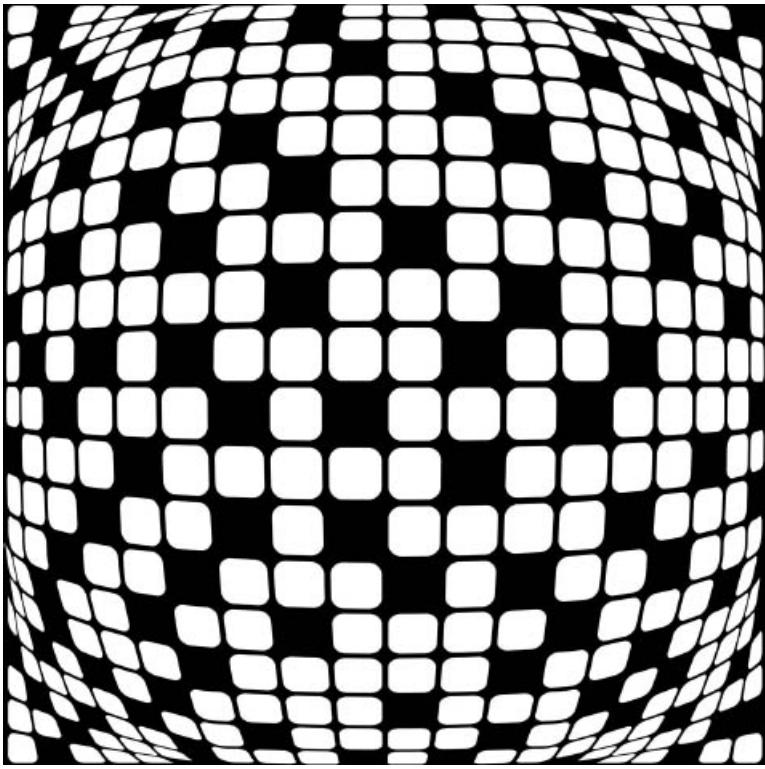


Figure 9.5 The textures of objects can provide a sense of shape and depth. The center of the image seems closer than the edges due to texture gradient.

Height relative to horizon. Height relative to the horizon makes objects appear to be further away when they are closer to the horizon (Figure 9.6).

Aerial perspective. Aerial perspective, also called atmospheric perspective, is a depth cue where objects with more contrast (e.g., brighter, sharper, and more colorful) appear closer than duller objects (Figure 9.7). This results from scattering of light by particles in the atmosphere.

Motion Depth Cues

Motion cues are relative movements on the retina.

Motion parallax. Motion parallax is a depth cue where the images of distal stimuli projected onto the retina (proximal stimuli) move at different rates depending on their distance. For example, when you look outside the side window of a moving car, nearby objects appear to speed by in a blur, whereas distant objects appear to move more slowly. If while moving, you focus on a single point called the fixation point, objects further and closer than that fixation point appear to move in opposite directions (closer objects move against the direction of physical motion and further objects appear to move with the direction of physical motion). Motion parallax enables us to look around objects and is a strong cue for relative perception of depth. Motion parallax works equally in all directions, whereas binocular disparity only occurs in head-horizontal directions. Even small motions of the head due to breathing can result in subtle motion parallax that adds to presence (Andrew Robinson and Sigurdur Gunnarsson, personal communication, May 11, 2015).

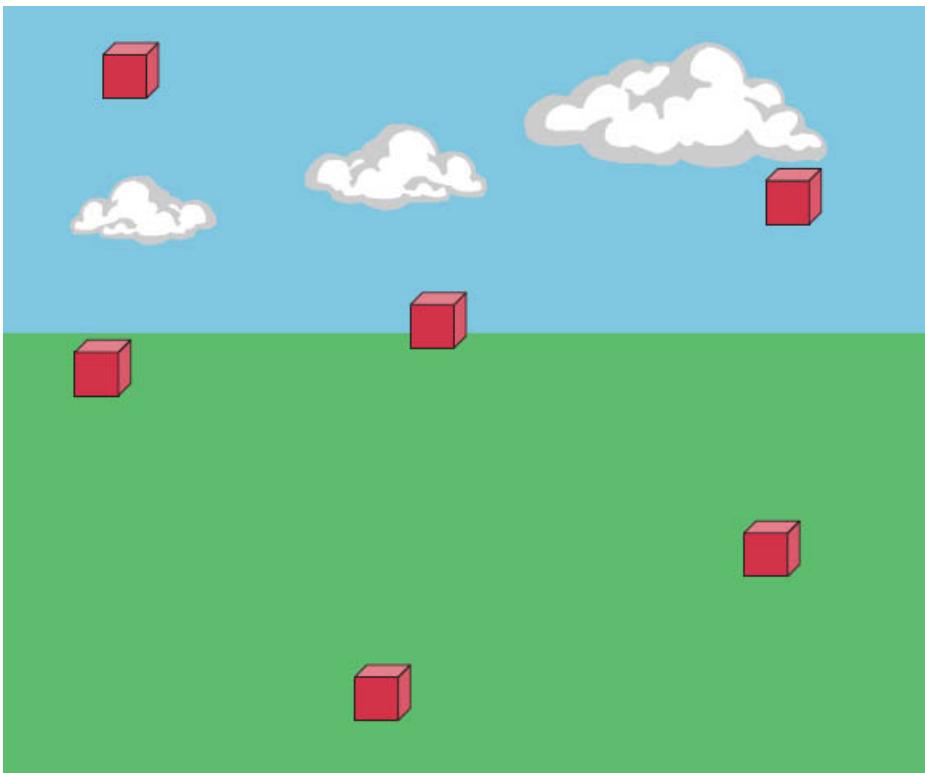


Figure 9.6 Objects closer to the horizon result in the perception that they are further away.

Motion parallax is a problem for world-fixed displays (Section 3.2.1) when there are multiple viewers but only one person's viewpoint is being tracked. The shape and motion of the scene is correct for the person being tracked but appears as distortion and objects swaying for the non-tracked viewers.



Figure 9.7 Objects further away have less color and detail due to aerial perspective. (Courtesy of NextGen Interactions)

Kinetic depth effect. The **kinetic depth effect** is a special form of motion parallax that describes how 3D structural form can be perceived when an object moves. For example, observers can perceive the 3D structure of a wire simply by the motion and deformation of the shadow cast by that wire as it is rotated.

Binocular Depth Cues

Those of us with two healthy eyes receive slightly different views of the world on each of our retinas, with the differences being a function of distance. **Binocular disparity** is the difference in image location of a stimulus seen by the left and right eyes resulting from the eyes' horizontal separation and the stimulus distance. **Stereopsis** (also known as binocular fusion) occurs when the brain combines separate and slightly different images from each eye to form a single percept that contains a vivid sense of depth.

Stimuli at a distance resulting in no disparity form a surface in space called the **horopter** (Figure 9.8). The surrounding space in front of and behind the horopter where we can perceive stereopsis is called **Panum's fusional area**. The area outside Panum's fusional area contains disparity that is too great to fuse, causing double images resulting in diplopia.

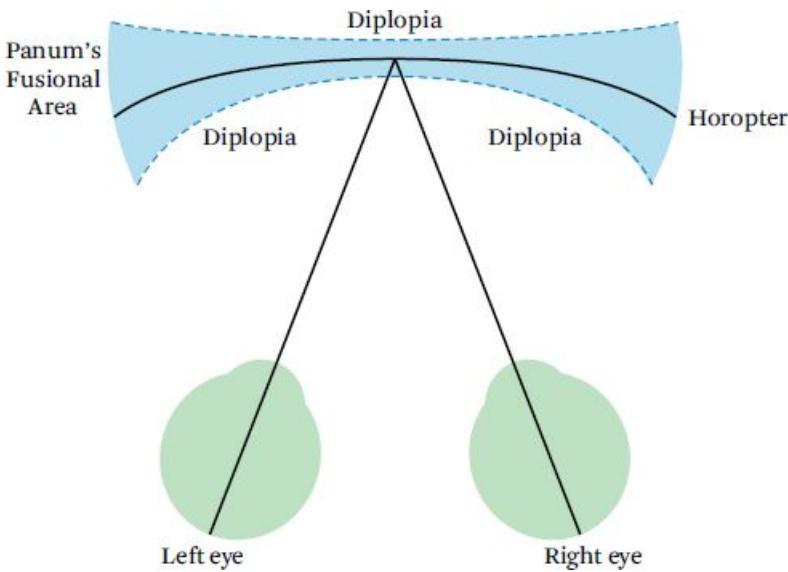


Figure 9.8 The horopter and Panum's fusional area. (Adapted from [Mirzaie \[2009 \]](#))

HMDs can be **monocular** (one image for a single eye), **biocular** (two identical images, one image for each eye), or **binocular** (two different images for each eye, providing a sense of stereopsis). Binocular images can cause conflicts with accommodation, vergence, and disparity (Section [13.1](#)). Biocular HMDs cause a conflict with motion parallax, but there are few complaints of the conflict.

The value of binocular cues is often debated for stereoscopic displays and VR. Part of this is likely due to some proportion of the population being **stereoblind**—it is estimated that 3–5% of the population is unable to extract depth signaled purely by disparity differences in modern 3D displays [[Badcock et al. 2014](#)]. The incidence of such stereoblindness increases with age, and there are likely different levels of stereoblindness beyond the 3–5%, resulting in disagreement on the value of binocular displays.

For most people, stereopsis becomes important when interacting in personal space through the use of the hands, especially when other depth cues are not strong. We do perceive some binocular depth cues at further distances (up to ~40 meters under ideal conditions—see Section [8.1.4](#)), but stereopsis is much less important at such distance.

Oculomotor Depth Cues

The musculature of the eyes, specifically those dealing with accommodation and vergence, can provide subtle depth cues at distances up to 2 meters. Under normal real-world conditions, accommodation and vergence automatically work together when looking at objects at different distances, known as the accommodation-vergence reflex. With VR, accommodation and vergence can be in conflict with each other (as well as with other binocular cues) when the display focus is set at a distance different than where the eyes feel like they are looking (Section [13.1](#)).

Vergence. As discussed in Section [8.1.5](#), **vergence** is the simultaneous rotation of the eyes in opposite directions, triggered by retinal disparity, with the primary purpose being to obtain or maintain both sharp and comfortable binocular vision [[Reichelt et al. 2010](#)]. We can feel inward (**convergence**) and outward (**divergence**) eye movements that provide depth cues. Evidence suggests the effectiveness of convergence alone as a source of information for depth is confined to a range of up to about 2 meters [[Cutting and Vishton 1995](#)]. Convergence is a more effective cue than accommodation, described below.

Accommodation. Our eyes contain elastic lenses that change curvature, enabling us to focus the image of objects sharply on the retina. **Accommodation** is the mechanism by which the eye alters its optical power to hold objects at different distances into focus on the retina. We can feel the tightening of eye muscles that change the shape of the lens to focus on nearby objects, and this muscular contraction can serve as a cue for distance for up to about 2 meters [[Cutting and Vishton 1995](#)]. Full accommodation response requires a minimum fixation time of one second or longer and is triggered primarily by retinal blur [[Reichelt et al. 2010](#)]. Once we have accommodated to a specific distance, objects much closer or further from that distance are out of focus. This blurriness can also provide subtle depth cues, but it is ambiguous if the blurred object is closer or further away from the object that is in focus. Objects having low contrast represent a weak stimulus for accommodation.

The ability to accommodate decreases with age, and most people cannot maintain close-up accommodation for any more than a short period of time. VR designers should not place essential stimuli that must be consistently viewed close to the eye.

Contextual Distance Factors

Although research on visual depth cues has been around for centuries, more recent research suggests distance perception is influenced not only by visual cues but also by environmental context and personal variables. For example, the accuracy of distance perception has been found to differ depending on whether objects being judged are indoors or outdoors, or even in different types of indoor environments [[Renner et al. 2013](#)].

Future effect distance cues are internal psychological beliefs of how distance may personally affect the viewer in the future. Such cues include one's intended actions and fear.

Intended action. Perception and action are closely intertwined (Section 10.4). **Action-intended distance cues** are psychological factors of future actions that influence distance perception. People in the same situation perceive the world differently depending upon the actions they intend to perform and benefits/costs of those actions [Proffitt 2008]. We see the world as “reachers” only if we intend to reach, as “throwers” only if we intend to throw, and as “walkers” only if we intend to walk. Objects within reach appear closer than those out of reach, and when reachability is extended by holding a tool that is intended to be used, apparent distances are diminished. Hills appear to be steeper than they actually are. Hills of 5° are typically judged to be about 20°, and 10° hills appear to be 30° [Proffitt et al. 1995]. A manipulation that influences the effort to walk will influence people’s perception of distance if they intend to walk. This overestimation increases for those encumbered by wearing a heavy backpack, fatigue, low physical health, old age, and/or declining health [Bhalla and Proffitt 1999]. Hills appear steeper and egocentric distances farther, as the anticipated metabolic energy costs associated with walking increase.

Fear. **Fear-based distance cues** are fear-causing situations that increase the sense of height. We overestimate vertical distance, and this overestimation is much greater when the height is viewed from above than from below. A sidewalk appears steeper for those fearfully standing on a skateboard compared to those standing fearlessly on a box [Stefanucci et al. 2008]. This is likely due to an evolved adaptation—enhancing the perception of vertical height motivates people to avoid falling. Fear influences spatial perception.

9.1.4 Measuring Depth Perception

There are several methods to test distance perception and those methods can be divided into three categories: (1) verbal estimation, (2) perceptual matching, and (3) visually directed actions [Renner et al. 2013]. The different methods often result in different estimates. Even the phrasing of instruction within the same method can influence how distance estimates are made. For example, “how far away does the object visually appear to be” or “how far away the object really is” may result in different estimates. With carefully designed methods, depth perception can be quite accurate for distances of up to 20 meters.

9.1.5 VR Challenges of Space Perception

Natural perception of space within VR still remains a challenge for display technology as well as content creation. For example, whereas egocentric distance perception can be quite good for the real world, VR egocentric distance estimates are reported to be compressed [Loomis and Knapp 2003 , Renner et al. 2013]. Veridical spatial perception is not essential for all VR applications, but it is extremely important where perception of distance and object size is essential, such as architecture, automotive, and military applications.

9.2 Time Perception

Time can be thought of as a person’s own perception or experience of successive change. Unlike objectively measured time, **time perception** is a construction of the brain that can be manipulated and distorted under certain circumstances. The concept of time is essential to most cultures, with languages having distinct tenses of past, present, and future, plus innumerable modifiers to specify the “when” more precisely.

9.2.1 A Breakdown of Time

The **subjective present** is always with us as the precious few seconds of our ongoing conscious experience. Everything else does not exist, but is either past or future. The subjective present consists of (1) the “now”—i.e., the current moment, and (2) the experience of passing time [Coren et al. 1999]. The experience of passing time can further be broken down into (a) duration estimation, (b) order/sequence (consisting of simultaneity and successiveness), and (c) anticipation/planning for the immediate future (e.g., knowing what note to play next with a musical instrument).

Psychological time can be considered a sequence of moments instead of a continuous experience. A **perceptual moment** is the smallest psychological unit of time that an observer can sense. Stimuli presented within the same perceptual moment are perceived as occurring simultaneously. Based on several research findings, Stroud [1986] estimated that perceptual moments are in the 100 ms range in duration. The length of these perceptual moments ranges from about 25 to 150 ms depending upon the sensory modality, stimulus, task, etc. Evidence suggests perceptual moments become unstable when one judges with more than one sensory modality, such as when judging the order of presentation of a sound and a light [Ulrich 1987].

An **event** is a segment of time at a particular location that is perceived to have a beginning and an end or a series of perceptual moments that unfolds over time. Our ability to understand relationships between actions and objects depends on this unfolding. Events are similar to sentences and phrases in language, where the sequence and timing of nouns and verbs gives meaning. There is a big difference between “bear eats man” and “man eats bear.” The two phrases contain the same elements, but it is the sequence and timing of these stimuli at the eye or ear that gives meaning.

9.2.2 The Brain and Time

Delayed Perception

Despite what most of us would like to believe, we all live in the past with our consciousness always lagging behind reality. It takes between 10 ms and 50 ms for visual signals to reach the LGN (Section 8.1.1), another 20–50 ms for those signals to reach the visual cortex, and an additional 50–100 ms to reach parts of the brain that are responsible for action and planning [Coren et al. 1999]. This can cause problems when response time is essential. For example, it takes a driver over 200 ms to perceive a stimulus that requires the decision to apply the brakes. This number can be even worse if the driver is dark adapted (Section 10.2.1). After perceiving the stimulus, it takes a relatively short amount of time at 10–15 ms to initiate muscle responses. For a driver traveling at 100 kph (62 mph), the car will have traveled 6 m (20 ft) before the foot starts moving toward

the brakes. For cases where we can anticipate events, this delay can be reduced via top-down processing that primes cells to respond faster.

Although we live in the past couple hundred milliseconds, this does not mean it is okay to have a couple hundred milliseconds of VR system delay. Such additional external-to-the-body latency is a major cause of motion sickness as discussed in Chapter [15](#).

Persistence

In addition to delayed perception, the length of an experience depends on neural activity, not necessarily the duration of a stimulus. **Persistence** is the phenomenon by which a positive afterimage (Section [6.2.6](#)) seemingly persists visually. A single 1 ms flash of light is often experienced as a 100 ms perceptual moment when the observer is light adapted and as long as 400 ms when dark adapted [[Coren et al. 1999](#)].

When two stimuli are close in time and space, the two stimuli can be perceived as a single moving stimulus (Section [9.3.6](#)). For example, pixels displayed in different locations at different times can appear as a single moving stimulus. **Masking** can also occur where one of two stimuli (usually visuals or sounds) are not perceived. The second stimulus is typically perceived more accurately (backward masking) as if the second stimulus goes back in time and erases the first stimulus.

Perceptual Continuity

Perceptual continuity is the illusion of continuity and completeness, even in cases when at any given moment only parts of stimuli from the world reach our sensory receptors [[Coren et al. 1999](#)]. The brain often constructs events after multiple stimuli have been presented, helping to provide continuity for the perception of a single event across breaks or interruptions that occur because of extraneous stimuli in the environment.

The phonemic restoration effect (Section [8.2.3](#)) causes us to perceive missing parts of speech, where what we hear in place of the missing parts is dependent on context. The visual system behaves in a similar way. When an object moves behind other occluding objects, then we perceive the entire object even when there is no single moment in time when the occluded object is completely seen. This can be seen by looking through a narrow slit created by a partially opened door (e.g., even for an opening of only 1 mm). As the head is moved or an object is moved behind the slit, we perceive the entire scene or object to be present due to the brain stitching together the stimuli over time. Perceptual continuity is one reason why we don't normally perceive the blind spot (Section [6.2.3](#)).

9.2.3 Perceiving the Passage of Time

Unlike the sensory modalities, there is no obvious time organ that enables us to perceive the passage of time. Instead it is through change that we perceive time. Two theories that describe how we are able to perceive time are biological clocks and cognitive clocks [[Coren et al. 1999](#)].

Biological Clocks

The world and its inhabitants have their own rhythms of change, including day-night cycles, moon cycles, seasonal cycles, and physiological changes. **Biological clocks** are body mechanisms that act in periodic manners, with each period serving as a tick of the clock, which enable us to sense the passing of time. Altering the speed of our physiological rhythm alters the perception of the speed of time.

A **circadian rhythm** (circadian comes from the root circa, meaning “approximate” and dies, meaning “days”) is a biologically recurring natural 24-hour oscillation. Circadian rhythms are largely controlled by the supra chiasmatic nucleus (SCN), a small organ near the optic chiasm of only a few thousand cells. Circadian rhythms result in sleep-wakefulness timed behaviors that run through a day-night cycle. Although circadian rhythms are endogenous, they are entrained by external cues called zeitgebers (German for “time-givers”), the most important being light. Without light, most of us actually have a rhythm of 25 hours.

Biological processes such as pulse, blood pressure, and body temperature vary as a function of the day-night cycle. Other effects on our mind and body include changes to mood, alertness, memory, perceptual motor tasks, cognition, spatial orientation, postural stability, and overall health. The experiences of VR users may not be as good at times when they are normally sleeping.

Slow circadian rhythms do not help us with our perception of short time periods. We have several biological clocks for judgments of different amounts of time and different behavioral aspects. Short-term biological timers that have been suggested for internal biological timing mechanisms include heartbeats, electrical activity in the brain, breathing, metabolic activities, and walking steps.

Our perception of time also seems to change when our biological clocks change. Durations seem to be longer when our biological clocks tick faster due to more biological ticks per unit time than normally occur. Conversely, time seems to whiz by when our internal clock is too slow. Evidence demonstrates body temperature, fatigue, and drugs cause change in time duration estimates.

Cognitive Clocks

Cognitive clocks are based on mental processes that occur during an interval. Time is not directly perceived but rather is constructed or inferred. The tasks a person engages in will influence the perception of the passage of time.

Cognitive clock factors affecting time perception are (1) the amount of change, (2) the complexity of stimulus events that are processed, (3) the amount of attention given to the passage of time, and (4) age. All these affect the perception of the passage of time, and all are consistent with the idea that the rate at which the cognitive clock ticks is affected by how internal events are processed.

Change. The ticking rate of cognitive clocks is dependent on the number of changes that occur. The more changes that occur during an interval, the faster the clock ticks, and thus the longer the estimate of the amount of time passed.

The **filled duration illusion** is the perception that a duration filled with stimuli is perceived as longer than an identical time period empty of stimuli. For example, an interval filled with tones is perceived as being longer than the same interval with fewer tones. This is also the case for light flashes, words, and drawings. Conversely, those in a soundproof darkened chamber tend to underestimate the amount of time they have spent in the chamber.

Processing effort. The ticking rate of cognitive clocks is also dependent on cognitive activity. Stimuli that are more difficult to process result in judgments of longer durations. Similarly, increasing the amount of memory storage required to process information results in judgments of longer durations.

Temporal attention. Time perception is complicated by the way observers attend to a task. The more attention one pays to the passage of time, the longer the time interval seems to be. Paying attention to time may even reverse the filled duration illusion mentioned above because it is difficult to process many events while attending at the same time to the passage of time.

When observers are told in advance that they will have to judge the time a task takes, they tend to judge the duration as being longer than they do when they are unexpectedly asked to judge the duration after the task is completed. This is presumably due to drawing attention to time.

Conversely, anything that draws attention away from the passage of time results in a faster perception of time. Duration estimates are typically shorter for difficult tasks than for easy tasks because it is more difficult to attend to both time and a difficult task.

Age. As people age, the passage of larger units of time (i.e., days, months, or even years) seems to speed up. One explanation for this is that the total amount of time someone has experienced serves as a reference. One year for a 4-year old is 20% of his life so time seems to drag by. One year for a 50-year old is 2% of his life, so time seems to move more quickly.

9.2.4 Flicker

Flicker is the flashing or repeating of alternating visual intensities. Retinal receptors respond to flickering light at up to several hundred cycles per second, although sensitivity in the visual cortex is much less [[Coren et al. 1999](#)]. Perception of flicker covers a wide range depending on many factors including dark adaptation, light intensity, retinal location, stimulus size, blanking time between stimuli, time of day (due to circadian rhythms), gender, age, bodily functions, and wavelength. We are most sensitive to a light-adapted eye with a high light intensity over a wide range of the visual periphery when the body is most awake. The **flicker-fusion frequency threshold** is the flicker frequency where flicker becomes visually perceptible.

9.3 Motion Perception

Perception is not static but varies continuously over time. Perception of motion at first seems quite trivial. For example, visual motion simply seems as a sense of stimuli moving across the retina. However, motion perception is a complex process that involves a number of sensory systems and physiological components [[Coren et al. 1999](#)].

We can perceive movement even when visual stimuli are not moving on the retina, such as when tracking a moving object with our eyes (even when there are no other contextual stimuli to compare to). At other times, stimuli move on the retina yet we do not perceive movement; for example, when the eyes move to examine different objects. Despite the image of the world moving on our retinas, we perceive a stable world (Section [10.1.3](#)).

A world perceived without motion can be both disturbing and dangerous. A person who has akinetopsia (blindness to motion) sees people and objects suddenly appearing or disappearing due to not being able to see them approach. Pouring a cup of coffee becomes nearly impossible. The lack of motion perception is not just a social inconvenience but can be quite dangerous; consider crossing a street where a car seems far away and then suddenly appears very close.

9.3.1 Physiology

The primitive visual pathway through the brain (Section [8.1.1](#)) is largely specialized for motion perception along with the control of reflexive responses such as eye and head movements. The primary visual pathway also contributes to motion detection through magno cells (and to a lesser extent parvo cells) as well as through both the ventral pathway (the “what” pathway) and the dorsal pathway (the “where/how/action” pathway).

The primate brain has visual receptor systems dedicated to motion detection [[Lisberger and Movshon 1999](#)], which humans use to perceive movement [[Nakayama and Tyler 1981](#)]. Whereas visual velocity (i.e., speed and direction of a visual stimulus on the retina) is sensed directly in primates, visual acceleration is not, but is instead inferred through processing of velocity signals [[Lisberger and Movshon 1999](#)]. Most visual perception scientists agree that for perception of most motion, visual acceleration is not as important as visual velocity [[Regan et al. 1986](#)]. However, visual linear acceleration can evoke motion sickness more so than visual linear velocity when there is a mismatch with linear acceleration detected by the vestibular system (Section [18.5](#)).

9.3.2 Object-Relative vs. Subject-Relative Motion

Object-relative motion (similar to exocentric judgments; Section [9.1.1](#)) occurs when the spatial relationship between stimuli changes. Because

the relationship is relative, which object is moving is ambiguous. Visual object-relative judgments depend solely upon stimuli on the retina and do not take into account extra-retinal information such as eye or head motion.

Subject-relative motion (similar to egocentric judgments; Section 9.1.1) occurs when the spatial relationship between a stimulus and the observer changes. This subject-relative frame of reference for visual perception is provided by extra-retinal information such as vestibular input. Even when the head is not moving the vestibular system still receives input. A purely subject-relative cue without any object-relative cues occurs only when a single visual stimulus is visible or all visual stimuli have the same motion.

People use both object-relative and subject-relative cues to judge motion of the external world. For all but the shortest intervals, humans are much more sensitive to object-relative motion than subject-relative motion [[Mack 1986](#)] due to perceiving displacement relative to a visual context rather than velocity. For example, one study found that observers can detect a luminous dot moving in the dark (subject-relative) at about 0.2°/s whereas observers can detect a moving target in the presence of a stationary visual context (object-relative) at about 0.03°/s.

Visuals in augmented reality are considered to be mostly object-relative to real-world cues, as the visuals can be directly compared with the real world. Incorrectly moving visuals are more easily noticed in optical-see-through HMDs (i.e., augmented reality) than non-see-through HMDs [[Azuma 1997](#)], since users can directly compare rendered visuals relative to the real world and hence see spurious object-relative motion. Visual error in VR due to problems such as latency and miscalibration is usually subject-relative, as such error causes the entire scene to move as the user moves the head (other than for motion parallax, where visuals move differently as a function of viewpoint translation and depth).

9.3.3 Optic Flow

Optic flow is the pattern of visual motion on the retina—the motion pattern of objects, surfaces, and edges on the retina caused by the relative motion between a person and the scene. As one looks to the left, optic flow moves to the right. As one moves forward, optic flow expands in a radial pattern. **Gradient flow** is the different speed of flow for different parts of the retinal image—fast near the observer and slower further away. The **focus of expansion** is the point in space around which all other stimuli seem to expand as one moves forward toward it.

9.3.4 Factors of Motion Perception

Many factors influence our perception of motion. For example, we are more sensitive to visual motion as contrast increases. Some other factors are discussed below.

Motion Perception in Peripheral vs. Central Vision

The literature conflicts as to whether sensitivity to motion increases or decreases with eye eccentricity (the distance from the stimulus on the retina to the center of the fovea). These conflicting claims are likely due to differences of experimental conditions as well as interpretations.

[Anstis \[1986 \]](#) states that it is sometimes mistakenly claimed that peripheral vision is more sensitive to motion than central vision. In fact, the ability to detect slow-moving stimuli actually decreases steadily with eye eccentricity. However, since sensitivity to static detail decreases even faster, peripheral vision is relatively better at detecting motion than form. A moving object seen in the periphery is perceived as something moving, but it is more difficult to see what that something is.

[Coren et al. \[1999 \]](#) states that the detection of movement depends on both the speed of the moving stimulus and eye eccentricity. A person's ability to detect slow-moving stimuli (less than 1.5°/s) decreases with eye eccentricity, which is consistent with Anstis. For faster-moving stimuli, however, the ability to detect moving stimuli increases with eye eccentricity. These differences are due to the dominance in the periphery of the dorsal or "action" pathway (Section 8.1.1), which consists of more transient cells (cells that respond best to fast-changing stimuli). Scene motion in the peripheral visual field is also important in sensing self-motion (Sections 9.3.10 and [10.4.3](#)).

Judging motion consists of more than just looking for moving stimuli. Qualitative evidence suggests slight visual motion in the periphery may more easily be judged by "feeling" (e.g.,vection as discussed in Section 9.3.10) whereas slight motion in central vision may more easily be judged by direct visual and logical thinking. For example, two subjects judging a VR scene motion perception task [[Jerald et al. 2008](#)] stated:

Most of the times I detected the scene with motion by noticing a slight dizziness sensation. On the very slight motion scenes there was an eerie dwell or 'suspension' of the scene; it would be still but had a floating quality.

At the beginning of each trial [sic: session—when scene velocities were greatest because of the adaptive staircases], I used visual judgments of motion; further along [later in sessions] I relied on the feeling of motion in the pit of my stomach when it became difficult to discern the motion visually.

These different ways of judging visual motion are likely due to the different visual pathways (Section 8.1.1).

Head Motion Suppresses Motion Perception

Although head motion suppresses perception of visual motion, perception of visual motion while the head is moving can be surprisingly accurate. [Loose and Probst \[2001 \]](#) found that increasing the angular velocity of the head significantly suppresses the ability to detect visual motion when the visual motion moves relative to the head. [Adelstein et al. \[2006 \]](#) and [Li et al. \[2006 \]](#) also showed that head motion suppresses perception of visual motion. They used an HMD without head tracking where the slight motion to detect was relative to the head. [Jerald \[2009 \]](#) measured how head motion suppresses perception of visual motion where the slight motion to detect was relative to the real world, and then mathematically

related that suppression to the perception of latency.

Depth Perception Affects Motion Perception

The **pivot hypothesis** [[Gogel 1990](#)] states that a point stimulus at a distance will appear to move as the head moves if its perceived distance differs from its actual distance. A related effect is demonstrated by focusing on a finger held in front of the eyes and noticing that the background further in the distance seems to move with the head. Likewise if one focuses on the background then the finger seems to move against the direction of the head.

Objects in HMDs tend to appear closer to users than their intended distance (Section [9.1.5](#)). According to the pivot hypothesis, if a user looks at an object as if it is close, but the system moves the object's image in the HMD as if it is further away when turning the head, then the object will appear to move in the same direction as the head (similar to the finger example above).

9.3.5 Induced Motion

Induced motion occurs when motion of one object induces the perception of motion in another object.

When a dot moves near an existing dot in an otherwise uniform visual field, it is clear one of the dots is moving but often it is difficult to identify which of the two dots is moving (due to the object-relative cues telling us how objects move relative to each other, but not relative to the world). However, when a rectangular frame and dot are observed, we tend to perceive the dot to be moving even if the dot is stable and the frame is what is moving. This is because the mind assumes smaller objects are more likely to move than larger surround stimuli and the larger surround stimuli serve as a stable context (see the moon-cloud illusion and the autokinetic effect in Section [6.2.7](#) and the rest frame hypothesis in Section [12.3.4](#)).

Induced motion effects are most dramatic when the context is moving slowly, the context is square vs. circular, the surround is larger, and the target and context are the same distance from the observer.

9.3.6 Apparent Motion

Perception of motion does not require continuously moving stimuli. **Apparent motion** (also known as stroboscopic motion) is the perception of visual movement that results from appropriately displaced stimuli in time and space even though nothing is actually moving [[Anstis 1986](#), [Coren et al. 1999](#)]. It's as if the brain is unwilling to conclude two similar-looking stimuli happened to disappear and appear next to each other close in time. Therefore, the original stimulus must have moved to the new location.

The ability to see discontinuous changes as if they were continuous motion has benefited technology since the 1800s and continues to this day, ranging from flashing neon signs (e.g., older Las Vegas signs giving the illusion of movement) to rolling text displays, television, movies, computer graphics, and VR. For VR creators, understanding apparent motion and the brain's rules of visual motion perception for filling in physically absent motion information is important for conveying not only a sense of motion from an array of pixels, but a sense of stability since the pixels must change appropriately as the head moves in order to appear stable in space.

The mind most often perceives apparent motion by following the shortest distance between two neighboring stimuli. However, in some cases the perceived path of motion can also seem to deflect around an object located between the two flashing stimuli. It is as if the brain tries to figure out how the object could have gotten from point A to point B. Two stimuli of different shapes, orientations, brightnesses, and colors can also be perceived to be a single stimulus that transforms while moving.

Strobing and Judder

Strobing and judder can be a problem for HMDs due to movement of the head and movement of the eyes relative to the screen and stimulus.

Strobing is the perception of multiple stimuli appearing simultaneously, even though they are separated in time, due to the stimuli persisting on the retina. If two adjacent stimuli are flashed too quickly, then strobing can result. Stimuli separated by larger distances require longer time intervals for motion to be perceived.

Judder is the appearance of jerky or unsmooth visual motion. If adjacent stimuli are flashed too slowly, then judder can result (for traditional video, judder can also result from converting between different timing formats).

Factors

Three variables, as shown in Figure [9.9](#), are involved with the temporal aspects of apparent motion.

Interstimulus interval: The blanking time between stimuli. Increasing the blanking time increases strobing and decreases judder.

Stimulus duration: The amount of time each stimulus is displayed. Increasing the stimulus duration reduces strobing and increases judder.

Stimulus onset asynchrony: The total time that elapses between the onset of one stimulus and the onset of the next stimulus (interstimulus interval + stimulus duration). This is the inverse of the display refresh rate.

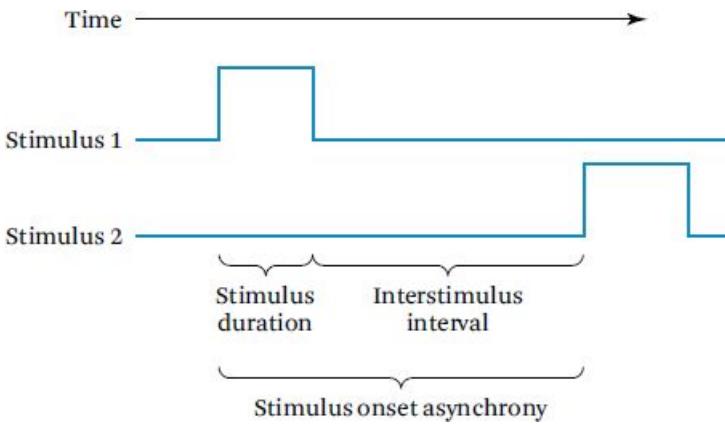


Figure 9.9 Three variables affecting apparent motion. (Adapted from [Coren et al. \[1999 \]](#))

The timings required to not notice judder or strobing depend not only on timings but on the spatial properties of the stimuli such as contrast and distance between stimuli. Filmmakers control many of these challenges through limited camera motion, object movement, lighting, and motion blur. The stimulus onset asynchrony required to perceive motion ranges from 200 ms for a flashing Vegas casino sign to ~10 ms for smooth motion in a high-quality HMD. VR is much more challenging, partly because the user controls the viewpoint that often is quite fast. The best way to minimize judder for VR is to minimize the stimulus onset asynchrony and stimulus duration, although strobing can appear if the stimulus duration is too short [[Abrash 2013](#)].

9.3.7 Motion Coherence

Seemingly randomly placed dots in individual frames can be perceived as having form and motion if those dots move in a coherent manner across frames. **Motion coherence** is the correlation between movements of dots in successive images. Complete random motion of the dots between frames has 0% motion coherence, and if all dots move in the same way there is 100% motion coherence. Humans are highly sensitive to motion coherence, and we can see form and motion with as little as 3% coherence [[Coren et al. 1999](#)]. The most optimal condition of perceiving motion coherence is when the dots move at 2%/s. Sensitivity to motion coherence increases as the visual field size increases up to 20° in diameter.

9.3.8 Motion Smear

Motion smear is the trail of perceived persistence that is left by an object in motion [[Coren et al. 1999](#)]. The persistence of the streak is long enough in duration to allow the path to be seen. Although motion can be seen when moving a small lit cigarette or hot campfire ember, motion smear does not require either dark conditions or bright stimuli to be seen. For example, motion smear can be seen by waving the hand back and forth in front of the eyes, resulting in the hand appearing to be in multiple locations simultaneously. Research has shown there is substantial motion smear suppression when a moving stimulus follows a predictable path, so motion smear cannot be due to persistence on the retina alone.

9.3.9 Biological Motion

Humans have especially fine-tuned perception skills when it comes to recognizing human motion. **Biological motion perception** refers to the ability to detect such motion.

[Johansson \[1976 \]](#) found that some observers were able to identify human motion represented by 10 moving point lights in durations as short as 200 ms and all observers tested were able to perfectly recognize 9 different motion patterns for durations of 400 ms or more.

In another study, people who were well acquainted with each other were filmed with only lit portions of several of their joints. Several months later, the participants were able to identify themselves and others on many trials just from viewing the moving lit joints [[Coren et al. 1999](#)]. When asked how they made their judgments, they mentioned a variety of motion components such as speed, bounciness, rhythm, arm swing, and length of steps. In other studies, observers could determine above chance whether moving dot patterns were from a male or female within 5 seconds of viewing even when only the ankles were represented!

These studies imply avatar motion in VR is extremely important, even when the avatar is not seen well or when presented over a short period of time. Any motion simulation of humans must be accurate in order to be believed.

9.3.10 Vection

Vection is an illusion of self-motion when one is not actually physically moving in the perceived manner. Vection is similar to induced motion (Section [9.3.5](#)), but instead of inducing an illusion of a small visual stimulus motion, vection induces an illusion of self-motion. In the real world, vection can occur when one is seated in a car and an adjacent stopped car pulls away. One experiences the car one is seated in, which is actually stationary, to move in the opposite direction of the moving car.

Vection often occurs in VR due to the entire visual world moving around the user even though the user is not physically moving. This is experienced as a compelling illusion of self-motion in the direction opposite the moving stimuli. If the visual scene is moved to the left, the user may feel that she is moving or leaning to the right and will compensate by leaning into the direction of the moving scene (the left). In fact, vection in carefully designed

VR experiences can be indistinguishable from actual self-motion (e.g., the Haunted Swing described in Section [2.1](#)).

Although not nearly as strong as vision, other sensory modalities can contribute tovection [[Hettinger et al. 2014](#)]. Auditoryvection can producevection by itself in the absence of other sensory stimuli or can enhancevection provided by other sensory stimuli. Biomechanicalvection can occur when one is standing or seated and repeatedly steps on a treadmill or an even, static surface. Haptic/tactile cues via touching a rotating drum surrounding the user or airflow via a fan can also add to the sense ofvection.

Factors

Vection is more likely to occur for large stimuli moving in peripheral vision. Increasing visual motions up to about $90^\circ/\text{s}$ result in increasedvection [[Howard 1986a](#)]. Visual stimuli with high spatial frequencies also increasevection.

Normally, users correctly perceive themselves to be stable and the visuals to be moving before the onset ofvection. This delay ofvection can last several seconds. However, at stimulus accelerations of less than about $5^\circ/\text{s}^2$,vection is not preceded by a period of perceived stimulus motion as it is at higher rates of acceleration [[Howard 1986b](#)].

Vection is not only attributed to low-level bottom-up processing. Research suggests higher-level top-downvection processing of globally consistent natural scenes with a convincing rest frame (Section [12.3.4](#)) dominates low-level bottom-upvection processing [[Riecke et al. 2005](#)]. Stimuli perceived to be further away cause morevection than closer stimuli, presumably because our experience with real life tells us background objects are typically more stable. Moving sounds that are normally perceived as landmarks in the real world (e.g., church bells) can cause greatervection than from artificial sounds or sounds typically generated from moving objects.

Ifvection depends on the assumption of a stable environment, then one would expect the sensation ofvection would be enhanced if the stimulus is accepted as stationary. Conversely, if we can reverse that assumption by thinking of the world as an object that can be manipulated (e.g., grabbing the world and moving it toward us as done in the 3D Multi-Touch Pattern; Section [28.3.3](#)), thenvection and motion sickness can be reduced (Section [18.3](#)). Informal feedback from users of such manipulable worlds indeed suggests this to be the case. However, further research is required to determine to what degree.

Perceptual Stability, Attention, and Action

10.1 Perceptual Constancies

Our perception of the world and the objects within it is relatively constant. A **perceptual constancy** is the impression that an object tends to remain constant in consciousness even though conditions (e.g., changes in lighting, viewing position, head turning) may change. Perceptual constancies occur partly due to the observer's understanding and expectation that objects tend to remain constant in the world.

Perceptual constancies have two major phases: registration and apprehension [[Coren et al. 1999](#)].

Registration is the process by which changes in the proximal stimuli are encoded for processing within the nervous system. The individual need not be consciously aware of registration. Registration is normally oriented on a focal stimulus, the object being paid attention to. The surrounding stimuli are the context stimuli.

Apprehension is the actual subjective experience that is consciously available and can be described. During apprehension, perception can be divided into two properties: object properties and situational properties. **Object properties** tend to remain constant over time. **Situational properties** are more changeable, such as one's position relative to an object or the lighting configuration.

Table [10.1](#) summarizes some common perceptual constancies that are further discussed below [Coren et al. 1999].

10.1.1 Size Constancy

Why do objects not seem to change size when we walk toward them? After all, the projected size of an object on the retina changes as we walk toward or away from it. For example, if we walk toward an animal in the distance then we do not gasp in horror as if a monster is growing. The sensory reality is that the animal projected on our retina is growing as we walk toward it, but we do not perceive the animal is changing size. Our past experiences and mental model of the world tells us things do not change size when we walk toward them—we have learned object properties remain constant independent of our movement through the world. This learned experience that objects tend to stay the same size is called **size constancy**.

Table 10.1 Perceptual constancies. (Adapted from [Coren et al. \[1999 \]](#))

Constancy	Registered Stimulus (may be unconscious)		Apprehended Stimulus (conscious)	
	Focal Stimulus	Context	Constant	Changes
Size constancy	retinal image size	distance cues	object size	object distance
Shape constancy	retinal image shape	orientation cues	object shape	object orientation
Position constancy	retinal image location	sensed head or eye pose	object position in space	head or eye pose
Lightness constancy	retinal image intensity	illumination cues	surface whiteness intensity	apparent illumination
Color constancy	retinal image color	illumination cues	surface colors	apparent illumination color
Loudness constancy	ear sound intensity	distance cues	loudness of sounds	distance from sound

Size constancy is largely dependent on the apparent distance of an object being correct, and in fact changes in misperceived distance can change perceived size. The more distance cues ([Section 9.1.3](#)) available, the better size constancy. When only a small number of depth cues are available, distant objects tend to be perceived as too small.

Size constancy does not seem to be consistent across all cultures. Pygmies, who live deep in the rain forest of tropical Africa, are not often exposed to wide-open spaces and do not have as much opportunity to learn size constancy. One Pygmy, outside of his typical environment, saw a herd of buffalo at a distance and was convinced he was seeing a swarm of insects [[Turnbull 1961](#)]. When driven toward the buffalo, he became frightened and was sure some form of witchcraft was at work as the insects "grew" into buffalo.

Size constancy does not occur only in the real world. Even with first-person video games containing no stereoscopic cues, we still perceive other characters and objects to remain the same size even when their screen size changes. With VR, size constancy can be even stronger due to motion parallax and stereoscopic cues. However, if those different cues are not consistent with each other, then the perception of size constancy can be broken (e.g., objects become smaller on the screen, but our stereo vision tells us the object is coming closer). The same applies for shape constancy and position constancy described below.

10.1.2 Shape Constancy

Shape constancy is the perception that objects retain their shape, even as we look at them from different angles resulting in a changing shape of their image on the retina. For example, we perceive the rim of a coffee mug as a circle even though it is only a circle on the retina when viewed from directly above (otherwise it is an ellipse when viewed at an angle or a line when viewed straight on). The reason for this is experience tells us cups are round so we label in our minds that the rims of cups are circular. Imagine the chaos that would result if we all described objects as the shapes that appear on our retinas with each changing moment.

Not only does shape constancy provide us with consistent perception of objects, but it helps us to determine the orientation and relative depth of those objects.

10.1.3 Position Constancy

Position constancy is the perception that an object appears to be stationary in the world even as the eyes and the head move [[Mack and Herman 1972](#)]. A perceptual matching process between motion on the retina and extra-retinal cues causes the motion to be discounted.

The real world remains stable as one rotates the head. Turning the head 10° to the right causes the visual field to rotate to the left by 10° relative to the head. The **displacement ratio** is the ratio of the angle of environmental displacement to the angle of head rotation. The stability of the real world is independent of head motion and thus has a displacement ratio of zero. If the environment rotates with the head then the displacement ratio is positive. Likewise, if the environment rotates against head direction the displacement ratio is negative.

Minifying glasses, i.e., glasses with concave lenses, increase the displacement ratio and cause objects to appear to move with head rotation. Magnifying glasses decrease the displacement ratio below zero and cause objects to appear to move in the opposite direction as head rotation. Rendering a different field of view than the displayed field of view causes similar effects as minifying/magnifying glasses. For HMDs, this is perceived as scene motion that can result in motion sickness (Chapter [12](#)).

The **range of immobility** is the range of displacement ratio where position constancy is perceived. [Wallach and Kravitz \[1965b\]](#) determined the range of immobility to be 0.04–0.06 displacement ratios wide. If an observer rotates her head 100° to the right, then the environment can move by up to 2°–3° with or against the direction of her head turn without her noticing that movement.

In an untracked HMD, the scene moves with the user's head and has a displacement ratio of one. In a tracked HMD with latency, the scene first moves with the direction of the head turn (a positive displacement ratio), then after the head decelerates the scene moves back toward its correct position, *against* the direction of the head turn (a negative displacement ratio), after the system has caught up to the user [[Jerald 2009](#)].

Lack of position constancy can result from a number of factors and can be a primary cause of motion sickness as discussed in Part [III](#).

10.1.4 Lightness and Color Constancy

Lightness constancy is the perception that the lightness (Section [8.1.2](#)) of an object is more a function of the reflectance of an object and intensity of the surrounding stimuli rather than the amount of light reaching the eye [[Coren et al. 1999](#)]. For example, white paper and black coal still appear to be white and black even as lighting conditions change. Lightness constancy based on relative intensity to surrounding objects holds over a million-to-one range of illumination. Lightness constancy is also affected by shadows, the shape of the object and distribution of illumination on it, and the relative spatial relationship among objects.

Color constancy is the perception for the colors of familiar objects to remain relatively constant even under changing illumination. When a shadow partially falls on a book, we do not perceive the shadowed part of the book to change colors. Like lightness constancy, color constancy is largely due to the relationship with surrounding stimuli. The greater the number of objects and colors in the scene that can serve as comparison stimuli, the greater the perception of color constancy. Prior experience (i.e., top-down processing) also is used to maintain color constancy. A banana almost always appears yellow even in different lighting conditions in part because we know bananas are yellow.

10.1.5 Loudness Constancy

Loudness constancy is the perception that a sound source maintains its loudness even when the sound level at the ear diminishes as the listener moves away from the sound source. Like any constancy, loudness constancy only works up to some threshold.

10.2 Adaptation

Humans must be able to adapt to new situations in order to survive. We must not only adapt to an ever-changing environment, but also adapt to internal changes such as neural processing times and spacing between the eyes, both of which change over a lifetime. When studying perception in VR, investigators must be aware of adaptation, as it may confound measurements and change perceptions.

Adaptation can be divided into two categories: sensory adaptation and perceptual adaptation [[Wallach 1987](#)].

10.2.1 Sensory Adaptation

Sensory adaptation alters a person's sensitivity to detect a stimulus. Sensitivity increases or decreases over time; one starts or stops detecting a stimulus after a period of constant stimulus intensity or after removal of that stimulus. Sensory adaptations are typically localized and only last for short periods of time. Dark adaptation is an example of sensory adaptation.

Dark adaptation increases one's sensitivity to light in dark conditions. **Light adaptation** decreases one's sensitivity to light in light conditions. The sensitivity of the eye changes by as much as six orders of magnitude, depending on the lighting of the environment.

The cones of the eye (Section [8.1.1](#)) reach maximum dark adaptation in approximately 10 min after initiation of dark adaptation. The rods reach maximum dark adaptation in approximately 30 min after initiation of dark adaptation. Complete light adaptation occurs within 5 min after initiation of light adaptation. Dark adaptation causes a person's perception of stimuli to be delayed as discussed in Section [15.3](#). Occasionally presenting bright large stimuli can keep users light-adapted. Rods are relatively insensitive to red light. Thus, if red lighting is used, rods will dark-adapt whereas cones will maintain high acuity.

10.2.2 Perceptual Adaptation

Perceptual adaptation alters a person's perceptual processes. [Welch \[1986 \]](#) defines **perceptual adaptation** to be “a semipermanent change of perception or perceptual-motor coordination that serves to reduce or eliminate a registered discrepancy between or within sensory modalities or the errors in behavior induced by this discrepancy.” Six major factors facilitate perceptual adaptation [[Welch and Mohler 2014](#)]:

- stable sensory rearrangement
- active interaction
- error-corrective feedback
- immediate feedback
- distributed exposures over multiple sessions
- incremental exposure to the rearrangement

Dual adaptation is the perceptual adaptation to two or more mutually conflicting sensory environments that occurs after frequently alternating between those conflicting environments. Going back and forth between a VR experience and the real world may be less of an issue for seasoned users.

One of the first examples of extreme perceptual adaptation goes back to the late 1800s. George Stratton wore inverting goggles that turned his visual world upside-down and right-side-left over a period of eight days [[Stratton 1897](#)]. At first, everything seemed backwards and it was difficult to function, but he slowly adapted over several days' time. Stratton eventually achieved near full adaptation so that the world seemed right-side-up and he could fully function while wearing the inverting goggles.

Position-Constancy Adaptation

The compensation process that keeps the environment perceptually stable during head rotation can be altered by perceptual adaptation. [Wallach \[1987 \]](#) calls this visual-vestibular sensory rearrangement “adaptation to constancy of visual direction,” and in this book it is called **position-constancy adaptation** to be consistent with position constancy described in Section [10.1.3](#).

VOR (Section [8.1.5](#)) adaptation is an example of position-constancy adaptation. Perhaps the most common form of position-constancy adaptation is eyeglasses that at first cause the stationary environment to seemingly move during head movements but is eventually perceptually stabilized after some use. In a more extreme case, [Wallach and Kravitz \[1965a \]](#) built a device where the visual environment moved against the direction of head turns with a displacement ratio (Section [10.1.3](#)) of 1.5. They found that in only 10 minutes time, subjects partially adapted such that perceived motion of the environment during head turns subsided. After removal of the device, subjects reported a negative aftereffect (Section [10.2.3](#)) where the environment appeared to move with head turns and appeared stable with a mean mean displacement ratio of 0.14. [Draper \[1998 \]](#) found similar adaptation for subjects in HMDs when the rendered field of view was intentionally modified to be different from the true field of view of the HMD.

People are able to achieve dual adaptation of position constancy so that they perceive a stable world for different displacement ratios if there is a cue (e.g., glasses on the head or scuba-diver face masks; [Welch 1986](#)).

Temporal Adaptation

As discussed in Section [9.2.2](#), our conscious lags behind actual reality by hundreds of milliseconds. **Temporal adaptation** changes how much time in the past our consciousness lags.

Until relatively recently, no studies have been able to show adaptation to latency (although the Pulfrich pendulum effect from Section 15.3 suggests dark adaptation also causes some form of temporal adaptation). [Cunningham et al. \[2001a\]](#) found behavioral evidence that humans can adapt to a new intersensory visual-temporal relationship caused by delayed visual feedback. A virtual airplane was displayed moving downward with constant velocity on a standard monitor. Subjects attempted to navigate through an obstacle field by moving a mouse that controlled only left/right movement. Subjects first performed the task in a pre-test with a visual latency of 35 ms. Subjects were then trained to perform the same task with 200 ms of additional latency introduced into the system. Finally, subjects performed the task in a post-test with the original minimum latency of 35 ms. The subjects performed much worse in the post-test than in the pre-test.

Toward the end of training, with 235 ms of visual latency, several subjects spontaneously reported that visual and haptic feedback seemed simultaneous. All subjects showed very strong negative aftereffects. In fact, when the latency was removed, some subjects reported the visual stimulus seemed to move before the hand that controlled the visual stimulus, i.e., a reverse of causality occurred, where effect seemed to occur before cause!

The authors reasoned that sensorimotor adaptation to latency requires exposure to the consequences of the discrepancy. Subjects in previous studies were able to reduce discrepancies by slowing down their movements when latency was present, whereas in this study subjects were not allowed to slow the constant downward velocity of the airplane.

These results suggest VR users might be able to adapt to latency, thereby changing latency thresholds over time. Susceptibility to VR sickness in general is certainly reduced for many individuals, and VR latency-induced sickness is likely specifically reduced. However, it is not clear if VR users truly perceptually adapt to latency.

10.2.3 Negative Aftereffects

Negative aftereffects are changes in perception of the original stimulus after the adapting stimulus has been removed. Negative aftereffects occur with both sensory adaptation and perceptual adaptation. These negative aftereffects provide the most common measure of adaptation [[Cunningham et al. 2001a](#)]. Aftereffects as they relate to VR usage are discussed in Section 13.4.

10.3 Attention

Our senses continuously provide an enormous amount of information that we can't possibly perceive all at once. **Attention** is the process of taking notice or concentrating on some entity while ignoring other perceivable information. Attention is far more than simply looking around at things—attending brings an object to the forefront of our consciousness by enhancing the processing and perception of that object. As we pay more attention, we start to notice more detail and perhaps realize the item is different from that which we first thought. Our perception and thinking becomes sharp and clear, and the experience is easy to recall. Attention informs us what is happening, enables us to perceive details, and reduces response time. Attention also helps bind different features, such as shape, color, motion, and location, into perceptible objects (Section 7.2.1).

10.3.1 Limited Resources and Filtering

Our brains have limited physiological capacity and processing capabilities, and attention can be thought of as the allocation of these limited resources. To prevent overloading, the mind-body has evolved to apply more resources to a single area of interest at a time. For example, the fovea is the only location on the retina that provides high-resolution vision. We also filter out events not relevant to our current interests, attending to only one of several available sources of information about the world.

Perceptual capacity is one's total capability for perceiving. **Perceptual load** is the amount of a person's perceptual capacity that is currently being used. Easy or well-practiced tasks have low perceptual loads whereas difficult tasks, such as learning a new skill, have higher perceptual loads.

As discussed in Section 7.9.3, deletion filtering omits certain aspects of incoming data by selectively paying attention to only parts of the world. Things we do not attend to are less distinct and more difficult to remember (or not even perceived as described below as inattentional blindness). Filtering out information allows little of that information to make a lasting impression.

The personal growth industry claims the reticular activating system serves as an automatic goal-seeking mechanism by bringing those things that we think about to the forefront of our attention and filtering out things that we don't care about. Unfortunately, none of these “backed by scientific evidence” claims include references, and a literature search results in little or no scientific evidence supporting such claims. Whereas the **reticular activating system** does enhance wakefulness and general alertness, this part of the brain is only one small piece of the attention puzzle. However, regardless of what parts of the brain control our attention, the concept of perceiving and attending to what we think about is important.

The Cocktail Party Effect

The **cocktail party effect** is the ability to focus one's auditory attention on a particular conversation while filtering out many other conversations in the same room. **Shadowing** is the act of repeating verbal input one is receiving, often among other conversations. Shadowing is easier if the messages come from two different spatial locations, are different pitches, and/or are presented at different speeds.

Inattentional Blindness

Inattentional blindness is the failure to perceive an object or event due to not paying attention and can occur even when looking directly at it.

Researchers created a video of two teams playing a game similar to basketball and observers were told to count the number of passes, causing them to focus their attention on one of the teams [[Simons and Chabris 1999](#)]. After 45 seconds, either a woman carrying an umbrella or a person dressed in a gorilla suit walked through the game over a period of 5 seconds. Amazingly, nearly half of all observers failed to report seeing the woman or the gorilla.

Inattentional blindness also refers to more specific types of perceptual blindness. Discussed below are change blindness, choice blindness, the video overlap phenomenon, and change deafness.

Change blindness. **Change blindness** is the failure to notice a change of an item on a display from one moment to the next. Change blindness can most easily be demonstrated by making a change in a scene when the display is blanked or there is a flash of light between images. Changes in the background are noticed less often than changes in the foreground.

Change blindness can occur even when observers are explicitly told to look for a change. **Continuity errors** are changes between shots in film, and many viewers fail to notice significant changes. Viewers also often miss changes when those changes are introduced during saccadic eye movement.

Most people believe they are good at detecting changes and are not aware of change blindness. This lack of awareness is called **change blindness blindness**.

Choice blindness. **Choice blindness** is the failure to notice that a result is not what the person previously chose. When people believe they chose a result, when in fact they didn't, they often justify with reasons why they "chose" that unchosen result.

The video overlap phenomenon. The **video overlap phenomenon** is the visual equivalent of the cocktail party effect. When observers pay attention to a video that is superimposed on another video, they can easily follow the events in one video but not both.

Change deafness. **Change deafness** is a physical change in an auditory stimulus that goes unnoticed by a listener. Attention influences change deafness.

10.3.2 Directing Attention

Whereas conscious attention is not required to get the "gist" of a scene, we do need to direct attention toward specific parts of a scene while ignoring other parts to perceive details.

Attentional Gaze

Attentional gaze [[Coren et al. 1999](#)] is a metaphor for how attention is drawn to a particular location or thing in a scene. Information is more effectively processed at the place where attention is directed. Attention is like a spotlight or zoom lens that improves processing when directed toward a specific location. Due to the ability to covertly focus attention, attentional gaze does not necessarily correspond to where the eye is looking. Attentional gaze can be directed to a local/small aspect of a scene or to a more global/larger area of a scene. We can also choose to attend to a single feature, such as texture or size, rather than the entirety of an object. However, we usually cannot be drawn to more than one location at any single point in time.

Attentional gaze can shift faster than the eye. Attentional gaze resulting from a visual stimulus starts in the primitive visual pathway (Section [8.1.1](#)) resulting in cells firing as fast as 50 ms after a target is flashed, whereas eye saccades often take more than 200 ms to start moving toward that stimulus.

Auditory attention also acts as if it has a direction and can be drawn to particular spatial locations.

Attention Is Active

We do not passively see or hear, but we actively look or listen in order to see and hear. Our experience is largely what we agree to attend to; we don't just passively sit, letting everything enter into our minds. Instead we actively direct our attention to focus on what is important to us—our interest and goals that guide top-down processing.

Top-down processing as it relates to attention is associated with scene schemas. A **scene schema** is the context or knowledge about what is contained in a typical environment that an observer finds himself in. People tend to look longer at things that seem out of place. People also notice things where they expect them—e.g., we are more likely to look for and notice stop signs at intersections than in the middle of a city block.

Sometimes our past experience or a cue tells us where or when an important event may happen in the near future. We prepare for the event's occurrence by aligning attention with the location and time that we expect. Preparing changes one's attentional state. Auditory cues are especially good for grabbing the attention of a user to prepare for some important event.

Attention also is affected by the task being performed. We almost always look at an object before taking action on that object, and eye movements typically precede motor action by a fraction of a second, providing just-in-time information needed to interact. Observers will also change their attention based on their probabilistic estimates of dynamic events. For example, people will pay more attention to a wild animal or someone acting suspiciously.

Capturing Attention

Attentional capture is a sudden involuntary shift of attention due to salience. **Salience** is the property of a stimulus that causes it to stick out from its neighbors and grab one's attention, such as a sudden flash of light, a bright color, or a loud sound. The attentional capture reflex occurs due to the brain having a need to update as quickly as possible its mental model of the world that has been violated [[Coren et al. 1999](#)]. If the same stimulus repeatedly occurs, then it becomes an expected part of the mental model and the orienting reflex is weakened.

A **saliency map** is a visual image that represents how parts of the scene stand out and capture attention. Saliency maps are created from characteristics such as color, contrast, orientation, motion, and abrupt changes that are different from other parts of the scene. Observers first fixate on areas that correlate with strong areas of the saliency map. After these initial reflexive actions, attention then becomes more active.

Attention maps provide useful information of what users actually look at ([Figure 10.1](#)). Measuring and visualizing attention maps can be extremely useful for determining what actually attracts users' attention. This can be extremely useful as part of the iterative content/interaction creation process by increasing creators' understanding of what draws users toward desired behaviors. Although eye tracking is ideal for creating quality attention maps, less precise attention maps can be created with HMDs by assuming the center of the field of view is what users are generally paying attention to.

Task-irrelevant stimuli are distracting information that is not relevant to the task with which we are involved and can result in decreased performance. Highly salient stimuli are more likely to cause distraction. Task-irrelevant stimuli have more of an impact on performance when a task is easy.



Figure 10.1 3D attention maps show what users most pay attention to. (From [Pfeiffer and Memili \[2015 \]](#))

We constantly monitor the environment by shifting both overtly and covertly.

Overt orienting is the physical directing of sensory receptors toward a set of stimuli to optimally perceive important information about an event. The eyes and head reflexively turn toward (with the head following the eyes) salient stimuli. Orienting reflexes include postural adjustments, skin conductance changes, pupil dilation, decrease in heart rate, pause in breathing, and constriction of the peripheral blood vessels.

Covert orienting is the act of mentally shifting one's attention and does not require changing the sensory receptors. Overt and covert orienting together result in enhanced perception, faster identification, and improved awareness of an event. It is possible to covertly orient without an overt sign that one is doing so. Attentional hearing is more often covert than seeing.

Visual Scanning and Searching

Visual scanning is the looking from place to place in order to most clearly see items of interest on the fovea. A **fixation** is a pause on an item of interest and a saccade is a jerky eye movement from one fixation to the next ([Section 8.1.5](#)). Even though we are often not aware of saccadic eye movements, we make such movements about three times per second. Scanning is a form of overt orienting. **Inhibition of return** is the decreased likelihood of moving the eyes to look at something that has already been looked at.

Searching is the active pursuit of relevant stimuli in the environment, scanning one's sensory world for particular features or combinations of features.

A **visual search** is looking for a feature or object among a field of stimuli and can be either a feature search or a conjunction search. A **feature search** is looking for a particular feature that is distinct from surrounding distractors that don't have that feature (e.g., an angled line). A **conjunction search** is looking for particular combinations of features. Feature searches are generally easier to do than conjunction searches.

For feature searches, the number of distracting items does not affect searching speed and the search is said to operate in parallel. In fact, feature searches sometimes result in the object being searched for to seem to "pop out" from a surrounding blur of irrelevant features. One explanation for this is that our perceptual system groups similar items together and divides scenes into figure and ground ([Section 20.4](#)) where the figure is what is being searched for.

For conjunction searches, we compare each object with what we are looking for and respond only when a match is found. Thus, conjunction searches are sometimes referred to as serial searches. Because conjunction searches are performed serially, they are slower than feature searches

and are dependent on the number of items being searched.

Vigilance is the act of maintaining careful attention and concentration for possible danger, difficulties, or perceptual tasks, often with infrequent events and prolonged periods of time. After an initial time of conducting a vigilant task, sensitivity to stimuli decreases with time due to fatigue.

Flow

When in the state of **flow**, people lose track of time and that which is outside the task being performed [[Csikszentmihalyi 2008](#)]. They are at one with their actions. Flow occurs at just the proper level of difficulty: difficult enough to provide a challenge that requires continued attention, but not so difficult that it invokes frustration and anxiety.

10.4 Action

Perception not only provides information about the environment but also inspires action, thereby turning an otherwise passive experience into an active one. In fact, some researchers believe the purpose of vision is not to create a representation of what is out there but to guide actions. Taking action on information from our senses helps us to survive so we can perceive another day. This section provides a high-level overview of action and how it relates to perception. Part [V](#) focuses on how we can design VR interfaces for users to interact with.

The continuous cycle of interaction consists of forming a goal, executing an action, and evaluating the results (the perception) and is broken down in more detail in Section [25.3](#) . Predicting the result of an action is also an important aspect of perception. See Section [7.4](#) for an example of how prediction and unexpected results can result in the misperception of the world moving about the viewer.

As discussed in Section [9.1.3](#) , intended actions affect distance perception. Performance can also affect perception—for example, successful batters perceive a ball to be bigger than less successful batters, recent tennis winners report lower nets, and successful field goal kickers estimate goal posts to be further apart [[Goldstein 2014](#)]. Perception of how an object might be interacted with (Section [25.2.2](#)) can influence our perception of it. Visual motion can also causevection (the sense of self-motion; Section [9.3.10](#)) and postural instability (Section [12.3.3](#)).

10.4.1 Action vs. Perception

A large body of evidence suggests visually guided action is primarily processed through the dorsal stream and visual perception without action is primarily processed through the ventral pathway (Section [8.1.1](#)). This results in humans using different metrics and different frames of reference depending on if they are purely observing or also taking action. For example, length estimation is biased when visually observing a Ponzo illusion (Section [6.2.4](#)), but the hand is not fooled by the illusion when reaching out to grab the Ponzo lines [[Ganel et al. 2008](#)]. That is, the illusion works for perception (the length estimation task) but not for action (the grasping task).

10.4.2 Mirror Neuron Systems

Although mirror neurons are controversial and individual mirror neurons have not yet been proven to exist in humans, the concept of mirror neurons is still useful to think about how action and perception are related. **Mirror neurons** respond in a similar way whether an action is performed or the same action is observed [[Goldstein 2014](#)]; i.e., mirror neurons seem to respond to what is happening, independently of who is performing the action. Most mirror neurons are specialized to respond to only one type of action, such as grasping or placing an object somewhere, and the specific type of object that is acted upon or observed has little effect on the neuron's response. Mirror neurons may be important for understanding actions and intentions of other people, learning new skills by imitation, and communicating emotions such as empathy.

Regardless of whether or not individual mirror neurons actually exist in humans, the concept is useful to think about when designing VR interactions. For example, watching a computer-controlled character perform an action can be useful for learning new interaction techniques.

10.4.3 Navigation

Navigation is determining and maintaining a course or trajectory to an intended location.

Navigation tasks can be divided into exploration tasks and search tasks. **Exploration** has no specific target, but is used to browse the space and build knowledge of the environment [[Bowman et al. 2004](#)]. Exploration typically occurs when entering a new environment to orient oneself to the world and its features. Searching (Section [10.3.2](#)) has a specific target, where the location may be unknown (**naive search**) or previously known (**primed search**). In the extreme, naive search is simply exploration, albeit with a specific goal.

Navigation consists of wayfinding (the mental component) and travel (the motoric component), and the two are intimately tied together [[Darken and Peterson 2014](#)].

Wayfinding is the mental element of navigation that does not involve actual physical movement of any kind but only the thinking that guides movement. Wayfinding involves spatial understanding and high-level thinking, such as determining one's current location, building a cognitive map of the environment, and planning/deciding upon a path from the current location to a goal location. Landmarks (distinct cues in the environment) and other wayfinding aids are discussed in Sections [21.5](#) and [22.1](#) . Eye movements and fMRI scans show that people look at environmental wayfinding aids more often than other parts of the scene and the brain automatically distinguishes decision point location/cues to guide navigation [[Goldstein 2014](#)].

Wayfinding works through

- **perception** : perceiving and recognizing cues along a route,
- **attention** : attending to specific stimuli that serve as landmarks,
- **memory** : using top-down information stored from past trips through the environment, and
- **combining** all this information to create cognitive maps to help relate what is perceived to where one currently is and wishes to go next.

Travel is the act of moving from one place to another and can be accomplished in many different ways (e.g., walking, swimming, driving, or for VR, pointing to fly). Travel can be seen as a continuum from passive transport to active transport. The extreme of passive transport occurs automatically without direct control from the user (e.g., immersive film). The extreme of active transport consists of physical walking and, to a lesser extent, interfaces that replicate human bipedal motion (e.g., treadmill walking or walking in place). Most VR travel is somewhere in between passive and active, such as point-to-fly or viewpoint control via a joystick. Various VR travel techniques are discussed in Section [28.3](#).

People rely on both optic flow (Section [9.3.3](#)) and egocentric direction (direction of a goal relative to the body—see Sections [9.1.1](#) and [26.2](#)) in a complementary manner as they travel toward a goal, resulting in robust locomotion control [[Warren et al. 2001](#)]. When flow is reduced or distorted, behavior tends to be influenced more by egocentric direction. When there is greater flow and motion parallax, behavior tends to be more influenced by optical flow.

Many senses can contribute to the perception of travel including visual, vestibular, auditory, tactile, proprioceptive, and podokinetic (motion of the feet). Visual cues are the dominant modality for perceiving travel. Chapter [12](#) discusses how motion sickness can result when visual cues do not match other sensory modalities.

Perception: Design Guidelines

Objective reality is not always how we perceive it to be and the mind does not always work in ways that we intuitively assume it works. For creating VR experiences, it is more important to understand how we perceive than it is for creating with any other medium. Studying human perception will help VR creators to optimize experiences, invent innovative techniques that are pleasant for humans, and troubleshoot perceptual problems as they arise.

General guidelines for applying concepts of human perception to creating VR experiences are organized below by chapter.

11.1 Objective and Subjective Reality (Chapter [6](#))

- Study perceptual illusions (Section [6.2](#)) to better understand human perception, identify and test assumptions, interpret experimental results, better design VR worlds, and help understand/solve problems as they come up.
- Make sensory cues consistent in space and time across sensory modalities so that unintended illusions stranger than real-world illusions do not occur (Sections [6.2.4](#) and [7.2.1](#)).

11.2 Perceptual Models and Processes (Chapter [7](#))

- Do not discount visceral (Section [7.7.1](#)) and emotional (Section [7.7.4](#)) processes. Use aesthetic intuition to drive initial attraction and positive emotions.
- Use both positive and negative feedback to minimize user frustration and to modify user behavior (Section [7.7.2](#)).
- Make sure users end a VR experience on an emotional high—the end of the experience is what they are more likely to remember (Section [7.7.4](#)).
- Encourage users to reflect (Section [7.7.3](#)) upon previous positive experiences by providing suggestive cues. For example, provide a review of the highlights of a user’s performance at the end of the VR experience through a third-person view.
- Make interfaces intuitive by enabling complexity to be understood with the simplest mental model possible that can achieve the desired result, but not simpler (Section [7.8](#)). This will help to minimize learned helplessness.
- Use signifying cues, feedback, and constraints to help the user form quality mental models and to make assumptions explicit (Sections [7.8](#) and [25.1](#)).
- Do not fall into the trap of focusing only on one’s own preferred sensory modality. Provide sensory cues across all modalities to cover a wider range of users (Section [7.9.2](#)).
- Use consistency as users will form filters that generalize perception of events and interactions (Section [7.9.3](#)).
- Provide cues to bring back memories in order to associate the experience with a previous real-life or virtual event (Section [7.9.3](#)).
- Be aware that users make decisions early in a VR experience and they are more likely to stick with those decisions rather than change them (Section [7.9.3](#)). Make it easy for users to decide that they like an experience from the very beginning.
- Define personas (Section [31.10](#)) so that the VR experience can be designed to take advantage of the target audience’s filters of general values, beliefs, attitudes, and memories (Section [7.9.3](#)).

11.3 Perceptual Modalities (Chapter [8](#))

- VR creators should be very cognizant of what colors they choose as arbitrary colors can result in unintended consequences (Section [8.1.3](#)).
- Use binaural cues with head pose to give a sense of sound location (Section [8.2.2](#)).
- Use audio when precise timing (Section [8.2](#)) is important and use visuals when precise location is important (Section [8.1.4](#)).

11.4 Perception of Space and Time (Chapter [9](#))

- Think in terms of personal space, action space, and vista space when designing environments and interactions (Section [9.1.2](#)).
- Consider how the relative importance of depth cues varies as a function of distance. Some depth cues are more relevant at different distances (Table 9.2).
- Use a variety of depth cues to improve spatial judgments and enhance presence (Section [9.1.3](#)).

- Use text located in space instead of 2D heads-up displays so the text has a distance from the user and occlusion is handled properly (Section [9.1.3](#)).
- Do not place essential stimuli that must be consistently viewed close to the eye (Section [9.1.3](#)).
- Do not forget that depth is not only a function of presented visual stimuli. Users perceive depth differently depending on their intentions and fear (Section [9.1.3](#)).
- Be careful of having two visuals or sounds occur too closely in time as masking can cause the second stimulus to perceptually erase the earlier stimulus (Section [9.2.2](#)).
- Be careful of sequence and timing of events. Changing the order can completely change the meaning (Section [9.2.1](#)).

11.5 Perceptual Stability, Attention, and Action (Chapter [10](#))

- Include a sufficient number of distance cues and keep them consistent with each other so that size constancy, shape constancy, and position constancy are maintained (Section [10.1](#)).
- Consider using primarily red colors and lighting for dark scenes to maintain dark adaptation while maintaining high visual acuity for foveal vision (Section [10.2.1](#)).
- Don't expect users to notice or remember events just because they are within their field of view (Section [10.3.1](#)). Use salience (e.g., a shiny/colorful object or a spatialized sound) to capture a person's attention (Section [10.3.2](#)).
- Consider getting a user's attention first through spatialized audio to prepare them in advance for an event (Section [10.3.2](#)).
- Attention can also be captured by objects that seem out of place and by putting objects where they expect them (Section [10.3.2](#)).
- To increase performance, remove task-irrelevant stimuli (especially highly salient stimuli and when the task is easy). To make a task more challenging, consider adding distracting stimuli (Section [10.3.2](#)).
- Collect data to build attention maps to determine what actually attracts users' attention (Section [10.3.2](#)).
- To optimize searching, emphasize feature searching by making the searched-for objects distinct from surrounding stimuli (e.g., objects having angled lines where surrounding features don't include angled lines) so the searched-for items seem to "pop out" from a surrounding blur of irrelevant features (Section [10.3.2](#)).
- To make a search task more challenging, emphasize conjunction searching by making the searched-for objects have particular combinations of features and including many similar objects (Section [10.3.2](#)).
- To maximize flow, match difficulty with the individual's skill level (Section [10.3.2](#)).
- Have computer-controlled characters perform interactions to help users learn the same interactions (Section [10.4.2](#)).

PART III

ADVERSE HEALTH EFFECTS

Many users report feeling sick after experiencing VR-type experiences, and this is perhaps the greatest challenge of VR. As an example, over an eight-month period, six people over the age of 55 were taken to the hospital for chest pain and nausea after going on Walt Disney's "Mission: Space," a \$100 million VR ride. This is the most hospital visits for a single ride since Florida's major theme parks agreed in 2001 to report any serious injuries to the state (Johnson, 2005). Disney also began placing vomit bags in the ride. More recently, John Carmack—legendary video game developer and CTO of Oculus—referenced his company's unwillingness to "poison the well" (i.e., to cause motion sickness on a massive scale) by introducing consumer VR hardware too quickly (Carmack, 2015).

Part [III](#) focuses on adverse health effects of VR and their causes. An **adverse health effect** is any problem caused by a VR system or application that degrades a user's health such as nausea, eye strain, headache, vertigo, physical injury, and transmitted disease. Causes include perceived self-motion through the environment, incorrect calibration, latency, physical hazards, and poor hygiene. In addition, such challenges can indirectly be more than a health problem. Users might adapt their behavior to avoid becoming sick, resulting in incorrect training for real-world tasks (Kennedy and Fowlkes, 1992). This is not just a danger for the user, as resulting aftereffects of VR usage (Section [13.4](#)) can lead to public safety issues (e.g., vehicle accidents). We may never eliminate all the negative effects of VR for all users, but by understanding the problems and why they occur, we can design VR systems and applications to, at the very least, reduce their severity and duration.

VR Sickness

Sickness resulting from VR goes by many names including motion sickness, cyber-sickness, and simulator sickness. These terms are often used interchangeably but are not completely synonymous, being differentiated primarily by their causes.

Motion sickness refers to adverse symptoms and readily observable signs that are associated with exposure to real (physical or visual) and/or apparent motion (Lawson, 2014).

Cybersickness is visually induced motion sickness resulting from immersion in a computer-generated virtual world. The word cybersickness sounds like it implies any sickness resulting from computers or VR (the word cyber refers to relating or involving computers and does not refer to motion), but for some reason cybersickness has been defined throughout the literature to be *only motion sickness* resulting from VR usage. That definition is quite limiting as it doesn't account for non-motion-sickness challenges of VR such as accommodation-vergence conflict, display flicker, fatigue, and poor hygiene. For example, a headache resulting from a VR experience due to accommodation-vergence conflict but no scene motion is not a form of cybersickness.

Simulator sickness is sickness that results from shortcomings of the simulation, but not from the actual situation that is being simulated (Pausch et al., 1992). For example, imperfect flight simulators can cause sickness, but such simulator sickness would not actually occur were the user in the real aircraft (although non-simulator sickness might occur). Simulator sickness most often refers to motion sickness resulting from apparent motion that does not match physical motion (rather than physical motion is actually zero motion, user motion, or motion created by a motion platform), but can also be caused by accommodation-vergence conflict (Section [13.1](#)) and flicker (Section [13.3](#)). Because simulator sickness does not include sickness that would result from the experience being simulated, sickness resulting from intense stressful games (e.g., that simulate life-and-death situations) is not simulator sickness. For example, acrophobia (a fear of heights that can cause anxiety) and vertigo resulting from looking down over a virtual cliff are not considered to be a form of simulator sickness (nor cybersickness). Some expert researchers also use the term simulator sickness to refer to sickness resulting from flight simulators implemented with world-fixed display and not sickness resulting from HMD usage (Stanney et al., 1997).

There is currently no generally accepted term that covers all sickness resulting from VR usage, and most users don't know or care about the specific terminology. A general term is needed that is not restricted by specific causes. Thus, this book tends to stay away from the terms cybersickness and simulator sickness, and instead uses the term **VR sickness** or simply "sickness" when discussing any sickness caused by using VR, irrespective of the specific cause of that sickness. Motion sickness is used only when specifically discussing motion-induced sickness.

Overview of Chapters

Part [III](#) is divided into several chapters about the negative health effects of VR that VR creators should be aware of.

Chapter 12, Motion Sickness, discusses scene motion and how such motion can cause sickness. Several theories are described that explain why motion sickness occurs. A unified model of motion sickness is presented that ties together all the theories.

Chapter 13, Eye Strain, Seizures, and Aftereffects, discusses how non-moving stimuli can also cause discomfort and adverse health effects. Known problems include accommodation-vergence conflict, binocular-occlusion conflict, and flicker. Aftereffects are also discussed, which can indirectly result from scene motion as well as non-moving visual stimuli.

Chapter 14, Hardware Challenges, discusses physical challenges involved with the use of VR equipment. This chapter discusses physical fatigue, headset fit, injury, and hygiene.

Chapter 15, Latency, discusses in detail how unintended scene motion and motion sickness can result from latency. Since latency is a primary contributor to VR sickness, latency and its sources are covered in detail.

Chapter 16, Measuring Sickness, discusses how researchers measure VR sickness. Collecting data to determine users' level of motion sickness is important to help improve upon the design of VR applications and to make VR experiences more comfortable. Three methods of measuring VR sickness are covered: The Kennedy Simulator Sickness Questionnaire, postural stability, and physiological measures.

Chapter 17, Summary of Factors That Contribute to Adverse Effects, summarizes the primary contributors to adverse health effects resulting from VR exposure. The factors are divided into system factors, individual user factors, and application design factors. In addition, trade-offs of presence vs. motion sickness are also briefly discussed.

Chapter 18, Examples of Reducing Adverse Effects, provides some examples of specific techniques that are used to make users more comfortable and reduce negative health effects.

Chapter 19, Adverse Health Effects: Design Guidelines, summarizes the previous seven chapters and lists a number of actionable guidelines for improving comfort and reducing adverse health effects.

Motion Sickness

Motion sickness refers to adverse symptoms and readily available observable signs that are associated with exposure to real and/or apparent motion [[Lawson 2014](#)]. Motion sickness resulting from apparent motion (also known as cybersickness) is the most common negative health effect resulting from VR usage. Symptoms of such sickness include general discomfort, nausea, dizziness, headaches, disorientation, vertigo, drowsiness, pallor, sweating, and, in the occasional worst case, vomiting [[Kennedy and Lilienthal 1995](#), [Kolasinski 1995](#)].

Visually induced motion sickness occurs due to visual motion alone, whereas physically induced motion sickness occurs due to physical motion. Visually induced motion can be stopped by simply closing the eyes, whereas physically induced motion cannot.

Travel sickness is a form of motion sickness involving physical motion brought on by traveling in any type of moving vehicle such as a car, boat, or plane. Physically induced motion sickness can also occur from an amusement ride, spinning oneself, or playing on a swing. Although visually induced motion sickness may be similar to physically induced motion sickness, the specific causes and effects can be quite different. For example, vomiting occurs less often from simulator sickness than traditional motion sickness [[Kennedy et al. 1993](#)]. Motion sickness can occur even when the user is not physically moving and results from numerous factors listed in Chapter [15](#).

This chapter discusses visual scene motion that occurs in VR, howvection relates to motion sickness, theories of motion sickness, and a unified model of motion sickness.

12.1 Scene Motion

Scene motion is visual motion of the entire virtual environment that would not normally occur in the real world [[Jerald 2009](#)]. A VR scene might be non-stationary due to

intentional scene motion injected into the system in order to make the virtual world behave differently than the real world (e.g., to virtually navigate through the world as discussed in Section [28.3](#)), and

unintentional scene motion caused by shortcomings of technology (where the scene motion often occurs only when moving the head), such as latency or inaccurate calibration (mismatched field of view, optical distortion, tracking error, incorrect interpupillary distance, etc.).

Although intentional scene motion is well defined by code and unintentional scene motion is well defined mathematically [[Adelstein et al. 2005](#), [Holloway 1997](#), [Jerald 2009](#)], perception of scene motion and how it relates to motion sickness is not as well understood. Researchers do know that noticeable scene motion can degrade a VR experience by causing motion sickness, reducing task performance, lowering visual acuity, and decreasing the sense of presence.

12.2 Motion Sickness and Vection

Vection (an illusion of self-motion; Section [9.3.10](#)) does not necessarily always cause motion sickness. For example, VR often evokesvection when the scene moves at a constant linear velocity, but such motion does not seem to be much of a factor for motion sickness. Yet, when the scene moves in other ways, motion sickness can be a major problem. Why is this the case?

Visual motion perception takes into account sensation from sensory modalities other than just the eyes. When a visual scene moves independently of how a user is physically moving, there can be mismatch between what is seen and what is physically felt. This mismatch is especially discontenting when the virtual motion accelerates because the otolith organs (Section [8.5](#)) do not sense that same acceleration. The mismatch is not nearly as discontenting when the scene moves with constant velocity because the otolith organs do not sense velocity so there is nothing to compare the visual velocity to. Virtual rotations can be more discontenting because the semicircular canals detect both velocity and acceleration [[Howard 1986b](#)]—one is more likely to experience motion sickness from constant angular velocity than from constant linear velocity.

Vection can be a powerful tool in VR to create a sense of travel. However, VR designers should be aware of the consequences of creating such a sense of travel—particularly linear accelerations of the entire scene and any type of angular motion. Contrary to what one might think would add to presence, adding artificial head bobbing should never be used as this can be significantly sickness inducing. Hill terrain and stairs can also be problematic, although to a lesser extent. Lateral movement can also be a problem, presumably because we do not often strafe in the real world [[Oculus Best Practices 2015](#)].

Incorrect scene motion also often occurs in VR due to imperfect implementation. Although vestibular stimulation suppressesvection [[Lackner and Teixeira 1977](#)], scene motion due to latency and head motion together can still causevection and is a major contributor to motion sickness. Misaligned systems can also result in incorrect scene motion as users move their heads.

12.3 Theories of Motion Sickness

There are several theories as to why motion sickness occurs. When designing and evaluating VR experiences, it is useful to consider all these theories.

12.3.1 Sensory Conflict Theory

The **sensory conflict theory** [[Reason and Brand 1975](#)] is the most widely accepted explanation for the initiation of motion sickness symptoms [[Harm 2002](#)]. The theory states that motion sickness may result when the environment is altered in such a way that incoming information across sensory modalities (primarily visual and vestibular) are not compatible with each other and do not match our mental model of expectations.

For VR, visual and auditory cues come from the simulation, whereas vestibular (Section [8.5](#)) and proprioceptive (Section [8.4](#)) cues come from body motion in the real world (motion platforms and haptics could be considered both simulation and real world). As a result, the visual and auditory synthesized cues are often in disagreement with proprioceptive and vestibular cues, and this disagreement is the primary cause of motion sickness. One of the great challenges of VR is to create visual and auditory cues that are consistent with vestibular and proprioceptive cues in order to minimize motion sickness.

The two primary conflicts at the root of motion sickness are the visual and vestibular senses; in most VR applications, the visual system senses motion whereas the vestibular does not. For example, when a user pushes forward on a game controller, it may look like she is moving forward but in reality, she is sitting in the same physical position. When these senses are inconsistent, then motion sickness may result.

12.3.2 Evolutionary Theory

The sensory conflict theory provides an explanation for motion sickness in certain situations, namely, those in which there is a conflict among the senses. However, the theory does not provide an explanation of why motion sickness occurs from an evolutionary point of view.

The **evolutionary theory** [[Treisman 1977](#)], also known as the poison theory, offers a reason for why motion makes us sick: it is critical to our survival to properly perceive our body's motion and the motion of the world around us. If we get conflicting information from our senses, it means something is not right with our perceptual and motor systems. Our bodies have evolved to protect us by minimizing physiological disturbances produced by absorbed toxins. Protection occurs by

- discouraging movement (e.g., laying down until we recover),
- ejecting the poison via sweating and vomiting, and
- causing nausea and malaise (a slight or general feeling of not being healthy or happy) in order to discourage us from ingesting similar toxins in the future.

The emetic response associated with motion sickness may occur because the brain interprets sensory mismatch as a sign of intoxication, and triggers a nausea/vomiting defense program to protect itself.

12.3.3 Postural Instability Theory

Although the sensory conflict theory states that sickness will occur where there is a mismatch between experienced stimuli and expected stimuli, it does not accurately predict which situations will result in a mismatch or how severe sickness will be.

The **postural instability theory** predicts that sickness results when an animal lacks or has not yet learned strategies for maintaining postural stability [[Riccio and Stoffregen 1991](#)]. Riccio and Stoffregen suggest maintaining posture is one of the major goals of animals, and animals tend to become sick in circumstances where they have not learned strategies to maintain balance. They suggest people need to learn new patterns in novel situations to control their postural stability. Until this learning is completed, sickness may result. They do acknowledge there are sickness problems with sensory stimulation in provocative situations, but argue such problems are caused by (not a cause of) postural instability; i.e., postural instability both precedes motion sickness and is necessary to produce symptoms. Furthermore, they argue the length of time one is unstable and the magnitude of that unstableness are predictors of motion sickness and the intensity of symptoms.

A person who thinks she is standing still is not completely motionless, but continuously and subconsciously adjusts muscles to provide the best balance. If the wrong muscle control is applied due to perceiving visual motion that is inconsistent with the physicality of the body, then postural instability occurs [[Laviola 2000](#)]. For example, if the visual scene is moving forward, users often lean forward to compensate [[Badcock et al. 2014](#)]. Since the user is standing still instead of moving forward as perceived, then the leaning forward makes the user less stable; postural instability and motion sickness increase.

Getting one's "sea legs" while traveling by boat to adjust to the boat's motion is an example of learning to cope (an adaptive mechanism) with postural stability in the real world. Similarly, VR users learn how to better control posture and balance over time, resulting in less motion sickness.

12.3.4 Rest Frame Hypothesis

Another criticism of the sensory conflict theory is there are some cue conflicts that do not provoke sickness. The **rest frame hypothesis** states that motion sickness does not arise from conflicting orientation and motion cues directly, but rather from conflicting stationary frames of reference implied by those cues [[Prothero and Parker 2003](#)].

The rest frame hypothesis assumes the brain has an internal mental model of which objects are stationary and which are moving. The **rest frame** is the part of the scene that the viewer considers stationary and judges other motion relative to. In the real world, the visual background/context/ground is normally the rest frame. A room one is in is normally considered stationary, so rooms are almost always used as a basis for spatial orientation and motion. It is more intuitive to think about the motion of a ball with respect to a room than it is to think of the room moving with respect to the ball. To perceive motion, the brain first decides which objects are stationary (i.e., the rest frame). Motion of other

objects and oneself is then perceived to move relative to this rest frame. When new incoming sensory motion cues do not fit the current mental model of the rest frame, motion sickness results. Motion sickness is inextricably tied to one's internal mental model of what should be stable.

Motion between an object and the self can be interpreted as either moving relative to the other. For example, an object moving to the right can also be perceived as the body moving to the left. If the object is considered the rest frame, then it is considered stable and any motion between the object and the self must be due to the self moving, andvection can result.

The rest frame hypothesis allows for the existence of sensory conflicts without causing motion sickness if those conflicting cues are not essential to the stability of the rest frame. However, if cues that are considered to be part of the rest frame are in conflict, then sickness can result. For example, when a user virtually moves through a world, the ground moves beneath his feet. If the ground is considered the rest frame that does not move, then it must be the user that is moving. Vestibular cues inform the brain that he is not actually moving over the ground, so his mental model of the world becomes confused by the conflicting information and motion sickness results.

For vision, humans have a bias toward certain things, such as meaningful parts of the scene and background, as being stationary (Section [20.4.2](#)). For example, larger backgrounds are considered stable and small stimuli less stable as demonstrated by the autokinetic effect (Section [6.2.7](#)).

VR creators should make rest frame cues consistent whenever possible rather than being overly concerned with making all orientation and all motion consistent. The visuals in VR can be divided into two components: one component being content and the other component being a rest frame that matches the users' physical inertial environment [[Duh et al. 2001](#)]. For example, since users are heavily influenced by backgrounds to serve as a rest frame, making the background consistent with vestibular cues, even if other parts of the scene move, can reduce motion sickness. Even when the background can't be made consistent with vestibular cues, then smaller foreground cues can help to serve as a rest frame. [Houben and Bos \[2010 \]](#) created an "anti-seasickness display" for use in enclosed ship cabins where no horizon is visible. The display served as an earth-fixed rest frame that matched vestibular cues better than the rest of the room. Subjects experienced less sickness due to ship motion as a result of the display. Such a rest frame can be thought of as an inverse form of augmented reality where real-world spatially fixed visual cues (even if they are computer-generated) that match vestibular cues are brought into the virtual world (vs. adding virtual-world cues into the real world as is done in augmented reality). Such techniques can be used to reduce VR motion sickness as discussed in Section [18.2](#).

Motion sickness is much less of an issue for augmented reality optical-see-through HMDs (but not video-see-through HMDS) because users can directly see the real world, which acts as a rest frame consistent with vestibular cues. Motion sickness is also reduced when the real world can be seen in the periphery of the outside edges of an HMD.

12.3.5 Eye Movement Theory

The eye movement theory states that motion sickness occurs due to the unnatural eye motion required to keep the scene's image stable on the retina. If the image moves differently than expected, such as often occurs in VR, then a conflict occurs between what the eyes expect and what actually occurs. The eyes then must move differently than they do in the real world in order to stabilize the image on the retina. As a result of this discrepancy, motion sickness results. See Section [7.4](#) for a quick demonstration of how the real world can seem to move, causing a slight sense ofvection due to the eye moving by an abnormal method.

The visual and vestibular systems are strongly connected as demonstrated by gaze-stabilizing eye movements that result from the vestibulo-ocular reflex (VOR) and visual suppression of the VOR through the optokinetic reflex (OKR) (Section [8.1.5](#)). Optokinetic nystagmus has been proposed [[Ebenholtz et al. 1994](#)] to affect the vagal nerve, a mixed nerve that contains both sensory and motor neurons [[Siegel and Sapru 2014](#)]. Such innervations lead to nausea and emesis.

Studies have shown that when observers fixate on a point to reduce eye movement induced by larger moving stimuli (i.e., reduction of optokinetic nystagmus, also discussed in Section [8.1.5](#)), thenvection is enhanced. Such a stationary fixation point may also result in users selecting that point as a rest frame [[Keshavarz et al. 2014](#)]. Thus, supplying fixation points for VR users to focus upon and maintain eye stability may help to reduce motion sickness.

12.4 A Unified Model of Motion Sickness

This section describes a unified model of motion perception and motion sickness that includes the true state of the world and self; sensory input; the estimated state of the world and self; actions; afferent and efferent signals; and the internal mental model of the individual [[Razzaque 2005](#)]. Integrating these various components together is not just important for perceiving motion but is critical for survival, navigation, balance/posture, and stabilizing the eyes so they function properly. The model is consistent with all the previously described theories of motion sickness and can be used to help understand how we perceive motion, whether that motion is perceived as external motion of the world or self-motion, and why motion sickness results. Figure [12.1](#) shows the relationship of the various components, which are discussed below.

12.4.1 The True State of the World

The "truth" of the world (which includes oneself) at any given moment can be considered a state describing everything in it. For this model, the state is the objective motion between the external world (distal stimuli) and the physical self.

12.4.2 Sensory Input

A person's perception of motion is largely a function of information coming in through multiple senses (proximal stimuli transduced into afferent

nerve impulses). Humans rely on visual, auditory, vestibular, proprioceptive, and tactile information in order to maintain balance and orientation, and to distinguish self-motion from external motion.

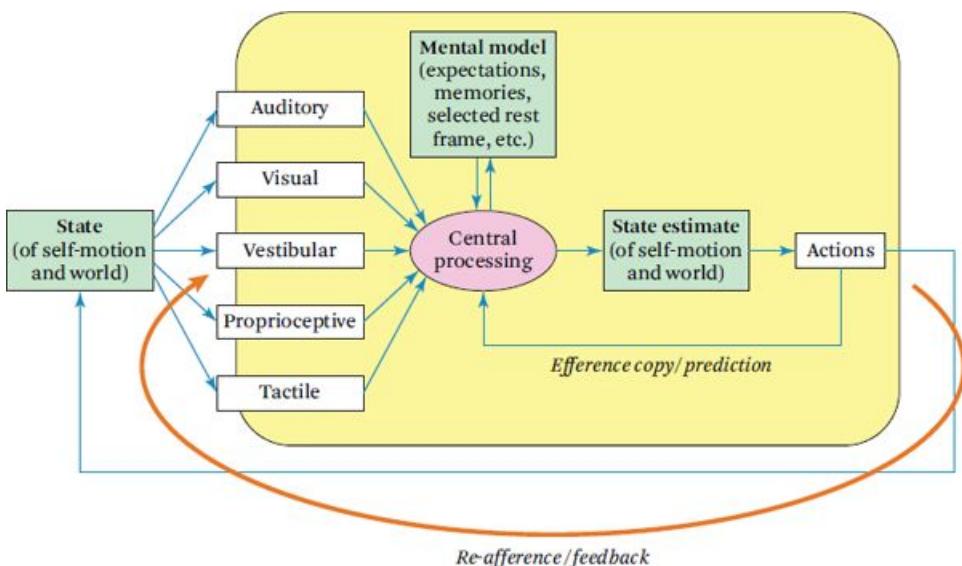


Figure 12.1 A unified model of motion perception and motion sickness. (Adapted from [Razzaque \[2005\]](#))

12.4.3 Central Processing

Brain processes integrate information from multiple senses to perceive what is happening (bottom-up processing). Any single sensory modality has limitations, so the brain integrates information from multiple senses together to form a better estimate of the world and the self. In fact, even information from all the senses is not enough (and sometimes not consistent) to accurately estimate both. Thus, additional input to central processing is needed. Additional input comes from an internal mental model (top-down processing) as well as prediction of sensory input from one's own actions (efference copy).

12.4.4 State Estimation

At any given moment, the brain contains an estimate of the world and the self. Interaction among the various sensory cues helps to disambiguate whether motion is a result of the world moving or the self moving through that world. One's estimate of this motion is constantly tested and revised as a function of central processing. A state estimate that is incorrect can result in visual illusions such asvection. An inconsistent or unstable state estimate (e.g., vestibular cues do not match visual cues) can result in motion sickness. If the state estimate changes to be consistent with itself even though input has not changed, then perceptual adaptation (Section [10.2.2](#)) has occurred. The state estimate includes not only what is currently happening but what is likely to happen.

12.4.5 Actions

The body continuously attempts to self-correct through physical actions in order to fit its state estimate. Several physical actions can occur when attempting to self-correct: (1) one attempts to stabilize himself by changing posture, (2) the eyes rotate to try to stabilize the virtual world's unstable image on the retina through the vestibulo-ocular and optokinetic reflexes, and (3) physiological responses occur, such as sweating and, in extreme cases, vomiting, to remove what the body thinks may be toxins causing the confused state estimate.

Once the mind and body have learned or adapted to new ways of taking action (e.g., a new navigation technique), then action, prediction, feedback, and the mental model become more consistent resulting in less motion sickness.

12.4.6 Prediction and Feedback

The perception of a visually stable world relies on prediction of how incoming sensory information will change due to one's action. For example, as one turns the head and eyes to the left, then copies of outgoing efferent signals (Section [7.4](#)) are sent to central processing that predict a left vestibular response, left-turning neck proprioception, and optical flow moving to the right. If this efference copy matches incoming afferent signals from the senses, then the mind perceives any changes as coming from one's own actions (the afferent signals are then called re-afference since the afference reconfirms that the intended action has taken place). If the efference copy and afference do not match, then the observer perceives the stimulus to have occurred from a change in the external world (e.g., the entire world appears to move in a strange way) instead of from one's actions.

At a higher level, drivers and pilots who control their vehicle motion rarely exhibit motion sickness although the same individuals sometimes report motion sickness when they are passengers [[Rohick and Lubow 1991](#)]. Such results are consistent with voluntary active movement initiating perceptual processes differently than passive movements (Section [10.4](#)) and re-afferent feedback. Similar to driving, VR users are less likely to get sick if they actively control their viewpoint.

12.4.7 Mental Model of Motion

Afference and efference are not perfect and cannot completely explain position constancy and self-motion. There is noise in muscles, external forces sometimes constrain motion, eye saccades cannot be completely predicted, etc.

Each of us has an internal mental model (Section 7.8) of how the world works based on previous experience and future expectations. New incoming information is evaluated in terms of this model, rather than continuously constructing new models from scratch. When new information does not match the model, then disorientation and confusion can result. A part of a person's mental model is transient; the mental model is often reevaluated and relearned. Other parts are more permanent/hardwired assumptions. For example, larger and further-away objects are more likely to be perceived as a rest frame, which can result in induced motion (Section 9.3.5) of smaller objects when the larger, further-away objects are in fact not stationary. Another assumption is consistency—one assumes object properties (Section 10.1) do not change over a short period of time. Breaking these assumptions can lead to strong illusions, even when one cognitively knows exactly which assumptions are false; the perceptual system does not always agree with the rational thinking cortex [[Gregory 1973](#)].

The internal mental model has expectations of how one's actions will affect the world and how stimuli will move. When one acts, the new incoming perceptual cues, efference copy, and mental model are all compared, and the mental model is verified, refined, or invalidated. This is consistent with the rest frame hypothesis—when the mental motion model is invalidated by new sensory cues, the rest frame seems to move and motion sickness can result.

Eye Strain, Seizures, and Aftereffects

Non-moving visual stimuli can also cause discomfort and adverse health effects. Known problems include accommodation-vergence conflict, binocular-occlusion conflict, and flicker. Aftereffects are also discussed, which can indirectly result from scene motion as well as non-moving visual stimuli.

13.1 Accommodation-Vergence Conflict

In the real world, accommodation is closely associated with vergence (Section 9.1.3), resulting in clear vision of closely located objects. If one focuses on a near object, the eyes automatically converge and if one focuses on a distant object, the eyes automatically diverge resulting in a sharp single fused image. The HMD **accommodation-vergence conflict** occurs due to the relationship between accommodation and vergence not being consistent with what occurs in the real world. Overriding the physiologically coupled oculomotor processes of vergence and accommodation can result in eye fatigue and discomfort [Banks et al. 2013].

13.2 Binocular-Occlusion Conflict

Binocular-occlusion conflict occurs when occlusion cues do not match binocular cues, e.g., when text is visible but appears at a distance behind a closer opaque object. This can be quite confusing and uncomfortable for the user.

Many desktop first-person shooter/perspective games contain a 2D overlay/heads-up display to provide information to users (Section 23.2.2). If such games are ported to VR, then it is essential that either such information be removed or the overlay should be given a depth such that information is occluded properly when behind scene geometry.

13.3 Flicker

Flicker (Section 9.2.4) is distracting and can cause eye fatigue, nausea, dizziness, headache, panic, confusion, and in rare cases seizures and loss of consciousness [Laviola 2000 , Rash 2004]. As discussed in Section 9.2.4, the perception of flicker is a function of many variables. Trade-offs in HMD design can result in flicker becoming more perceivable (e.g., wider field of view and less display persistence), so it is essential to maintain high refresh rates so that flicker does not cause health problems. Flicker, however, is not only a function of the refresh rate of the display. Flashing lights in a virtual environment can also cause flicker, and it is important that VR creators be careful not to create such flashing. Dark scenes can also be used to increase dark adaptation, which reduces the perception of flicker.

A **photic seizure** is an epileptic event provoked by flashing lights that can occur in situations ranging from driving by a picket fence in the real world to flashing stimuli in VR, and occurs in about 0.01% of the population [Viirre et al. 2014]. Those experiencing seizures generally show a brief period of absence where they are awake but do not respond. Such seizures do not always result in convulsions so are not always obvious. Repeated seizures can lead to brain injury and a lower threshold for future episodes. Reports of the frequency at which seizures most commonly occur vary widely from a range of 1 to 50 Hz, likely due to different conditions. Because the conditions that will most likely cause seizures within a specific HMD or application are not known, flickering or flashing of lights at any rate should be avoided. Anyone with a history of epilepsy should not use VR.

13.4 Aftereffects

Immediate effects during immersive experiences are not the only dangers of VR usage. Problems may persist after returning to the real world. A **VR aftereffect** (Section 10.2.3) of VR usage is any adverse effect that occurs after VR usage but was not present during VR usage. Such aftereffects include perceptual instability of the world, disorientations, and flashbacks.

About 10% of those using simulators experience negative aftereffects [Johnson 2005]. Those who experience the most sickness during VR exposure usually experience the most aftereffects. Most aftereffect symptoms disappear within an hour or two. The persistence of symptoms longer than six hours has been documented but fortunately remains statistically infrequent.

[Kennedy and Lilienthal \[1995 \]](#) compared effects of simulator sickness to effects of alcohol intoxication using postural stability measurements after VR usage and suggested to restrict subsequent activities in a similar way that is done for alcohol intoxication. There is a danger that such effects could contribute to problems that involve more than the individual, such as vehicular accidents. In fact, many VR entertainment centers require that users not drive for at least 30–45 min after exposure [Laviola 2000]. We, as the VR community, should do the same in doing our part to not only minimize VR sickness but also educate those not familiar with VR sickness of potentially dangerous aftereffects.

13.4.1 Readaptation

As described in Section 10.2.1, a perceptual adaptation is a change in perception or perceptual-motor coordination that serves to reduce or eliminate sensory discrepancies. A **readaptation** is an adaptation back to a normal real-world situation after adapting to a non-normal situation. For example, after a sea voyage, many travelers need to readapt to land conditions, something that can last weeks or even months [Keshavarz et al. 2014].

Over time VR users adapt to shortcomings of VR. After adapting to those shortcomings, when the user returns to the real world, a readaptation occurs due to the real world being treated as something novel compared to the VR world to which the user has adapted. An example of adaptation and readaptation occurs when the rendered field of view is not properly calibrated to match the physical field of view. This causes the image in the HMD to appear distorted. Because of this distortion, when the user turns the head everything but the center of the display will seem to move in an odd way. Once a user becomes accustomed to this odd motion and the world seems stable, then adaptation has occurred. Unfortunately, after adaptation to the incorrect field of view, the inverse may very well occur when the user returns to the real world. That is, the person might perceive an inverse distortion and the world to not appear stable when turning the head. Such adaptation (called position-constancy adaptation; Section 10.2.2) have been found to occur with minifying/magnifying glasses [[Wallach and Kravitz 1965b](#)] and HMDs where the rendered field of view and physical field of view don't match [[Draper 1998](#)].

Until the user has readapted to the real world, aftereffects may persist, such as drowsiness, disturbed locomotor and postural control, and lack of hand-eye coordination [[Kennedy et al. 2014](#), [Keshavarz et al. 2014](#)].

Two approaches are possible for helping with readaptation: natural decay and active readaptation [[DiZio et al. 2014](#)]. **Natural decay** is the refraining of any activity such as relaxing with little movement and the eyes closed. This approach can be less sickness inducing but can prolong the readadaptation period. **Active readadaptation** involves the use of real-life targeted activities aimed at recalibrating the sensory systems affected by the VR experiences, such as hand-eye coordination tasks. Such activity might be more sickness inducing but can speed up the readadaptation process.

Fortunately, as stated in Section 10.2.2, users are able to adapt to multiple conditions (called dual adaptation). Expert users who frequently go back and forth between a VR experience and the real world may have less of an issue with aftereffects and other forms of VR sickness.

Hardware Challenges

It is important to consider physical issues involved with the use of VR equipment. This chapter discusses physical fatigue, headset fit, injury, and hygiene.

14.1 Physical Fatigue

Physical fatigue can be a function of multiple causes including the weight of worn/held equipment, holding unnatural poses, and navigation techniques that require physical motion over an extended period of time.

Some HMDs of the 1990s weighed as much as 2 kg [[Costello 1997](#)]. Fortunately, the weights of HMDs have decreased and are not as much of an issue as they used to be. HMD weight is more of a concern when the HMD center of mass is far from the head center of mass (especially in the horizontal direction). If the HMD mass is not centered, then the neck must exert extra torque to offset gravity. In such a case, the worst problems occur when moving the head. This can tire the neck, which can lead to headaches. The offset mass not only can cause fatigue and headaches but may also modify the way in which the user perceives distance and self-motion [[Willemsen et al. 2009](#)].

Controller weight is typically not a major issue. However, some devices have their center of mass offset more than others. For such devices, trade-offs should be carefully considered before finalizing the decision on selecting controllers.

In the real world, humans naturally rest their arms when working. In many VR applications, there is not as much opportunity to rest the arms on physical objects. **Gorilla arm** is arm fatigue resulting from extended use of gestural interfaces without resting the arm(s). This typically starts to occur within about 10 seconds of having to hold the hand(s) up above the waist in front of the user. Line-of-sight issues should be considered when selecting hand-tracking devices (Section [27.1.9](#)) so users are able to work comfortably with their hands in their lap and/or to the side of their body. Interaction techniques should be designed to not require the hands to be high and in front of the user for more than a few seconds at a time (Section [18.9](#)).

Some forms of navigation via walking (Section [28.3.1](#)) can be exhausting after a period of time, especially when users are expected to walk long distances. Even standing can be tiring for some users, and often they will ask for a seat after some period of time. Sitting options should be provided for longer sessions, and the experience might be optimized for sitting with only occasional need to stand up and walk around. Weight becomes more of an issue for walking interfaces.

14.2 Headset Fit

Headset fit refers to how well an HMD fits a user's head and how comfortable it feels. HMDs often cause discomfort due to pressure or tightness at contact points between the HMD and head. Such pressure points vary depending on the HMD and user but typically occur around the eye sockets and on the ears, nose, forehead, and/or back of the head. Not only can this pressure be uncomfortable on the skin, but it can cause headaches. Headset fit can especially be an issue for those who wear eyeglasses.

Loose-fitting HMDs are also a concern as anything mounted on the head (unless bolted into the skull!) will wiggle if jolted enough because the human head has skin on it and the skin is not completely taut. When the user moves her head quickly, the HMD will move slightly. Such HMD slippage can aggravate the skin and cause small scene motions. Slippage can be reduced by making lighter HMDs but will not completely go away. The small motions can be compensated for by calculating the amount of wiggle and moving the scene appropriately to remain stable relative to the eyes.

Most HMDs are able to be physically adjusted, and users should make sure to adjust the HMD at the start of use as discomfort can increase over time and users are less likely to adjust when they are fully engaged in the experience. Such adjustments do not, however, guarantee a quality fit for all users. Although basic head sizes can be used for different people, head sizes vary considerably more than along a single measurement. For optimal fit, scanning of the head can be used for customized HMD design [[Melzer and Moffit 2011](#)]. Other options currently being investigated by Eric Greenbaum (personal communication, April 29, 2015) of Jema VR (and creator of About Face VR hygiene solutions) include pump-inflatable liners, and ergonomic inserts designed to spread the force of the HMD along the most robust parts of the user's face (forehead and cheekbones), and different forms of foam.

14.3 Injury

Various types of injury are a risk in fully immersive VR due to multiple factors.

Physical trauma is an injury resulting from a real-world physical object that is an increased risk with VR due to being blind and deaf to the real world (e.g., trauma resulting from falling or hitting physical objects). Sitting is preferred from a safety perspective as standing can be more risky due to the potential of colliding with real-world walls/objects/cables and tripping or losing balance. Hanging the wires from the ceiling can reduce problems if the user does not rotate too much. A human **spotter** should closely watch a standing or walking user to help stabilize them if necessary. Harnesses, railing, or padding might also be considered. HMD cables can also be dangerous as there is a possibility of minor whiplash if a short or tangled/stuck cable pulls on the head when the user makes a sudden movement. Even if physical injury does not occur, something as simple as softly bumping against an object or a cable touching the skin can cause a break-in-presence. Active haptic devices can be especially dangerous if the forces are too strong. Hardware designers should design the device so the physical forces are not able to exceed some maximum

force. Physical safety mechanisms can also be integrated into the device.

Repetitive strain injuries are damage to the musculoskeletal and/or nervous systems, such as tendonitis, fibrous tissue hyperplasia, and ligamentous injury, resulting from carrying out prolonged repeated activities using rapid carpal and metacarpal movements [[Costello 1997](#)]. Such injuries have been reported with using standard input devices such as mice, gamepads, and keyboards. Any interaction technique that requires continual repetitive movements is undesirable (as it is for a real-world task). Properly designed VR interfaces have the potential for less repetitive strain injuries than traditional computer mouse/keyboard input due to not being constrained to a plane. Fine 3D translation when holding down a button is stressful (Paul Mlyniec, personal communication, April 28, 2015), so button presses are sometimes performed by the non-translating hand at the cost of the interaction not being as intuitive or direct.

Noise-induced hearing loss is decreased sensitivity to sound that can be caused by a single loud sound or by lesser audio levels over a long period of time. Permanent ear damage can occur with continuous sound exposure (over 8 h) that has an average level in excess of 85 dB or with a single impulse sound in excess of 130 dB [[Sareen and Singh 2014](#)]. Ideally, sounds should not be continuously played and maximum audio levels should be limited to prevent ear damage. Unfortunately, limiting maximum levels can be difficult to guarantee as developers cannot fully control specific systems of individual users. For example, some users may have their own sound system and speakers or have their computer volume gain set differently. However, audio levels can be set to a maximum in software. For example, as users move close to a sound source, code can confirm the audio resulting from the simulation does not exceed a specified level.

14.4 Hygiene

VR hardware is a **fomite** —an inanimate physical object that is able to harbor pathogenic organisms (e.g., bacteria, viruses, fungi) that can serve as an agent for transmitting diseases between different users of the same equipment [[Costello 1997](#)]. Hygiene becomes more important as more users don the same HMD. Such challenges are not unique to VR, as traditional interfaces such as keyboards, mice, telephones, and restrooms have similar issues, albeit interfaces close to or on the face may be more of an issue.

The skin on the face produces oils and sweats. Further, facial skin is populated by a variety of beneficial and pathogenic bacteria. Heat from the electronics and lack of ventilation around the eyes can add to sweating. This may result in an unpleasant user experience and in the most extreme cases can present health risks. Kevin [Williams \[2014 \]](#) divides hygiene into wet, dry, and unusual:

- **Wet** includes sweat, oils, and makeup/hair products.
- **Dry** includes skin, scalp, and hair.
- **Unusual** includes lice/eggs and earwax.

Some professionals clean their 3D polarized glasses with an ultraviolet sterilization process, but unfortunately such a process breaks down plastic material (known as UV degradation) and thus the process can only be used so many times before making the hardware unusable (Eric Greenbaum, personal communication, April 29, 2015). The film industry primarily uses industrial washing machines to clean their glasses. Neither of these methods is appropriate for HMDs (with possibly the exception of the cheaper cell-phone holder HMDs).

Although wiping down the system with alcohol can be helpful, this does not necessarily remove all dangerous biological agents as it can be difficult to clean cracks and porous material (material often used where the skin touches the HMD). Metal, plastic, and rubber are more resistant than porous materials and are often used, but such impermeable material is not as breathable or comfortable for users. The lenses should be wiped with a nonabrasive microfiber cloth with a mild detergent/antibacterial soap and dried before reuse. An inner cleanable cap for each individual user that prevents direct contact with the top of the head, as used by Disney, can be used to further reduce risk [[Pausch et al. 1996](#), [Mine 2003](#)]. For audio, earbuds should not be used for more than one user. Instead larger earphones should be used that cover the entire ear and can be easily wiped clean.

Eric Greenbaum has created multiple removable-from-the HMD solutions to the hygiene problem through About Face VR. About Face VR provides several options of placing material between the HMD and face that drastically reduces transfer of biological agents. Options include personal textile pads/liners that are washed as part of one's laundry, and impermeable liners that can be wiped down with alcohol or other disinfectant for public use.



Figure 14.1 An About Face VR ergonomic insert (center gray) and removable/washable liner (blue). (Courtesy of Jema VR)

If using controllers, provide hand sanitizer for users to sanitize their hands before and after use. Be prepared for the worst case of users becoming

ill by keeping sick bags, plastic gloves, mouthwash, drinks, light snacks, air freshener, and cleaning products nearby. Keep such items out of view as such cues can cause users to think about getting sick, which can cause them to get sick.

Latency

A fundamental task of a VR system is to present worlds that have no unintended scene motion even as the user's head moves. Today's VR applications often produce spatially unstable scenes, most notably due to latency.

Latency is the time a system takes to respond to a user's action, the true time from the start of movement to the time a pixel resulting from that movement responds [[Jerald 2009](#)]. Note depending on display technology different pixels can show at different times and with different rise and fall times (Section [15.4](#)). Prediction and warping (Section [18.7](#)) can reduce effective latency where the goal is to stabilize the scene in real-world space although the actual latency is greater than zero.

15.1 Negative Effects of Latency

For low latencies below ~100 ms, users do not perceive latency directly, but rather the consequences of latency—a static virtual scene appears to be unstable in space when users move their heads [[Jerald 2009](#)]. Latency in an HMD-based system causes visual cues to lag behind other perceptual cues (e.g., visual cues get out of phase with vestibular cues), creating sensory conflict. With latency and some head motion, the visual scene presented to a subject in an HMD moves incorrectly. This unintended scene motion (Section [12.1](#)) due to latency is known as “swimming” and has serious usability consequences. VR latency is a major contributor to motion sickness, and it is essential for VR creators to understand latency in order to minimize it.

In addition to causing scene motion and being a primary cause of motion sickness, latency has other negative effects as described below.

15.1.1 Degraded Visual Acuity

Latency can cause degraded vision. Given some latency, as an HMD user moves her head then stops, the scene's image relative to the head is still moving when the head has stopped moving. If the image velocity on the retina is greater than 2–3°/s, then motion blur and degraded visual acuity result. Typical head motions and latencies result in scene motion greater than 3°/s. For example, a sinusoidal head motion of 0.5 Hz at ±20° and 133 ms of latency results in a peak scene motion of ±8.5°/s [[Adelstein et al. 2005](#)]. It is not known if users' eyes tend to follow a lagging image of the scene, resulting in no retinal image slip, or if their eyes tend to stay stabilized in space, resulting in retinal image slip.

15.1.2 Degraded Performance

The level of latency necessary to negatively impact performance may be different from the level of latency that can be perceived. [So and Griffin \[1995\]](#) studied the relationship between latency and operator learning in an HMD. The task consisted of tracking a target with the head. Training did not improve performance when latency was ≥ 120 ms—subjects were unable to learn to compensate for these latencies in the task.

15.1.3 Breaks-in-Presence

Latency detracts from the sense of presence in HMDs [[Mechan et al. 2003](#)]. Latency combined with head movement causes the scene to move in a way not consistent with the real world. This incorrect scene motion can distract the user, who might otherwise feel present in the virtual environment, and cause her to realize the illusion is only a simulation.

15.1.4 Negative Training Effects

A negative training effect is an unintended decrease in performance that results from training for a task. Latency has been shown to result in negative training effects with desktop displays [[Cunningham et al. 2001a](#)] and driving simulators utilizing large screens [[Cunningham et al. 2001b](#)].

15.2 Latency Thresholds

Often engineers build HMD systems with a goal of “low latency” without specifically defining what that “low latency” is. There is little consensus among researchers what latency requirements should be. Ideally, latency should be low enough so that users are not able to perceive scene motion. Latency thresholds decrease as head motion increases—latency is much easier to detect when making quick head movements.

Various experiments conducted at NASA Ames Research Center [[Adelstein et al. 2003](#), [Ellis et al. 1999](#), [2004](#), [Mania et al. 2004](#)] reported latency thresholds during quasi-sinusoidal head yaw. They found absolute thresholds to vary by 85 ms due to bias, type of head movement, individual differences, differences of experimental conditions, and other known and unknown factors. Surprisingly, they found no differences in latency thresholds for different scene complexities, ranging from single, simple objects to detailed, photorealistic rendered environments. They found just-noticeable differences to be more consistent at ~4–40 ms and that users are just as sensitive to changes in latency with a low base latency as those with a higher base latency. Consistent latency is important even when average latency is high.

After working with NASA on measuring motion thresholds during head turns [[Adelstein et al. 2006](#)], [Jerald \[2009\]](#) built a VR system with 7.4 ms of latency (i.e., system delay) and measured HMD latency thresholds for various conditions. He found his most sensitive subject to be able to discriminate between latency differences as small as 3.2 ms. Jerald also developed a mathematical model relating latency, scene motion, head

motion, and latency thresholds and then verified that model through psychophysical measurements. The model demonstrates that even though our sensitivity to scene motion decreases as head motion increases (Section 9.3.4), our sensitivity to latency increases. This is because as head motion increases, latency-induced scene motion increases more quickly than scene motion sensitivity decreases.

Note the above results apply to fully immersive VR. Optical-see-through displays have much lower latency thresholds (under 1 ms) due to users being able to directly discriminate between real-world cues and the synthesized cues (i.e., judgments are object-relative instead of subject-relative —see Section 9.3.2).

15.3 Delayed Perception as a Function of Dark Adaptation

The **Pulfrich pendulum effect** is a depth illusion that occurs when one eye is dark adapted (Section 10.2.1) by a different amount than the other eye [Gregory 1973] or when one eye is covered with a dark filter [Arditi 1986]. A pendulum swinging in a plane orthogonal to the line of sight appears to swing in an ellipse (i.e., a greater or less distance from the observer at the bottom of its arc, when maximum velocity is reached) instead of a flat arc.

The “dark” eye trades its acuity in space and time for increased light sensitivity. The dark eye’s retina integrates the incoming light over a longer period of time resulting in a time delay. The dark eye perceives the pendulum to be further in the past, and thus further behind its true position than the light eye. As the pendulum speeds up in the middle of the arc, the dark eye sees its position further and further behind the position seen by the light eye. This difference of effective position creates the illusion of an ellipse lying in depth. This illusion of depth is shown in Figure 15.1.

Delayed perception for dark adaptation brings up the question of how/if we perceive latency in VR differently for different lighting conditions. The differences in delay between a light and dark environment can be up to 400 ms [Anstis 1986]. This delay produces a lengthening of reaction time for automobile drivers in dim light [Gregory 1973]. Given that visual delay varies for different amounts of dark adaptation or stimulus intensity, then why do people not perceive the world to be unstable or experience motion sickness when they experience greater delays in the dark or with sunglasses as they do in delayed HMDs? Two possible answers to this question are as follows [Jerald 2009].

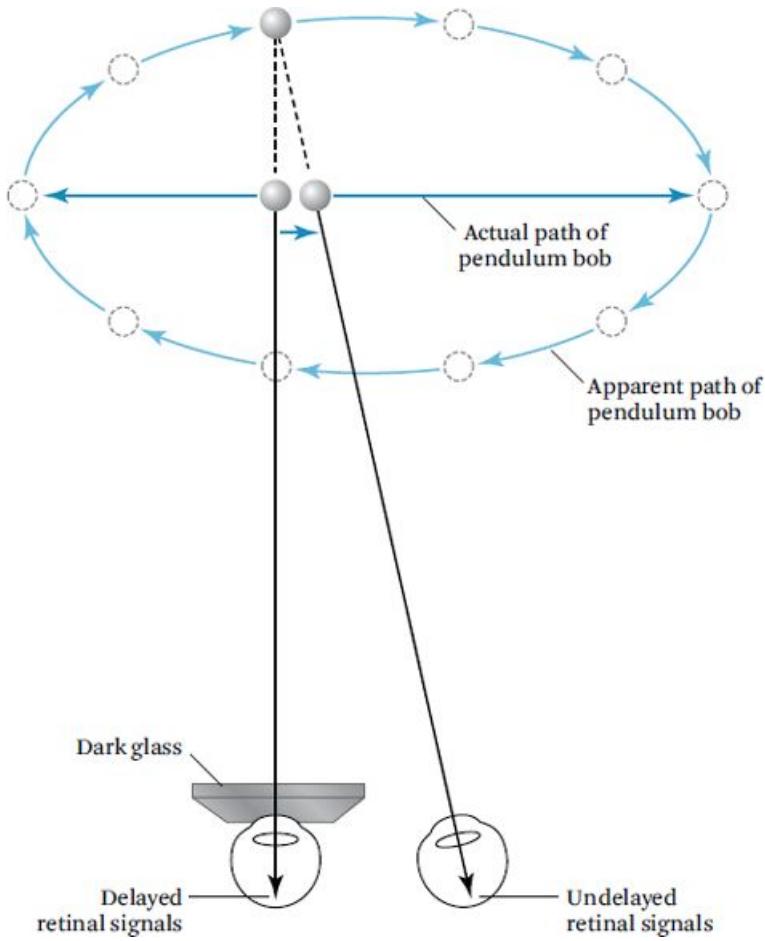


Figure 15.1 The Pulfrich pendulum effect: A pendulum swinging in a straight arc across the line of sight appears to swing in an ellipse when one eye is dark adapted due to the longer delay of the dark-adapted eye. (Based on Gregory [1973].)

- The brain might recognize stimuli to be darker and calibrates for the delay appropriately. If this is true, this suggests users can adapt to latency in HMDs. Furthermore, once the brain understands the relationship between wearing an HMD and delay, position constancy could exist for both the delayed HMD and the real world; the brain would know to expect more delay when an HMD is being worn. However, the brain has had a lifetime of correlating dark adaptation and/or stimulus intensity to delay. The brain might take years to correlate the wearing of an HMD to delay.
- The delay inherent in dark adaptation and/or low intensities is not a precise, single delay. Stimuli 1 ms in duration can be perceived to be as long as 400 ms in duration (Section 9.2.2). This imprecise delay results in motion smear (Section 9.3.8) and makes precise localization of moving

objects more difficult. Perhaps observers are biased to perceive objects to be more stable in such dark situations, but not for the case of brighter HMDs. If this is true, perhaps darkening the scene or adding motion blur to the scene would cause it to appear to be more stable and reduce sickness (although presenting motion blur typically adds to average latency unless prediction is used).

15.4 Sources of Delay

[Miné \[1993\]](#) and [Olano et al. \[1995\]](#) characterize system delays in VR systems and discuss various methods of reducing latency. **System delay** is the sum of delays from tracking, application, rendering, display, and synchronization among components. Note the term system delay is used here because some consider latency to be the effective delay that can be less than system delay, accomplished by using delay compensation techniques (Section 18.7). Thus, system delay is equivalent to true latency, rather than effective latency. Figure 15.2 shows how the various delays contribute to total system delay.

Note that system delay is greater than the inverse of the update rate; i.e., a pipelined system can have a frame rate of 60 Hz but have a delay of several frames.

15.4.1 Tracking Delay

Tracking delay is the time from when the tracked part of the body moves until movement information from the tracker's sensors resulting from that movement is input into the application or rendering component of the VR system. Tracking products can include techniques that complicate delay analysis. For example, many tracking systems incorporate filtering to smooth jitter. If filters are used, the resulting output pose is only partially determined by the most recent tracker reading, so that precise delay is not well defined. Some trackers use different filtering models that are selected depending on the current motion estimate for different situations—delay during some movements may differ from that during other movements. For example, the 3rdTech HiBall tracking system allows the option of using multi-modal filtering. A low-pass filter is used to reduce jitter if there is little movement, whereas a different model is used for larger velocities.

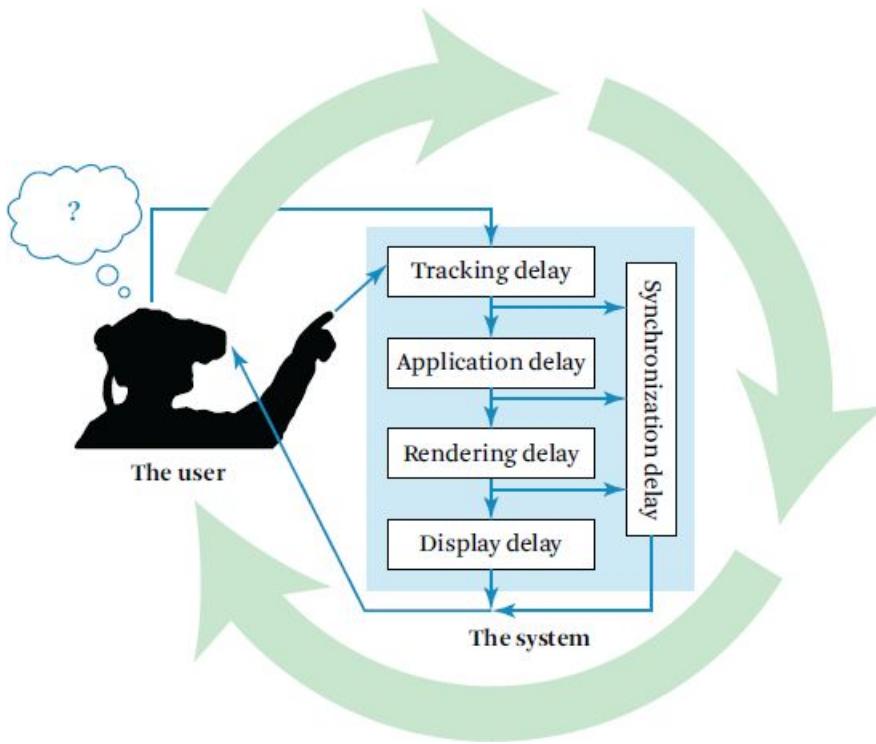


Figure 15.2 End-to-end system delay comes from the delay of the individual system components and from the synchronization of those components. (Adapted from [Jerald \[2009\]](#))

Tracking is sometimes processed on a different computer from the computer that executes the application and renders the scene. In that case network delay can be considered to be a part of tracking delay.

15.4.2 Application Delay

Application delay is the time from when tracking data is received until the time data is passed onto the rendering stage. This includes updating the world model, computing the results of a user interaction, physics simulation, etc. This application delay can vary greatly depending on the complexity of the task and the virtual world. Application processing can often be executed asynchronously from the rest of the system [[Bryson and Johan 1996](#)]. For example, a weather simulation with input from remote sources could be delayed by several seconds and computed at a slow update rate, whereas rendering needs to be tightly coupled to head pose with minimal delay. Even if the simulation is slow, the user should be able to naturally look around and into the slow updating simulation.

15.4.3 Rendering Delay

A **frame** is a full-resolution rendered image. **Rendering delay** is the time from when new data enters the graphics pipeline to the time a new frame resulting from that data is completely drawn. Rendering delay depends on the complexity of the virtual world, the desired quality of the resulting image, the number of rendering passes, and the performance of the graphics software/hardware. The **frame rate** is the number of times the system renders the entire scene per second. **Rendering time** is the inverse of the frame rate, and in non-pipelined rendering systems is equivalent to rendering delay.

Rendering is normally performed on graphics hardware in parallel with the application. For simple scenes, current graphics cards can achieve frame rates of several thousand hertz. Fortunately, rendering delay is what content creators and software developers have the most control over. If geometry is reasonable, the application is optimized, and high-end graphics cards are used, then rendering will be a small portion of end-to-end delay.

15.4.4 Display Delay

Display delay is the time from when a signal leaves the graphics card to the time a pixel changes to some percentage of the intended intensity defined by the graphics card output. Various display technologies are used for HMDs. These include CRTs (cathode ray tubes), LCDs (liquid crystal displays), OLEDs (organic light-emitting diodes), DLP (digital light processing) projectors, and VRD (virtual retinal displays). Different display technologies also have different advantages and disadvantages. See [Jerald \[2009\]](#) for a summary of these display technologies as they relate to delay. The basics of general display delays are discussed below.

Refresh Rate

The **refresh rate** is the number of times per second (Hz) that the display hardware scans out a full image. Note this can be different from the frame rate described in Section [15.4.3](#). The **refresh time** (also known as stimulus onset asynchrony—see Section [9.3.6](#)) is the inverse of the refresh rate (seconds per refresh). A display with a refresh rate of 60 Hz has a refresh time of 16.7 ms. Typical displays have refresh rates from 60 to 120 Hz.

Double Buffering

In order to avoid memory access issues, the display should not read the frame at the same time it is being written to by the renderer. The frame should not scan out pixels until the rendering is complete, otherwise geometric primitives may not be occluded properly. Furthermore, rendering time can vary, depending on scene complexity, implementation, and hardware, whereas the refresh rate is set solely by the display hardware.

A solution to this problem of dual access to the frame can be solved by using a **double-buffer** scheme. The display processor renders to one buffer while the refresh controller feeds data to the display from an alternate buffer. The **vertical sync** signal occurs just before the refresh controller begins to scan an image out to the display. Most commonly, the system waits for this vertical sync to swap buffers. The previously rendered frame is then scanned out to the display while a newer frame is rendered. Unfortunately, waiting for vertical sync causes additional delay, since rendering must wait up to 16.7 ms (for a 60 Hz display) before starting to render a new frame.

Raster Displays

A **raster display** sweeps pixels out to the display, scanning out left to right in a series of horizontal scanlines from top to bottom [[Whitton 1984](#)]. This pattern is called a raster. Timings are precisely controlled to draw pixels from memory to the correct locations on the screen.

Pixels on raster displays have inconsistent intra-frame delay (i.e., different parts of the frame have different delay), because pixels are rendered from a single time-sampled viewpoint but are presented at different times; if the system waits on vertical sync, then the bottommost pixels are presented nearly a full refresh time after the top-most pixels are presented.

Tearing. If the system does not wait for vertical sync to swap buffers, then the buffer swap occurs while the frame is being scanned out to the display hardware. In this case, **tearing** occurs during viewpoint or object motion and appears as a spatially discontinuous image, due to two or more frames (each rendered from a different sampled viewpoint) contributing to the same displayed image. When the system waits for vertical sync to swap buffers, no tearing is evident, because the displayed image comes from a single rendered frame.

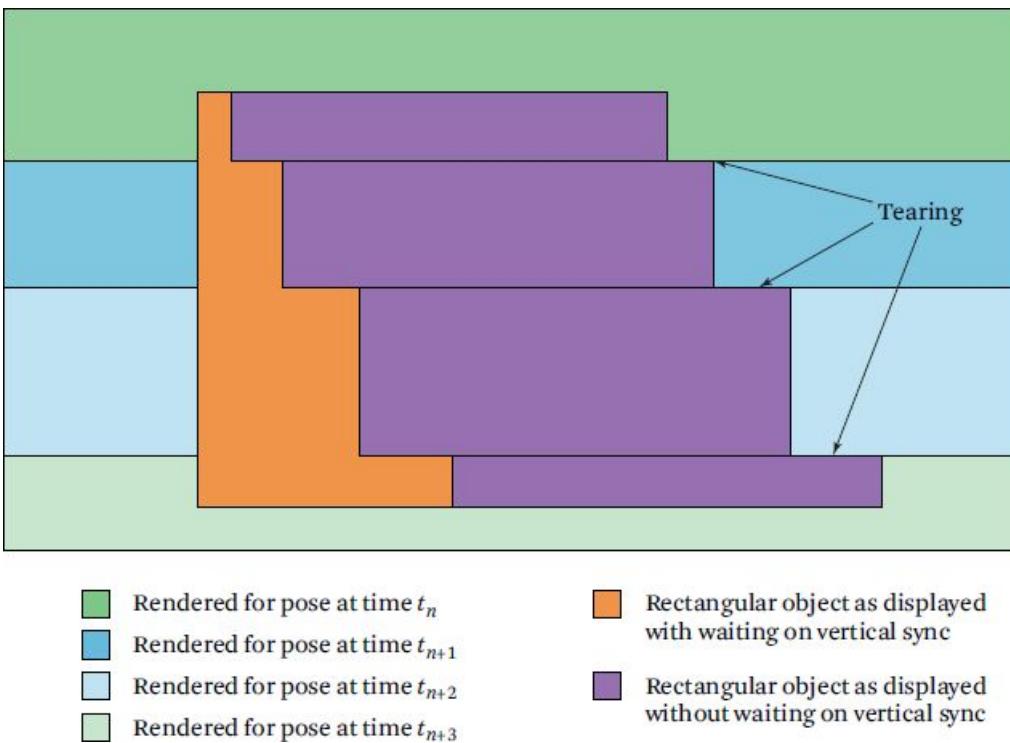


Figure 15.3 A representation of a rectangular object as seen in an HMD as the user is looking from right to left with and without waiting for vertical sync to swap buffers. Not waiting for vertical sync to swap buffers causes image tearing. (Adapted from [Jerald \[2009\]](#))

Figure 15.3 shows a simulated image that would occur with a system that does not wait for vertical sync superimposed over a simulated image that would occur with a system that does wait for vertical sync. The figure shows what a static virtual block would look like when a user is turning her head from right to left. The tearing is obvious when the swap does not wait for vertical sync, due to four renderings of the images with four different head poses. Thus, most current VR systems avoid tearing at the cost of additional and variable intra-frame delay.

Just-in-time pixels. Tearing decreases with decreasing differences of head pose. As the sampling rate of tracking increases and the frame rate increases, pose coherence increases and the tearing becomes less evident. If the system were to render each pixel with the correct up-to-date viewpoint, then the tearing would occur between pixels. The tearing would be small compared to the pixel sizes, resulting in a smooth image without perceptual tearing. [Miné and Bishop \[1993\]](#) call this **just-in-time pixels**.

Rendering could conceivably occur at a rate fast enough that buffers would be swapped for every pixel. Although the entire image would be rendered, only a single pixel would be displayed for each rendered image. However, a 1280×1024 image at 60 Hz would require a frame rate of over 78 MHz—clearly impossible for the foreseeable future using standard rendering algorithms and commodity hardware. If a new image were rendered for every scanline, then the rendering would need to occur at approximately 1/1000 of that or at about 78 kHz. In practice, today's systems can render at rates up to 20 kHz for very simple scenes, which make it possible to show a new image every few scanlines.

With some VR systems, delay caused by waiting for vertical sync can be the largest source of system delay. Ignoring vertical sync greatly reduces overall delay at the cost of image tearing. Note some HMDs rely on vertical sync to perform delay compensation (Section 18.7), so not waiting on vertical sync is not appropriate for all hardware.

Response Time

Response time is the time it takes for a pixel to reach some percentage of its intended intensity. Each technology behaves differently with respect to pixel response time. For example, liquid crystals take time to turn resulting in slow response times—in some cases over 100 ms! Slow-response displays make it impossible to define a precise delay. Typically, a percentage of the intended intensity is defined, although even that is often not precise due to response time being a function of the starting and intended intensity on a per pixel basis.

Persistence

Display persistence is the amount of time a pixel remains on a display before going away. Many displays hold pixel values/intensities until the next refresh and some displays have a slow dissipation time, causing pixels to remain visible even after the system attempts to turn off those pixels. This results in delay not being well defined. Is the delay up to the time that the pixel first appears or up to the average time that the pixel is visible?

A slow response time and/or persistence on the display can appear as motion blur and/or “ghosting.” Some systems only flash the pixels for some portion of the refresh time (e.g., some OLED implementations) in order to reduce motion blur and judder (Section 9.3.6).

15.4.5 Synchronization Delay

Total system delay is not simply a sum of component delays. **Synchronization delay** is the delay that occurs due to integration of pipelined components. Synchronization delay is equal to the sum of component delays subtracted from total system delay. Synchronization delay can be due to components waiting for a signal to start new computations and/or asynchrony among components.

Pipelined components depend upon data from the previous component. When a component starts a new computation and the previous component has not updated data, then old data must be used. Alternatively, the component can in some cases wait for a signal or wait for the input component to finish.

Trackers provide a good example of a synchronization problem. Commercial tracker vendors report their delays as the tracker response time—the minimum delay incurred assumes the tracker data is read as soon as it becomes available. If the tracker is not synchronized with the application or rendering component, then the tracking update rate is also a crucial factor and affects both average delay and delay consistency.

15.5 Timing Analysis

This section presents an example timing analysis and discusses complexities encountered when analyzing system delay. Figure 15.4 shows a timing diagram for a typical VR system where each colored rectangle represents a period of time that some piece of information is being created. In this example, the scanout to the display begins just after the time of vertical sync. The display stage is shown for discrete frames, even though typical displays present individual pixels at different times. The image delays are the time from the beginning of tracking until the start of scanout to the display. Response time of the display is not shown in this diagram.

As can be seen in the figure, asynchronous computations often produce repeated images when no new input is available. For example, the rendering component cannot start computing a new result until the application component provides new information. If new application data is not yet available, then the rendering stage repeats the same computation.

In the figure, the display component displays frame n with the results from the most up-to-date rendering. All the component timings happen to line up fairly well for frame n , and image i delay is not much more than the sum of the individual component delays. Frame $n+1$ repeats the display of an entire frame because no new data is available when starting to display that frame. Frame $n+1$ has a delay of an additional frame time more than the image i delay because a newly rendered frame was not yet available when display began. Frame $n+4$ is delayed even further due to similar reasons. No duplicate data is computed for frame $n+5$, but image $i+2$ delay is quite high because the rendering and application components must complete their previous computations before starting new computations.

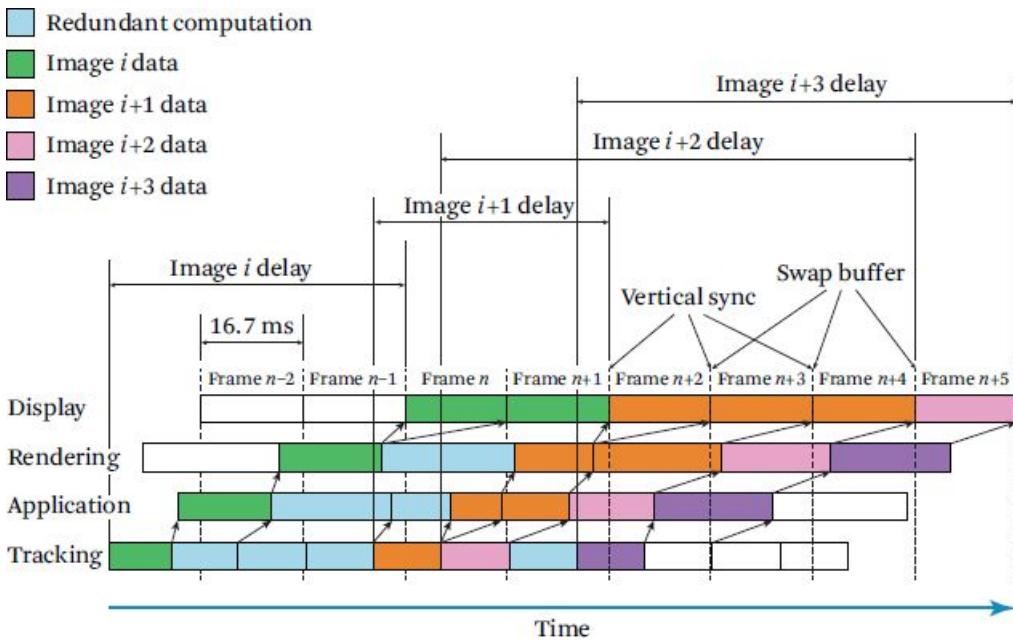


Figure 15.4 A timing diagram for a typical VR system. In this non-optimal example, the pipelined components execute asynchronously. Components are not able to compute new results until preceding components compute new results themselves. (Based on [[Jerald 2009](#)])

15.5.1 Measuring Delays

To better understand system delay, one can measure timings not only for the end-to-end system delay but also for sub-components of the system. Means and standard deviations can be derived from several such measurements. **Latency meters** can be used to measure system delay. For an example of an open-source, open-hardware latency meter, see [Taylor \[2015 \]](#).

Timings can be further analyzed by sampling signals at various stages of the pipeline and measuring the time differences. The parallel port on PCs can be used to output timing signals. These signals are precise in time since there is no additional delay due to a protocol stack; writing to the parallel port is equivalent to writing to memory.

Synchronization delays between two adjacent components of the pipeline can also be measured indirectly. If the delays of individual components

are known, then the sum of two adjacent components can be compared with the measured delay across both components. The difference is the synchronization delay between the two components.

Measuring Sickness

VR sickness can be difficult to measure. One reason for this is VR sickness is polysymptomatic and cannot be measured by a single variable [[Kennedy and Fowlkes 1992](#)]. Another challenge is the large variance between individuals. When effects of VR sickness do occur, they are typically small and weak impacts that disappear quickly upon completing the experience. Because many participants eventually adapt, testing the same participants can be a problem. Thus, researchers often choose to use between-subject study designs (Section [33.4.1](#)). These factors of large individual differences, weak effects, adaptation, and between-subject designs invariably lead to the conclusion that large sample sizes are required. A difficulty with experimenting over such a broad range of users is keeping experiment control conditions consistent.

Questionnaires or symptom checklists are the usual means of measuring VR sickness. These methods are relatively easy to do and have a long history of usage, but are subjective and rely on the participant's ability to recognize and report changes. Postural stability tests and physiological measures are occasionally used due to their objective nature.

16.1 The Kennedy Simulator Sickness Questionnaire

The **Kennedy Simulator Sickness Questionnaire (SSQ)** was created from analyzing data from 1,119 users of 10 US Navy flight simulators. Twenty-seven commonly experienced symptoms were identified that resulted in the creation and validation of 16 symptoms [[Kennedy et al. 1993](#)]. The 16 symptoms were found to cluster into three categories: oculomotor, disorientation, and nausea. The oculomotor cluster includes eye strain, difficulty focusing, blurred vision, and headache. The disorientation cluster includes dizziness and vertigo. The nausea cluster includes stomach awareness, increased salivation, and burping. When taking the questionnaire, participants rank each of the 16 symptoms on a 4-point scale as "none," "slight," "moderate," or "severe." The SSQ results in four scores: total (overall) sickness and three subscores of oculomotor, disorientation, and nausea.

The Kennedy SSQ has become a standard for measuring simulator sickness, and the total severity factor may reflect the best index of VR sickness. See Appendix [A](#) for an example questionnaire containing the SSQ. Although the three subscales are not independent, they can provide diagnostic information as to the specific nature of the sickness in order to provide clues as to what needs improvement in the VR application. These scores can be compared to scores from other similar applications, or can be used as a measure of application improvement over time.

Contrary to suggestions by [Kennedy et al. \[1993 \]](#), the SSQ has often been given before exposure to the VR experience in order to provide a baseline to compare the SSQ given after the experience. [Young et al. \[2007 \]](#) found that when pre-questionnaires suggest VR can make users sick, then the amount of sickness those users report after the VR experience increases. Thus, it is now standard practice to only give a simulator sickness questionnaire after the experience to remove bias, although this comes at the cost of not having a baseline to compare with and risks users not being aware that VR can cause sickness. Those not in their usual state of good health and fitness before starting the VR experience should not use VR and definitely should not be included in the SSQ analysis.

16.2 Postural Stability

Motion sickness can be measured behaviorally through **postural stability tests**. Postural stability tests can be separated into those testing static and dynamic postural stability. A common static test is the Sharpened Romberg Stance: one foot in front of the other with heel touching toe, weight evenly distributed between the legs, arms folded across the chest, chin up. The number of times a subject breaks the stance is the postural instability measure [[Prothero and Parker 2003](#)]. There are currently several commercial systems to assess body sway.

16.3 Physiological Measures

Physiological measures can provide detailed data over the entire VR experience. Examples of physiological data that changes as sickness occurs include heart rate, blink rate, electroencephalography (EEG), and stomach upset [[Kim et al. 2005](#)]. Changes in skin color and cold sweating (best measured via skin conductance) also occur with increased sickness [[Harm 2002](#)]. The direction of change for most physiological measures of sickness is not consistent across individuals, so measuring absolute changes is recommended. Little research currently exists for physiological measures of VR sickness, and there is a need for more cost-effective and objective physiological measures of both resulting VR sickness and a person's susceptibility to it [[Davis et al. 2014](#)].

Summary of Factors That Contribute to Adverse Effects

Adverse health effects due to sensory incongruences do not only occur from VR. About 90% of the population have experienced some form of travel sickness and about 1% of the population get sick from riding in automobiles. [[Lawson 2014](#)]. About 70% of naval personnel experience seasickness, with the greater problems occurring on smaller ships and rougher seas. Up to 5% of those traveling by sea fail to adapt through the duration of the voyage, and some may show effects weeks after the end of the voyage (this is known as land sickness), resulting in postural instability and perceived instability of the visual field during self-movement. VR systems of the 1990s resulted in as many as 80%–95% of users reporting sickness symptoms, and 5%–30% had symptoms severe enough to discontinue exposure [[Harm 2002](#)]. Fortunately, extreme cases of vomiting from VR is not as bad as real-world motion sickness—75% of those suffering from seasickness vomit whereas vomiting in simulators is rare at 1% [[Kennedy et al. 1993](#), [Johnson 2005](#)].

This chapter summarizes contributing factors to adverse health effects, several of which have been previously discussed. It is extremely important to understand these in order to create the most comfortable experiences and to minimize sickness. Contributing factors and interaction between these factors can be complex—some may not affect users until combined with other problems (e.g., fast head motion may not be a problem until latency exceeds some value). The contributing factors are divided into groups of system factors, individual user factors, and application design factors. Each group of factors is an important contributor to VR sickness, as without one of them there would be no adverse effects (e.g., turn off the system, provide no content, or remove the user), but then there would be no VR experience. These lists were created from a number of publications, the author’s personal experience, discussions with others, and editing/additions from John Baker (personal communication, May 27, 2015) of Chosen Realities LLC and former chief engineer of the army’s Dismounted Soldier Training System, a VR/HMD system that trained over 65,000 soldiers with over a million man-hours of use.

17.1 System Factors

System factors of VR sickness are technical shortcomings that technology will eventually solve through engineering efforts. Even small optical misalignment and distortions may cause sickness. Factors are listed in approximate order of decreasing importance.

Latency. Latency in most VR systems causes more error than all other errors combined [[Holloway 1997](#)] and is often the greatest cause of VR sickness. Because of this, latency is discussed in extensive detail in Chapter [15](#).

Calibration. Lack of precise calibration can be a major cause of VR sickness. The worst effect of bad calibration is inappropriate scene motion/scene instability that occurs when the user moves her head. Important calibration includes accurate tracker offsets (the world-to-tracker offset, tracker-to-sensor offset, sensor-to-display offset, display-to-eye offset), mismatched field-of-view parameters, misalignment of optics, incorrect distortion parameters, etc. [Holloway \[1995\]](#) discusses error due to miscalibration in detail.

Tracking accuracy. Low-accuracy head trackers result in seeing the world from an incorrect viewpoint. Inertial sensors used in phones result in yaw drift over time, causing heading to become less accurate over time. A magnetometer can sense the earth’s magnetic field to reduce drift, but rebar in floors, I-beams in tall buildings, computers/monitors, etc. can cause distortion to vary as the user changes position. Error in tracking the hands does not cause motion sickness, although such error can cause usability problems.

Tracking precision. Low precision results in **jitter**, which for head tracking is perceived as shaking of the world. Precision/jitter can be reduced via filtering at the cost of latency.

Lack of position tracking. VR systems that only track orientation (i.e., not position) result in the virtual world moving with the user when the user translates. For example, as the user leans down to pick something off the floor, the floor extends downward. Users can learn to minimize changing position, but even subtle changes in position can contribute to VR sickness. If no position tracking is available, calibrate the sensor-to-eye offset to estimate motion parallax as the user rotates her head.

Field of view. Displays with wide field of view result in more motion sickness due to (1) users being more sensitive tovection in the periphery, (2) the scene moving over a larger portion of the eyes, and (3) scene motion (either unintentional due to other shortcomings or intentional by design—e.g., when navigating through a world). A perfect wide field-of-view VR system with no scene motion would not cause motion sickness.

Refresh rate. High display refresh rates can reduce latency, judder, and flicker. Ideally refresh rates should be as fast as possible.

Judder. Judder is the appearance of jerky or unsmooth visual motion and is discussed in Section [9.3.6](#).

Display response time and persistence. Display response time is the time it takes for a pixel to reach some percentage of its intended intensity (Section [15.4.4](#)). Persistence is the time that pixels remain on the display after reaching their intended intensity. Such timings have trade-offs of judder, motion smear/blur, flicker, and latency—all of which are a problem, so these should all be considered when choosing hardware.

Flicker. Flicker (Sections [9.2.4](#) and [13.3](#)) is distracting, induces eye fatigue, and can cause seizures. As luminance increases, more flicker is perceived. Flicker is also more perceivable with wider-field-of-view HMDs. Displays that have longer response times and longer persistence (e.g., LCDs) can reduce flicker at the cost of adding motion blur, latency, and/or judder.

Vergence/accommodation conflict. For conventional HMDs, accommodative distance remains constant whereas vergence does not (Section [13.1](#)). Visuals should not be placed close to the eye, and if they are then only for short periods of time to reduce strain on the eyes.

Binocular images. HMDs can present monocular (one image for a single eye), binocular (identical images for each eye), or binocular images (two different images for each eye that provide a sense of depth) (Section 9.1.3). Incorrect binocular images can result in double images and eye strain.

Eye separation. For many VR systems, the inter-image distance, inter-lens distance, and interpupillary distance are often in conflict with each other. This conflict can give rise to visual discomfort symptoms and muscle imbalances in users' eyes [[Costello 1997](#)].

Real-world peripheral vision. HMDs that are open in the periphery where users can see the real world reducevection due to the real world acting as a stable rest frame (Section 12.3.4). Theoretically, this factor would not make a difference for an HMD that is perfectly calibrated.

Headset fit. Improperly fitting HMDs (Section 14.2) can cause discomfort and pressure, which can lead to headaches. For those who wear eyeglasses, HMDs that push up against the glasses can be very uncomfortable or even painful. Improper fit also results in larger HMD slippage and corresponding scene motion.

Weight and center of mass. A heavy HMD or HMD with an offset center of mass (Section 14.1) can tire the neck, which can lead to headaches and also modify the way distance and self-motion are perceived.

Motion platforms. Motion platforms can significantly reduce motion sickness if implemented well (Section 18.8). However, if implemented badly then motion sickness can increase due to adding physical motion cues that are not congruent with visual cues.

Hygiene. Hygiene (Section 14.4) is especially important for public use. Bad smells are a problem as nobody wants a stinky HMD and bad smells can cause nausea.

Temperature. Reports of discomfort go up as temperature increases beyond room temperature [[Pausch et al. 1996](#)]. Isolated heat near the eyes can occur due to lack of ventilation.

Dirty screens. Dirty screens can cause eye strain due to not being able to see clearly.

17.2 Individual User Factors

Susceptibility to VR sickness is polygenic (a function of multiple genes, although specific genes contributing to sickness are not yet known), and perhaps the greatest factor for VR sickness is the individual. In fact, [Lackner \[2014\]](#) claims the range of vulnerability varies by a factor of 10,000 to 1. Some people immediately exhibit all the signs of VR sickness within moments of using a VR system, others exhibit only a few signs after some period of use, and some exhibit no signs of sickness after extended use for all but the most extreme conditions. Almost anyone exposed to provocative physical body motion or who is exposed to dramatic inconsistencies between the vestibular and visual systems can to some extent be made sick. The only exception is those who have a total loss of vestibular function [[Lawson 2014](#)].

Not only is there variability to susceptibility of VR sickness between individuals, but individuals are affected differently by different causes. For example, lateral (e.g., strafing) motions make some users uncomfortable [[Oculus Best Practices 2015](#)] where other users have no problem whatsoever with such motions, even though they are susceptible to other provocative motions. Other users seem more susceptible to problems caused by accommodation-vergence conflict—perhaps this is related to the decreased ability to focus with age. Furthermore, new users are almost always more susceptible to motion sickness than more experienced users. Because of these differences, different users often have very different opinions of what is important for reducing sickness. One way to deal with this is by providing configuration options so that users can choose between different configurations to optimize their comfort. For example, a smaller field of view can be more appropriate for new users.

Even though there are dramatic differences between individuals, three key factors seem to affect how much individuals get sick [[Lackner 2014](#)]:

- sensitivity to provocative motion,
- the rate of adaptation, and
- the decay time of elicited symptoms.

These three key factors may explain why some individuals are better able to adapt to VR over a period of time. For example, a person with high sensitivity but with a high rate of adaptation and short decay time can eventually experience less sickness than an individual with moderate sensitivity but a low rate of adaptability and long decay time.

Listed below are descriptions of more specific individual factors that correlate to VR sickness in approximate order of decreasing importance.

Prior history of motion sickness. In the behavioral sciences, past behavior is the best predictor of future behavior. Evidence suggests it is also the case that an individual's history of motion sickness (physically induced or visually induced) predicts VR sickness [[Johnson 2005](#)].

Health. Those not in their usual state of health and fitness should not use VR as unhealthiness can contribute to VR sickness [[Johnson 2005](#)]. Symptoms that make individuals more susceptible include hangover, flu, respiratory illness, head cold, ear infection, ear blockage, upset stomach, emotional stress, fatigue, dehydration, and sleep deprivation. Those under the influence of drugs (legal or illegal) or alcohol should not use VR.

VR experience. Due to adaptation, increased experience with VR generally leads to decreased incidence of VR sickness.

Thinking about sickness. Users who think about getting sick are more likely to get sick. Suggesting to users that they may experience sickness

can cause them to dwell on adverse effects, making the situation worse than if they were not informed [[Young et al. 2007](#)]. This causes an ethical dilemma as not warning users can be a problem as well (e.g., users should be aware of sickness so they can make an informed decision and know to stop the experience if they start to feel ill). Users should be casually warned that people experience some “discomfort” and not to try to “tough it out” if they start to experience any adverse effects, but do not overemphasize sickness so they are less likely to become overly concerned.

Gender. Females are as much as three times more susceptible to VR sickness than males [[Stanney et al. 2014](#)]. Possible explanations include hormonal differences, field-of-view differences (females have a larger field of view which is associated with greater VR sickness), and biased self-report data (males may tend to under-report symptoms).

Age. Susceptibility to physically induced motion sickness is greatest at ages 2–12 (kids are the ones who get sick in a car), decreases rapidly from ages 12–21, and then decreases more slowly thereafter [[Reason and Brand 1975](#)]. VR sickness, however, increases with age [[Brooks et al. 2010](#)]. The reason for the difference of age effects on physically induced motion sickness vs. VR sickness is not clear, but it could be due to confounding factors such as experience (flight hours for flight simulators and video game experience for immersive entertainment), excitement, lack of the ability to accommodate, concentration level, balance capability, etc.

Mental model/expectations. Users are more likely to get sick if scene motion does not match their expectations (Section [12.4.7](#)). For example, gamers accustomed to first-person shooter games may have different expectations than what occurs with commonly used VR navigation methods. Most first-person shooter games have the view direction, weapon/arm, and forward direction always pointing in the same direction. Because of this, gamers often expect their bodies to go wherever their head is pointing, even if walking on a treadmill. When navigation direction does not match viewing/head direction, their mental model can break down and confusion along with sickness can result (sometimes called **Call of Duty Syndrome**).

Interpupillary distance. The distance between the eyes of most adults ranges from 45 to 80 mm and goes as low as 40 mm for children down to five years of age [[Dodgson 2004](#)]. VR systems should be calibrated for each individual’s interpupillary distance. Otherwise, eye strain, headaches, general discomfort, and associated problems can result.

Not knowing what looks correct. Simply telling someone to put an HMD on doesn’t mean that it is properly secured on the head and in the correct position. The user’s eyes may not be in the sweet spots, the headset can be at an odd angle, the headset can be loose, etc. Educating users how to put on the headset and asking the user to look at objects and stating “adjust the headset until comfortable and the objects in the display are crisp and clear” can help reduce negative effects.

Sense of balance. Postural instability (Section [12.3.3](#)) correlates well with motion sickness. Pre-VR postural stability is most strongly associated with the nausea and disorientation subscale of the Kennedy SSQ [[Kolasinski 1995](#)].

Flicker-fusion frequency threshold. A person’s flicker-fusion frequency threshold is the flicker frequency where flicker becomes visually perceptible (Section [9.2.4](#)). There is a wide individual variability in flicker-fusion threshold across individuals due to time of day, gender, age, and intelligence [[Kolasinski 1995](#)].

Real-world task experience. More experienced pilots (i.e., those who have more real flight hours) are more susceptible to simulator sickness [[Johnson 2005](#)]. This is likely due to having higher sensitivity to expectations of how the real world should behave for the task being simulated.

Migraine history. Individuals with a history of migraines have a tendency to experience higher levels of VR sickness [[Nichols et al. 2000](#)].

There are likely several other individual user factors that contribute to VR sickness. For example, it has been hypothesized that ethnicity, concentration, mental rotation ability, and field independence may correlate with susceptibility to VR sickness [[Kolasinski 1995](#)]. However, there has been little conclusive research so far to support these claims.

17.3 Application Design Factors

Even if all technical issues were perfectly solved (e.g., zero latency, perfect calibration, and unlimited computing resources), content can still cause VR sickness. This is due to the human physiological hardware of our bodies and brains instead of the technical limitations of human-made hardware. Note users can drive content via head motions and navigation techniques so those actions are included here. Factors are listed in approximate order of decreasing importance.

Frame rate. A slow frame rate contributes to latency (Section [15.4.3](#)). Frame rate is listed here as an application factor vs. a system factor as frame rate depends on the application’s scene complexity and software optimizations. A consistent frame rate is also important. A frame rate that jumps back and forth between 30 Hz and 60 Hz can be worse than a consistent frame rate of 30 Hz.

Locus of control. Actively being in control of one’s navigation reduces motion sickness (Section [12.4.6](#)). Motion sickness is often less for pilots and drivers than for passive co-pilots and passengers [[Kolasinski 1995](#)]. Control allows one to anticipate future motions.

Visual acceleration. Virtually accelerating one’s viewpoint can result in motion sickness (Sections [12.2](#) and [18.5](#)), so the use of visual accelerations should be minimized (e.g., head bobbing that simulates walking causes oscillating visual accelerations and should never be used). If visual accelerations must be used, the period of acceleration should be as short as possible.

Physical head motion. For imperfect systems (e.g., a system with latency or inaccurate calibration), motion sickness is significantly reduced when individuals keep their head still, due to those imperfections only causing scene motion when the head is moved (Section [15.1](#)). If the system has more than ~30 ms of latency, content should be designed for infrequent and slow head movement. Although head motion could be considered an

individual user factor, it is categorized here since content can suggest head motions.

Duration. VR sickness increases with the duration of the experience [[Kolasinski 1995](#)]. Content should be created for short experiences, allowing for breaks between sessions.

Vection. Vection is an illusion of self-motion when one is not actually physically moving (Section [9.3.10](#)). Vection often, but not always, causes motion sickness.

Binocular-occlusion conflict. Stereoscopic cues that do not match occlusion clues can lead to eye strain and confusion (Section [13.2](#)). 2D heads-up displays that result from porting from existing games often have this problem (Section [23.2.2](#)).

Virtual rotation. Virtual rotations of one's viewpoint can result in motion sickness (Section [18.5](#)).

Gorilla arm. Arm fatigue can result from extended use of above-the-waist gestural interfaces when there is not frequent opportunity to rest the arms (Section [14.1](#)).

Rest frames. Humans have a strong bias toward worlds that are stationary. If some visual cues can be presented that are consistent with the vestibular system (even if other visual cues are not), motion sickness can be reduced (Sections 12.3.4 and 18.2).

Standing/walking vs. sitting. VR sickness rates are greater for standing users vs. sitting users, which correlates with postural stability [[Polonen 2010](#)]. This is in congruence with the postural instability theory (Section [12.3.3](#)). Due to being blind and deaf to the real world, there is also more risk of injury due to falling down and collisions with real-world objects/cables when standing. Requiring users to always stand or navigate excessively via some form of walking also results in fatigue (Section [14.1](#)).

Height above the ground. Visual flow is directly related to velocity and inversely related to altitude. Low altitude in flight simulators has been found to be one of the strongest contributions to simulator sickness due to more of the visual field containing moving stimuli [[Johnson 2005](#)]. Some users have reported that being placed visually at a different height than their real-world physical height causes them to feel ill.

Excessive binocular disparity. Objects that are too close to the eyes result in seeing a double image due to the eyes not being able to properly fuse the left and right eye images.

VR entrance and exit. The display should be blanked or users should shut their eyes when putting on or taking off the HMD. Visual stimuli that are not part of the VR application should not be shown to users (i.e., do not show the operating system desktop when changing applications).

Luminance. Luminance is related to flicker. Dark conditions may result in less problems for displays with little persistence (e.g., CRTs or OLEDs) and a low refresh rate [[Pausch et al. 1992](#)].

Repetitive strain. Repetitive strain injury results from carrying out prolonged repeated activities (Section [14.3](#)).

17.4 Presence vs. Motion Sickness

The rest frame hypothesis (Section [12.3.4](#)) claims the sense of presence (Section [4.2](#)) correlates with the stability of rest frames [[Prothero and Parker 2003](#)]. For example, a perfectly calibrated system with the entire scene acting as a rest frame that is truly stable relative to the real world (i.e., no scene motion) can indeed have high presence. However, this claim does not hold under all conditions. For example, adding non-realistic real-world stabilized cues to the environment as one flies forward in order to reduce sickness (Section [18.2](#)) can reduce presence, especially if the rest of the environment is realistic.

Factors that enhance presence (e.g., wider field of view) can also enhancevection and in many casesvection is a VR design goal. Unfortunately, some of these same presence-enhancing variables can also increase motion sickness. Likewise, design techniques can be used to reduce the sense of self-motion and motion sickness. Such trade-offs should be considered when designing VR experiences.

Examples of Reducing Adverse Effects

This chapter provides several examples of reducing sickness based off of previous chapters. The intention is not only that these examples be used, but that they serve as motivation for understanding the theoretical aspects of VR sickness so that new methods and techniques can be created.

18.1 Optimize Adaptation

Since visual and perceptual-motor systems are modifiable, sickness can be reduced through dual adaptation (Sections 10.2 and 13.4.1). Fortunately, most people are able to adapt to VR, and the more one has adapted to VR the less she will get sick [[Johnson 2005](#)]. Of course, adaptation does not occur immediately and adaptation time depends on the type of discrepancy [[Welch 1986](#)] and the individual. Individuals who adapt quickly might not experience any sickness whereas those who are slower to adapt may become sick and give up before completely adapting [[McCauley and Sharkey 1992](#)]. Incremental exposure and progressively increasing the intensity of stimulation over multiple exposures is a very effective way to reduce motion sickness [[Lackner 2014](#)]. To maximize adaptation, sessions should be spaced 2–5 days apart [[Stanney et al. 2014](#), [Lawson 2014](#), [Kennedy et al. 1993](#)]. In addition to verbally and textually telling novice users to make only slow head motions, they can also be more subtly encouraged to do so through relaxing/soothing environments. Latency should also be kept constant to maximize the chance of temporal adaptation (Section [10.2.2](#)).

18.2 Real-World Stabilized Cues

Seasickness and car sickness are triggered by the motion of the vehicle. When one looks at something that is stable within the vehicle, such as a book when reading, visual cues imply no movement, but the motion of the vehicle stimulates the vestibular system implying there is movement. Vection can be thought of as the inverse of this—the eyes see movement, but the vestibular system is not stimulated. Although inverses of each other, visual and vestibular cues are in conflict in both cases, and as a result similar sickness often occurs.



Figure 18.1 The cockpit from EVE Valkyrie serves as a rest frame that helps users to feel stabilized and reduces motion sickness. (Courtesy of CCP Games)

Looking out the window in a car, or at the horizon in a boat, results in seeing visual motion cues that match vestibular cues. Since vestibular-ocular cues are no longer in conflict, motion sickness is dramatically reduced. Inversely, motion sickness can be reduced by adding stable visual cues to virtual environments (assuming low latency and the system is well calibrated) that act as a rest frame (Section [12.3.4](#)). A cockpit that is stable relative to the real world substantially reduces motion sickness. The virtual world outside the cockpit might change as one navigates their vehicle through space, but the local vehicle interior is stable relative to the real world and the user (Figure [18.1](#)). Visual cues that stay stable relative to the real world and the user but are not a full cockpit can also be used to help the user feel more stabilized in space and reduce motion sickness. Figure [18.2](#) shows an example—the blue arrows are stable relative to the real world no matter how the user virtually turns or walks through the environment. Another example is a stabilized bubble with markings surrounding the user.

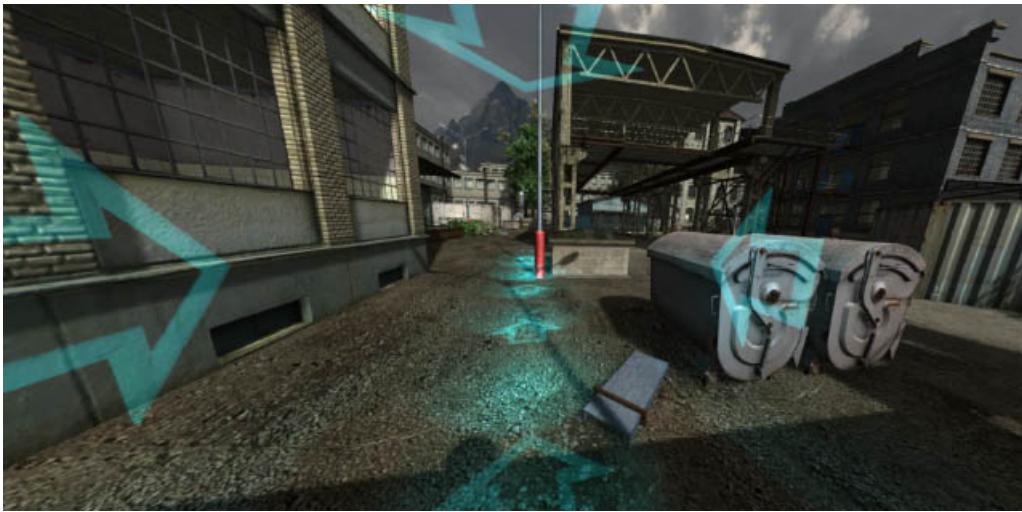


Figure 18.2 For a well-calibrated system with low latency, the blue arrows are stationary relative to the real world and are consistent with vestibular cues even as the user virtually turns or walks around. In this case, the blue arrows also serve to indicate which way the forward direction is. (Courtesy of NextGen Interactions)

18.3 Manipulate the World as an Object

Although quite different than how the real world works, 3D Multi-Touch (Section 28.3.3) enables users to push, pull, turn, and stretch the viewpoint with their hands. Such viewpoint control can be thought about and perceived in two subtly different but important ways:

- *self-motion*, where the user pulls himself through the world, or
- *world motion*, where the user pushes and pulls the world as if it were an object.

While these two alternatives are technically equivalent in terms of relative movement between the user and the world, anecdotal evidence suggests the latter way of thinking can alleviate motion sickness. When the user thinks about viewpoint movement in terms of manipulating the world as an object from a stationary vantage point (i.e., the user considers gravity and the body to be the rest frame), the mind less expects the vestibular system to be stimulated. In fact, Paul Mlyniec (personal communication, April 28, 2015) claims “at least in the absence of roll and pitch, there is no more visual-vestibular conflict than if a user picked up a large object in the real world.” Maintaining a mental model of the world being an object can be a challenge for some users, so visual cues that help with that mental model are important. For example, the center point of rotation and/or scaling at the midpoint between the hands can be important to help users more easily predict motion and give them a sense of direct and active control. Constraints can also be added for novice users. For example, keeping the world vertically upright can help to increase control and reduce disorientation.

18.4 Leading Indicators

When a pre-planned travel path must be used, adding cues that help users to reliably predict how the viewpoint will change in the near future can help to compensate for disturbances. This is known as a **leading indicator**. For example, adding an indicator object that leads a passive user by 500 ms along a motion trajectory of a virtual scene (similar to a driver following another car when driving) alleviates motion sickness [[Lin et al. 2004](#)].

18.5 Minimize Visual Accelerations and Rotations

Constant visual velocity (e.g., virtually moving straight ahead at a constant speed) is not a major contributor to motion sickness due to the vestibular organs not being able to detect linear velocity. However, visual accelerations (e.g., transitioning from no virtual movement to virtually moving in the forward direction) can be more of a problem and should be minimized (assuming a motion platform is not being used). If the design requires visual acceleration, the acceleration should be infrequent and occur as quickly as possible. For example, when transitioning from a stable position to some constant velocity, the period of acceleration to the constant velocity should occur quickly. Change in direction while moving, even when speed is kept constant, is a form of acceleration although it’s not clear if change in direction is as sickness inducing as forward linear acceleration.

Minimizing accelerations and rotations is especially important for passive motion where the user is not actively controlling the viewpoint (e.g., immersive film where the capture point is moving). Slower orientation changes do not seem to be nearly as much of a problem as faster orientation changes. For real-world panoramic capture (Section 21.6.1), unless the camera can be carefully controlled and tested for small comfortable motions, the camera should only be moved with constant linear velocity, if moved at all. Likewise, for computer-generated scenes, the user’s viewpoint should not be attached to the user’s own avatar head when it is possible for the system to affect the avatar’s head pose (e.g., from a physics simulation).

Visual accelerations are less of a problem when the user actively controls the viewpoint (Section 12.4.6) but can still be nauseating. When using the analog sticks on a gamepad or using other steering technique (Section 28.3.2), it is best to have discrete speeds and the transition between

those speeds to occur quickly.

Unfortunately, there are no published numbers as to what is acceptable for VR. VR creators should perform extensive testing across multiple users (particularly users new to VR) to determine what is comfortable with their specific implementation.

18.6 Ratcheting

Denny Unger (personal communication, March 26, 2015) has found that discrete virtual turns (e.g., instantaneous turns of 30°) he calls **ratcheting** result in far fewer reports of sickness compared to smooth virtual turns. This technique results in breaks-in-presence and can feel very strange when one is not accustomed to it, but can be worth the reduction in sickness. Interestingly, not just any angles work. Denny had to experiment and fine-tune the rotation until he reached 30°. Rotation by other amounts turned out to be more sickness inducing.

18.7 Delay Compensation

Because computation is not instantaneous, VR will always have delay. **Delay compensation techniques** can reduce the deleterious effects of system delay, effectively reducing latency [[Jerald 2009](#)]. Prediction and some post-rendering techniques are described below. These can be used together by first predicting and then correcting any error of that prediction with a post-rendering technique.

18.7.1 Prediction

Head-motion prediction is a commonly used delay compensation technique for HMD systems. Prediction produces reasonable results for low system delays or slow head movements. However, prediction increases motion overshoot and amplifies sensor noise [[Azuma and Bishop 1995](#)]. Displacement error increases with the square of the angular head frequency and the square of the prediction interval. Predicting for more than 30 ms can be more detrimental than doing no prediction for fast head motions. Furthermore, prediction is incapable of compensating for rapid transients.

18.7.2 Post-rendering Techniques

Post-rendering latency reduction techniques first render to geometry larger than the final display and then, late in the display process, select the appropriate subset to be presented to the user.

The simplest post-rendering latency reduction technique is accomplished via a **2D warp** (what Oculus calls time warping; Carmack 2015). As done in standard rendering, the system first renders the scene to a single image plane. The pixels are then selected from the larger image plane or reprojected according to the newly desired viewing parameters just before the pixels are displayed [[Jerald et al. 2007](#)].

A cylindrical panorama, as in Quicktime VR [[Chen 1995](#)], or spherical panorama can be used, instead of a single image plane, to remove error due to rotation. However, rendering to a panorama or sphere is computationally expensive, because standard graphics pipelines are not optimized to render to such surfaces.

A compromise between a single image plane and a spherical panorama is a **cubic environment map**. This technique extends the image plane technique by rendering the scene onto the six sides of a large cube [[Greene 1986](#)]. Head rotation simply alters what part of display memory is accessed—no other computation is required.

[Regan and Pose \[1994 \]](#) took environment mapping further by projecting geometry onto concentric cubes surrounding the viewpoint. Larger cubes that contain projected geometry far from the viewpoint do not require re-rendering as often as smaller cubes that are close to the viewpoint. However, in order to minimize error due to large translations or close objects, a full 3D warp [[Mark et al. 1997](#)] or a pre-computed light field [[Regan et al. 1999](#)] is required, although such techniques do produce other types of visual artifacts.

18.7.3 Problems of Delay Compensation

None of the delay compensation techniques described above are perfect. For example, 2D warping a single image plane minimizes error at the center of the display, but error increases toward the periphery [[Jerald et al. 2007](#)]. Ultra-wide-field-of-view HMDs may especially be a problem due to this peripheral error that results in incorrect motion as the head turns and the mind's sensitivity to motion in the periphery. Cubic environment mapping helps by reducing error in the periphery, but such techniques only compensate for rotations, and natural head rotations include some motion parallax. Thus, visual artifacts are worse the closer geometry is to the viewpoint. Furthermore, many latency compensation techniques assume the scene is static; object motion is often not corrected for. Regardless, most latency compensation techniques, if implemented well, can reduce error with visual artifacts being less detrimental than with no correction.

18.8 Motion Platforms

Motion platforms (Section [3.2.4](#)), if implemented well, can help to reduce simulator sickness by reducing the inconsistency between vestibular and visual cues. However, motion platforms do add a risk of increasing motion sickness by (1) causing incongruity between the physical motion and visual motion (the primary cause), and (2) increasing physically induced motion sickness independent of how visuals move (a lesser cause). Thus motion platforms are not a quick fix for reducing motion sickness—they must be carefully integrated as part of the design.

Even passive motion platforms (Section [3.2.4](#)) can reduce sickness (Max Rheiner, personal communication, May 3, 2015). Birdly by Somniacs

(Figure 3.14) is a flying simulation where the user becomes a bird flying above San Francisco. The user controls the tilt of the platform by leaning. The active control of flying with the arms also helps to reduce motion sickness. Motion sickness is further reduced because the user is lying down, so less postural instability occurs than if the user were standing.

18.9 Reducing Gorilla Arm

Using VR devices can become tiresome after extended use due to holding the controllers out and in front of the user (Section 14.1). However, when the interaction is designed so users can mostly interact with their hands in a relaxed manner at their sides and/or in their laps, then they can interact for several hours at a time without experiencing gorilla arm [[Jerald et al. 2013](#)].

18.10 Warning Grids and Fade-Outs

For a user approaching the edge of tracking, a real-world physical object not visually represented in the virtual world (e.g., a wall), or the edge of a safe zone, a **warning grid** (Figure 18.3) or the real world can fade in to provide a cue to the user that he should back off. Whereas such feedback can cause a break-in-presence, it is better than the alternative of the loss of quality tracking that results in a worse break-in-presence and injury.

Once tracking is no longer solid or latency exceeds some value, the system should quickly **fade out** the display to a single color to prevent motion sickness. Such problems are most easily detected by detecting tracking jitter or a decreased frame rate. Alternatively, for systems with low-latency video-see-through capability, the system can quickly fade in the real world.

18.11 Medication

Anti-motion-sickness medication can potentially enhance the rate of adaptation by allowing progressive exposure to higher levels of stimulation without symptoms being elicited. Unfortunately, many of the various medications that have been developed to minimize motion sickness lead to serious side effects that dramatically restrict their field of application [[Keshavarz et al. 2014](#)].

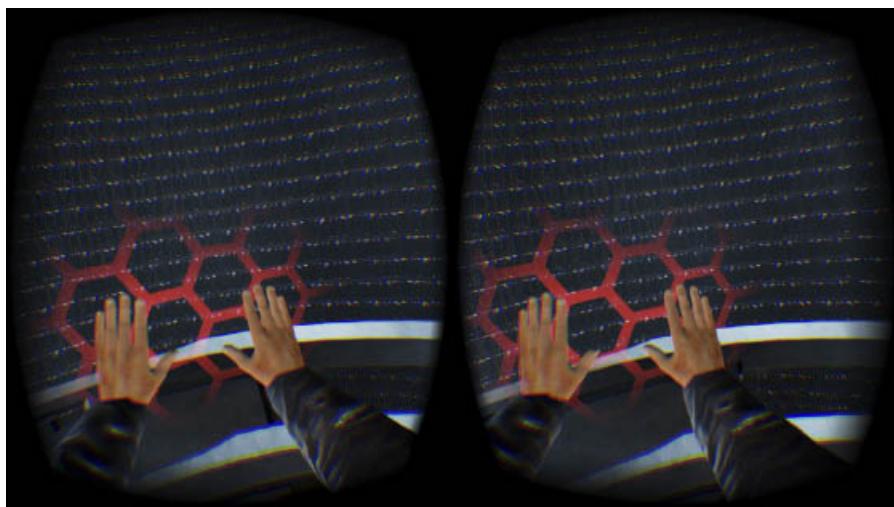


Figure 18.3 A warning grid can be used to inform the user he is nearing the edge of tracked space or approaching a physical barrier/risk. (Courtesy of Sixense)

Ginger has also been touted as a remedy, but its effects are marginal [[Lackner 2014](#)]. Relaxation feedback training has been used in conjunction with incremental exposure to increasingly provocative stimulation as a way of decreasing susceptibility to VR sickness.

In drug studies of motion sickness, there are usually large benefits of placebo effects, typically in the 10%–40% range of the drug effects. Wrist acupressure bands and magnets sold to “alleviate” or prevent motion sickness could potentially provide placebo benefits for some people.

Adverse Health Effects: Design Guidelines

The potential for adverse health effects of VR is the greatest risk for individuals as well as for VR achieving mainstream acceptance. It is up to each of us to take any adverse health effects seriously and do everything possible to minimize problems. The future of VR depends on it.

Practitioner guidelines that summarize chapters from Part III are organized below as hardware, system calibration, latency reduction, general design, motion design, interaction design, usage, and measuring sickness.

19.1 Hardware

- Choose HMDs that have no perceptible flicker (Section [13.3](#)).
- Choose HMDs that are light, have the weight centered just above the center of the head, and are comfortable (Section [14.1](#)).
- Choose HMDs without internal video buffers, with a fast pixel response time, and with low persistence (Section [15.4.4](#)).
- Choose trackers with high update rates (Section [15.4.1](#)).
- Choose HMDs with tracking that is accurate, precise, and does not drift (Section [17.1](#)).
- Use HMD position tracking if the hardware supports it (Section [17.1](#)). If position tracking is not available, estimate the sensor-to-eye offset so motion parallax can be estimated as users rotate their heads.
- If the HMD does not convey binocular cues well, then use binocular cues, as they can be more comfortable even though not technically correct (Section [17.1](#)).
- Use wireless systems if possible. If not, consider hanging the wires from the ceiling to reduce tripping and entanglement (Section [14.3](#)).
- Choose haptic devices that are not able to exceed some maximum force and/or that have safety mechanisms (Section [14.3](#)).
- Add code to prevent audio gain from exceeding a maximum value (Section [14.3](#)).
- For multiple users, use larger earphones instead of earbuds (Section [14.4](#)).
- Use motion platforms whenever possible in a way that vestibular cues correspond to visual motion (Section [18.8](#)).
- Choose hand controllers that do not have line-of-sight requirements so that users can work with their hands comfortably to their sides or in their laps with only occasional need to reach up high above the waist (Section [18.9](#)).

19.2 System Calibration

- Calibrate the system and confirm often that calibration is precise and accurate (Section [17.1](#)) in order to reduce unintended scene motion (Section [12.1](#)).
- Always have the virtual field of view match actual field of view of the HMD (Section [17.1](#)).
- Measure for interpupillary distance and use that information for calibrating the system (Section [17.2](#)).
- Implement options for different users to configure their settings differently. Different users are prone to different sources of adverse effects (e.g., new users might use a smaller field of view and experienced users may not mind more complex motions) (Section [17.2](#)).

19.3 Latency Reduction

- Minimize overall end-to-end delay as much as possible (Section [15.4](#)).
- Study the various types of delay in order to optimize/reduce latency (Section [15.4](#)) and measure the different components that contribute to latency to know where optimization is most needed (Section [15.5](#)).
- Do not depend on filtering algorithms to smooth out noisy tracking data (i.e., choose trackers that are precise so there is no tracker jitter). Smoothing tracker data comes at the cost of latency (Section [15.4.1](#)).
- Use displays with fast response time and low persistence to minimize motion blur and judder (Section [15.4.4](#)).
- Consider not waiting on vertical sync (note that this is not appropriate for all HMDs) as reducing latency can be more important than removing tearing artifacts (Section [15.4.4](#)). If the rendering card and scene content allow it, render at an ultra-high rate that approaches just-in-time pixels to reduce tearing.

- Reduce pipelining as much as possible as pipelining can add significant latency (Sections [15.4](#) and [15.5](#)). Likewise, beware of multi-pass rendering as some multi-pass implementations can add delay.
- Be careful of occasional dropped frames and increases in latency. Inconsistent latency can be as bad as or worse than long latency and can make adaptation difficult (Section [18.1](#)).
- Use latency compensation techniques to reduce effective/perceived latency, but do not rely on such techniques to compensate for large latencies (Section [18.7](#)).
- Use prediction to compensate for latencies up to ~30 ms (Section [18.7.1](#)).
- After predicting, correct for prediction error with a post-rendering technique (e.g., 2D image warping; Section [18.7.2](#)).

19.4 General Design

- Minimize visual stimuli close to the eyes as this can cause vergence/accommodation conflict (Section [13.1](#)).
- For binocular displays, do not use 2D overlays/heads-up displays. Position overlays/text in 3D at some distance from the user (Section [13.2](#)).
- Consider making scenes dark to reduce the perception of flicker (Section [13.3](#)).
- Avoid flashing lights 1 Hz or above anywhere in the scene (Section [13.3](#)).
- To reduce risk of injury, design experiences for sitting. If standing or walking is used, provide protective physical barriers (Section [14.3](#)).
- For long walking or standing experiences, also provide a physical place to sit, with visual representation in the virtual world so users can find it (Section [14.1](#)).
- Design for short experiences (Section [17.3](#)).
- Fade in a warning grid when the user approaches the edge of tracking, a real-world physical object not visually represented in the virtual world, or the edge of a safe zone (Section [18.10](#)).
- Fade out the visual scene if latency increases or head-tracking quality decreases (Section [18.10](#)).

19.5 Motion Design

- If the highest-priority goal is to minimize VR sickness (e.g., when the intended audience is non-experienced VR users), then do not move the viewpoint in any way whatsoever that deviates from actual head motion of the user (Chapter [12](#)).
- If latency is high, do not design tasks that require fast head movements (Section [18.1](#)).

Both Active and Passive Motion

- Be careful of adding visual motion that results in users trying to incorrectly physically balance themselves in order to compensate for the visual motion (Section [12.3.3](#)).
- Design for the user to be seated or lying down to reduce postural instability (Section [12.3.3](#)).
- Focus on making rest frame cues consistent with vestibular cues rather than being overly concerned with making all orientation and all motion consistent (Section [12.3.4](#)).
- Consider dividing visuals into two components, one component that represents content and one component that represents the rest frame (Section [12.3.4](#)).
- When a large portion of the background is visible and realism is not required, consider making the entire background a real-world stabilized rest frame that matches vestibular cues (Section [12.3.4](#)).
- Use a stable cockpit for vehicle experiences or consider using non-realistic world-stabilized cues for non-vehicle experiences (Section [18.2](#)).
- Perform extensive testing across multiple users (particularly users new to VR) to determine what motions are comfortable with the specific implementation or experience (Chapter [16](#)).
- Do not put the viewpoint near the ground when the viewpoint moves in any way other than from direct head motion (Section [17.3](#)).

Passive Motion Design

- Only passively change the viewpoint when absolutely required (Section [12.4.6](#)).

- If passive motion is required, then minimize any motion other than linear velocity (Sections [12.2](#) and [18.5](#)).
- Never add visual head bobbing (Section [12.2](#)).
- Do not attach the viewpoint to the user's own avatar head if the system can affect the pose of the avatar's head (Section [18.5](#)).
- For real-world panoramic capture, unless the camera can be carefully controlled and tested for small comfortable motions, the camera should only be moved with constant linear velocity, if moved at all (Section [18.5](#)).
- Slower orientation changes are not as much of a problem as faster orientation changes (Section [18.5](#)).
- If virtual acceleration is required, ramp up to a constant velocity quickly (e.g., instantaneous acceleration is better than gradual change in velocity) and minimize the occurrence of these accelerations (Section [18.5](#)).
- Use leading indicators so the user can know what motion is about to occur (Section [18.4](#)).

Active Motion Design

- Visual accelerations are less of a problem when the user actively controls the viewpoint (Section [12.4.6](#)) but can still be nauseating.
- When using analog sticks on a gamepad to navigate or when using other steering technique, it is best to have discrete speeds and the transition between those speeds to occur quickly (Section [18.5](#)).
- Design for physical rotation instead of virtual rotation whenever possible (Section [12.2](#)). Consider using a swivel chair (although this can be troublesome for wired systems).
- Considering offering a non-realistic ratchet mode for angular rotations (Section [18.6](#)).
- Be careful of the viewpoint moving up and/or down due to hilly terrain, stairs, etc. (Section [12.2](#)).
- For 3D Multi-Touch, add cues that imply the world is an object to be manipulated instead of moving oneself through the environment (Section [18.3](#)).

19.6 Interaction Design

- Design interfaces so users can work with their hands comfortably at their sides or in their laps with only occasional need to reach up high above the waist (Section [18.9](#)).
- Design interactions to be non-repetitive to reduce repetitive strain injuries (Section [14.3](#)).

19.7 Usage

Safety

- Consider forcing users to stay within safe areas via physical constraints such as padded circular railing (Section [14.3](#)).
- For standing or walking users, have a human spotter carefully watch the users and help stabilize them when necessary (Section [14.3](#)).

Hygiene

- Consider providing to users a clean cap to wear underneath the HMD and removable/cleanable padding that sits between the face and the HMD (Section [14.4](#)).
- Wipe down equipment and clean the lenses after use and between users (Section [14.4](#)).
- Keep sick bags, plastic gloves, mouthwash, drinks, light snacks, and cleaning products nearby, but keep these items hidden so they don't suggest to users that they might get sick (Section [14.4](#)).

New Users

- For new users, be especially conservative with presenting any cues that can induce sickness.
- Consider decreasing the field of view for new users (Section [17.2](#)).
- Unless using gaze-directed steering (Section [28.3.2](#)), tell gamers something like "Forget your video games. If you think you're going to move where your head is pointing, then think again. This isn't Call of Duty." (Section [17.3](#)).
- Encourage users to start with only slow head motions (Section [18.1](#)).

- Begin with short sessions and/or use timeouts/breaks excessively. Do not allow prolonged sessions until the user has adapted (Sections [17.2](#) and [17.3](#)).

Sickness

- Do not allow anyone to use a VR system that is not in their usual state of health (Section [17.2](#)).
- Inform users in a casual way that some users experience discomfort and warn them to stop at the first sign of discomfort, rather than “tough it out.” Do not overemphasize adverse effects (Sections [16.1](#) and [17.2](#)).
- Respect a user’s choice to not participate or to stop the experience at any time (Section [17.2](#)).
- Do not present visuals while the user is putting on or taking off the HMD (Section [17.2](#)).
- Do not freeze, stop the application, or switch to the desktop without first having the screen go blank or having the user close her eyes (Section [17.3](#)).
- Carefully pay attention for early warning signs of VR sickness, such as pallor and sweating (Chapter [12](#)).

Adaptation and Readaptation

- For faster readaptation to the real world after VR usage, use active readaptation by providing activities targeted at recalibrating the user’s sensory systems (Section [13.4.1](#)).
- For reducing readaptation discomfort after VR usage, use natural decay by having the user gradually readapt to the real world—encourage the user to sit and relax with little movement and eyes closed (Section [13.4.1](#)).
- After VR usage, inform users not to drive, operate heavy machinery, or participate in high-risk behavior for at least 30–45 min and after all aftereffects have passed (Section [13.4](#)).
- To maximize adaptation, use VR at spaced intervals of 2–5 days (Section [18.1](#)).

19.8 Measuring Sickness

- For easiest data collection, use symptom checklists or questionnaires. The Kennedy Simulator Sickness Questionnaire is the standard (Section [16.1](#)).
- Postural stability tests are also quite easy to use if the person administering the test is trained (Section [16.2](#)).
- For objectively measuring sickness, consider using physiological measures (Section [16.3](#)).

PART IV

CONTENT CREATION

We graphicists choreograph colored dots on a glass bottle so as to fool eye and mind into seeing desktops, spacecraft, molecules, and worlds that are not and never can be .

—Frederick P. Brooks, Jr. ([1988](#))

Humans have been creating content for thousands of years. Cavemen painted inside caves. The ancient Egyptians created everything from pottery to pyramids. Such creations vary widely, from the magnificent to the simple, across all cultures and ages, but beauty has always been a goal. Philosophers and artists argue that delight is equal to usefulness, and this is even truer for VR.

VR is a relatively new medium and is not yet well understood. Most other disciplines have been around much longer than VR and although many aspects of other crafts are very different from VR, there are still many elements of those disciplines that we can learn from. For content creation, studying fields such as architecture, urban planning, cinema, music, art, set design, games, literature, and the sciences can add great benefit to VR creations. The chapters within Part [IV](#) utilize some concepts from these disciplines that are especially pertinent to VR. Different pieces from different disciplines all must come together in VR to form the experience—the story, the action and reaction, the characters and social community, the music and art. Integrating these different pieces together, along with some creativity, can result in experiences that are more than the sum of the individual pieces.

Part [IV](#) consists of five chapters that focus on creating the assets of virtual worlds.

Chapter 20, High-Level Concepts of Content Creation, discusses core concepts of content creation common to the most engaging VR experiences: the story, the core experience, conceptual integrity, and gestalt principles.

Chapter 21, Environmental Design, discusses design of the environment ranging from different types of geometry to non-visual cues. Designing the environment is more than just creating pleasing stimuli—the design of the environment provides clues to affordances, constrains interaction, and enhances wayfinding.

Chapter 22, Affecting Behavior, discusses how content creation can affect the states and actions of users. Influencing behavior is especially important for VR because the experience cannot be directly controlled by the system like many other mediums often do. The chapter includes sections on personal wayfinding aids, directing attention, how hardware choice relates to content creation and behavior, and social aspects of VR.

Chapter 23, Transitioning to VR Content Creation, discusses how creating for VR is different than creating for other mediums. Ideally, the VR content should be designed from the beginning for VR. However, when that is not possible, several tips are provided to help port existing content to VR.

Chapter 24, Content Creation: Design Guidelines, summarizes the previous four chapters and lists a number of actionable guidelines for those creating VR content.

High-Level Concepts of Content Creation

Virtual worlds would be very lonely places without content. This chapter discusses general high-level tips on creating content—creating a compelling story, focusing on the core experiences, keeping the integrity of the world consistent, and gestalt principles of perceptual organization.

20.1 Experiencing the Story

We need to consider a user's experiential arc through time. Virtual worlds can be intense or contemplative, ideally experiences oscillate between these, thus giving the author the ability to punctuate story elements with different design elements of VR.

—Mark Bolas (personal communication, June 13, 2015)

In traditional media ranging from plays to books to film to games, stories are compelling because they resonate with the audience's experiences. VR is extremely experiential and users can become part of a story more so than with any other form of media. However, simply telling a story using VR does not automatically qualify it to be an engaging experience—it must be done well and in a way that resonates with the user.

To convey a story, all of the details do not need to be presented. Users will fill in the gaps with their imaginations. People give meaning to even the most basic shapes and motions. For example, [Heider and Simmel \[1944 \]](#) created a two-and-a-half-minute video of some lines, a small circle, a small triangle, and a large triangle (Figure 20.1). The circle and triangles moved around both inside the lines and outside the lines, and sometimes touched each other. Participants were simply told “write down what happened in the picture.” One participant gave the following story.

A man has planned to meet a girl and the girl comes along with another man. The first man tells the second to go; the second tells the first, and he shakes his head. Then the two men have a fight, and the girl starts to go into the room to get out of the way and hesitates and finally goes in. She apparently does not want to be with the first man. The first man follows her into the room after having left the second in a rather weakened condition leaning on the wall outside the room. The girl gets worried and races from one corner to the other in the far part of the room . . .

All 34 participants created stories with 33 participants interpreting the actions of the geometric shapes as animate beings. Only one participant created a story consisting of geometric shapes. What is most surprising, however, is how similar some parts of their stories were. The 33 subjects who had stories about animate beings all stated similar things for specific points in the animation: two characters had a fight, one character was shut in some structure and tries to get out, and one character chased another.

VR creators will never be able to take the subjectivity out of a story experience. Users will always distort any stimuli presented to them based on their filters of the values, beliefs, memories, etc. (Section [7.9.3](#)). However, the above experiment suggests that in some cases, even simple cues can provide fairly consistent interpretations. Although we can't control exactly how users will make up all the details of the story in their own mind, content creators can lead the mind through suggestion. Content creators should focus on the most important aspects of the story and attempt to make them consistently interpreted across users.

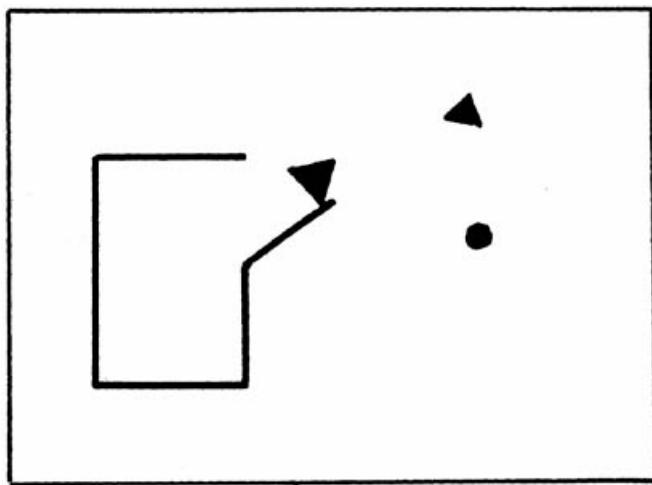


Figure 20.1 A frame from a simple geometric animation. Subjects came up with quite elaborate stories based on the animation. (From [Heider and Simmel \[1944 \]](#))

Experiential fidelity is the degree to which the user's personal experience matches the intended experience of the VR creator [[Lindeman and Beckhaus 2009](#)]. Experiential fidelity can be increased by better technology, but is also increased through everything that goes with the technology. For example, priming users before they enter the virtual world can structure anticipation and expectation in a way that will increase the impact of what is presented. Prior to the exposure, give the user a backstory about what is about to occur. The mind provides far better fidelity than any technology can, and the mind must be tapped to provide quality experiences. Providing perfect realism is not necessary if enough clues are provided for them to fill in the details with their imagination. Note experiential fidelity for all parts of the experience is not always the goal. A social free-roaming world can allow users to create their own stories—perhaps the content creator simply provides the meta-story where users create their own personal stories within.

Skeuomorphism is the incorporation of old, familiar ideas into new technologies, even though the ideas no longer play a functional role [[Norman 2013](#)]. This can be important for a wide majority of people who may be more resistant to change than early adopters. If designed to match the real world, VR experiences can be more acceptable than other technologies because of the lower learning curve—one interacts in a similar way to the real world rather than having to learn a new interface and new expectations. Then real-world metaphors can be incorporated into the story, gradually teaching how to interact and move through the world in new ways. Eventually, even conservative users will be open to more magical interactions (Section [26.1](#)) that have less relationship to the old, through skeuomorphic designs that help with the gradual transition.

In June of 2008, approximately 50 leading VR researchers at the “Designing the Experience” session at the Dagstuhl Seminar on Virtual Realities identified four primary attributes that characterize great experiences [[Lindeman and Beckhaus 2009](#)] as described below. To achieve the best experiences, shoot for the following concepts through powerful storytelling and sensory input.

Strong emotion can cover a range of emotions and feeling of extremes. These include any strong feeling such as joy, excitement, and surprise that occurs without conscious effort. Strong emotions result from anticipating an event, achieving a goal, or fulfilling a childhood dream. Most people are more aroused by and interested in emotional stories than logical stories. Stories aimed at emotions distract from the limits of technology.

Deep engagement occurs when users are in the moment and experience a state of flow as described by [Csikszentmihalyi \[2008 \]](#). Engaged users are extremely focused on the experience, losing track of space and time. Users become hyper-sensitive, with a heightened awareness without distraction.

Massive stimulation occurs when there is a sensory input across all the modalities, with each sense receiving a large amount of stimulation. The entire body is immersed and feels the experience in multiple ways. The stimulation should be congruent with the story.

Escape from reality is being psychologically removed from the real world. There is a lack of demands of the real world and the people in it. Users fully experience the story when not paying attention to sensory cues from the real world or the passage of time.

VR does not exist in isolation. Disney understands this extremely well. Over a period of 14 months in the mid-1990s, approximately 45,000 users experienced the Aladdin attraction, which used VR as a medium to tell stories [[Pausch et al. 1996](#)]. The authors came to the following conclusions.

- Those not familiar with VR are unimpressed with technology. Focus on the entire experience instead of the technology.
- Provide a relatable background story before immersing users into the virtual world.
- Make the story simple and clear.
- Focus on content, why users are there, and what there is to do.
- Provide a specific goal to perform
- Focus on believability instead of photorealism
- Keep the experience tame enough for the most sensitive users.
- Minimize breaks-in-presence such as interpenetrating objects and characters not responding to the user.
- Like film, VR appeals to everyone but content may segment the market, so focus on the targeted audience.

20.2 The Core Experience

The **core experience** is the essential moment-to-moment activity of users making meaningful choices resulting in meaningful feedback. Even though the overall VR experience might contain an array of various contexts and constraints, the core experience should remain the same. Because the core experience is so essential, it should be the primary focus of VR creators to make that core experience pleasurable so users stay engaged and want to come back for more.

A hypothetical example is a VR ping-pong game. A traditional ping-pong game very much has a core experience of continuously competing via simple actions that can be improved upon over time. A VR game might go well beyond a traditional real-world ping-pong game via adding obstacles, providing extra points for bouncing the ball differently, increasing the difficulty, presenting different background art, rewarding the player with a titanium racquet, etc. However, the core experience of hitting the ping-pong ball still remains. If that core experience is not implemented well so that it is enjoyable, then no number of fancy features will keep users engaged for more than a couple of minutes.

Even for a serious non-game application, making the core experience pleasurable is just as crucial for users to think of it as more than just an interesting demo, and to continue using the system. **Gamification** is, at least in part, about taking what some people consider mundane tasks and making the core of that task enjoyable, challenging, and rewarding.

Determining what the core experience is and making it enjoyable is where creativity is required. However, that creativity rarely results in the core experience being enjoyable on a first implementation. Instead VR creators must continuously improve upon the core experience, build various prototype experiences focused on that core, and learn from observing and collecting data from real users. These concepts are covered in Part VI .

20.3 Conceptual Integrity

No style will appeal to everyone; a mishmash of styles delights no one . . . Somehow consistency brings clarity, and clarity brings delight .

—Frederick P. Brooks, Jr. [[2010](#)]

Brooks argues in the *Mythical Man-Month* [[Brooks 1995](#)] “conceptual integrity is the most important consideration in system design.” This is even more so with VR content creation. **Conceptual integrity** is also known as coherence, consistency, and sometimes uniformity of style [[Brooks 2010](#)].

The basic structure of the virtual world should be directly evident and self-explanatory so users can immediately understand and start experiencing and using the world. There might be complexity added as users become experts, but the core conceptual model should be the same.

Extraneous content and features that are immaterial to the experience, even if good, should be left out. It is better to have one good idea that carries through the entire experience than to have several loosely related good ideas. If there are many but incompatible ideas where the sum of those ideas is better than the existing primary concept, then the primary concept should be reconsidered to provide an overarching theme that encompasses all of the best ideas in a coherent manner.

The best artists and designers create conceptual integrity with consistent conscious and subconscious decisions. How can conceptual integrity be enforced when the project is complex enough to require a team? Empower a single individual director who controls the basic concepts and has a clear passionate vision to direct the project’s vision and content.

The **director** primarily acts as an agent to serve the users of the experience, but also serves to fulfill the project goals and to direct the team in focusing on the right things. The director takes input and suggestions from the team but is the final decision maker on the concepts and what the experience will be. She leads many steps in defining the project (Chapter [31](#)). However, the director is not a dictator of how implementation will occur (Chapter [32](#)) or how feedback will be collected from users (Chapter [33](#)), but instead suggests to and takes suggestions from those working in those stages. The director should always be prepared to show an example implementation, even if not an ideal implementation, of how his ideas might be implemented. Even if the makers and data collectors accept the director’s suggestions, then the director should give credit to them.

Makers and data collectors should have immediate access to the director to make sure the right things are being implemented and measured. Questions and communication should be highly encouraged. This communication should be maintained throughout the project as the project definition will change and assumptions may drift apart as the project evolves.

20.4 Gestalt Perceptual Organization

Gestalt is a German word that roughly translates into English as *configuration*. The central principle of **gestalt psychology** states that perception depends on a number of organizing principles, which determine how we perceive objects and the world. This principle maintains that the human mind considers objects in their entirety before, or in parallel with, perception of their individual parts, suggesting the whole is different than the sum of its parts. Gestalt psychology is even more important for VR as compared to pictures alone as the best VR applications bring together all the senses into a cohesive experience that could not be obtained by one sensory modality alone.

The gestalt effect is the capability of our brain to generate whole forms, particularly with respect to the visual recognition of global figures, instead of just collections of simpler and unrelated elements such as points, lines, and polygons. As demonstrated in Section [6.2](#) and with every VR session anyone has ever experienced, we are pattern-recognizing machines seeing things that do not exist.

Perceptual organization involves two processes: grouping and segregation. Grouping is the process by which stimuli are put together into units or objects. Segregation is the process of separating one area or object from another.

20.4.1 Principles of Grouping

Gestalt groupings are rules of how we naturally perceive objects as organized patterns and objects, enabling us to form some degree of order from the chaos of individual components.

The **principle of simplicity** states that figures tend to be perceived in their simplest form instead of complicated shapes. As can be seen in Figure [20.2](#), perception of an object being 3D varies depending on the simplicity of its 2D projection; the simpler the 2D interpretation, the more likely that figure will be perceived as 2D. At a higher level, we tend to recognize 3D objects even if they are not realistic. The hydrant in Figure [20.3](#) is intentionally bent in an unrealistic manner to provide character, but we still perceive it as a simple recognizable object instead of abstract individual components.

The **principle of continuity** states aligned elements tend to be perceived as a single group or chunk, and are interpreted as being more related than unaligned elements. Points that form patterns are seen as belonging together, and the lines tend to be seen in such a way as to follow the smoothest path. An ‘X’ symbol or two crossing lines (Figure [20.4](#)) are perceived as two lines that cross instead of angled parts. Our brains automatically and subconsciously integrate cues as whole objects that are most likely to be in reality. Objects that are partially covered by other objects are also seen as continuing behind the covering object, as shown in Figure [20.5](#).

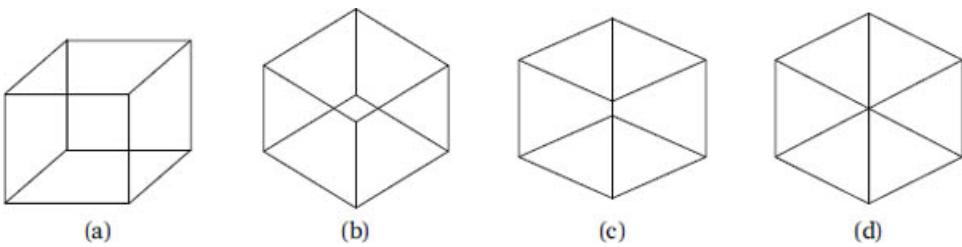


Figure 20.2 The principle of simplicity states that figures tend to be perceived in their simplest form, whether that is 3D (a) or 2D (d). (From [Lehar \[2007\]](#))



Figure 20.3 A MakeVR user applies the principle of simplicity when creating (the yellow/blue geometry is the user's 3D cursor) a fire hydrant. (Courtesy of Sixense)

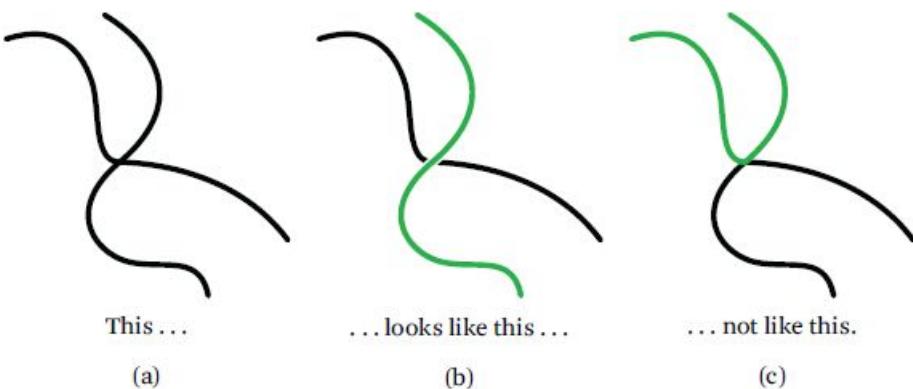


Figure 20.4 The principle of continuity states that elements tend to be perceived as following the most aligned or smoothest path. (Based on [Wolfe \[2006\]](#))

The **principle of proximity** states that elements that are close to one another tend to be perceived as a shape or group (Figure 20.6). Even if the shapes, sizes, and objects are radically different from each other, they will appear as a group if they are close together. Society evolved by using proximity to create written language. Words are groups of letters, which are perceived as words because they are spaced close together in a deliberate way. While learning to read, one recognizes words not only by individual letters but by the grouping of those letters into larger recognizable patterns. Proximity is also important when creating graphical user interfaces, as shown in Figure 20.7.

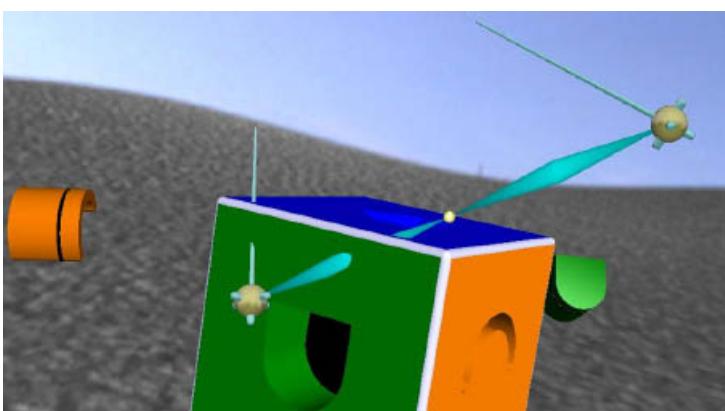


Figure 20.5 The blue geometry connecting the two yellow 3D cursors are perceived as being connected even when occluded by the cubic object. The “poker” part of the left cursor is also perceived as continuing through the cube. (From [[Yoganandan et al. 2014](#)])

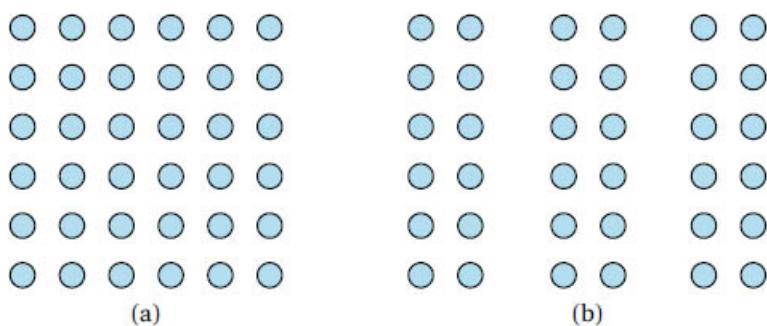


Figure 20.6 The principle of proximity states that elements close together tend to be perceived as a shape or group.

The **principle of similarity** states that elements with similar properties tend to be perceived as being grouped together (Figure 20.8). Similarity depends on form, color, size, and brightness of the elements. Similarity helps us organize objects into groups in realistic scenes or abstract art scenes, as shown in Figure 20.9.



Figure 20.7 The panel in MakeVR uses the principles of proximity and similarity to group different user interface elements together (drop-down menu items, tool icons, boolean operations, and apply-to options). (Courtesy of Sixense)

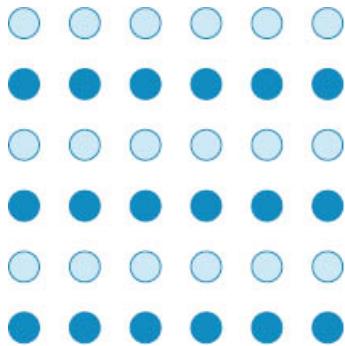


Figure 20.8 The principle of similarity states that elements with similar properties tend to be perceived as being grouped together.

The **principle of closure** states that when a shape is not closed, the shape tends to be perceived as being a whole recognizable figure or object (Figure 20.10). Our brains do everything to perceive incomplete objects and shapes as whole recognizable objects, even if that means filling in empty space with imaginary lines such as in the Kanizsa illusion (Section 6.2.2). When the viewer’s perception completes a shape, closure occurs.

Temporal closure is used with film, animation, and games. When a character begins a journey and then suddenly in the next scene the character arrives at his destination, then the mind fills the time between those scenes—we don’t need to watch the entire journey. A similar concept is used in immersive games and film. At shorter timescales, apparent motion can be perceived when in fact no motion occurred or the screen is blank between displayed frames (Section 9.3.6).

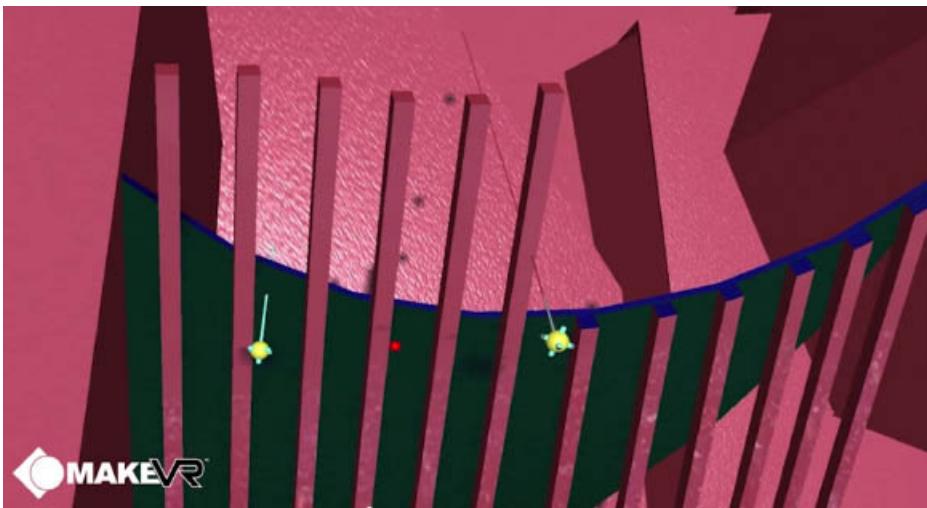


Figure 20.9 A MakeVR user using the principle of similarity to build abstract art. (Courtesy of Sixense)

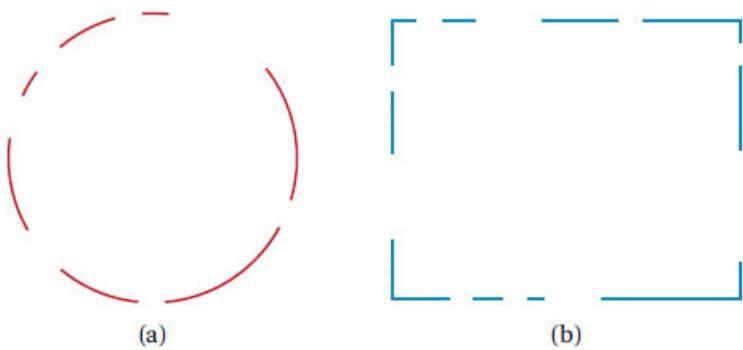


Figure 20.10 The principle of closure states that when a shape is not closed, the shape tends to be perceived as being a whole recognizable figure or object.

The principle of common fate states that elements moving together tend to be perceived as grouped, even if they are in a larger, more complex group (Figure 20.11). Common fate applies not only to groups of moving elements but also to the form of objects. For example, common fate helps to perceive depth of objects when moving the head through motion parallax (Section 9.1.3).

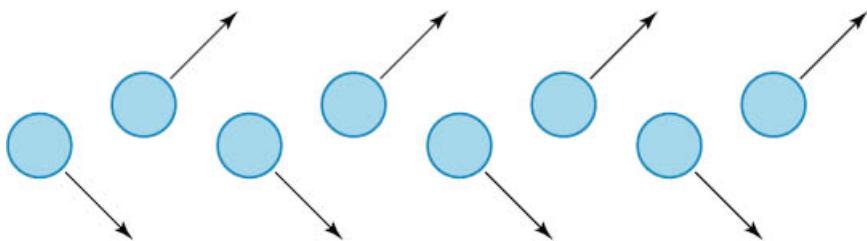


Figure 20.11 The principle of common fate states that elements moving in a common way (signified by arrows) tend to be perceived as being grouped together.

20.4.2 Segregation

The perceptual separation of one object from another is known as **segregation**. The question of what is the foreground and what is the background is often referred to as the **figure-ground problem**.

Objects tend to stand out from their background. In perceptual segregation terms, a **figure** is a group of contours that has some kind of object-like properties in our consciousness. The **ground** that is the background is perceived to extend behind a figure. Borders between figure and ground appear to be part of the figure (known as ownership).

Figures have distinct form whereas ground has less form and contrast. Figures appear brighter or darker than equivalent patches of light that form part of the ground. Figures are seen as richer, are more meaningful, and are remembered more easily. A figure has the characteristics of a “thing,” more apt to suggest meaning, whereas the ground has less conscious meaning but acts as a stable reference that serves as a rest frame for motion perception (Section 12.3.4). Stimuli perceived as figures are processed in greater detail than stimuli perceived as ground.

Various factors affect whether observers view stimuli as figure or ground. Objects tend to be convex and thus convex shapes are more likely to be perceived as figure. Areas lower in the scene are more likely to be perceived as ground. Familiar shapes that represent objects are more likely to be seen as figure. Depth cues such as partial occlusions/interposition, linear perspective, binocular disparity, motion parallax, etc. can have significant impact on what we perceive as figure and ground. Figures are more likely to be perceived as objects and signifiers (Section 25.2.2),

giving us clues that the form can be interacted with.

Environmental Design

The environment users find themselves in defines the context of everything that occurs in a VR experience. This chapter focuses on the virtual scene and its different aspects that environmental designers should keep in mind when creating virtual worlds.

21.1 The Scene

The **scene** is the entire current environment that extends into space and is acted within.

The scene can be divided into the background, contextual geometry, fundamental geometry, and interactive objects (similar to that defined by [Eastgate et al. \[2014 \]](#)). Background, contextual geometry, and fundamental geometry correspond approximately to vista space, action space, and personal space as described in Section [9.1.2](#).

The background is scenery in the periphery of the scene located in far vista space. Examples are the sky, mountains, and the sun. The background can simply be a textured box since non-pictorial depth cues (Section [9.1.3](#)) at such a distance are non-existent.

Contextual geometry helps to define the environment one is in. Contextual geometry includes far landmarks (Section [21.5](#)) that aid in wayfinding and are typically located in action space. Contextual geometry has no affordances (i.e., can't be picked up; Section [25.2.1](#)). Contextual geometry is often far enough away that it can consist of simple faked geometry (e.g., 2D billboards/textures often used to portray more complex geometry such as trees).

Fundamental geometry consists of nearby static components that add to the fundamental experience. Fundamental geometry includes items such as tables, instructions, and doorways. Fundamental geometry often has some affordances, such as preventing users from walking through walls or providing the ability to set an object upon something. This geometry is most often located in personal space and action space. Because of its nearness, artists should focus on fundamental geometry. For VR, 3D details are especially important at such distances (Section [23.2.1](#)).

Interactive objects are dynamic items that can be interacted with (discussed extensively in Part [V](#)). These typically small objects are in personal space when directly interacted with and most commonly in action space when indirectly interacted with.

All objects and geometry scaling should be consistent relative to each other and the user. For example, a truck in the distance should be scaled appropriately relative to a closer car in 3D space; the truck should not be scaled smaller because it is in the distance. A proper VR rendering library should handle the projection transformation from 3D space to the eye; artists need not be concerned with such detail. For realistic experiences, include familiar objects with standard sizes that are easily seen by the user (see relative/familiar size in Section [9.1.3](#)). Most countries have standardized dimensions for sheets of paper, cans of soda, money, stair height, and exterior doors.

21.2 Color and Lighting

Color is so pervasive in the world that we often take it for granted. However, we constantly perceive and interact with colors throughout the day whether getting dressed or driving. We associate color to emotional reactions (purple with rage, green with envy, feeling blue) and special meanings (red signifies danger, purple signifies royalty, greens signifies ecology).

Other factors influence our perception of color. For example, the surrounding context/background can influence our perception of an object's color. Exaggerated coloring of the entire scene or extreme lighting conditions can result in loss of lightness and color constancy (Section [10.1.1](#)) and objects seen as being different than they would be under normal conditions. Unless intentional for special situations, content creators should use multiple colors and only slight variations of white light so users perceive the intended color of objects and maintain color constancy.

Color enables us to better distinguish between objects. For example, bright colors can capture users' attention through salience (salience refers to physical properties of a scene that reflexively grabs our attention; Section [10.3.2](#)). Figure [21.1](#) (left) shows an example of how non-relevant background can be made a dull gray to draw attention to more relevant objects. Figure [21.1](#) (right) shows how color is turned on, signifying that brain lobes can be grabbed after the system provides audio instructions.

Consider that real-world painters choose from only a little over 1,000 colors (the Pantone Matching System has about 1,200 color choices). This does not mean we only need 1,200 pixel options for VR. Much of the subtle changes in color come from gradual changes over surfaces—for example, change in lighting intensity across a surface. However, ~1,000 colors is fine for creating content before lighting is added to the scene (unless one is attempting to perfectly match a color from outside the 1,000-color set).



Figure 21.1 Color is used as salience to direct attention. The user's eyes are drawn toward the colored items in the scene (left). After the system provides audio instructions, the brain lobes and table are colored to draw attention and to signify they can be interacted with (right). (Courtesy of Digital ArtForms)

21.3 Audio

As discussed in Section 8.2, sound is quite complex with many factors affecting how we perceive it. Auditory cues play a crucial role in everyday life as well as VR, including adding awareness of surroundings, adding emotional impact, cuing visual attention, conveying a variety of complex information without taxing the visual system, and providing unique cues that cannot be perceived through other sensory systems. Although deaf people have learned to function quite well, they have a lifetime of experience learning techniques to interact with a soundless world. VR without sound is equivalent to making someone deaf without the benefit of having years of experience learning to cope without hearing.

The entertainment industry understands audio well. For example, George Lucas, known for his stunning visual effects, has stated sound is 50% of the motion picture experience. Like a great movie, music is especially good at evoking emotion. Subtle **ambient sound effects** such as birds chirping along with rustling of trees by the wind, children playing in the distance, or the clanking of an industrial setting can provide a surprisingly strong effect on a sense of realism and presence.

When sounds are presented in an intelligent manner, they can be informing and extremely useful. Sounds work well for creating situational awareness and can obtain attention independent of visual properties and where one is looking. Although sound is important, sound can be overwhelming and annoying if not presented well. It is also difficult to ignore sounds like what can be done with vision (e.g., closing the eyes). Aggressive warning sounds that are infrequent with short durations are designed to be unpleasant and attention-getting, and thus are appropriate if that is the intention. However, if overused such sounds can quickly become annoying rather than useful.

Vital information can be conveyed through spoken language (Section 8.2.3) that is either synthesized, prerecorded by a human, or live from a remote user. The spoken information might be something as simple as an interface responding with “confirmed” or could be as complex as a spoken interface where the system both provides information and responds to the user’s speech. Other verbal examples include clues to help users reach their goals, warnings of upcoming challenges, annotations describing objects, help as requested from the user, or adding personality to computer-controlled characters.

At a minimum, VR should include audio that conveys basic information about the environment and user interface. Sound is a powerful feedback cue for 3D interfaces when haptic feedback is not available, as discussed in Section 26.8.

For more realistic audio and where auditory cues are especially important, **auralization**—the rendering of sound to simulate reflections and binaural differences between the ears—can be used. The result of auralization is **spatialized audio**—sound that is perceived to come from some location in 3D space (Section 8.2.2). Spatialized audio can be useful to serve as a wayfinding aid (Section 21.5), to provide cues of where other characters are located, and to give feedback of where the element of a user interface is located. A **head-related transfer function** (HRTF) is a spatial filter that describes how sound waves interact with the listener’s body, most notably the outer ear, from a specific location. Ideally, HRTFs model a specific user’s ears, but in practice a generic ear is modeled. Multiple HRTFs from different directions can be interpolated to create an HRTF from any direction relative to the ear. The HRTF can then modify a sound wave from a sound source, resulting in a realistic spatialized audio cue to the user.

21.4 Sampling and Aliasing

Aliasing is an artifact that occurs due to approximating data with discrete sampling. With computer graphics, aliasing occurs due to approximating edges of geometry and textures with discretely sampled/rendered pixels. This occurs because the geometry or texture representing the point sample either does or does not project onto the pixel, i.e., a pixel either represents a piece of geometry or color of a texture or it does not. Edges in the environment that are projected onto the display cause discontinuities called “jaggies” or “staircasing,” as shown in Figure 21.2. Other artifacts also occur, such as moire patterns, as seen in Figure 21.3. Such artifacts are worse in VR due to the artifact patterns continuously fluctuating as a result of viewpoint movement, each eye being rendered from different views, and the pixels being spread out over a larger field of view. Even subconscious small head motion that is normally imperceptible can be distracting due to the eye being drawn toward the continuous motion of the jaggies and other artifact patterns.



Figure 21.2 Jaggies/staircasing as seen on a display that represents the edge of an object. Such artifacts result due to discrete sampling of the object.



Figure 21.3 Aliasing artifacts can clearly be seen at the left-center area of the chain-link fence. Such artifacts are even worse in VR due to the artifact patterns continuously moving as a result of viewpoint movement as small as normally imperceptible head motion. (Courtesy of NextGen Interactions)

Such artifacts can be reduced through various anti-aliasing techniques, such as mipmapping, filtering, rendering, and jittered multi-sampling. Although such techniques can reduce aliasing artifacts, they unfortunately cannot completely obviate all artifacts from every possible viewpoint. Some of these techniques also add significantly to rendering time, which can add latency. In addition to using anti-aliasing techniques, content creators can help to reduce such artifacts by not including high-frequency repeating components in the scene. Unfortunately, it is not possible to remove all aliasing artifacts. For example, linear perspective causes far geometry to have high spatial frequency when projected/rendered onto the display (although artifacts can be somewhat reduced by implementing fog or atmospheric attenuation). Swapping in/out models with different levels of detail (e.g., removing geometry with higher-frequency components when it becomes further from the viewpoint) can help but often occurs at the cost of geometry “popping” into and out of view, which can cause a break-in-presence. The ideal solution is to have ultra-high resolution displays, but until then content creators should do what they can to minimize such artifacts by not creating or using high-frequency components.

21.5 Environmental Wayfinding Aids

Wayfinding aids [[Darken and Sibert 1996](#)] help people form cognitive maps and find their way in the world (Section [10.4.3](#)). Wayfinding aids help to maintain a sense of position and direction of travel, to know where goals are located, and to plan in the mind how to get to those goals. Examples of wayfinding aids include architectural structures, markings, signposts, paths, compasses, etc. Wayfinding aids are especially important for VR because it is very easy to get disoriented with many VR navigation techniques. Virtual turns can be especially disorienting (e.g., turning with a hand controller but not physically turning the body) due to the lack of vestibular cues and other physical sensations of turning. The lack of physically stepping/walking can also cause incorrect judgments of distance.

Fortunately, there is much opportunity for creating wayfinding aids in VR that can't be done in the real world. Wayfinding aids are most often visual but do not need to be. Floating arrows, spatialized audio, and haptic belts conveying direction are examples of VR wayfinding aids that would be difficult to implement in the real world. The aids may or may not be consciously noticed by users but are often useful in either case. Like the real world, there is far too much sensory information in VR for users to consciously take in information from every single element in the environment. Understanding application and user goals can be useful for designers to know what to put where to help users find their way.

The remaining portion of this section focuses on environmental wayfinding aids. Section [22.1](#) focuses on personal wayfinding aids.

Environmental wayfinding aids are cues within the virtual world that are independent of the user. These aids are primarily focused on the organization of the scene itself. They can be overt such as signs along a road, church bells ringing, or maps placed in the environment. They can also be subtle such as buildings, characters traveling in some direction, and shadows from the sun. The sound of vehicles on a highway can be a useful cue when occlusions, such as trees, are common to the environment. Although not typical for VR, smell can also be a strong cue that a user is in a general area. Well-constructed environments do not happen by accident and good level designers will consciously include subtle environmental wayfinding aids even though many users may not consciously notice. There is much that can be done in designing VR spaces that enhances spatial understanding of the environment so users can comprehend and operate effectively.

Although VR designers can certainly create more than what can be done within the limitations of physical reality, we can learn much from non-VR designers. Just because more is possible with VR does not mean we should not understand real spaces and construct space in a meaningful way. Architectural designers and urban planners have been dealing with wayfinding aids for centuries through the relationship between people and the environment. [Lynch \[1960\]](#) found there are similarities across cities as described below.

Landmarks are disconnected static cues in the environment that are unmistakable in form from the rest of the scene. They are easy to recognize and help users with spatial understanding. Global landmarks can be seen from about anywhere in the environment (such as a tower), whereas local landmarks provide spatial information closer to the user.

The strongest landmarks are strategically placed (e.g., on a street corner) and often have the most salient properties (highly distinguishable characteristics), such as a brightly colored light or pulsing glowing arrows showing a path. However, landmarks do not need to be explicit. They might be more subtle like colors or lighting on the floor. In fact, landmarks that are overdominating may cause users to move from one location to another without paying attention to other parts of the scene, discouraging acquisition of spatial knowledge [[Darken and Sibert 1996](#)]. The structure and form of a well-designed natural environment provides strong spatial cues without clutter.

Landmarks are sometimes directional, meaning they might be perceived from one side but not another. Landmarks are especially important when a user enters a new space, because they are the first things users pay attention to when getting oriented. Landmarks are important for all environments ranging from walking on the ground to sea travel to space travel. Familiar landmarks, such as a recognizable building, aid in distance estimation due to their familiarity and relation to surrounding objects.

Regions (also known as districts and neighborhoods) are areas of the environment that are implicitly or explicitly separated from each other. They are best differentiated perceptually by having different visual characteristics (e.g., lighting, building style, and color). **Routes** (also known as paths) are one or more travelable segments that connect two locations. Routes include roads, connected lines on a map, and textual directions. Directing users can be done via channels. **Channels** are constricted routes, often used in VR and video games (e.g., car-racing games) that give users the feeling of a relatively open environment when it is not very open at all. **Nodes** are the interchanges between routes or entrances to regions. Nodes include freeway exits, intersections, and rooms with multiple doorways. Signs giving directions are extremely useful at nodes. **Edges** are boundaries between regions that prevent or deter travel. Examples of edges are rivers, lakes, and fences. Being in a largely void environment is uncomfortable for many users, and most people like to have regular reassurance they are not lost [[Darken and Peterson 2014](#)]. A visual **handrail** is a linear feature of an environment, such as a side of a building or a fence, which is used to guide navigation. Such handrails are used to psychologically constrain movement, keeping the user to one side and traveling along it for some distance.

How these parts of the scene are classified often depends on user capabilities. For walkers, paths are routes and highways tend to be edges, whereas for drivers, highways are routes. For flying users, paths and routes might only serve as landmarks. Often these distinctions are also in the mind of the user, and geometry and cues can be perceived differently by different users or even the same user under different conditions (e.g., when one starts driving a vehicle/aircraft after walking).

Constructing geometric structure is not enough. The user's understanding of overarching themes and structural organization are important for wayfinding. Users should know the structure of the environment for best results, and making such structure explicit in the minds of users is important as it can help give cues meaning, affect what strategies users employ, and improve navigation performance. Street numbers or names in alphabetical order make more sense after understanding streets follow a grid pattern and are named in numerical and alphabetical order.

Once the person builds a mental model of the meta-structure of the world, violations of the metaphor used to explain the model can be very confusing and it should be made explicit when such violations do occur. A person who lived his entire life in Manhattan (a grid-like structure of city blocks) traveling to Washington, DC, will become very confused until the person understands the hub-and-spoke structure of many of the city's streets, at which point wayfinding becomes easier. However, due to having many more violations of the metaphor, Washington, DC, is much more confusing than New York for most people.

Adding concepts of city-like structures to abstract data such as information or scientific visualization can help users better understand and navigate that space [[Ingram and Benford 1995](#)]. Adding cues such as a simple rectangular grid or radial grid can improve performance. Adding paths suggests leading to somewhere useful and interesting. Sectioning data into regions can emphasize different parts of the dataset. However, developers should be careful of imposing inappropriate structure onto abstract or scientific data as users will grasp onto anything they perceive as structure, and that may result in perceiving structure in the data that does not exist in the data itself. Subject-matter experts familiar with the data should be consulted before adding such cues.

21.5.1 Markers, Trails, and Measurement

Markers are user-placed cues. If using a map (Section [22.1.1](#)), markers can be placed on the map (e.g., as colored pushpins) as well as in the environment. This helps users remember which preexisting landmarks are important. **Breadcrumbs** are markers that are dropped often by the user when traveling through an environment [[Darken and Sibert 1993](#)]. A **trail** is evidence of a path traveled by a user and informs the person who left the path as well as other users that they have already been to that place and how they traveled to and from there. Trails consisting of individual directional cues (such as footprints) are generally better than non-directional cues. Multiple trails inform of well-traveled paths. Unfortunately, many trails can result in visual clutter. Footprints can fade away after some time to reduce clutter. Sometimes it is better to identify areas searched instead of paths followed.

Data understanding and spatial comprehension via interactive analysis often requires markup and quantification of dataset features within the environment. Different types of markup tools such as paintbrushes and skewers that contain symbolic, textual, or numerical information can be provided to interactively mark, count, and measure features of the environment. Users may want to simply indicate areas of interest for later investigation, or to count the number of vessels exiting a mass (e.g., a tumor) by having the system automatically increment a counter as markers are placed. Linear segments, surface area, and angles can also be measured and the resulting values placed in the environment with a line attached to the markups. Figure [21.4](#) shows examples of drawing on the world and placing skewers. Figure [21.5](#) shows an example of measuring the circumference of an opening within a medical dataset.



Figure 21.4 A user observes a colleague marking up a terrain. (Courtesy of Digital ArtForms)

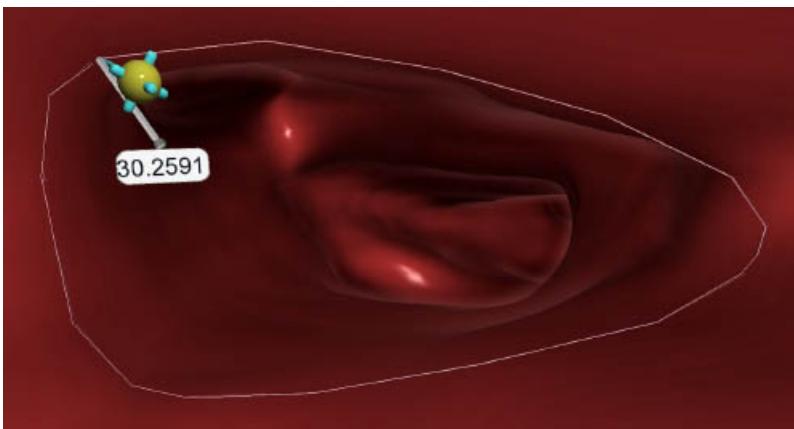


Figure 21.5 A user measures the circumference of an opening in a CT medical dataset. (Courtesy of Digital ArtForms)

21.6 Real-World Content

Content does not necessarily need to be created by artists. One way of creating VR environments is not to build content but to reuse what the real world already provides. Real-world data capture can result in quite compelling experiences.



Figure 21.6 A 360° image from the VR experience *Strangers with Patrick Watson*. (© Strangers with Patrick Watson / Félix & Paul Studios)

21.6.1 360° Cameras

Specialized panoramic cameras can be used to capture the world in 360° from one or more specific viewpoints (Figure 21.6). Capturing the world in this way is forcing filmmakers to rethink what it means to experience content. Instead of watching a movie through a “window,” viewers of immersive film are in and part of the scene. An example of capturing 360° data that is very different from traditional filming is having to carefully place equipment in a way so that it is not seen by the camera in all directions. In a similar way, all individuals that are not part of the story must leave the set or hide behind objects so they do not unintentionally become part of the story. Those capturing 360° should do so from static poses unless they are extremely careful in controlling the motion of the camera (Section 18.5).

Creating stereoscopic 360° content is technically challenging due to the cameras capturing content from only a limited number of views. Such content can work reasonably well with VR when the viewer is seated in a fixed position and primarily only makes yaw (left-to-right or right-to-left) type of head motions. However, if the viewer pitches the head (looks down) or rolls the head (twists the head while still looking straight ahead), then the assumptions for presenting the stereoscopic cues no longer hold and the viewer will perceive strange results. A common method of dealing with the pitch problem is to have the disparity between the left and right eyes in the upper and lower parts of the scene converge. This increases comfort at the cost of everything above the viewer and below the viewer appearing far away. Another method is to smooth out the ground beneath the viewer to a single color such that it contains no depth cues. Or a virtual object can be placed beneath the viewer to prevent seeing the real-world data at that location. These methods all work well enough when the content of interest is not above or below the viewer.

Another solution for the stereo problem is simply not to show the captured content in stereo. This makes content capture a lot easier and reduces artifacts. Computer-generated content that has depth information can be added to the scene but will always appear in front of the captured content.

21.6.2 Light Fields and Image-Based Capture/Rendering

Capture from a single location cannot properly convey a fully immersive VR experience when users move their heads—for example, when leaning left or right. A **light field** describes light that flows in multiple directions through multiple points in space. An array of cameras can be coupled with light field capture and rendering techniques to portray multiple viewpoints from different locations, including those not originally captured [[Gortler et al. 1996](#), [Levoy and Hanrahan 1996](#)]. Examples include a system that captures a static scene with an outward-looking spherical capture device [[Debevec et al. 2015](#)] and a system for portraying animated stop motion puppets by capturing sequences of a 360° ring of images around each puppet [[Bolas et al. 2015](#)].

21.6.3 True 3D Capture

Depth cameras or laser scanners capture true 3D data and can result in more presence-inducing experiences at the cost of the scene not seeming quite as real due to artifacts such as gaps or skins (interpolating color where the camera can't see). Data used from multiple depth cameras can be used to help reduce some of these artifacts. Data acquired from 3rdTech laser scanners is used for crime scenes where investigators can go back and take measurements after data has been collected (Figure 21.7). When 3rdTech CEO Nick England (personal communication, June 4, 2015) viewed himself in an HMD lying dead on the floor in a mock murder scene, he witnessed quite a jarring out-of-body experience.

21.6.4 Medical and Scientific Data

Figure 21.8 shows an image from iMedic—immersive medical environment for distributed interactive consultation [[Mlyniec et al. 2011](#), [Jerald 2011](#)]. iMedic enables real-time interactive exploration of volumetric datasets by “crawling” through the data with a 3D multi-touch interface (Section 28.3.3). The mapping of voxel (3D volumetric pixels) source density values to visual transparency can also be controlled via a virtual panel held in the non-dominant hand, enabling real-time changes of what structures can be seen in the dataset.



Figure 21.7 Mock crime scenes captured with a 3rdTech laser scanner. Even though artifacts such as cracks and shadows can be seen in parts of the scene that the laser scanner(s) could not see (e.g., behind the victim's legs), the results can be quite compelling. (Courtesy of 3rdTech)

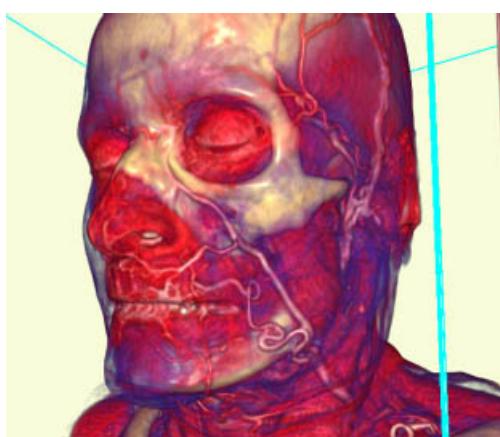


Figure 21.8 A real-time voxel-based visualization of a medical dataset. The system enables the user to explore the dataset by “crawling” through it with a 3D multi-touch interface. (Courtesy of Digital ArtForms)

Visualizing data from within VR can provide significant insight for scientists to understand their data. Multiple depth cues such as head-motion parallax and real walking that are not available with traditional displays enable scientists to gain insight by better seeing shape, protrusions, and relationships they previously did not know existed even when they were, or so they thought, intimately knowledgeable about their data.

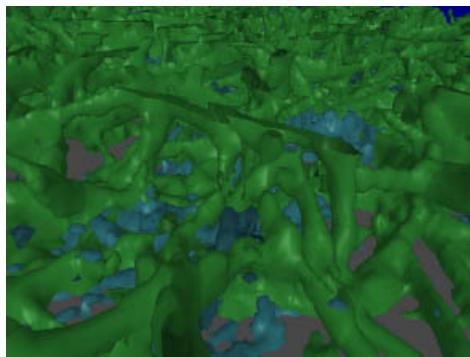


Figure 21.9 Whereas this dataset of a fibrin network (green) grown over platelets (blue) may seem like random polygon soup to those not familiar with the subject, scientists intimately familiar with the dataset gained significant insight after physically walking through the dataset with an HMD. Even with such simple rendering, the wide range of depth cues and interactivity provides understanding not possible with traditional tools. (Courtesy of UNC CISMM NIH Resource 5-P41-EB002025 from data collected in Alisa S. Wolberg’s laboratory under NIH award HL094740.)

From March 2008 to March 2010, scientists from various departments and organizations used the UNC-Chapel Hill Department of Computer Science VR Lab to physically walk through their own datasets [Taylor 2010] (Figure 21.9). In addition to clarifying understanding of their datasets, several scientists were able to gain insight that was not possible with traditional displays they typically used. They understood the complexity of structural protrusions better; saw concentrated material next to small pieces of yeast that led to ideas for new experiments; saw branching protrusions and more complex structures that were completely unexpected; more easily tracked and navigated branching structures of lungs, nasal passages, and fibers; and more easily saw clot heterogeneity to get an idea of overall branch density. In one case, a scientist stated, “We didn’t do the experiment right to answer the question about clumping . . . we didn’t know that these tools existed to view the data this way.” If they were able to view the data in an HMD before the experiment, then the experiment design could have been corrected.

Affecting Behavior

VR users who feel fully present inside the creator's world can be affected by content more than in any other medium. This chapter discusses several ways that the designer can influence and affect users through mechanisms linked to content, as well as how the designer should consider hardware and networking when creating content.

22.1 Personal Wayfinding Aids

22.1.1 Maps

A **map** is a symbolic representation of a space, where relationships between objects, areas, and themes are conveyed. The purpose of a map is not necessarily to provide a direct mapping to some other space. Abstract representations can often be more effective in portraying understanding. Consider, for example, subway maps that only show relevant information and where scale might not be consistent.

Maps can be static or dynamic and looked at before navigation or concurrently while navigating. Maps that are used concurrently involve placement of oneself on the map (mentally or through technology) and help to answer the questions "Where am I?" and "What direction am I facing?"

Dynamic "you-are-here" markers on a map are extremely helpful for users to understand where they are in the world. Showing the user's direction is also important. This can be done with an arrow or the user's field of view on the map.

For large environments, it can be difficult to see details if the map represents the entire environment. The scale of the map can be modified, such as by a pinch gesture or 3D Multi-Touch Pattern (Section [28.3.3](#)). To prevent the map from getting in the way/being annoying when it is not of use, consider putting the map on the non-dominant hand and give the option to turn it on/off.

Maps can also be used to more efficiently navigate to a location by entering the map through the World-In-Miniature Pattern (Section [28.5.2](#))—point on the map where to go and then "enter into" the map with the map becoming the surrounding world.

How a map is oriented can have a big impact on spatial understanding. Map use during navigation is different from map use for spatial knowledge extraction. While driving or walking, people in an unfamiliar area generally turn maps in different directions, and misaligned maps cause problems for some people in judgments of direction. However, when looking over a map to plan a trip, the map is normally not turned. Depending on the task, maps are best used with either a forward-up orientation or a north-up orientation.

Forward-up maps align the information on the map to match the direction the user is facing or traveling. For example, if the user is traveling southeast, then the top center of the map corresponds to that southeast direction. This method is best used when concurrently navigating and/or searching in an egocentric manner [[Darken and Cevik 1999](#)], as it automatically lines up the map with the direction of travel and enables easy matching between points on the map and corresponding landmarks in the environment. When the system does not automatically align the map (i.e., the users have to manually align the map), many users will stop using the map.

North-up maps are independent of the user's orientation; no matter how the user turns or travels, the information on the map does not rotate. For example, if the user holds the map in the southeast direction, then the top center of the map still represents north. North-up maps are good for exocentric tasks. Example exocentric uses of north-up maps are (1) familiarizing oneself with the entire layout of the environment independent of where one is located, and (2) route planning before navigating. For egocentric tasks, a mental transformation is required from the egocentric perspective to the exocentric perspective.

22.1.2 Compasses

A **compass** is an aid that helps give a user a sense of exocentric direction. In the real world, many people have an intuitive feel of which way is north after only being in a new location for a short period of time. If virtual turns are employed in VR, then keeping track of which way is which is much more difficult due to the lack of physical turning cues. By having easy access to a virtual compass, users can easily redirect themselves after performing a virtual turn and return to the intended direction of travel. Compasses are similar to forward-up maps except compasses only consist of direction, not location.

A virtual compass can be held in the hand as in the real world. Holding the compass in the hand enables one to precisely determine the heading of a landmark by holding the compass up to directly line up ticks on the compass onto landmarks in the environment. In contrast to the real world, a virtual compass can hang in space relative to the body (Section [26.3.3](#)) so there is no need to hold it with the hand, although this comes at the disadvantage of not being able to easily line up the compass with landmarks. The hand, however, can easily grab and release the compass as needed. Figure [22.1](#) (left) shows a standard compass showing the cardinal directions over a terrain.

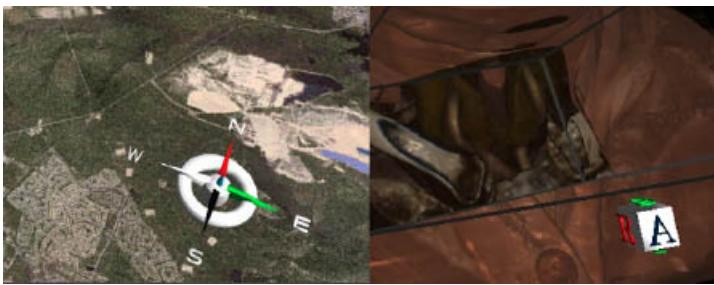


Figure 22.1 A traditional cardinal compass (left) and a medical compass with a human CT dataset (right). The letter *A* on the medical compass represents the anterior (front) side of the body, and the letter *R* represents the right lateral side of the body. (Courtesy of Digital ArtForms)

A compass does not necessarily provide cardinal directional information. Dataset visualization sometimes uses a cubic compass that represents direction relative to the dataset. For example, medical visualization sometimes uses a cube with the first letter of the directions on each face of the cube to help users understand how the close-up anatomy relates to the context of the entire body. Figure 22.1 (right) shows such a compass.

If the compass is placed at eye level surrounding the user, then the user can overlay the compass ticks on landmarks by simply rotating the head (in yaw and pitch). Alternatively, the compass can be placed at the user's feet as shown in Figure 22.2. In this case the user simply looks down to determine the way he is facing. This has the advantage of not being distracting as the compass is only seen when looking in the downward direction.

22.1.3 Multiple Wayfinding Aids

Individual wayfinding aids by themselves are not particularly useful. Combining multiple aids together can significantly help. Effectiveness of wayfinding depends on the number and quality of wayfinding cues or aids provided to users [Bowman et al. 2004]. However, too many wayfinding aids can be overwhelming; choose what is most appropriate for the goals of the project.

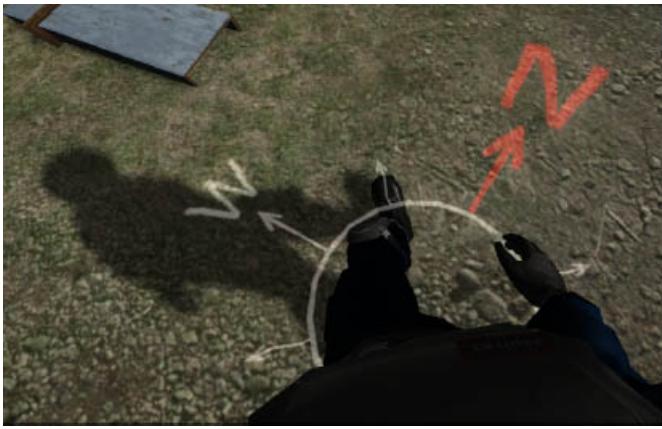


Figure 22.2 A compass surrounding the user's feet. (Courtesy of NextGen Interactions)



Figure 22.3 Visionary VR splits the scene into different action zones. (Courtesy of Visionary VR)

22.2 Center of Action

Not being able to control what direction users are looking is a new challenge for filmmakers as well as other content makers. Visionary VR, a start-up based in Los Angeles, splits the world around the user into different zones such as primary and secondary viewing directions (Figure 22.3) [Lang 2015]. The most important action occurs in the primary direction. As a user starts looking toward another zone, cues are provided to inform the user that she is starting to look at a different part of the scene and the action and/or content is about to change if they keep looking in that direction. For example, glowing boundaries between zones can be drawn, the lights might dim for a zone as the user starts to look toward a different zone, or time might slow down to a halt as users look toward a different zone. Zones are also aware if the user is looking in their direction so actions within that zone can react when being looked at. These techniques can be used to enable users to experience all parts of the scene

independent of the order in which they view the different zones.

22.3 Field of View

There are a number of techniques designers can use to take advantage of the power of field of view (Mark Bolas, personal communication, June 13, 2015). One is to artificially constrain the field of view for certain parts of a VR experience, and then allow it to go wide for others. This is similar to how cinematographers use light or color in a movie; they control the perceptual arc to correlate with moments in the story being experienced. Another is to shape the frame around the field of view—rounded, asymmetric shapes feel better than rectilinear ones. You want the periphery to feel natural, even including the occlusion due to the nose. The qualitative result is the ability to author content with greater range and emotional depth. Its power turns up in unexpected ways. For example, anecdotal evidence suggests virtual characters cause a greater sense of presence when looking at them with a wide field of view compared to a narrow field of view. When there is a wide field of view, there is more of an obligation to obey social norms—for example, not staring at characters for too long and keeping an appropriate distance.

22.4 Casual vs. High-End VR

Different types of systems each have their own advantages and disadvantages. The experiences will be very different depending on the system type, and the experience should be targeted and designed for the primary type (although multiple types might be supported, the experience should be most optimized for one). This section focuses on wired vs. wireless and mobile vs. location-based systems. Chapter [27](#) discusses system details specific to user input.

22.4.1 Wired vs. Wireless VR

Wired and wireless systems result in different experiences whether the designer considers the trade-offs or not. For example, users will feel the tug of wires on the HMD resulting in a break-in-presence if they attempt to physically turn too far with a wired system. To optimize the user experience, design should take into account whether the targeted system is wired or wireless.

Wired seated systems should place content in front of the user and/or use an interface that enables virtual turning so that wires do not become tangled. An advantage of not enabling virtual turning (in addition to minimizing motion sickness) is that the designer can assume the user is facing in a general forward direction. Many seated experiences can be designed so that non-tracked hand controllers can be assumed to be held in the lap, such that some visual representation of the controller and user's hands can be drawn at the approximate location of the physical hands (Section [27.2.2](#)). If virtual turning is used, the visual hands/controller should turn with the virtual turn (e.g., draw some form of chair-like device with the controller and body attached to the chair).

Freely turnable systems are wireless systems that allow users to easily look in any direction just as they would in the real world without concern of which direction is forward in the real world. Real turning has the advantages that it does not cause motion sickness and does not cause disorientation. A system that includes a swivel chair or omnidirectional treadmill should use a wireless system as cables will otherwise quickly become tangled after more than a 180° turn.

Fully walkable systems enable users to stand up and physically walk around. Fully walkable untethered systems are known as **nomadic VR** (Denise Quesnel, personal communication, June 8, 2015). Whereas fully walkable systems can work extremely well, a non-immersed spotter should follow behind the user to prevent injury, and in the case of wired systems to prevent entanglement, tugging of the cables on the user's head, and breaks-in-presence by the wires touching any part of the user's body. Even with a wireless system or wires attached to a rail system, such as that at UNC-Chapel Hill, a spotter should still be used.

22.4.2 Mobile vs. Location-Based VR

Mobile VR vs. location-based VR is the equivalent of mobile apps/games vs. more detailed and higher-powered desktop software. Although there is overlap, each should be designed to be optimized to take advantage of its specific characteristics (similar to input devices).

Mobile VR is defined by being able to place all equipment in a small bag and being able to immerse oneself almost instantaneously at about any time and any location when engagement with the real world is not required. Cell phones that snap into a head-mounted device are currently the standard for mobile VR. Mobile VR is also much more social and casual as VR experiences can be shared on a plane, at a party, or in a meeting.

Location-based VR requires a suitcase or more, takes time to set up, and ranges from one's living room or office to out-of-home experiences [[Williams and Mascioni 2014](#)] such as VR arcades or theme parks. Location-based VR has the potential to be higher quality and the most immersive as it can include high-end equipment and tracking technologies.



Figure 22.4 A user drives a virtual car in a CAVE while speaking with an avatar controlled by a remote user. (From [Daily et al. \[2000 \]](#))

22.5 Characters, Avatars, and Social Networking

A character can be either a computer-controlled character (an agent) or an avatar. An **avatar** is a character that is a virtual representation of a real user. This section discusses characters and how they influence behavior. Section [32.3](#) discusses technical implementation of social networking. Figure [22.4](#) shows an example of a user and a remote user rendered locally as an avatar.

As mentioned in Section [4.3](#), how we perceive ourselves can be quite distorted. In VR, we can explicitly define and design ourselves from a third-person perspective to look any way we wish. In addition, audio filters and prerecorded audio can be used to modify the sound of our voice. Providing the capability for users to personalize their own avatar has been proven to be very popular in both 2D virtual worlds (e.g., Second Life) as well as more fully immersive worlds. The Xbox Avatar store alone has 20,000 items for free or purchase.¹ **Caricature** is a representation of a person or thing in which certain striking characteristics are exaggerated and less important features are omitted or simplified in order to create a comic or grotesque effect. Cartoonlike rendering with caricature can be enjoyable, effective, compelling, and presence inducing. Caricature can be especially effective for avatars because it can avoid the Uncanny Valley (Section [4.4.1](#)).



Figure 22.5 Robotic characters without legs prevent breaks-in-presence that often occur when characters walk or run due to the difficulty of animating walking/running in a way that humans perceive as real. (Courtesy of NextGen Interactions)

As discussed in Section [9.3.9](#), the human ability to perceive the motion of others is extremely good. Although motion capture can result in character animation that is extremely compelling, this is difficult to do when mixing motion capture data with characters that are able to arbitrarily move and turn at different speeds. A simple solution is to use characters without legs, such as legless robotic characters, so that no breaks-in-presence occur due to walking and animations not always looking correct. Figure [22.5](#) shows an example of an enemy robot that floats above the ground and only tilts when moving. Although not as impressive as a human-like character with legs, simple is better in this case to prevent breaks-in-presence resulting from awkward-looking leg movement.

Character head motion is extremely important for maintaining a sense of social presence [[Pausch et al. 1996](#)]. Fortunately, since the use of HMDs requires head tracking, this information can be directly mapped to one's avatar as seen by others (assuming quality bandwidth). Eye animations can add effect but are not necessary as any three-year-old child can attest to after watching *Sesame Street*. Moving an avatar's eyes that does not result from eye tracking also risks the false impression that the character is not paying attention.



Figure 22.6 An example of a fully immersive VR social experience. (Courtesy of VRChat)

To naturally convey attention, computer-controlled characters should first turn their heads before turning their bodies. Rotating the eyes before head turning can also add effect. However, if a computer-controlled character has eye motion and a user-controlled avatar does not, then that can provide a cue to users of which other characters are user controlled and which are not. Characters can also be used to encourage users to look in certain directions by pointing or lining themselves up between the user and the object to be looked at.

Social networking across distances through technology is not new. Like online 2D virtual worlds, people now spend hours at a time in fully immersive social VR such as VRChat (Figure 22.6). Jesse Joudrey (personal communication, April 20, 2015), CEO of VRChat, claims the urge to communicate with others in VR to be much stronger than in other digital mediums. For example, there is an urge to back up in VR if someone gets in one's personal space. Although he believes face-to-face interaction is the best way of communicating, VR is the second-best way to interact with real people. Interestingly, even though users can visually define themselves differently than what they look like in real life, behavior is more difficult to change. Involuntary body language is difficult to hide even with only basic motion. People who are fidgety in real life also fidget in VR.

1 . <http://marketplace.xbox.com/en-US/AvatarMarketplace> ; accessed June 16, 2015.

Transitioning to VR Content Creation

It is interesting to realize that while the VR industry is rushing to develop VR game systems, the real task will be to fill these systems with interesting and entertaining environments. This task is challenging because the medium is much richer and demanding than existing video games. While a conventional game requires the designer to create a two-dimensional video environment, the nature of VR requires that a completely interactive and rich environment be created.

—Mark Bolas [[1992](#)]

VR development is very different than traditional product or software development. This chapter provides some of the most important points to consider when creating VR content.

23.1 Paradigm Shifts from Traditional Development to VR Development

VR is very different than any other form of media. In order to fully understand and take advantage of VR, it is very useful to approach it as a new artistic medium for expression [[Bolas 1992](#)]. VR is extremely cross-disciplinary and thus it is useful to study and understand other disciplines. However, no single discipline is sufficient by itself. There will be things that work from other disciplines and some that won't. Be prepared to give up on those things that don't work. The following are some key points that should always be at the forefront of a VR content creator's mind.

Focus on the user experience. For VR, the user experience is more important than for any other medium. A bad website design is still usable. A bad VR design will make users sick. The experience must be extremely compelling and engaging to convince people to stick a piece of hardware on their face.

Minimize sickness-inducing effects. Whereas real-world motion sickness is common (e.g., seasickness or from a carnival ride), sickness is almost never an issue with other forms of digital media. More traditional digital media does occasionally cause sickness (e.g., IMAX and 3D film), but it is minor compared to what sickness can occur with VR if not designed properly (Part [III](#)).

Make aesthetics secondary. Aesthetics are nice to have, but other challenges are more important to focus on at the beginning of a project. Start with basic content and incrementally build upon it while the essential requirements are maintained. For example, frame rate is absolutely essential in reducing latency, so geometry should be optimized as necessary. In fact, frame rate is so important that a core requirement of all VR systems should be to match or exceed the refresh rate of the HMD, and the scene should fade out when this requirement is not met (Section [31.15.3](#)).

Study human perception. Perceptions are very different in VR than they are on a regular display. Film and video games can be viewed from different angles with almost no impact on the experience. VR scenes are rendered from a very specific viewpoint of where the user is looking from. Depth cues are important for presence (Section [9.1.3](#)), and inconsistent sensory cues can cause sickness (Section [12.3.1](#)). Understanding how we perceive the real world is directly relevant to creating immersive content.

Give up on having all the action in one part of the scene. No other form of media completely surrounds and encompasses the user. The user can look in any direction at any time. Many traditional first-person video games allow the user to look around with a controller, but there is still the option for the system to control the camera when necessary. Proper VR does not offer such luxury. Provide cues to direct users to important areas but do not depend upon them.

Experiment excessively. VR best practices are not yet standardized (and they may never be) nor are they mature—there are a lot of unknowns so there must be room for experimentation. What works in one situation might not work in another situation. Test across a variety of people that fit within your target audience and make improvements based on their feedback. Then iterate. Iterate. Iterate . . . (Part [VI](#)).

23.2 Reusing Existing Content

Traditional digital media is very different from VR. With that being said, some content can be included or even ported to VR. 2D images or video is very straightforward to apply onto a texture map within a virtual environment. In fact, watching 2D movies within VR has proven to be surprisingly popular. Web content or any other content can even be updated in real time and interacted with on a 2D surface such as that described in Section [28.4.1](#).

Although there are similarities, video game design is very different than VR design, primarily due to being designed for a 2D screen and not having head tracking. Concepts such as head tracking, hand tracking, and the accurate calibration required to match how we perceive the real world are core elements for VR experiences that the designer should consider from the start of the project. If any of these essential VR elements are not implemented well, then the mind will quickly reject any sense of presence and sickness may result. Because of these issues, porting existing games to VR without a redesign and rewrite of the game engine where appropriate is not recommended. Reusing assets and refactoring code where appropriate is the correct solution. With that being said, implementing the right solution requires resources and there will be those who port existing games with no or minimal modification of code. Thus, the following information is provided to help port games for those choosing this path.

23.2.1 Geometric Detail

Games typically use texture excessively in place of 3D geometry. More depth cues, especially stereo cues, and motion parallax cues make textures

obviously 2D. Whereas such flat cardboard-appearing worlds are not necessarily realistic, there is no concern for flat textures to cause sickness.

In addition, many standard video game techniques and geometric hacks simplify 3D calculations for better performance. Simplified geometry and lighting created with screen space techniques, normal maps, textures, billboard sprites, some shaders and shadows (depending on implementation), and other illusion of 3D in games are not actually 3D and often look flat or strange in VR. Many of these techniques fortunately often do work at further distances but should be avoided for close distances. For close objects to be most convincing, they should be modeled in detail. Such detailed objects can be swapped out with lower levels of detailed models and with geometric hacks when they become further from the viewpoint. Some code may need to be modified or rewritten to appear correct in VR, even for further objects.

23.2.2 Heads-Up Displays

Heads-up displays (HUDs) provide visual information overlaid on the scene in the general forward direction. Traditional video-game implementations are especially problematic for VR. This is because most HUDs are implemented as a 2D overlay that is in screen space, not part of the scene. Because there is no depth, most ports simply place the same HUD image to the left and right eyes, resulting in binocular cues being at an infinite depth. Unfortunately, this results in a binocular-occlusion conflict (Section [13.2](#)), which is a major problem. When the two eyes see the same image of the HUD, the HUD appears at an infinite distance so the entire scene should occlude the HUD, but it does not. Some solutions for this are described below.

1. The easiest solution is to turn off all HUD elements. Some game engines enable the user to do this without modifying source code. However, some of the HUD information is important for status and game play, so this solution can leave players quite confused.
2. Draw or render the HUD information to a texture and place onto a transparent quad at some depth in front of the user (and possibly angled and below eye level so the information is accessed by looking down) so that it becomes occluded when other geometry passes in front of it. Factors such as readability, occlusion of the scene, and accommodation-vergence conflict can occur, but these are relatively minor compared to binocular-occlusion conflict and adjustments can be made until reasonably comfortable values are found. This solution works reasonably well, except that the HUD information becomes occluded when other geometry passes in front of it.
3. Separate out the different HUD elements and place them on the user's body and/or in different areas relative to the user's body (e.g., an information belt or at the user's feet). This is the ideal solution but can take substantial work depending on the game and game engine.

23.2.3 Targeting Reticle

A targeting **reticle** is a visual cue used for aiming or selecting objects and is typically implemented as part of the HUD. However, the solutions for a VR reticle may be different.

1. The reticle can be implemented in the same way as option #2 for HUDs as described above. This results in the reticle behaving as a helmet with the reticle on the front of the helmet in front of the eyes. Like a HUD, a challenge with this solution is the reticle is seen as a double image when looking at the target in the distance. This is due to the eyes not being converged on the reticle and is congruent with how one sights a weapon in the real world. This is a more realistic solution as the real world requires one to close one eye to properly aim a weapon. One eye (ideally the system should enable the user to configure the reticle for his dominant eye) must be set to be the sighting eye so that the reticle crosshairs directly overlap the target. In order to know what the user is targeting, the system must know what eye is being used in order to cast a ray from the eye through the reticle.
2. Another option is to project the reticle onto (or just in front of) whatever is being aimed at. Although not as realistic and the reticle jumps large distance in depth, this is preferred by some users as being more comfortable and is less confusing when one doesn't understand the need to close one eye to aim.

23.2.4 Hand/Weapon Models

Similar to HUDs, the hands and/or weapons in desktop games are placed directly in front of the user, either as 2D textures or full 3D objects. The obvious problem is that 2D textures representing the hands/weapons will appear flat and not work well with VR.

Although not realistic and perhaps a bit disturbing at first, replacing 2D hands and arms with 3D hands without arms that are directly controlled with a tracked hand-held controller can work quite well in VR. However, hands with arms are more of a problem. In most first-person desktop video games there is no need to have a full body, so when the user looks down in a game ported to VR there is no body, or at best only some portion of the body. If the game includes hands, then the arms will likely be hanging in space. Rotating a tracked hand-held controller cannot be directly mapped to an entire static hand/arm model because the point of rotation should be about the hand, and rotation of the hand results in the entire arm rotating. In addition, many game engines have a completely different implementation for the hands and weapons than the rest of the world, so generalizing the hands and weapons to behave as other 3D objects may not be possible without re-architecting or refactoring the code. If arms are required, then some form of inverse kinematics will be required.

23.2.5 Zoom Mode

Some games have a zoom mode with a tool or weapon such as a sniper rifle. A zoom lens that moves with the head causes a difference between the physical field of view and the rendered field of view, which results in an unstable virtual world and sickness when the head moves (a similar result can occur with real-world magnifying glasses; Section [10.1.3](#)). Because of this, the zoom should be independent of head pose or only a

small portion of the screen should be zoomed in (i.e., a small part of the screen where the scope is located). If the zoom is affected by head motion and it is not possible to prevent a large part of the scene from zooming in, then zooming should be disabled.

Content Creation: Design Guidelines

Virtual worlds begin as an entirely open slate for the content designer to build upon. The preceding four chapters and the following guidelines will help to lay the foundation and paint the details of newly created worlds.

24.1 High-Level Concepts of Content Creation (Chapter 20.)

Experiencing the Story (Section 20.1)

- Focus on creating and refining a great experience that is enjoyable, challenging, and rewarding.
- Focus on conveying the key points of the story instead of the details of everything. The users should all consistently get those essential points. For the non-essential points, their minds will fill in the gaps with their own story.
- Use real-world metaphors to teach users how to interact and move through the world.
- To create a powerful story, focus on strong emotions, deep engagement, massive stimulation, and an escape from reality.
- Focus on the experience instead of the technology.
- Provide a relatable background story before immersing users into the virtual world.
- Make the story simple and clear.
- Focus on why users are there and what there is to do.
- Provide a specific goal to perform
- Focus on believability instead of photorealism.
- Keep the experience tame enough for the most sensitive users.
- Minimize breaks-in-presence.
- Focus on the targeted audience.

The Core Experience (Section 20.2)

- Keep the core experience the same even with an array of various contexts and constraints.
- Make the core experience enjoyable, challenging, and rewarding so users want to come back for more.
- If the core experience is not implemented well, then no number of fancy features will keep users engaged for more than a few minutes.
- Continuously improve the core experience, build various prototypes focused on that core, and learn from observing and collecting data from real users.

Conceptual Integrity (Section 20.3)

- The core conceptual model should be consistent.
- Make the basic structure of the world directly evident and self-explanatory.
- Eliminate extraneous content and features that are immaterial to the intended experience.
- Have one good idea rather than several loosely related good ideas. If there are too many good incompatible ideas, then reconsider the primary idea to provide an overarching theme that encompasses all of the best ideas in a coherent manner.
- Empower a director who controls the basic concept to lead the project.
- The director defines the project, but is not a dictator of implementation or data collection.
- The director should always give full credit, even when others have implemented his ideas.
- The director should be fully accessible to the makers and data collectors, and highly encourage questions.

Gestalt Perceptual Organization (Section 20.4)

- Use gestalt principles of grouping when creating assets and user interfaces.
- Use gestalt principles of grouping at a higher conceptual level, such as keeping objects simple and temporal closure.
- Use concepts of segregation—ground acts as a stable reference and figure emphasizes objects that can be interacted with.

24.2 Environmental Design (Chapter [21](#))

The Scene (Section [21.1](#))

- Simplify the background and contextual geometry. Focus on fundamental geometry and interactive objects.
- Make sure geometry scaling is consistent. For realistic experiences, include familiar objects with standard sizes that are easily seen by the user.

Color and Lighting (Section [21.2](#))

- Use color to bring out emotions.
- Use multiple colors and only slight variations of white light so users perceive the intended color of objects and color constancy is maintained.
- Capture users' attention with bright colors that stand out.
- Turn on and off color on specific objects to signify use or disable use.

Audio (Section [21.3](#))

- Use audio for situational awareness from all directions, adding emotional impact, cuing visual attention, conveying information without taxing the visual system, and providing unique cues that cannot be perceived through other sensory systems.
- Do not make people deaf by not providing audio.
- Do not overwhelm with sound.
- Use aggressive warning sounds only occasionally to grab attention.
- Use ambient sound effects to provide a sense of realism and presence.
- Use music to evoke emotion.
- Use audio with interfaces as feedback to the user.
- Use audio along with color to replace touch when haptics are not available.
- Convey information through the spoken word.

Sampling and Aliasing (Section [21.4](#))

- Avoid spatially high-frequency visual elements.

Environmental Wayfinding Aids (Section [21.5](#))

- Use environmental wayfinding aids to help users maintain a sense of their position and direction of travel, know where goals are located, and plan in their minds how to get there.
- Use both explicit and subtle wayfinding aids.
- Study the work of non-VR designers such as architectural design and urban planners.
- Place landmarks strategically.
- Provide tall landmarks so they can be seen from most places in the environment.
- Differentiate areas of the environment with regions that have different characteristics.
- Use channels to restrict navigation options while providing a feeling of openness.
- Place signs at nodes.
- Use edges to deter travel.

- Use visual handrails to guide navigation.
- Utilize simple spatial organization such as grid-like patterns.
- Minimize violations of the overarching metaphor of the world structure.
- Adding structure to abstract or scientific data can provide context, but be careful of imposing inappropriate structure on data that does not exist in the data itself. Consult with subject-matter experts before adding such cues to datasets.
- Implement breadcrumbs, trails, markup tools, user-placed markers, and measurement tools for users to better understand and think about the environment, as well as to share information with other users.

Real-World Content (Section [21.6](#))

- For immersive film, reconsider the entire movie experience as the viewer being within and a part of the scene.
- When capturing the world in 360°, make sure that all equipment and people that are not part of the story are not visible to the camera.
- For 360° camera capture, remove stereoscopic cues in the down and up directions.
- Capture light fields or true 3D data to enable motion parallax.
- Empower scientists by enabling them to walk through their own datasets and see that data in new ways.

24.3 Affecting Behavior (Chapter [22](#))

Personal Wayfinding Aids (Section [22.1](#))

- Use abstract representations in maps to portray understanding of essential environmental concepts.
- Use “you-are-here” maps with arrows or field-of-view markings to convey both location and direction.
- For large environments, allow the map to be scaled.
- Use forward-up maps when concurrently navigating and/or searching in an egocentric manner in order to enable easy matching between points on the map and corresponding landmarks in the environment.
- Use north-up maps for exocentric tasks (e.g., when users want (1) to familiarize themselves with the entire layout of the environment independent of where they are located or (2) to plan routes before navigating).
- Use compasses so users can easily direct themselves in the direction of intended travel.
- Place a compass around the user, such as floating in space where it can be easily grabbed, surrounding the user at eye level, or surrounding the user at the feet.

Center of Action (Section [22.2](#))

- Consider breaking the scene around the user into zones.

Casual vs. High-End VR (Section [22.4](#))

- When creating content, take into account whether the system is wired or wireless.
- If virtual turning is implemented, move the controls/hands with the torso.
- For wired systems, put the majority of content in the general forward direction.
- Design and optimize for a single system, even when supporting multiple types of systems.

Characters, Avatars, and Social Networking (Section [22.5](#))

- When the bandwidth is available, map head tracking data to avatars’ heads.
- Unless the intent is for users to distinguish between computer-controlled character behavior and user-controlled avatars, make the computer-controlled characters behave in a similar manner as avatars (e.g., similar head movements).
- Be careful of adding artificial eye movements to avatars to avoid the risk of conveying that the remote user is not paying attention.
- To convey natural attention of a computer-controlled character, first move the eyes (if implemented), then the head, then the body.

- To get users to look in certain directions, use computer-controlled characters to place themselves between the user and the intended look direction, or have the characters look and/or point in the intended direction.
- For non-realistic experiences, use caricature to exaggerate the most important features of characters.
- Consider not using legs for characters that move in order to reduce breaks-in-presence caused by awkward-looking leg movement.
- Provide the ability for users to personalize their own avatars.

24.4 Transitioning to VR Content Creation (Chapter [23](#))

Paradigm Shifts from Traditional Development to VR Development (Section [23.1](#))

- Study human perception and other disciplines, but be prepared to give up on concepts that do not work with VR.
- Focus on the user experience as the experience is more important for VR than for any other medium.
- Minimize sickness-inducing effects.
- Calibrate often and excessively. Poor calibration is a major cause of motion sickness.
- Make aesthetics secondary. Start instead with basic content and incrementally build while focusing on ease of use and maintaining essential requirements.
- Give up on having all the action in one part of the scene.
- Experiment excessively. There are unknowns and various forms of VR—what works in one situation might not work in another situation.
- Consider core concepts of VR such as head and hand tracking from the start of the project.

Reusing Existing Content (Section [23.2](#))

- Instead of directly porting desktop applications to VR, reuse assets and refactor code where needed.
- Understand that geometric hacks (e.g., representing depth with textures) that work well with desktop systems often appear strange in VR.
- Be very careful of using desktop heads-up displays and targeting reticles as straightforward porting/implementation causes binocular-occlusion conflicts.
- Replace desktop 2D hands/weapons with 3D models.
- When using tracked hand-held controllers, do not render the arms unless inverse kinematics is available.
- Be careful of using a zoom mode.

PART V

INTERACTION

For this book, interaction is defined to be a subset of general communication (Section 1.2). **Interaction** is the communication that occurs between a user and the VR application that is mediated through the use of input and output devices. An **interface** is the VR system side of the interaction that exists whether a user is interacting or not.

Interaction in VR at first seems obvious—simply interact with the virtual world in a similar way that we interact with the real world. Unfortunately, natural real-world interfaces (and Hollywood interfaces) often do not work well within VR as one would expect. This is not only because virtual worlds do not model the real world as closely as we would like, but even if they did, abstract interfaces are often superior (e.g., if one wanted to look up information in a virtual world, one would not want to have to travel to a virtual library to find a virtual book). Whereas some of the basic concepts from traditional desktop interfaces can be used within immersive environments, there are few similarities between the two. Creating quality interactions is currently one of the greatest challenges for VR.

Well-implemented interaction techniques enable high levels of performance and comfort while diminishing the impact of human and hardware limitations. It is the role of the interaction designer to make complex interactions intuitive and effective.

Part V consists of five chapters that provide an overview of the most important aspects of VR interaction design.

Chapter 25, Human-Centered Interaction, reviews core concepts for general human-centered interaction: intuitiveness, Norman's principles of interaction design, direct vs. indirect interaction, the cycle of interaction, and the human hands.

Chapter 26, VR Interaction, discusses some of the most important distinctions especially relevant to VR interaction, such as interaction fidelity, proprioceptive and egocentric interaction, reference frames, multimodal interaction, and unique challenges of VR interaction.

Chapter 27, Input Devices, provides an overview of input device characteristics and classes of various input devices. No single input device is appropriate for all VR applications, and choosing the most appropriate input device for the intended experience is essential for optimizing interactions.

Chapter 28, Interaction Patterns and Techniques, describes several interaction patterns and examples of various interaction techniques. An interaction pattern is a generalized high-level interaction concept that can be used over and over again across different applications to achieve common user goals. An interaction technique is a more specific implementation of an interaction pattern.

Chapter 29, Interaction: Design Guidelines, summarizes the previous four chapters and lists a number of actionable guidelines for developing interaction techniques that largely depend on application needs.

Human-Centered Interaction

VR has the potential to provide experiences and deliver results that cannot be otherwise achieved. However, VR interaction is not just about an interface for the user to reach their goals. It is also about users working in an intuitive manner that is a pleasurable experience and devoid of frustration. Although VR systems and applications are incredibly complex, it is up to designers to take on the challenge of having the VR application effectively communicate to users how the virtual world and its tools work so that those users can achieve their goals in an elegant manner.

Perhaps the most important part of VR interaction is the person doing the interacting. Human-centered interaction design focuses on the human side of communication between user and machine—the interface from the user’s point of view. Quality interactions enhance user understanding of what has just occurred, what is happening, what can be done, and how to do it. In the best case, not only will goals and needs be efficiently achieved, but the experiences will be engaging and enjoyable.

This section describes several general human-centered design concepts that are essential for interaction designers to consider when designing VR interactions.

25.1 Intuitiveness

Mental models (Section [7.8](#)) of how a virtual world works are almost always a simplified version of some complexity. When interacting, there is no need for users to understand the underlying algorithms—just the high-level relationship between objects, actions, and outcomes. Whether a VR interface attempts to be realistic or not, it should be intuitive. An intuitive interface is an interface that can be quickly understood, accurately predicted, and easily used. Intuitiveness is in the mind of the user, but the designer can help form this intuitiveness by conveying through the world and interface itself concepts that support the creation of a mental model.

An **interaction metaphor** is an interaction concept that exploits specific knowledge that users already have of other domains. Interaction metaphors help users quickly develop a mental model of how an interaction works. For example, VR users typically think of themselves as “walking” through an environment, even though in most implementations they are not physically moving their feet (e.g., they may be controlling the walking with a hand-held controller). For such an interface, users assume they are traveling at a set height above a surface, and if that height changes, then the interaction does not fit the mental model and the user will likely become confused.

Mental models can be made more consistent across users by providing manuals, communicating with other users, and from the virtual world itself. For fully immersive VR, users are cut off from the real world and there is no guarantee they will look at a manual or talk with others. Thus, the virtual world should be sufficient in projecting appropriate information needed to create a consistent conceptual model of how things work in the mind of each user, without requiring external explanation. Otherwise, problems will occur when an application does not fit with the user’s mental model. It should not be assumed that an expert will always be available to directly explain how an interface works, answer questions, or correct mistakes. Too often, a VR creator expects the user’s model to be identical to what she thinks she has designed, but unfortunately this is rarely the case since creators and users think differently (as they should). Tutorials that cannot be completed until the user has indicated a clear understanding and effective interaction are a great method of inducing mental models into the minds of users.

25.2 Norman’s Principles of Interaction Design

When users interact in VR, they need to figure out how to work the system. **Discoverability** is exploring what something does, how it works, and what operations are possible [[Norman 2013](#)]. Discoverability is especially important for fully immersive VR because the user is blind and deaf to the real world, cut off from those real-world humans who want to help. Essential tools can lead the mind into discovering how an interface works through consistent affordances, unambiguous signifiers, constraints to guide actions and ease interpretation, immediate and useful feedback, and obvious and understandable mappings. These principles as defined by Norman are summarized below along with how they relate to VR and are a good starting point when designing and refining interactions.

25.2.1 Affordances

Affordances define what actions are possible and how something can be interacted with by a user. We are used to thinking that properties are associated with objects, but an affordance is not a property; an affordance is a *relationship* between the capabilities of a user and the properties of a thing. Interface elements afford interaction, such as a virtual hand affords selection. An affordance between an object and one user may be different between that object and another user. Light switches on a wall offer the ability to control the lighting in a room, but only for those who are able to reach the switches. Some objects in virtual environments afford selecting, moving, controlling, etc. Good interaction design focuses on creating appropriate affordances to make desired actions easily doable with the technology used (e.g., a tracking system that is able to track the hand near the point of the light switch) and by the intended user base (e.g., the designer may intentionally place a light switch to not be reachable by a certain class of users in order to encourage collaboration).

25.2.2 Signifiers

Some affordances are perceivable, others are not. To be effective, an affordance should be perceivable. A **signifier** is any perceivable indicator (a signal) that communicates appropriate purpose, structure, operation, and behavior of an object to a user. A good signifier informs a user what is

possible before she interacts with its corresponding affordance.

Examples of signifiers are signs, labels, or images placed in the environment indicating what is to be acted upon, which direction to gesture, or where to navigate toward. Other signifiers directly represent an affordance, such as the handle on a door or the visual and/or physical feel of a button on a controller. A misleading signifier can be ambiguous or not represent an affordance—something may look like a drawer to be opened when in fact it cannot be opened. Such a false signifier is usually accidental or not yet implemented. But a misleading signifier can also be purposeful—such as to motivate users to find a key in order to turn the non-accessible drawer into something that can be opened. In such cases, the content creator should be aware of such anti-signifiers and be careful not to frustrate users.

Signifiers are most often intentional, but, as mentioned above, they may also be accidental. An example of an intentional signifier is a sign giving directions. In the real world, an example of an accidental and unintentional (but useful) signifier is garbage on a beach representing unhealthy conditions. At first thought, we might think signifiers are only intentionally created in VR, for the VR creator created everything from that which does not actually exist. However, this is not always the case. An unintended VR signifier might be an object that looks like it is designed to be picked up to be placed into a puzzle, but it can also be perceived as an object that can be picked up and thrown (a common occurrence much to the frustration of content creators). Or an unintended signifier in a social VR experience might be a gathering of users at an area of interest, signifying to others to navigate to that location to investigate what is happening and what affordance might be available there.

Signifiers might not be attached to a specific object. Signifiers can be general information. Conveying what interaction mode a user is currently using can help prevent confusion.

Regardless of how and why signifiers are created, signifiers are important for communicating to users whether or not action is possible, and what those actions are. Good VR design ensures signifiers are effectively discoverable through signifiers that are well communicated and intelligible.

25.2.3 Constraints

Interaction constraints are limitations of actions and behaviors. Such constraints include logical, semantic, and cultural limitations to guide actions and ease interpretation. With that being said, this section focuses on the physical and mathematical constraints as such constraints most directly apply to VR interactions. For an overview of general project constraints, see Section [31.10](#).

Proper use of constraints can limit possible actions, which makes interaction design feasible and can simplify interaction while improving accuracy, precision, and user efficiency [[Bowman et al. 2004](#)]. A commonly used method for simplifying VR interactions is to constrain interfaces to only work in a limited number of dimensions. The **degrees of freedom** (DoF) for an entity are the number of independent dimensions available for the motion of that entity (also see Section [27.1.2](#)). The DoF of an interface can be constrained by the physical limitations of an input device (e.g., a physical dial has one DoF, a joystick has two DoF), the possible motion of a virtual object (e.g., a slider on a panel is constrained to move along a single axis to control a single value), or travel that is constrained to the ground, enabling one to more easily navigate through a scene.

Physical limitations constrain possible actions and these limitations can also be simulated in software. In addition to being useful, constraints can also add more realism. For example, a virtual hand can be stopped even when there is nothing physically stopping a user's real hand from going through the object (although this results in visual-physical conflict; Section [26.8](#)). However, physics should not always necessarily be used as physics can sometimes make interactions more difficult. For example, it can be useful to leave virtual tools hanging in the air when not using them.

Interaction constraints are more effective and useful if appropriate signifiers make them easy to perceive and interpret, so users can plan appropriately before taking action. Without effective signifiers, users might effectively be constrained because they are not able to determine what actions are possible. Consistency of constraints can also be useful as learning can be transferred across tasks. People are highly resistant to change; if a new way of doing things is only slightly better than the old, then it is better to be consistent. For experts, providing the ability to remove constraints can be useful in some situations. For example, advanced flying techniques might be enabled after users have proved they can efficiently maneuver by being constrained to the ground.

25.2.4 Feedback

Feedback communicates to the user the results of an action or the status of a task, helps to aid understanding of the state of the thing being interacted with, and helps to drive future action. In VR, timely feedback is essential. Something as simple as moving the head requires immediate visual feedback, otherwise the illusion of a stable world is broken and a break-in-presence results (and, even worse, motion sickness). Input devices capture users' physical motion that is then transformed into visual, auditory, and haptic feedback. At the same time, feedback internal to the body is generated from within—proprioceptive feedback that enables one to feel the position and motion of the limbs and body. Unfortunately, it is quite difficult for VR to provide all possible types of feedback. Haptic feedback is especially difficult to implement in a way similar to how real forces occur in the real world. Section [26.8](#) discusses how sensory substitution can be used in place of strong haptic cues. For example, objects can make a sound, become highlighted, or vibrate a hand-held controller when a hand collides with them or they have been selected.

Feedback is essential for interaction, but not when it gets in the way of interaction. Too much feedback can overwhelm the senses, resulting in cluttered perception and understanding. Feedback should be prioritized so less important information is presented in an unobtrusive manner and essential information always captures attention. Instead of putting information on a heads-up display in the head reference frame, place it near the waist or toward the ground in the torso reference frame so the information is easily accessed when needed but not in the way otherwise. If information must always be visible with a heads-up display, then only provide the most essential minimal information as anything directly in front of a user reduces awareness of the virtual world. Too many audio beeps or, worse, overlapping audio announcements, can cause users to ignore all of them or even cause them to be indecipherable even if users wanted to listen to them. Such clutter not only can result in unusable applications, but

can be annoying (think of backseat drivers!) and is inappropriate. In many cases, users should have the option to turn off or turn down feedback that is not important to them.

25.2.5 Mappings

A **mapping** is a relationship between two or more things. The relationship between a control and its results is easiest to learn where there is an obvious understandable mapping between controls, action, and the intended result. Mappings are useful even when the person is not directly holding the thing being manipulated, for example, when using a screwdriver to lift a lever up that cannot be directly reached.

Mappings from hardware to interaction techniques defined by software are especially important for VR. Often a device will have a natural mapping to one technique but poor mapping to another technique. For example, hand-tracked devices work well for a virtual hand technique and for pointing (Section 28.1) but not as well for a driving simulator where a physical steering wheel would be more appropriate.

Compliance

Compliance is the matching of sensory feedback with input devices across time and space. Maintaining compliance improves user performance and satisfaction. Compliance results in perceptual binding (Section 7.2.1) so that interaction feels as if one is interacting with a single coherent object. Visual-vestibular compliance is especially important for reducing motion sickness (Section 12.3.1). Compliance can be divided into spatial compliance and temporal compliance [Bowman et al. 2004] as described below.

Spatial compliance. Direct spatial mappings lead immediately to understanding. For example, we intuitively understand that once we have grabbed an object, to move the object up, we simply move the hand holding the object up. **Spatial compliance** consists of position compliance, directional compliance, and nulling compliance.

Position compliance is the co-location of sensory feedback with the input device position. An example of spatial compliance is when the proprioceptive sense of where the hand is matches the visual sense of where the hand is. Not all interaction techniques require position compliance, but position compliance results in direct intuitive interaction and should be used whenever appropriate. Labels placed at the location where physical controls are located on a hand-held device is an example of position compliance. Position compliance is important for tracked physical devices that the user can pick up and/or interact with. Consider users who have just placed an HMD on their head but have not yet picked up the hand controllers (or who have previously set the controller down); the user must be able to see the controller in the correct location in order to pick it up.

Directional compliance is the most important of the three spatial compliances. **Directional compliance** states virtual objects should move and rotate in the same direction as the manipulated input device. This results in correspondence of what is seen and what is felt by the body, resulting in more direct interaction. This enables the user to effectively anticipate motion in response to physical input and therefore plan and execute appropriately.

An example of directional compliance is the mapping from a mouse to a cursor on the screen. Even though a mouse is spatially dislocated from the screen (i.e., it lacks position compliance), the hand/mouse movement results in an immediate and isomorphic movement of the cursor on the screen, which makes the user feel as if he is directly moving the cursor itself [van der Veer and del C. P. Melguizo 2002]. The user does not think about the offset between the mouse and the cursor when the mouse is used as intended. Even though the mouse typically sits on a flat horizontal surface and the screen sits vertically (i.e., the screen is rotated 90° from the mouse), intuitively we expect if we move the mouse forward/back then the cursor will move up/down in the same way because we think of both as up/down movements. However, for anyone who has attempted to use a mouse when it is rotated by 90° to the right, they know simple manipulations can be extremely difficult due to up becoming left and right becoming up. The same is true for moving objects in VR; the direction of movement for a grabbed object, even if at a distance, should be mapped directly to that selected object whenever possible. Likewise, when a user rotates a virtual object with an input device, the virtual object should rotate in the same direction; that is, both should rotate around the same axis of rotation [Poupyrev et al. 2000].

Nulling compliance states that when a device returns to its initial placement, the corresponding virtual object should also return to its initial placement [Buxton 1986]. Nulling compliance can be accomplished with absolute devices (relative to some reference), but not relative devices (relative to itself)—see Section 27.1.3. For example, if a device is attached to the user’s belt, nulling compliance is important, as the user can use “muscle memory” to remember the initial, neutral placement of the device and corresponding virtual object (Section 26.2).

Temporal compliance. **Temporal compliance** states that different sensory feedback corresponding to the same action or event should be synced appropriately in time. Viewpoint feedback should be immediate to match vestibular cues, otherwise motion sickness may result (Chapter 15). But even for reasons unrelated to sickness, feedback should be immediate; otherwise users may become frustrated or give up before tasks are completed. Even if the entire action cannot be completed immediately, there should be some form of feedback implying the problem is being worked on. Without such information, users can become annoyed and computing resources can be wasted since the user may have forgotten about the task and moved on to something else. In fact, slow or poor feedback may be worse than no feedback, as it can be distracting, irritating, and anxiety provoking. Anyone who has attempted to browse the Web on an extremely slow Internet connection can attest to this.

Non-spatial Mappings

Non-spatial mappings are functions that transform a spatial input into a non-spatial output or a non-spatial input into a spatial output. Some indirect spatial to non-spatial mappings are universal, such as moving the hand up signifies more and moving the hand down signifies less. Other mappings are personal, cultural, or task dependent. For example, some people think of time moving from left to right whereas others think of time

moving from behind the body to the front of it.

25.3 Direct vs. Indirect Interaction

Different types of interaction can be thought of as being on a continuum from indirect interaction to direct interaction. Both direct and indirect interactions are useful for VR depending upon the task. Use each where appropriate and do not use one where not appropriate, e.g., don't try to force everything to be direct.

Direct interaction is an impression of direct involvement with an object rather than of communicating with an intermediary [[Hutchins et al. 1986](#)]. The most direct interaction occurs when the user directly interacts with a physical object in the hands. Well-designed hand-held tools (e.g., a knife) that directly affect an object are only slightly less direct, as once a user understands the tool, it seems to become one with the user as if the tool was an extension of the body instead of an intermediary object.

An example of direct interaction is when a user moves a virtual object on a touch screen with his finger. However, the virtual object does not follow the finger when the finger leaves the plane of the screen. VR enables more direct interactions than other digital technologies because virtual objects can be directly mapped to the hands in 3D space with full spatial (both directional and positional) and temporal compliance (Section [25.2.5](#)). Directional compliance and temporal compliance are more important than position compliance; consider the feeling of directness with a mouse (albeit not quite as directly as moving the cursor with a finger on a touch screen), even though there is no position compliance. Similarly, manipulating an object at a distance can feel like one is directly manipulating the object.

Indirect interaction requires more cognition and conversion between input and output. Performing a search for an image through a typed or verbal query is an example of an indirect interaction. The user must put thought into what is being searched for, provide the query, wait for a response, and then interpret the result. Not only must the query be thought about but the conversion from words to and from visual imagery must be considered. Even though such indirect interaction requires more cognition than direct interactions, that does not mean direct interaction is always better. Indirect interactions are more effective for their intended purposes.

Somewhere in the middle of the extremes of direct and indirect interactions are semi-direct interactions. An example is a slider on a panel that controls the intensity of a light. The user is directly controlling the intermediary slider but less directly controlling the light. However, once the slider is grabbed and moved back and forth a couple of times it feels more direct due to the mental mapping of up for lighter and down for darker.

25.4 The Cycle of Interaction

Interaction can be broken down into three parts: (1) forming the goal, (2) executing the action, and (3) evaluating the results [[Norman 2013](#)].

Execution bridges the gap between the goal and result. This feedforward is accomplished through the appropriate use of signifiers, constraints, mappings, and the user's mental model. There are three stages of execution that follow from the goal: plan, specify, and perform.

Evaluation enables judgment about achieving a goal, making adjustments, or the creation of new goals. This feedback is obtained by perceiving the impact of the action. Evaluation consists of three stages: perceive, interpret, and compare.

As can be seen in Figure [25.1](#), the seven stages of interaction consist of one stage for goals, three stages for execution (plan, specify, and perform), and three stages for evaluation (perceive, interpret, and compare). Quality interaction design considers requirements, intentions, and desires at each stage. An example interaction is as follows.

1. Form the **goal**. The question is "What do I want to accomplish?" Example: Move a boulder blocking an intended route of travel.
2. **Plan** the action. Determine which of many possible plans of action to follow. Question: "What are the alternative set of action sequences and what do I choose?" Example: Navigate to the boulder or select from a distance to move it.
3. **Specify** an action sequence. Even after the plan is determined, the specific sequence of actions must be determined. Question: "What are the specific actions of the sequence?" Example: Shoot a ray from the hand, intersect the boulder, push the grab button, move the hand to the new location, release the grab button.

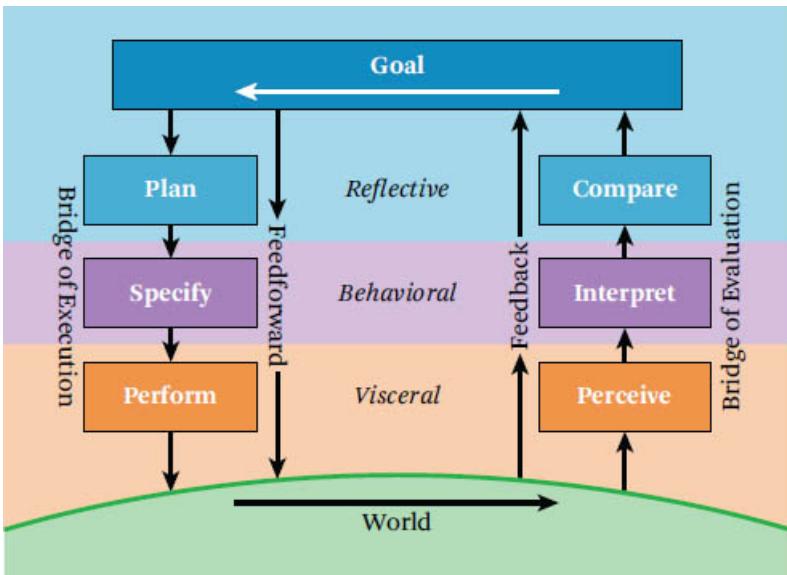


Figure 25.1 The cycle of interaction. (Adapted from [Norman \[2013\]](#))

4. **Perform** the action sequence. Question: “Can I take action now?” One must actually take action to achieve a result. Example: Move the boulder.

5. **Perceive** the state of the world. Question: “What happened?” Example: The boulder is now in a new location.

6. **Interpret** the perception. Question: “What does it mean?” Example: The boulder is no longer in the path of desired travel.

7. **Compare** the outcome with the goal. Questions: “Is this okay? Have I accomplished my goal?” Example: The route has been cleared so navigation along the route is possible.

The interaction cycle can be initiated by establishing a new goal (goal-driven behavior). The cycle can also initiate from some event in the world (data-driven or event-driven behavior). When initiated from the world, goals are opportunistic rather than planned. Opportunistic interactions are less precise and less certain than explicitly specified goals, but they result in less mental effort and more convenience.

Not all activities in these seven stages are conscious. Goals tend to be reflective (Section 7.7) but not always. Often, we are only vaguely aware of the execution and evaluation stages until we come across something new or run into an obstacle, at which point conscious attention is needed. At a high reflective level of goal setting and comparison, we assess results in terms of cause and effect. The middle levels of specifying and interpreting are often only semi-conscious behavior. The visceral level of performing and perceiving is typically automatic and subconscious unless careful attention is drawn to the actual action.

Many times the goal is known, but it is not clear how to achieve it. This is known as the **gulf of execution**. Similarly, the **gulf of evaluation** occurs when there is a lack of understanding about the results of an action. Designers can help users bridge these gulfs by thinking about the seven stages of interaction, performing task analysis for creating interaction techniques (Section 32.1), and utilizing signifiers, constraints, and mappings to help users create effective mental models for interaction.

25.5 The Human Hands

The human hands are remarkable and complex input and output devices that naturally interact with other physical objects quickly, precisely, and with little conscious attention. Hand tools have been perfected over thousands of years to perform intended tasks effectively. Figure 8.8 shows that a large portion of the sensory cortex is devoted to the hands. It is not surprising, then, that the best fully interactive VR applications make use of the hands. Using interaction techniques that the hands can intuitively work with can add significant value to VR users.

25.5.1 Two-Handed Interaction

It is natural and intuitive to reach out with both hands to manipulate objects in the real world, so common sense says two-handed 3D interfaces should be appropriate and intuitive for VR [[Schultheis et al. 2012](#)]. Whereas such intuition tells us that two hands are better than one, two hands can in fact be worse than one hand if the interaction is designed inappropriately [[Kabbash et al. 1994](#)]. Perhaps this is a reason why, even though devices like the mouse have been around for decades, most computer interactions are one handed. Anyone who has developed 3D systems knows that 3D devices alone do not guarantee superior performance, and this is even more true for two hands, in part because the hands do not necessarily work in parallel [[Hinckley et al. 1998](#)]. Iterative design with feedback from real users (Part VI) is essential for creating high-quality bimanual interfaces.

Bimanual Classifications

Bimanual interaction (two-handed interaction) can be classified as **symmetric** (each hand performs identical actions) or **asymmetric** (each hand performs a different action), with asymmetric tasks being more common in the real world [[Guisard 1987](#)].

Bimanual symmetric interactions can further be classified as synchronous (e.g., pushing on a large object with both hands) or asynchronous (e.g., climbing a ladder with one arm reaching up at a time). Scaling by grabbing two sides of an object and spreading the hands apart is an example of a bimanual symmetric interaction.

Bimanual asymmetric interactions occur when both hands work differently but in a coordinated way to accomplish a task. The **dominant hand** is the user's preferred hand for performing fine motor skills. The **non-dominant hand** provides the reference frame, giving the ergonomic benefit of placing the object being worked upon in a way that is comfortable (often subconsciously) for the dominant hand to work with and that does not force the dominant hand to work in a single locked position. The non-dominant hand also typically initiates manipulation of the task and performs gross movements of the object being manipulated, in order to provide convenient, efficient, and precise manipulation by the dominant hand. The most commonly given example is writing; the non-dominant hand controls the orientation of the paper for writing by the dominant hand. Or consider peeling a potato—the potato is much easier to peel when held in the non-dominant hand than when the potato is sitting on a table (whether the potato is locked in place or not). In a similar manner, unimanual interactions in VR can be awkward when the non-dominant hand does not control the reference frame for the dominant hand.

By using two hands in a natural manner, the user can specify spatial relationships, not just absolute positions in space. Bimanual interactions ideally should be designed so the two hands work together in a fluid manner, switching between symmetric and asymmetric modes depending on the current task.

VR Interaction Concepts

VR interactions are not without their challenges. Trade-offs must be considered that may result in interactions being different than in the real world. However, there are enormous advantages VR has over the real world as well. This chapter focuses upon interaction concepts, challenges, and benefits specific to VR.

26.1 Interaction Fidelity

VR interactions are designed on a continuum ranging from an attempt to imitate reality as closely as possible to in no way resembling the real world. Which goal to strive toward depends on the application goals, and most interactions fall somewhere in the middle. **Interaction fidelity** is the degree to which physical actions used for a virtual task correspond to the physical actions used in the equivalent real-world task [[Bowman et al. 2012](#)].

On the high end of the interaction fidelity spectrum are **realistic interactions** —VR interactions that work as closely as possible to the way we interact in the real world. Realistic interactions strive to provide the highest level of interaction fidelity possible given the hardware being used. Holding one hand above the other as if holding a bat and swinging them together to hit a virtual baseball has high interaction fidelity. Realistic interactions are often important for training applications, so that which is learned in VR can be transferred to the real-world task. Realistic interactions can also be important for simulations, surgical applications, therapy, and human-factors evaluations. If interactions are not realistic in such applications, problems such as adaptation (Section [10.2](#)) may occur, which can lead to negative training effects for the real-world task being trained for. An advantage of using realistic interactions is that there is little learning required of users since they already know how to perform the actions.

On the other end of the interaction fidelity spectrum are **non-realistic interactions** that in no way relate to reality. Pushing a button on a non-tracked controller to shoot a laser from the eyes is an example of an interaction technique that has low interaction fidelity. Low interaction fidelity is not necessarily a disadvantage as it can increase performance, cause less fatigue, and increase enjoyment.

Somewhere in the middle of the interaction fidelity spectrum are **magical interactions** where users make natural physical movements but the technique makes users more powerful by giving them new and enhanced abilities or intelligent guidance [[Bowman et al. 2012](#)]. Such magical hyper-natural interactions attempt to create “better” ways of interacting by enhancing usability and performance through superhuman capabilities and unrealistic interactions [[Smith 1987](#)]. Although not realistic, magical often uses interaction metaphors (Section [25.1](#)) to help users quickly develop a mental model of how an interaction works. Consider interaction metaphors as a source of inspiration for creating new magical interaction techniques. Grabbing an object at a distance, pointing to fly through a scene, or shooting fireballs from the hand are examples of magical interactions. Magic interactions strive to enhance the user experience by reducing interaction fidelity and circumventing the limitations of the real world. Magic works well for games and teaching of abstract concepts.

Interaction fidelity is a multi-dimensional continuum of components. The Framework for Interaction Fidelity Analysis [[McMahan et al. 2015](#)] categorizes interaction fidelity into three concepts: biomechanical symmetry, input veracity, and control symmetry.

Biomechanical symmetry is the degree to which physical body movements for a virtual interaction correspond to the body movements of the equivalent real-world task. Biomechanical symmetry makes heavy use of posture and gestures that replicate how one positions and moves their body in the real world. This provides a high sense of proprioception resulting in a high sense of presence since the user feels his body physically acting in the environment as if he were performing the task in the real world. Real walking for VR navigation purposely has a biomechanical symmetry to how we walk in the real world. Walking in place has a lower biomechanical symmetry due to its less realistic movements. Pressing a button or joystick to walk forward has no biomechanical symmetry.

Input veracity is the degree to which an input device captures and measures users’ actions. Three aspects that dictate the quality of input veracity are accuracy, precision, and latency. A system with low input veracity can significantly affect performance due to difficulty capturing quality input.

Control symmetry is the degree of control a user has for an interaction as compared to the equivalent real-world task. High-fidelity techniques provide the same control as the real world without the need for different modes of interaction. Low control symmetry can result in frustration due to the need to switch between techniques to obtain full control. For example, directly manipulating object position and rotations (6 DoF) with a tracked hand controller has greater control symmetry than indirectly manipulating the same object with gamepad controls because the gamepad controls (less than 6 DoF) require using multiple translation and rotational modes. However, low control symmetry can also have superior performance if implemented well. For example, non-isomorphic rotations (Section [28.2.1](#)) can be used to increase performance by amplifying hand rotations.

26.2 Proprioceptive and Egocentric Interaction

As described in Section [8.4](#), proprioception is the physical sense of the pose and motion of the body and limbs. Because most VR systems do not provide a sense of touch outside of hand-held devices, proprioception can be especially important for exploiting the one real object every user has —the human body [[Miné et al. 1997](#)]. The body provides an egocentric frame of reference (Section [26.3.3](#)) in which to work, and interactions relative to the body’s reference frame are more effective than techniques relying solely on visual information. In fact, eyes-off interactions can be performed in peripheral vision or even outside the field of view of the display, which reduces visual clutter. The user also has a more direct sense of control within personal space—it is easier to place an object directly with the hand rather than through less direct means.

26.2.1 Mixing Egocentric and Exocentric Interactions

Exocentric interactions consist of viewing and manipulating a virtual model of the environment from outside of it. With egocentric interaction, the user has a first-person view of the world and typically interacts from within the environment. Don't assume one or the other must be chosen. These egocentric and exocentric interactions can be mixed together so that the user can view himself on a smaller map (Sections [22.1.1](#) and [28.5.2](#)), and/or manipulate the world in an exocentric manner but from an egocentric perspective (Figure [26.1](#)).

26.3 Reference Frames

A **reference frame** is a coordinate system that serves as a basis to locate and orient objects. Understanding reference frames is essential to creating usable VR interactions. This section describes the most important reference frames as they relate specifically to VR interaction. The virtual-world reference frame, real-world reference frame, and torso reference frame are all consistent with one other when there is no capability to rotate, move, or scale the body or the world (e.g., no virtual body motion, no torso tracking, and no moving the world). The reference frames diverge when that is not the case.

Although thinking abstractly about reference frames and how they relate can be difficult, reference frames are naturally perceived and more intuitively understood when one is immersed and interacting with the different reference frames. The reference frames discussed in this section will be more intuitively understood by actually experiencing them.



Figure 26.1 An exocentric map view from an egocentric perspective. (Courtesy of Digital ArtForms)

26.3.1 The Virtual-World Reference Frame

The **virtual-world reference frame** matches the layout of the virtual environment and includes geographic directions (e.g., north) and global distances (e.g., meters) independent of how the user is oriented, positioned, or scaled. When creating content over a wide area, forming a cognitive map, determining global position, or planning travel on a large scale (Section [10.4.3](#)), it is typically best to think in terms of the exocentric virtual-world reference frames. Care should be taken when placing direct hand interfaces relative to the virtual-world reference frame as reaching specific locations can be difficult and awkward unless the user is able to easily and precisely navigate and turn through the environment.

26.3.2 The Real-World Reference Frame

The **real-world reference frame** is defined by real-world physical space and is independent of any user motion (virtual or physical). For example, as a user virtually flies forward the user's physical body is still located in the real-world reference frame. A physical desk, computer screen, or keyboard sitting in front of the user is in the real-world reference frame. A consistent physical location should be provided to set any tracked or non-tracked hand-held controller when not being used. For tracked controllers or other tracked objects, make sure to match the virtual model with the physical controller in form and position/orientation in the real-world reference frame (i.e., full spatial compliance) so users can see it correctly and more easily pick it up. In order for virtual objects, interfaces, or rest frames to be solidly locked into the real-world reference frame, the VR system must be well calibrated and have low latency. Such interfaces often, but not always, provide output cues only to help provide a rest frame (Section [12.3.4](#)) in order for users to feel stabilized in physical space and to reduce motion sickness. Automobile interiors, cockpits (Figure [18.1](#)), or non-realistic stabilizing cues (Figure [18.2](#)) are examples of cues in the real-world reference frame. In some cases it makes sense to add the capability to input information through real-world reference-framed elements (e.g., buttons located on a virtual cockpit). A big advantage of real-world reference frames is that passive haptics (Section [3.2.3](#)) can be added to provide a sense of touch that matches visually rendered elements.

26.3.3 The Torso Reference Frame

The **torso reference frame** is defined by the body's spinal axis and the forward direction perpendicular to the torso. The torso reference frames are especially useful for interaction because of proprioception (Sections [8.4](#) and [26.2](#))—the sense of where one's arms and hands are felt relative

to the body. The torso reference frame can also be useful for steering in the direction the body is facing (Section 28.3.2).

The torso reference frame is similar to the real-world reference frame in the sense that both frames move with the user through the virtual world as the user virtually translates or scales. The difference is that virtual objects in the torso reference frame, virtual objects rotate with the body (both virtual and physical body turns) and move with physical translation whereas objects in the real-world reference frame do not. The chair a user is seated in can be tracked instead of the torso if it can be assumed the torso is stable relative to the chair. Systems with head tracking but not torso or chair tracking can assume the body is always facing forward (i.e., the torso reference frame and real-world reference frame are consistent). However, physical turning of the body can cause problems due to the system not knowing if only the head turned or the entire body turned.

If no hand tracking is available, the hand reference frame can be assumed to be consistent with the torso reference frame. For example, a visual representation of a non-tracked hand-held controller should move and rotate with the body (hand-held controllers are often assumed to be held in the lap). For VR, information displays often work better in torso reference frames rather than head reference frames as commonly done with heads-up displays in traditional first-person video games. Figure 26.2 shows an example of a visual representation of a non-tracked hand-held controller and other information at waist level in the torso reference frame.



Figure 26.2 Information and a visual representation of a non-tracked hand-held controller in the torso reference frame. (Courtesy of NextGen Interactions)

Body-Relative Tools

Just like in the real world, tools in VR can be attached to the body so that they are always within reach no matter where the user goes. This is done in VR by simply placing the tool in the torso reference frame. This not only provides convenience of the tool always being available but also takes advantage of the user's body acting as a physical mnemonic, which helps in recall and acquisition of frequently used control [Miné et al. 1997].

Items should be placed outside the forward direction so that they do not get in the way of viewing the scene (e.g., the user simply looks down to see the options and then can select via a point or grab). Advanced users should be able to turn off items or make them invisible. Examples of physical mnemonics are pull-down menus located above the head (Section 28.4.1), tools surrounding the waste as a utility belt, audio options at the ear, navigation options at the user's feet, and deletion by throwing an object behind the shoulder (and/or object retrieval by reaching behind the shoulder).

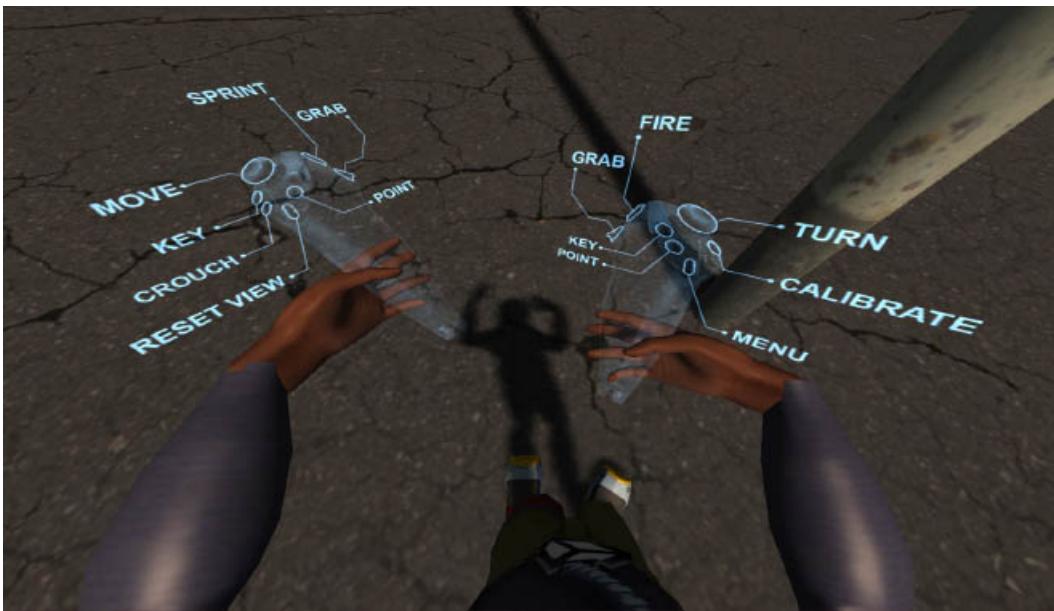


Figure 26.3 A simple transparent texture in the hand can convey the physical interface. (Courtesy of NextGen Interactions)

26.3.4 The Hand Reference Frames

The **hand reference frames** are defined by the position and orientation of the user’s hands, and hand-centric judgments occur when holding an object in the hand. Hand-centric thinking is especially important when using a phone, tablet, or VR controller. Placing a visual representation of a tracked hand-held controller (Section 27.2.3) in the hand(s) can help add a sense of presence due to the sense of touch matching the visuals. Placing labels or icons/signifiers in the hand reference frame that point to buttons, analog sticks, or fingers is extremely helpful (Figure 26.3), especially for new users. The option to turn on/off such visuals should be provided so as to not occlude/clutter the scene when not using the interface or after the interface has been memorized. Although both the left and right hands can be thought of as separate reference frames, the non-dominant hand is useful for serving as a reference frame for the dominant hand to work in (Section 25.5.1), especially for hand-held panels (Section 28.4.1).

26.3.5 The Head Reference Frame

The **head reference** frame is based on the point between the two eyes and a reference direction perpendicular to the forehead. In psychological literature, this reference frame is known as the **cyclopean eye**, which is a hypothetical position in the head that serves as our reference point for the determination of a head-centric straight-ahead [Coren et al. 1999]. People generally tend to think of this straight-ahead as a direction in front of themselves, oriented around the midline of the head, regardless of where the eyes are actually looking. From an implementation point of view, the head reference frame is equivalent to the head-mounted-display reference frame, but from the user’s point of view (assuming a wide field of view), the display is not visually perceived. A world-fixed secondary display that shows what the user is seeing matches the head reference frame.

Heads-up displays (HUDs) are often located in the head reference frame. Such heads-up display information should be minimized, if used at all, other than a selection pointer for gaze-directed selection (Section 28.1.2). If used, it is important to make cues small (but large enough to be easily perceived/readable), minimize the number of visual cues to not be annoying or distracting, not make the cues too far in the periphery, give the cues depth so they are occluded properly by other objects (Section 13.2), and place the cues at a far enough distance so that there is not an extreme accommodation-vergence conflict (Section 13.1). It can also be useful to make the cues transparent. Figure 26.4 shows an example HUD in the head reference frame that serves as a virtual helmet that helps the user to target objects.

26.3.6 The Eye Reference Frames

The **eye reference frames** are defined by the position and orientation of the eyeballs. Very few VR systems support the orientation portion of eye reference frames due to the requirement for eye tracking (Section 27.3.2). However, the left/right horizontal offset of the eyes is easily determined by the interpupillary distance of the specific user measured in advance of usage.

When looking at close objects (for example, when sighting down the barrel of a gun) and assuming a binocular display, eye reference frames are important to consider because they are differentiated from the head reference frame due to the offset from the forehead to the eye. The offset between the left and right eyes results in double images for close objects when looking at an object further in the distance (or double images of the further object when looking at the close object). Thus, when users are attempting to line up close objects with further objects (e.g., a targeting task), users should be advised to close the non-dominant eye and to sight with the dominant eye (Sections 9.1.1 and 23.2.2).



Figure 26.4 A heads-up display in the head reference frame. No matter where the user looks with the head, the cues are always visible. (Courtesy of NextGen Interactions)

26.4 Speech and Gestures

The usability of speech and gestures depends upon the number and complexity of the commands. More commands require more learning—the number of voice commands and gestures should be limited to keep interaction simple and learnable.

Voice interfaces and gesture recognition systems are normally invisible to the user. Use explicit signifiers, such as a list of possible commands or icons of gestures, in the users' view so they know and remember what is possible. Neither speech nor gesture recognition is perfect. In many cases it is appropriate to have users verify commands to confirm the system understands correctly before taking action. Feedback should also be provided to let the user know a command has been understood (e.g., highlight the signifier if the corresponding command has been activated).

Use a set of well-defined, natural, easy-to-understand, and easy-to-recognize gestures/words. Pushing a button to signal to the computer that a word or gesture is intended to start (i.e., push-to-talk or push-to-gesture) can keep the system from recognizing unintended commands. This is especially true when the user is also communicating with other humans, rather than just the system itself (for both voice and gestures as humans subconsciously make gestures as they talk).

26.4.1 Gestures

A **gesture** is a movement of the body or body part whereas a **posture** is a single static configuration. Each conveys some meaning whether intentional or not. Postures can be considered a subset of gestures (i.e., a gesture over a very short period of time or a gesture with imperceptible movement). Dynamic gestures consist of one or more tracked points (consider making a gesture with a controller) whereas a posture requires multiple tracked points (e.g., a hand posture).

Gestures can communicate four types of information [[Hummels and Stappers 1998](#)].

Spatial information is the spatial relationship that a gesture refers to. Such gestures can manipulate (e.g., push/pull), indicate (e.g., point or draw a path), describe form (e.g., convey size), describe functionality (e.g., twisting motion to describe twisting a screw), or use objects. Such direct interaction is a form of structural communication (Section [1.2.1](#)) and can be quite effective for VR interaction due to its direct and immediate effect on objects.

Symbolic information is the sign that a gesture refers to. Such gestures can be concepts like forming a V shape with the fingers, waving to say hello or goodbye, and explicit rudeness with a finger. The formation of such gestures is structural communication (Section [1.2.1](#)) whereas the interpretation of the gesture is indirect communication (Section [1.2.2](#)). Symbolic information can be useful for both human-computer interaction and human-human interaction.

Pathic information is the process of thinking and doing that a gesture is used with (e.g., subconsciously talking with one's hands). Pathic

information is most commonly visceral communication (Section 1.2.1) added on to indirect communication (Section 1.2.1) that is useful for human-human interaction.

Affective information is the emotion a gesture refers to. Such gestures are more typically body gestures that convey mood such as distressed, relaxed, or enthusiastic. Affective information is a form of visceral communication (Section 1.2.1) most often used for human-human interaction, although pathic information is less commonly recognized with computer vision as discussed in Section 35.3.3.

In the real world, hand gestures often augment communication with gestures such as okay, stop, size, silence, kill, goodbye, pointing, etc. Many early VR systems used gloves as input and gestures to indicate similar commands. Advantages of gestures include flexibility, the number of degrees of freedom of the human hand, the lack of having to hold a device in the hand, and not necessarily having to see (or at least look directly at) the hand. Gestures, like voice, can also be challenging due to having to remember them and most current systems have low recognition rates for more than a few gestures. Although gloves are not as comfortable, they are more consistent than camera-based systems due to not having line-of-sight issues. Push-to-gesture systems can drastically reduce false positives. This is especially true when the user is communicating with other humans, rather than just the system itself.

Direct vs. Indirect Gestures

Direct gestures are immediate and structural (Section 1.2.1) in nature and convey spatial information; they can be interpreted and responded to by the system as soon as the gesture starts. Direct manipulation, such as pushing an object, and selection via hand pointing are examples of direct gestures. **Indirect gestures** indicate more complex semantic meaning over a period of time so the start of the gesture is not sufficient—the application interprets over a range of movement so there is a delay from the start of the gesture. Indirect gestures convey symbolic, pathic, and affective information. A single posture command is somewhere between direct and indirect because the system response is immediate but not structural (the posture is interpreted as a command).

26.4.2 Speech Recognition

Speech recognition translates spoken words into textual and semantic form. If implemented well, voice commands have many advantages including keeping the head and hands free to interact while giving commands to the system. Voice recognition does have significant challenges including limited recognition capability, not always obvious command options, difficulty in selecting from a continuous scale, background noise, variability between speakers, and distraction to other individuals [McMahan et al. 2014]. Regardless, speech can work well for multimodal interactions (Section 26.6). Speech recognition categories, strategies, and errors are discussed below as described by Hannema [2001].

Speech Recognition Categories

Speech recognition is often categorized into the following groups.

Speaker-independent speech recognition has the flexibility to recognize a small number of words from a wide range of users. This type of speech recognition is used with telephone navigation systems and is best used with VR when there are only a small number of options provided to the user (a VR system should visually show available commands so the user knows what the options are).

Speaker-dependent speech recognition recognizes a large number of words from a single user where the system has been extensively trained to recognize words from that specific user. This type of speech recognition can work well with VR when the user has a personal system that she uses often.

Adaptive recognition is a mix of speaker-independent and speaker-dependent speech recognition. The system does not need explicit training but learns the characteristics of the specific user as he speaks. This often requires that the user corrects the system when words are misinterpreted. Use adaptive recognition when users have their own system but don't want to bother with explicitly training the voice recognizer.

Speech Recognition Strategies

Each of the speech recognition categories listed above can use one or more of the following strategies to recognize words.

Discrete/isolated strategies recognize one word at a time from a predefined vocabulary. This strategy works well when only one word is used or there is a silence between consecutive words. Examples include commands such as "save," "undo," "restart," or "freeze."

Continuous/connected strategies recognize consecutive words from a predefined vocabulary. This is more challenging to implement than a discrete/isolated strategy.

Phonetic strategies recognize individual phonemes (small, perceptually distinct sounds; Section 8.2.3), diphones (combinations of two adjacent phonemes), or triphones (combinations of three adjacent phonemes). Triphones are computationally expensive, and the system may be slow to respond due to the number of combinations that must be recognized so are rarely used.

Spontaneous/conversational strategies attempt to determine the context of the words in sentences in a way similar to what humans do. This results in a natural spoken dialogue with the computer. This strategy can be difficult to implement well.

Speech Recognition Errors

There are several reasons why speech recognition is difficult. By being aware of the common types of errors listed below, the system can be better designed to minimize such errors. Errors can also be reduced by using a microphone designed for speech recognition (Section [27.3.3](#)).

Deletion/rejection occurs when the system fails to match or recognize a word from the predetermined vocabulary. The advantage of this type of error is that the system recognizes the failure and can request the user to repeat the word.

Substitution occurs when the system misrecognizes a word for a different word than that which was intended. If the error is not caught, the system might execute a wrong command. This error is difficult to detect, but statistical measures can be used to calculate confidence.

Insertion occurs when an unintended word is recognized. This most often occurs when the user is not intentionally speaking to the system, such as when thinking out loud or when speaking to another human. Similar to a substitution error, this can execute an unintended command. Requiring the user to push a button (e.g., a push-to-talk interface) can drastically reduce this type of error.

Context

The specific context that the user is engaged in at any particular time can help improve accuracy and better match the user's intention. This is important as words can be homonyms (the same word with multiple meanings, e.g., volume can be a volume of space or audio volume) or homophones (different words with the same sound, e.g., die and dye). Context-sensitive systems with a large vocabulary can be implemented by having the system only able to recognize a subset of that vocabulary at any particular time.

26.5 Modes and Flow

Although ideally the same metaphors should be applied across all interactions for a single application, this is often not possible. Complex applications with different types of tasks may require different interaction techniques. In such a case, different techniques might be combined together. The mechanism to choose a technique may be as simple as pressing a different button or a mode selection from a hand-held panel, or the technique may be order dependent (e.g., a specific manipulation technique only occurs after a specific selection technique). Whatever the mode, that mode should be made clear to the user.

All interactions should also integrate and flow together well. The overall usability of a system depends on the seamless integration of various tasks and techniques provided by the application. One way to think about flow is to consider the sequence of basic actions. People may more often verbally state commands with the action coming before the object to be acted upon, but they tend to think about the object first. Objects are more concrete in nature so they are easier to first think about, whereas verbs are more abstract and are more easily thought about when being applied to something. For example, someone might think "pick up the book," but before thinking about picking up the book the person must first perceive and think about the book. Users prefer object-action sequences over action-object sequences as it requires less mental effort [[McMahan and Bowman 2007](#)]. Thus, when designing interaction techniques, the selection of the object to be acted upon should be performed (at least in most cases) before taking action upon that object. The interaction technique should also enable easy and smooth transition between selecting an object and manipulating or using that object.

At a higher level, the flow of longer interactions should occur without distractions so the user can give full attention to the primary task. Ideally, users should not have to physically (whether the eyes, head, or hands) or cognitively move between tasks. Lightweight mode switching, physical props, and multimodal techniques can help to maintain the flow of interaction.

26.6 Multimodal Interaction

No single sensory input or output is appropriate for all situations. **Multimodal interactions** combine multiple input and output sensory modalities to provide the user with a richer set of interactions. The "put-that-there" interface is known as the first human-computer interface to effectively and naturally mix voice and gesture [[Bolt 1980](#)]. Note although "put-that-there" is an action-object sequence, as discussed above in Section [26.5](#), better flow often occurs by first selecting the object to be moved. The better implementation might be called a "that-moves-there" interface.

When choosing or designing multimodal interactions, it can be helpful to consider different ways of integrating the modalities together. Input can be categorized into six types of combinations: specialized, equivalence, redundancy, concurrency, complementarity, and transfer [[Laviola 1999](#), [Martin 1998](#)]. All of the input modality types are multimodal other than specialized.

Specialized input limits input options to a single modality for a specific application. Specialization is ideal when there is clearly a single best modality for the task. For example, for some environments, selecting an object might only be performed by pointing.

Equivalent input modalities provide the user a choice of which input to use, even though the result would be the same across modalities. Equivalence can be thought of as the system being indifferent to user preferences. For example, a user might be able to create the same objects either by voice or through a panel.

Redundant input modalities take advantage of two or more simultaneous types of input that convey the same information to perform a single command. Redundancy can reduce noise and ambiguous signals, resulting in increased recognition rates. For example, a user might select a red cube with the hand while saying "select the red cube" or physically move an object with the hand while saying "move."

Concurrent input modalities enable users to issue different commands simultaneously, and thus enable users to be more efficient. For example, a user might be pointing to fly while verbally requesting information about an object in the distance.

Complementarity input modalities merge different types of input together into a single command. Complementarity often results in faster interactions as the different modalities are typically close in time or even concurrent. For example, to delete an object, the application might require the user to move the object behind the shoulder while saying “delete.” Another example is a “put-that-there” interface [Bolt 1980] that merges voice and gesture to place an object.

Transfer occurs when information from one input modality is transferred to another input modality. Transfer can improve recognition and enable faster interactions. A user may achieve part of a task by one modality but then determine a different modality would be more appropriate to complete the task. In such a case, transfer would prevent the user from needing to start over. An example is verbally requesting a specific menu to appear, which can be interacted with by speaking or pointing. Another example is a “push-to-talk” interface. Transfer is most appropriate to use when hardware is unreliable or does not work well in some situations.

26.7 Beware of Sickness and Fatigue

Some interaction techniques, especially those controlling the viewpoint, can cause motion sickness. When choosing or creating navigation techniques, designers should carefully understand and consider scene motion and motion sickness as discussed in Part III. If motion sickness is a primary concern, then changing the viewpoint should only occur through one-to-one mapping of real head motion or teleportation (Section 28.3.4).

Some users are not comfortable looking at interfaces close to the face for extended periods of time due to the accommodation-vergence conflict (Section 13.1) that occurs in most of today’s HMDs. Visual interfaces close to the face should be minimized.

As mentioned in Section 14.1, gorilla arm can be a problem for interactions that require the user to hold their hands up high and out in front of themselves for more than a few seconds at a time. This occurs even with bare-hand systems (Section 27.2.5) where the user is not carrying any additional weight. Interactions should be designed to minimize holding the hands above the waist for more than a few seconds at a time. For example, shooting a ray from the hand held at the hip is quite comfortable.

26.8 Visual-Physical Conflict and Sensory Substitution

Most VR experiences offer little haptic feedback, and when they do the feedback is quite limited compared to the sense of touch in the real world. Not having full haptic feedback is more of a problem than just not feeling objects. The hand or other body part (or physical device) continues to move through the object since there is no (or limited) physical force stopping it from doing so. As a result, the physical location of the hand may no longer match the visual location.

Enforcing simulated physics so the hand does not visually pass through visual geometry is often preferred by users when the penetrations are only slight (shallow penetration). When deeper penetration occurs, then users prefer the visual hands to match the physical hands even though that breaks the intuition that hands do not pass through objects [Lindeman 1999]. Stopping the visual hand for deep penetration can be especially confusing when the visual hand pops out of a different part of the penetrated object than where the visual hand has previously been visually stopped. A compromise solution for non-realistic interactions is to draw two hands when the physical hand and physically simulated hand diverge (see ghosting below).

In some cases, the virtual hand can be considerably offset from the physical hand without the user noticing as visual representation tends to dominate proprioception [Burns et al. 2006]. However, this is not always the case. Vision is generally stronger than proprioception when moving the hand in a left/right and/or up/down direction, but proprioception can be stronger when moving the hand in depth (forward/back) [Van Beers et al. 2002].

Sensory substitution is the replacement of an ideal sensory cue that is not available with one or more other sensory cues. Examples of sensory substitution that work well with VR are described below.



Figure 26.5 In the game “The Gallery: Six Elements,” the bottle is highlighted to show the object can be grabbed. (Courtesy of Cloudhead Games)

Ghosting is a second simultaneous rendering of an object in a different pose than the actual object. In some cases it is appropriate to render the hand twice—both where the physical hand is located and where the physics simulation states the hand is located. Ghosting is also often used to provide a clue of where a virtual object will be snapped into place if released. Be careful of using ghosting for training applications as users can depend on ghosting as a crutch that will not be available for the real-world task.

Highlighting is visually outlining or changing the color of an object. Highlighting is most often used to show the hand has intersected with an object so that it can be selected or picked up. Highlighting is also used to convey that an object is able to be selected or grabbed when the hand is close even though a collision has not yet occurred. Figure 26.5 shows an example of highlighting.

Audio cues are very effective in conveying to a user that one of his hands has collided with some geometry. Audio might be as simple as a tone sound or real-world recorded audio track. In some cases, providing multiple audio files with variations (e.g., random grunt sounds when colliding with a virtual wall or when shot by an enemy) can help with a sense of realism and reduce annoyance. Continuous contact sounds can also be used to convey sliding along surfaces. Sound properties such as pitch or amplitude might also change depending on penetration depth.

Passive haptics (static physical objects that can be touched; Section 3.2.3) are effective when the virtual world is limited to the physical space where no virtual navigation can occur (i.e., when the real-world reference frames and virtual-world reference frames are consistent; Section 26.3) or when tracked physical tools travel with the user (i.e., the physical and virtual objects are spatially compliant; Section 25.2.5). Because vision often dominates proprioception, perfect spatial compliance is not always required [Burns et al. 2006]. Redirected touching warps virtual space to map many differently shaped virtual objects onto a single real object (i.e., hand or finger tracking is not one-to-one) in a way that the discrepancy between virtual and physical is below the user’s perceptual threshold [Kohli 2013]. For example, when one’s real hand traces a physical object, the virtual hand can trace a slightly differently shaped virtual object.

Rumble causes an input device to vibrate. Although not the same haptic force that would occur in the real world, rumble feedback can be quite an effective cue for informing the user she has collided with an object.

Input Devices

Input devices are the physical tools/hardware used to convey information to the application and to interact with the virtual environment. Some interaction techniques work more or less across different input devices whereas other techniques work only with input devices that have specific characteristics. Thus, appropriately choosing input hardware that best fits the application’s interaction techniques is an important design decision (or, conversely, designing and implementing interaction techniques depends upon available input hardware).

This chapter describes some general characteristics of input devices and then describes the primary classes of input devices.

27.1 Input Device Characteristics

Input devices can be very different, and the characteristics of each should be considered when choosing hardware and designing interactions.

27.1.1 Size and Shape

The most obvious characteristics to a new VR user are the basic shape and size of the input device. The shape and size has more to do with just how the controller looks and feels in the hand. Large hand-held devices are primarily controlled by large muscle groups of the shoulder, elbow, and wrist, whereas smaller hand-held devices utilize smaller and faster muscle groups in the fingers [[Bowman et al. 2004](#)]. Smaller devices can also decrease **clutching**—the releasing and regrasping of an object in order to complete a task due to not being able to complete it in a single motion (such as a wrench). Gloves also use these smaller muscle groups and have the advantage of being able to freely touch and feel other items.

27.1.2 Degrees of Freedom

Input devices are often classified by the number of degrees of freedom (DoF) they report. **Degrees of freedom (DoF)** are the number of dimensions that an input device is capable of manipulating (also see Section [25.2.3](#)). Devices range from a single DoF (e.g., an analog trigger), to 6 DoF that measure full 3D translation (up/down, left/right, forward/backward) and rotation (roll, pitch, and yaw), to full hand or body tracking with many DoFs. A traditional mouse, joystick, trackball, and touchpad (a rotatable ball that is essentially an upside-down mouse) are examples of 2 DoF devices. VR hand tracking should have a minimum of 6 DoF (multiple points tracked on the hand have more than 6 DoF). For a majority of active VR experiences, one or more 6 DoF hand-held controllers is often the most appropriate choice. For some simple tasks only requiring navigation and no direct interaction, a non-tracked hand-held controller is good enough.

27.1.3 Relative vs. Absolute

Relative input devices measure the differences between the current and last measurement. Mice, trackballs, and inertial trackers are examples of relative devices. Relative devices drift over time and thus are not nulling compliant (Section [25.2.5](#)). Although limited, the Nintendo Wii proved relative devices can work well for natural interactions under some circumstances if the applications are carefully designed. VR relative devices typically use inertial measurement units (IMUs) that have the advantage of having a higher update rate (e.g., 1,000 Hz) and faster response (e.g., 1 ms) than absolute measurements. **Absolute input devices** sense pose relative to a constant point of reference independent of past measurements and are nulling compliant. **Hybrid tracking systems** fuse both relative and absolute trackers to provide the advantages of both. VR head and hand tracking should sense pose via absolute measurements (although relative devices can arguably estimate absolute pose of the hands from modeling the constraints imposed by the physicality of the arms and hand).

27.1.4 Separable vs. Integral

Integral input devices enable users to control all DoFs simultaneously from a single motion (a single composition) whereas **separable input devices** contain at least one DoF that cannot be controlled simultaneously from a single motion (two or more distinct compositions). A gamepad with two different analog sticks is an example of a separable device. A 2D device that enables control of more than two dimensions via mode switching is also an example of a separable device. VR hand tracking should be integral.

27.1.5 Isometric vs. Isotonic

Isometric input devices measure pressure or force that contains no or little actual movement. **Isotonic input devices** measure deflection from a center point and may or may not have some resistance. Mice are isotonic input devices. Joysticks can be either isotonic or isometric. Isotonic input devices are best for controlling position, whereas isometric input devices are best for controlling rates such as navigation speed. An isometric joystick, for example, works well for controlling velocity (i.e., hold to continue moving).

27.1.6 Buttons

Buttons control one DoF via pushing with a finger and typically take on one of two states (e.g., pushed or not pushed) although some buttons can take on analog values (also known as analog triggers). Buttons are often used to change modes, to select an object, or to start an action. Although buttons can be useful for VR applications, too many buttons can lead to confusion and error—especially when the button-mapping functionality is unclear or inconsistent (although attaching labels to virtual controllers can help—see Figure [26.3](#) as an example). Consider the

capabilities and intuitiveness of desktop applications that are controlled via no more than three buttons on a mouse.

There is a debate between bare-hand system (Section 27.2.5) and hand-held controller (Sections 27.2.2 and 27.2.3) advocates as to the utility of buttons. For example, Microsoft Kinect and Leap Motion developers believe buttons are a primitive and unnatural form of input whereas Playstation Move and Sixense Stem developers believe buttons are an essential part of game play. Like any great debate, the answer is “it depends” [Jerald et al. 2012]. Buttons can be abstracted to indirectly trigger nearly any action, but this abstraction can cause a disconnect between the user and the application. Buttons are most effective when an action is binary, when the action needs to occur often, when reliability is required, and when physical feedback to the user is essential. Buttons are also ideal for time-sensitive actions, since very little time is required for the physical action (i.e., an entire dynamic gesture does not need to be completed in order to register the action). Gestures can be slower and more fatiguing than button presses, particularly in command-intensive tasks such as modeling or radiology. Natural buttonless hand manipulation is most effective for providing a sense of realism and presence, when abstraction is not appropriate, or when detailed tracking of the entire hand is required.

27.1.7 Encumbrance

Unencumbered input devices do not require physical hardware to be held or worn. Such systems are implemented with camera systems. Thus no “suit-up” time is required (although calibration might be necessary). Unencumbered systems can also reduce hygiene issues (Section 14.4) since a physical device is not passed between users. Non-encumbrance is not always a design goal; holding something physically in the hand can add to the sense of presence (Section 3.2.3); consider holding a controller in the hand vs. not holding a controller in the hand for a shooting or golf experience.

27.1.8 Ability to Fully Interact with Physical Objects

Some devices enable one to touch the real world in natural ways without the device getting in the way. Bare-hand tracking systems (i.e., camera systems) and gloves are the most common examples of this. Held devices and world-grounded devices (Section 27.2.1) must be released before the hand can fully interact with other physical objects. In such cases, the physical devices should be tracked and rendered in the virtual world so the user can reach out to grab them again.

27.1.9 Device Reliability

Device **reliability** is the extent to which an input device can consistently work within the user’s entire personal space (and a larger volume if the user is expected to physically move around). Devices should ideally have 100% reliability anywhere the user can reach with no loss in tracking acquisition. Reliability should carefully be considered when choosing an input device as unreliable devices can result in frustration, fatigue (e.g., due to having to hold the hand high and in front of the body; Section 14.1), increased cognitive load (e.g., because the user must think about holding the device in a certain way), breaks-in-presence (Section 4.2), and reduced performance.

Unreliable tracking can be due to multiple reasons and divided into two sets of factors: (1) implementation limitations and (2) inherent physical limitations. An inherent physical limitation is the best a device will ever be able to achieve given an optimal engineering solution/implementation. Some devices cannot possibly provide 100% reliability no matter what the engineering effort. A system that requires line of sight from a sensor to the tracked device can be occluded by another physical object (such as a hand or torso), in which case there is no way for the system to reliably determine the pose of the device (although the state of the device can be estimated for short periods of time). VR devices should ideally work in all orientations and hand postures (e.g., hands covering sensors or tracking the fingers when a fist is made).

Another challenge of reliability occurs for users attempting to work in a tracked volume smaller than the user’s personal space. An example is a vision-based system with a limited field of view. Lighting can also be a challenge for some vision-based systems, especially in uncontrolled environments outside of the laboratory or some other highly controlled space. Other systems only recognize gestures when the hand is oriented or held in a certain way. Many camera-based hand tracking systems only reliably recognize poses and gestures when the hands are perpendicular to the camera with the fingers visible.

27.1.10 Haptics Capable

Active haptics can easily be added to physical devices that are worn or held. However, the degree of haptics might be limited depending on the size of the device and if attached to the world in some way (Section 3.2.3).

27.2 Classes of Hand Input Devices

The most important VR input devices are the human hands, and this section explores how different types of devices can integrate the hands into VR. An **input device class** is a set of input devices that share the same essential characteristics that are crucial to interaction. Input device classes described in this section focus on the hands and are classified as world-grounded input devices, non-tracked hand-held controllers, tracked hand-held controllers, hand-worn devices, and bare-hand controllers.

Table 27.1 summarizes some of the most essential characteristics of the hand input device classes described in this section and non-hand input device classes described in Section 27.3. As can be seen, no single input device class is universally advantageous although some classes can be combined to create a hybrid system with more advantages. For example, a bare-hands camera-based system might be used with tracked hand-held controllers (although simultaneous usage can be difficult due to different characteristics, so bare hands might be used until the controller is

picked up). Note the table is based upon inherent physical limitations rather than today's existing implementations. Technical specifications such as update rate, latency, etc. are not included here as such specs are independent of the hardware (e.g., a fast update rate could potentially be implemented on any device class).

27.2.1 World-Grounded Input Devices

World-grounded input devices, like world-grounded haptics (Section 3.2.3), are designed to be constrained or fixed in the real world and are most often used to interact with desktop systems.

Keyboards and mice are considered to be world-grounded devices and are the most popular form of input that works extremely well for its intended task—2D desktop manipulation. However, such input is not a good way of interacting for most all immersive applications (with the possible exception of video-see-through augmented reality where users can see the mouse and keyboard).

Trackballs and joysticks mounted to a permanent location are world-grounded devices. Other devices offer up to 6 DoF through pushing, pulling, twisting, and/or buttons for mode changes. However, these devices suffer from similar limitations for VR as the mouse due to not being designed to be held comfortably and freely in the hands. There are exceptions (e.g., to mount a joystick on the arm of a chair or if simulating a desktop environment where the physical controls precisely match a virtual desktop), but from a human-centered design perspective there should be a solid reason to choose such devices, instead of “because it is available” or “it is what someone else is using.”

Table 27.1 Comparison of hand and non-hand input device classes.

	Proprioception	Consistent	Usable in Lap or the Side	Haptics Capable	Unencumbered	Physical Buttons	Hands Free to Interact with Real World	General Purpose
Hand Input Device Class								
World-Grounded Devices	✓	✓		✓	✓	✓	✓	
Non-Tracked Hand-Held Controllers	✓	✓	✓			✓		
Bare Hands	✓				✓		✓	✓
Tracked Hand-Held Controllers	✓	✓	✓	✓		✓		✓
Hand Worn	✓	✓	✓	✓		✓	✓	✓
Non-Hand Input Device Class								
Head Tracking	✓	✓				✓	✓	
Eye Tracking							✓	
Microphone			✓		✓		✓	✓
Full-Body Tracking	✓	✓	✓	✓			✓	✓
Treadmills	✓	✓			✓		✓	

World-grounded devices that work extremely well for VR are specialty devices such as handlebars, steering wheels, gas and brake pedals, cockpits, and automotive interior controls. Such devices can be especially good for travel, as most users already have real-world experience with devices. Even if world-grounded devices aren't actually used in the real world, they can still be quite effective and presence inducing if designed well. For example, Disney's Aladdin Magic Carpet Ride, 3 DoF controls (Figure 27.1) provides an intuitive physical interface for travel. One reason such controls are so effective is due to having physical signifiers, affordances, and feedback, e.g., the user feels what can be done and how she is doing it. Some challenges of these devices are that creators can't assume there is a wide user base that owns such hardware and it can be difficult to generalize the devices to work with a wide range of tasks. Thus, such devices are more commonly used at location-based entertainment venues where large groups of people use the same device(s) and the device can be designed or modified for the particular VR experience.

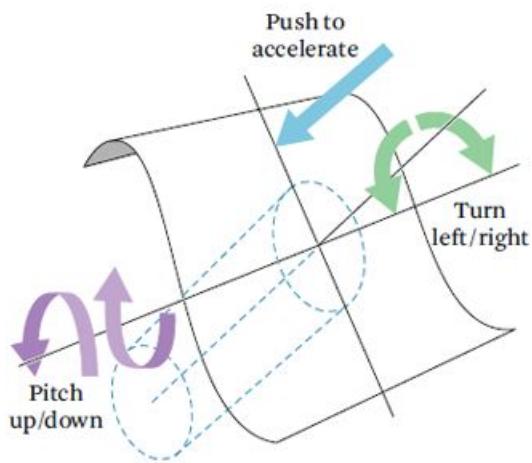


Figure 27.1 The Disney Aladdin world-grounded input device along with its mapping for viewpoint control. (From [Pausch et al. \[1996 \]](#))

27.2.2 Non-Tracked Hand-Held Controllers

Non-tracked hand-held controllers are devices held in the hand that include buttons, joysticks/analog sticks, triggers, etc. but are not tracked in 3D space. Traditional video game input devices such as joysticks and gamepads are the most common form of non-tracked hand-held controllers (Figure 27.2.). Many VR applications are starting to support such game controllers. These controllers work much better than the mouse and keyboard since the controller can be held comfortably in the lap where users can continually hold the controller. Many gamers have an intuitive feel of where the buttons are through years of use. Controllers with analog sticks work surprisingly well for navigating within VR (Section 28.3.2.).



Figure 27.2 The Xbox One controller is an example of a non-tracked hand-held controller.

Although not tracked, such controllers can increase presence for seated experiences by placing visual hands and controllers at the approximate location of the user's lap since most users hold the controller in such a position. The visual controller and hands have also been found to cause users to subconsciously move the hands and controller to the visual controller (Andrew Robinson and Sigurdur Gunnarsson, personal communication, May 11, 2015). Unfortunately, when a user moves his hands away from the assumed position, a break-in-presence typically occurs if the user sees the virtual hands stay in place.

27.2.3 Tracked Hand-Held Controllers

Tracked hand-held controllers are typically 6 DoF devices (known as “wands” in the VR research community where they have been used for decades) and can also contain functionality offered by non-tracked hand-held controllers. Tracked hand-held controllers are currently the best option for a majority of interactive VR applications. Tracked hand-held controllers are easy to use for many 3D tasks due to their natural, direct mapping to hand motion. Because the controllers are tracked, they can be visually co-located with the real hands (i.e., spatially and temporally compliant; Section 25.2.5) as well as physically felt, providing proprioceptive and passive haptics/ touch cues. Labels can also be attached to the virtual representation to provide immediate instruction of what the buttons do by simply looking at where the hands are physically located (Section 26.3.4 and Figure 26.3.), adding a big advantage over traditional desktop and gamepad input. Viewpoint manipulation with these devices is typically achieved using buttons and a trackball, integrated analog sticks (as on the Sixense STEM and Oculus Touch controllers as shown in Figure 27.3.), or by flying with the hands. Such techniques are described in Section 28.3. Other types of physical controls and feedback can also be added to these devices, such as trackpads and active haptics (e.g., vibration).



Figure 27.3 The Sixense STEM (left) and Oculus Touch (right) tracked hand-held controllers. (Courtesy of Sixense (left) and Oculus (right))

Tracked hand-held controllers have the advantage of acting as a physical prop, which enhances presence through physical touch. Not only do such controls facilitate communication with the virtual world, but they also help to make spatial relationships seem more concrete to the user [[Hinckley et al. 1998](#)]. However, such props come at the cost of not being able to directly/fully touch and feel other passive objects in the world and world-grounded input devices, such as seats, handlebars, and cockpit controls (Section [27.2](#)), without first setting down the controller/prop.

Tracked hand-held devices typically use inertial, electromagnetic, ultrasonic, or optical (camera) technologies. Each of these technologies has advantages and disadvantages, and hand-held trackers ideally use some hybrid method of integrating multiple technologies together (sensor fusion) to provide both high precision and accuracy.

27.2.4 Hand-Worn Devices

Hand-worn input devices include gloves, muscle-tension sensors (electromyographic or EMG sensors), such as what Thalmic Labs recently made popular with the Myo (which is worn on the arm but measures hand motion), and rings.

Many believe gloves (Figure [27.4](#)) to be the ultimate VR interface as they theoretically have many advantages, such as not having line-of-sight, sensor field-of-view, or lighting requirements so the hands can be held comfortably to the side or in the lap with no concern of losing tracking, resulting in less gorilla arm if the interaction techniques are designed well (Section [18.9](#)). Like bare hands, gloves also have the advantage that the hands and fingers can still fully interact with other physical objects.

Unfortunately, like bare-hand systems, full-hand tracked gloves are lacking in their current form and will require dramatic improvement to be used by the masses. Consistent recognition of more than a few gestures is still challenging due to the lack of consistent finger tracking accuracy. Recognizing more than a few gestures requires the user to recalibrate often due to the glove moving on the hand. Gloves must also be put on and worn, which can become uncomfortable and result in sweaty hands. There is also a risk of social resistance to wearing gloves similar to the resistance of Google Glass—although those willing to wear an HMD on their face are unlikely to care what others think of wearing gloves. If such challenges can be solved, then gloves may eventually become the input device of choice for VR.

The Fakespace Pinch Gloves have functions more like buttons, with near 100% consistent recognition, more than typical gloves that do full hand and finger tracking. They work via a conductive cloth sewn into the tip of each finger. When two or more fingers touch then the circuit closes resulting in a signal. This simple design provides the capability for a large number of pinch gestures; combinations range from two to ten fingers touching each other plus poses involving separate but simultaneous pinches (e.g., left thumb to left index finger and right thumb to right index finger pinched at the same time). In practice, due to the usability of physical hand constraints and users' willingness to memorize gestures, applications will only use so many of these gestures, similar to how too many buttons on a hand controller would be too confusing. Pinch gloves can be used quite well with many of the example techniques described in Chapter [28](#).

Perhaps one of the most significant advantages of gloves is that both full hand tracking and button simulation via pinch gestures can be combined such as demonstrated by [LaViola and Zeleznik \[1999\]](#). Haptics can also be used with gloves, such as is done with the CyberGlove CyberTouch (as shown in Figure [27.4](#) but with buzzers added to provide a sense of haptics).



Figure 27.4 The CyberGlove is an example of a hand-worn device. (Courtesy of CyberGlove Systems LLC)

If EMG sensors and rings can be made more accurate, then they too might become an ideal fit for many applications.

27.2.5 Bare Hands

Bare-hand input devices work via sensors aimed at the hands (mounted in the world or on the HMD). Figure 27.5 shows the hands and a skeletal model fit to the hands, as seen by the user. The obvious major advantage is that the user’s hands are completely unencumbered. Many believe bare-hand systems will ultimately be the ideal VR interface. Although the bare hands work extremely well in the real world, it turns out consistent interaction with the bare hands in VR is an enormous challenge. Challenges include not having a sense of touch, fatigue from holding the hands in front of the sensor, line-of-sight requirements, and consistent recognition of gestures across a wide range of users. Such technical challenges lead to usability challenges, such as being able to comfortably work with the hands in the lap without concern for where the sensor(s) is located. The bare hands also lack physical buttons, which is important for some applications but not important for other applications (Section 27.1.6.).

Regardless of the challenges of effectively working with the bare hands in VR, seeing the entirety of one’s hands in 3D is extremely compelling and provides a nice sense of presence for the periods of time when the tracking does consistently work. It remains to be seen if such challenges will be overcome and/or accepted by a wide range of users.

27.3 Classes of Non-hand Input Devices

VR input can occur through more than just the hands. This section describes head tracking, eye tracking, microphones, and full-body tracking.

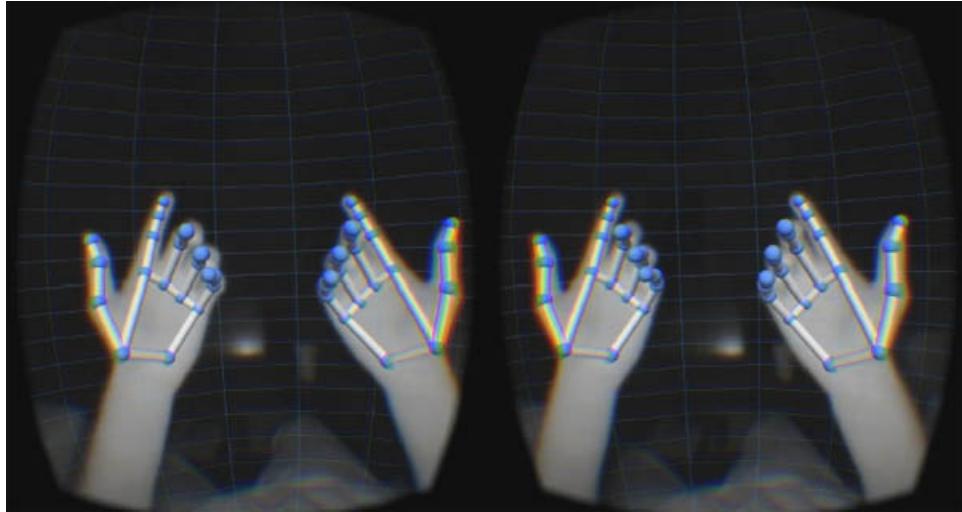


Figure 27.5 A depth-sensing camera on the HMD looking out enables one to see her own hands in detail. Here a skeleton model is also fit to the hands. (Courtesy of Leap Motion)

27.3.1 Head Tracking Input

Head tracking must be accurate, precise, fast, and well calibrated for the virtual world to appear stable. World stability is essential for VR and assumed to work well, but is not the focus of this chapter. Here, **head-tracking input** refers to interaction that modifies or provides feedback beyond just seeing the virtual environment. The most common form of head-tracking interaction is to aim by looking. One way of doing this is to provide a reticle or pointer in the middle of the screen (Section 26.3.5.) that is triggered by a button press, firing in the direction of the reticle, or selecting an option the reticle projects onto. Other more subtle interactions can be used, such as having an action occur in the direction the user is looking, having characters respond when looked at, or simple head gestures such as nodding the head yes or no.

27.3.2 Eye Tracking Input

An **eye-tracking input device** tracks where the eyes are looking. Eye-tracking input for VR is a largely unexplored topic other than the obvious of selecting (Section 28.1.2.) or firing where one is looking as mostly demonstrated with interactive eye-tracking systems that are integrated within some of today’s HMDs.

The **Midas Touch problem** refers to the fact that people expect to look at things without that look “meaning” something [Jacob 1991]. Interacting via eye tracking alone is usually not a good idea—eye tracking works better with multimodal input. For example, signaling via a clutch (e.g., push a button, blink, or say “select”) typically works better than signaling via a dwell time. Even with a clutch, designing interactions that work well with gaze is a challenge. Straightforward feedback in the form of a pointer/reticle that moves with the eyes can be annoying to users and occludes viewing in the part of vision with the highest visual acuity. Eye saccades can also make the pointer jitter or jump in unintended ways. These problems can be mitigated by only turning on the pointer when a button is held down and filtering out high-frequency motions.

Eye tracking can be more effective for specialized tasks and for subtle interactions, such as how a character responds when looked at. The following guidelines are useful to consider when designing eye-tracking interactions [Kumar 2007].

Maintain the natural function of the eyes. Our eyes are meant for looking, and interaction designers should maintain the natural function of the eye. Using the eyes for other purposes overloads the visual channel.

Augment rather than replace. Attempting to replace existing interfaces with eye tracking is typically not appropriate. Instead, think about how to

add functionality onto already existing and newly created interfaces. Gaze can provide context and inform the system that the user is paying attention to a specific object or area in the scene.

Focus on interaction design. Focus on the overall experience instead of eye tracking alone. Consider the number of steps in the interaction, the amount of time it takes, the cost of an error/failure, cognitive load, and fatigue.

Improve the interpretation of eye movements. Gaze data is noisy. Consider how to best filter eye movements, classify gaze data, recognize gaze patterns, and take other input modalities into account.

Choose appropriate tasks. Don't try to force gaze to solve every problem. Eye tracking is not appropriate for all tasks. Consider the task and scenario before choosing gaze.

Use passive gaze over active gaze. Consider ways in which gaze can be used more passively so the eyes can better maintain their natural function.

Leverage gaze information for other interactions. Leverage the system's knowledge of where the user is paying attention in order to provide context for non-gaze interactions.

Although typically not interactive, eye tracking will have significant benefit for VR usability by providing clues to content creators of what most engages a user's attention (see attention maps in Section [10.3.2](#)), similar to the use of eye tracking to inform and enhance website design.

27.3.3 Microphones

A **microphone** is an acoustic sensor that transforms physical sound into an electric signal. Use a headset microphone specifically designed for speech recognition that includes noise-canceling features that remove/reduce ambient noise. The microphone should be able to be easily adjusted/positioned while the HMD is on and while not being able to see it. Accuracy improves as the microphone comes closer to and in front of the mouth. Microphones should be comfortable and light.

To prevent speech recognition errors (Section [26.4.2](#)), such as sounds picked up when thinking aloud or another person's voice, use a push-to-talk interface. If hand-held controllers are used, then the push-to-talk button should be located on the controller.

27.3.4 Full-Body Tracking

Full-body tracking consists of tracking more than just the head and hands. Full-body tracking can significantly add to the illusion of self-embodiment as well as the illusion of social presence (Section [4.3](#)). Tracking a large number of features can also be used to enhance interaction (e.g., a game that allows kicking a ball).

VR full-body tracking is typically done with a motion capture suit, similar to what is used in the film industry. Different suits utilize different technologies such as electromagnetic sensors, retro-reflective markers, and inertial sensors. Depth cameras such as Microsoft Kinect can theoretically track the entire body, but it can be difficult to capture the entire body unless multiple cameras are used. Regardless, full-body camera capture systems, like Microsoft Kinect, can be used for creating extremely compelling experiences, even though resolution is not yet great and much of the body might go in and out of view. Figure [27.6](#) shows real-time capture of the real world and display of the resulting point-cloud data within an HMD.

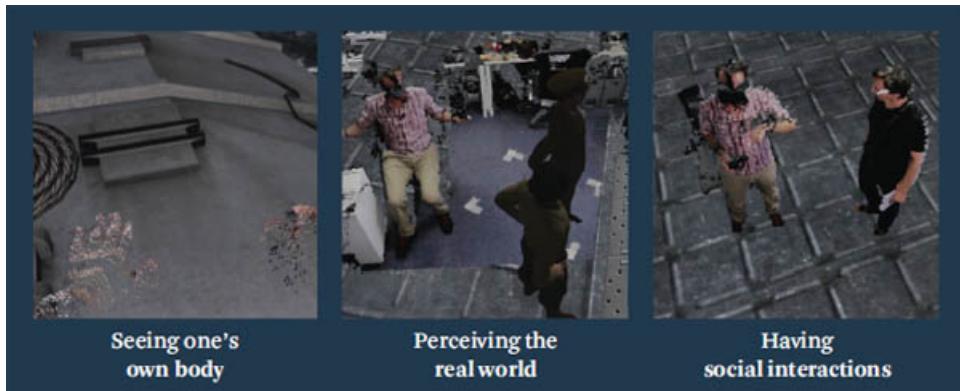


Figure 27.6 Depth cameras enable users to see their own body, the real world, and/or other people from the real world. (Courtesy of Dassault Systèmes, iV Lab)

Interaction Patterns and Techniques

An **interaction pattern** is a generalized high-level interaction concept that can be used over and over again across different applications to achieve common user goals. The interaction patterns here are intended to describe common approaches to general VR interaction concepts at a high level. Note interaction patterns are different than software design patterns (Section [32.2.5](#)) that many system architects are familiar with. Interaction patterns are described from the user's point of view, are largely implementation independent, and state relationships/interactions between the user and the virtual world along with its perceived objects.

An **interaction technique** is more specific and more technology dependent than an interaction pattern. Different interaction techniques that are similar are grouped under the same interaction pattern. For example, the walking pattern (Section [28.3.1](#)) covers several walking interaction techniques ranging from real walking to walking in place. The best interaction techniques consist of high-quality affordances, signifiers, feedback, and mappings (Section [25.2](#)) that result in an immediate and useful mental model for users.

Distinguishing between interaction patterns and interaction techniques is important for multiple reasons [[Sedig and Parsons 2013](#)].

- There are too many existing interaction techniques with many names and characterizations to remember, and many more will be developed in the future.
- Organizing interaction techniques under the umbrella of a broader interaction pattern makes it easier to consider appropriate design possibilities by focusing on conceptual utility and higher-level design decisions before worrying about more specific details.
- Broader pattern names and concepts make it easier to communicate interaction concepts.
- Higher-level groupings enable easier systematic analysis and comparison.
- When a specific technique fails, then other techniques within the same pattern can be more easily thought about and explored, resulting in better understanding of why that specific interaction technique did not work as intended.

Both interaction patterns and interaction techniques provide conceptual models to experiment with, suggestions and warnings of use, and starting points for innovative new designs. Interaction designers should know and understand these patterns and techniques well so they have a library of options to choose from depending on their needs and a base to innovate upon. Do not fall into the trap that there is a single best interaction pattern or technique. Each pattern and technique has strengths and weaknesses depending on application goals and the type of user [[Wingrave et al. 2005](#)]. Understanding distinctions and managing trade-offs of different techniques is essential to creating high-quality interactive experiences.

This chapter divides VR interaction patterns into selection, manipulation, viewpoint control, indirect control, and compound patterns. The interaction patterns and their interaction techniques as organized in this chapter are shown in Table [28.1](#). The first four patterns are often used sequentially (e.g., a user may travel toward a table, select a tool on the table, and then use that tool to manipulate other objects on the table) or can be integrated together into compound patterns. Interaction techniques that researchers and practitioners have found to be useful are then described within the context of the broader pattern. These techniques are only examples, as many other techniques exist, and such a list could never be exhaustive. The intent for describing these patterns and techniques is for readers to directly use them, to extend them, and to serve as inspiration for creating entirely new ways of interacting within VR.

Table 28.1 Interaction patterns as organized in this chapter. More specific example interaction techniques are described for each interaction pattern.

- Selection Patterns (Section [28.1](#))
 - Hand Selection Pattern
 - Pointing Pattern
 - Image-Plane Selection Pattern
 - Volume-Based Selection Pattern
- Manipulation Patterns (Section [28.2](#))
 - Direct Hand Manipulation Pattern
 - Proxy Pattern
 - 3D Tool Pattern
- Viewpoint Control Patterns (Section [28.3](#))
 - Walking Pattern

- Steering Pattern
- 3D Multi-Touch Pattern
- Automated Pattern
- Indirect Control Patterns (Section [28.4](#))
- Widgets and Panels Pattern
- Non-Spatial Control Pattern
- Compound Patterns (Section [28.5](#))
- Pointing Hand Pattern
- World-in-Miniature Pattern
- Multimodal Pattern

28.1 Selection Patterns

Selection is the specification of one or more objects from a set in order to specify an object to which a command will be applied, to denote the beginning of a manipulation task, or to specify a target to travel toward [[McMahan et al. 2014](#)]. Selection of objects is not necessarily obvious in VR, especially when most objects are located at a distance from the user. **Selection patterns** include the Hand Selection Pattern, Pointing Pattern, Image-Plane Selection Pattern, and Volume-Based Selection Pattern. Each has advantages over the others depending on the application and task.

28.1.1 Hand Selection Pattern

Related Patterns

Direct Hand Manipulation Pattern (Section [28.2.1](#)) and 3D Tool Pattern (Section [28.2.3](#)).

Description

The **Hand Selection Pattern** is a direct object-touching pattern that mimics real-world interaction—the user directly reaches out the hand to touch some object and then triggers a grab (e.g., pushing a button on a controller, making a fist, or uttering a voice command).

When to Use

Hand selection is ideal for realistic interactions.

Limitations

Physiology limits a fully realistic implementation of hand selection (the arm can only be stretched so far and the wrist rotated so far) to those objects within reach (personal space), requiring the user to first travel to place himself close to the object to be selected. Different user heights and arm lengths can make it uncomfortable for some people to select objects that are at the edge of personal space. Virtual hands and arms often occlude objects of interest and can be too large to select small items. Non-realistic hand selection techniques are not as limiting.



Figure 28.1 A realistic hand with arm (left), semi-realistic hands with no arms (center), and abstract hands (right). (Courtesy of Cloudhead Games (left), NextGen Interactions (center), and Digital ArtForms (right))

Exemplar Interaction Techniques

Realistic hands. **Realistic hands** are extremely compelling for providing an illusion of self-embodiment (Section [4.3](#)). Although ideally the entire arm would be tracked, inverse kinematics can estimate the pose of the arm quite well where users typically don't notice differences in arm pose if the head and/or torso is tracked along with the hand. Figure [28.1](#) (left) shows an example of a view from a user who has grabbed a bottle.

Modeling users (e.g., measuring arm length) and placing objects within a comfortable range depending on the measured arm length is ideal. However, Digital ArtForms found no complaints from ages 10 to adult in a semi-immersive world after setting a single arm-length scale value that was reasonable for the entire range of body sizes.

Non-realistic hands. Hands do not need to necessarily look real, and trying to make hands and arms realistic can limit interaction. **Non-realistic hands** does not try to mimic reality but instead focuses on ease of interaction. Often hands are used without arms (Figure 28.1, center) so that reach can be scaled to make the design of interactions easier. Although the lack of arms can be disturbing at first, users quickly learn to accept having no arms. The hands also need not look like hands. For abstract applications, users are quite accepting of abstract 3D cursors (Figure 28.1, right) and still feel like they are directly selecting objects. Such hand cursors reduce problems of visual occlusion. Hand occlusion can also be mitigated by making the hand transparent (although proper transparency rendering of multiple objects can be technically challenging to do correctly).

Go-go technique. The **go-go technique** [[Poupyrev et al. 1996](#)] expands upon the concept of a non-realistic hand by enabling one to reach far beyond personal space. The virtual hand is directly mapped to the physical hand when within 2/3 of the full arm's reach and when extended further, the hand "grows" in a nonlinear manner enabling the user to reach further into the environment. This technique enables closer objects to be selected (and manipulated) with greater accuracy while allowing further objects to be easily reached. Physical aspects of arm length and height have been found to be important for the go-go technique, so measuring arm length should be considered when using this technique [[Bowman 1999](#)]. Measuring arm length can be done by simply asking the user to hold out the hands in front of the body at the start of the application. [Bowman and Hodges \[1997 \]](#) describes extensions to the go-go technique, such as providing rate control (i.e., velocity) options that enable infinite reach, and compares these to pointing techniques. Non-isomorphic hand rotations (Section 28.2.1) are similar but scale rotations instead of position for manipulation.

28.1.2 Pointing Pattern

Related Patterns

Widgets and Panels Pattern (Section 28.4.1) and Pointing Hand Pattern (Section 28.5.1).

Description

The Pointing Pattern is one of the most fundamental and often-used patterns for selection. The **Pointing Pattern** extends a ray into the distance and the first object intersected can then be selected via a user-controlled trigger. Pointing is most typically done with the head (e.g., a crosshair in the center of the field of view) or a hand/finger.

When to Use

The Pointing Pattern is typically better for selection than the Hand Selection Pattern unless realistic interaction is required. This is especially true for selection beyond personal space and when small hand motions are desired. Pointing is faster when speed of remote selection is important [[Bowman 1999](#)], but is also often used for precisely selecting close objects, such as pointing with the dominant hand to select components on a panel (Section 28.4.1) held in the non-dominant hand.

Limitations

Selection by pointing is usually not appropriate when realistic interaction is required (the exception might be if a laser pointer or remote control is being modeled).

Straightforward implementations result in difficulty selecting small objects in the distance. Pointing with the hand can be imprecise due to natural hand tremor [[Bowman 1999](#)]. Object snapping and precision mode pointing described below can mitigate this problem.

Exemplar Interaction Techniques

Hand pointing. When hand tracking is available, **hand pointing** with a ray extending from the hand or finger is the most common method of selection. The user then provides a signal to actually select the item of interest (e.g., a button press or gesture with the other hand).

Head pointing. When no hand tracking is available, selection via head pointing is the most common form of selection. **Head pointing** is typically implemented by drawing a small pointer or reticle at the center of the field of view so the user simply lines up the pointer with the object of interest and then provides a signal to select the object. The signal is most commonly a button press but when buttons are not available then the item is often selected by **dwell selection**, that is, by holding the pointer on the object for some defined period of time. Dwell selection is not ideal due to having to wait for objects to be selected and accidental selection when looking at an object of interest.

Eye gaze selection. **Eye gaze selection** is a form of pointing implemented with eye tracking (Section 27.3.2). The user simply looks at an item of interest and then provides a signal to select the looked-at object. In general, eye gaze selection is typically not a good selection technique, primarily due to the Midas Touch problem discussed in Section 27.3.2.

Object snapping. **Object snapping** [[Haan et al. 2005](#)] works by objects having scoring functions that cause the selection ray to snap/bend toward the object with the highest score. This technique works well when selectable objects are small and/or moving.

Precision mode pointing. Precision mode pointing [[Kopper et al. 2010](#)] is a non-isomorphic rotation technique that scales down the rotational mapping of the hand to the pointer, as defined by the **control/display (C/D) ratio**. The result is a “slow motion cursor” that enables fine pointer control. A zoom lens can also be used that scales the area around the cursor to enable seeing smaller objects (but the zoom should not be affected by head pose unless the zoom area on the display is small; Section [23.2.5](#)). The user can control the amount of zoom with a scroll wheel on a hand-held device.

Two-handed pointing. Two-handed pointing originates the selection at the near hand and extends the ray through the far hand [[Miné et al. 1997](#)]. This provides more precision when the hands are further apart and fast rotations about a full 360° range when the hands are closer together (whereas a full range of 360° for a single hand pointer is difficult due to physical hand constraints). The distance between the hands can also be used to control the length of the pointer.

28.1.3 Image-Plane Selection Pattern

Also Known As

Occlusion and Framing.

Related Pattern

World-in-Miniature Pattern (Section [28.5.2](#)).

Description

The **Image-Plane Selection Pattern** uses a combination of eye position and hand position for selection [[Pierce et al. 1997](#)]. This pattern can be thought about as the scene and hand being projected onto a 2D image plane in front of the user (or on the eye). The user simply holds one or two hands between the eye and the desired object and then provides a signal to select the object when the object lines up with the hand and eye.

When to Use

Image-plane techniques simulate direct touch at a distance, thus are easy to use [[Bowman et al. 2004](#)]. These techniques work well at any distance as long as the object can be seen.

Limitations

Image-plane selection works for a single eye, so users should close one eye while using these techniques (or use a monoscopic display). Image-plane selection results in fatigue when used often due to having to hold the hand up high in front of the eye. As in the Hand Selection Pattern, the hand often occludes objects if not transparent.

Exemplar Interaction Techniques

Head crusher technique. In the **head crusher technique** (Figure [28.2](#)), the user positions his thumb and forefinger around the desired object in the 2D image plane.

Sticky finger technique. The **sticky finger technique** provides an easier gesture—the object underneath the user’s finger in the 2D image is selected.



Figure 28.2 The head crusher selection technique. The inset shows the user’s view of selecting the chair. (From [Pierce et al. \[1997\]](#))

Lifting palm technique. In the **lifting palm technique**, the user selects objects by flattening his outstretched hand and positions the palm so that it appears to lie below the desired object.

Framing hands technique. The **framing hands technique** is a two-handed technique where the hands are positioned to form the two corners of a frame in the 2D image surrounding an object.

28.1.4 Volume-Based Selection Pattern

Related Patterns

3D Multi-Touch Pattern (Section 28.3.3) and World-in-Miniature Pattern (Section 28.5.2).

Description

The **Volume-Based Selection Pattern** enables selection of a volume of space (e.g., a box, sphere, or cone) and is independent of the type of data being selected. Data to be selected can be volumetric (voxels), point clouds, geometric surfaces, or even space containing no data (e.g., to follow with filling that space with some new data). Figure 28.3 shows how a selection box can be used to carve out a volume of space from a medical dataset.

When to Use

The Volume-Based Selection Pattern is appropriate when the user needs to select a not-yet-defined set of data in 3D space or to carve out space within an existing dataset. This pattern enables selection of data when there are no geometric surfaces (e.g., medical CT datasets), whereas geometric surfaces are required for implementing many other selection patterns/techniques (e.g., pointing requires intersecting a ray with object surfaces).

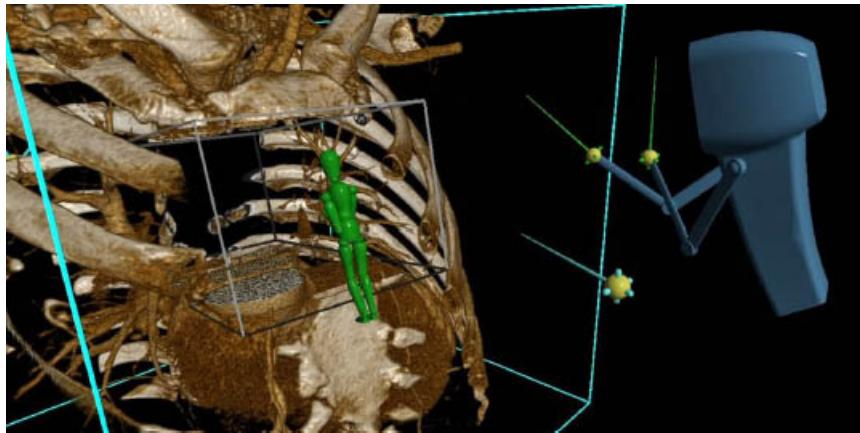


Figure 28.3 The blue user/avatar on the right has carved out a volume of space within a medical dataset via snapping, nudging, and reshaping a selection box (the gray box in front of the green avatar). The green avatar near the center is stepping inside the dataset to examine it from within. (Courtesy of Digital ArtForms)

Limitations

Selecting volumetric space can be more challenging than selecting a single object with the other more common selection patterns.

Exemplar Interaction Techniques

Cone-casting flashlight. The **cone-casting flashlight technique** uses pointing, but instead of using a ray, a cone is used. This results in easier selection of small objects than standard pointing via ray casting. If the intent is a single object, then the object closest to the cone's center line or the object closest to the user can be selected [[Liang and Green 1994](#)]. A modification of this technique is the aperture technique [[Forsberg et al. 1996](#)], which enables the user to control the spread of the selection volume by bringing the hand closer or further away.

Two-handed box selection. **Two-handed box selection** uses both hands to position, orient, and shape a box via snapping and nudging. Snap and nudge are asymmetric techniques where one hand controls the position and orientation of the selection box, and the second hand controls the shape of the box [[Yoganandan et al. 2014](#)]. Both snap and nudge mechanisms have two stages of interaction—grab and reshape. Grab influences the position and orientation of the box. Reshape changes the shape of the box.

Snap immediately brings the selection box to the hand and is designed to quickly access regions of interest that are within arm's reach of the user. Snap is an absolute interaction technique, i.e., every time snap is initiated the box position/orientation is reassigned. Therefore, snap focuses on setting the initial pose of the box and placing it comfortably within arm's reach.

Nudge enables incremental and precise adjustment and control of the selection box. Nudge works whether the selection box is near or far away from the user by maintaining the box's current position, orientation, and scale for the initial grab, but subsequent motion of the box is locked to the hand. Once attached to the hand, the box is positioned and oriented relative to its initial state—because of this, nudge can be thought of as a relative change in box pose. The box can then be simultaneously reshaped with the other hand while holding down a button.

28.2 Manipulation Patterns

Manipulation is the modification of attributes for one or more objects such as position, orientation, scale, shape, color, and texture. Manipulation typically follows selection, such as the need to first pick up an object before throwing it. **Manipulation Patterns** include the Direct Hand Manipulation Pattern, Proxy Pattern, and 3D Tool Pattern.

28.2.1 Direct Hand Manipulation Pattern

Related Patterns

Hand Selection Pattern (Section [28.1.1](#)), Pointing Hand Pattern (Section [28.5.1](#)), and 3D Tool Pattern (Section [28.2.3](#)).

Description

The **Direct Hand Manipulation Pattern** corresponds to the way we manipulate objects with our hands in the real world. After selecting the object, the object is attached to the hand moving along with it until released.

When to Use

Direct positioning and orientation with the hand have been shown to be more efficient and result in greater user satisfaction than other manipulation patterns [[Bowman and Hodges 1997](#)].

Limitations

Like the Hand Selection Pattern (Section [28.1.1](#)), a straightforward implementation is limited by the physical reach of the user.

Exemplar Interaction Techniques

Non-isomorphic rotations. Some form of clutching is required to rotate beyond certain angles, and clutching can hinder performance due to wasted motion [[Zhai et al. 1996](#)]. Clutching can be reduced by using non-isomorphic rotations [[Poupyrev et al. 2000](#)] that allow one to control larger ranges of 3D rotation with smaller wrist rotation. Non-isomorphic rotations can also be used to provide more precision by mapping large physical rotations to smaller virtual rotations.

Go-go technique. The go-go technique (Section [28.1.1](#)) can be used for manipulation as well as selection with no mode change.

28.2.2 Proxy Pattern

Related Patterns

Direct Hand Manipulation Pattern (Section [28.2.1](#)) and World-in-Miniature Pattern (Section [28.5.2](#)).

Description

A **proxy** is a local object (physical or virtual) that represents and maps directly to a remote object. The **Proxy Pattern** uses a proxy to manipulate a remote object. As the user directly manipulates the local object(s), the remote object(s) is manipulated in the same way.

When to Use

This pattern works well when a remote object needs to be intuitively manipulated as if it were in the user's hands or when viewing and manipulating objects at multiple scales (e.g., the proxy object can stay the same size relative to the user even as the user scales himself relative to the world and remote object).

Limitations

The proxy can be difficult to manipulate as intended when there is a lack of directional compliance (Section [25.2.5](#)); that is, when there is an orientation offset between the proxy and the remote object.

Exemplar Interaction Technique

Tracked physical props. **Tracked physical props** are objects directly manipulated by the user (a form of passive haptics; Section [3.2.3](#)) that map to one or more virtual objects and are often used to specify spatial relationships between virtual objects. [Hinckley et al. \[1998 \]](#) describe an asymmetric two-handed 3D neurosurgical visualization system where the non-dominant hand holds a doll's head and the dominant hand holds a planar object or a pointing device (Figure [28.4](#)). The doll's head directly maps to a remotely viewed neurological dataset and the planar object controls a slicing plane to see inside the dataset. The pointing device controls a virtual probe. Such physical proxies provide direct action-task correspondence, facilitate natural two-handed interactions, provide tactile feedback to the user, and are extremely easy to use without requiring any training.



Figure 28.4 A physical proxy prop used to control the orientation of a neurological dataset. (From [Hinckley et al. \[1994 \]](#))

28.2.3 3D Tool Pattern

Related Patterns

Hand Selection Pattern (Section [28.1.1](#)) and Direct Hand Manipulation Pattern (Section [28.2.1](#)).

Description

The **3D Tool Pattern** enables users to directly manipulate an intermediary 3D tool with their hands that in turn directly manipulates some object in the world. An example of a 3D tool is a stick to extend one's reach or a handle on an object that enables the object to be reshaped.

When to Use

Use the 3D Tool Pattern to enhance the capability of the hands to manipulate objects. For example, a screwdriver provides precise control of an object by mapping large rotations to small translations along a single axis.

Limitations

3D tools can take more effort to use if the user must first travel and maneuver to an appropriate angle in order to apply the tool to an object.

Exemplar Interaction Techniques

Hand-held tools. **Hand-held tools** are virtual objects with geometry and behavior that are attached to/held with a hand. Such tools can be used to control objects from afar (like a TV remote control) or to work more directly on an object. A paintbrush used to draw on a surface of an object is an example of a hand-held tool. Hand-held tools are often easier to use and understand than widgets (Section [28.4.1](#)) due to being more direct.

Object-attached tools. An **object-attached tool** is a manipulable tool that is attached/colocated with an object. Such a tool results in a more coupled signifier representing the affordance between the object, tool, and user. For example, a color icon might be located on an object, and the user simply selects the icon at which point a color cube appears so the user can choose the color of the object. Or if the shape of a box can be changed, then an adjustment tool can be made available on the corners of the box (e.g., dragging the vertex).

Jigs. Precisely aligning and modeling objects can be difficult with 6 DoF input devices due to having no physical constraints. One way to enable precision is to add virtual constraints with jigs. **Jigs**, similar to real-world physical guides used by carpenters and machinists, are grids, rulers, and other referenceable shapes that the user attaches to object vertices, edges, and faces. The user can adjust the jig parameters (e.g., grid spacing) and snap other objects into exact position and orientation relative to the object the jig is attached to. Figure [28.5](#) shows some examples of jigs. Jig kits support the snapping together of multiple jigs (e.g., snapping a ruler to a grid) for more complex alignments.

28.3 Viewpoint Control Patterns

Viewpoint control is the task of manipulating one's perspective and can include translation, orientation, and scale. Travel (Section [10.4.3](#)) is a form of viewpoint control that does not allow scaling.

Controlling one's viewpoint is equivalent to moving, rotating, or scaling the world. For example, moving the viewpoint to the left is equivalent to moving the world to the right, or scaling oneself to be smaller is equivalent to scaling the world to be larger. Thus, users perceive themselves either as moving through the world (self-motion) or as the world moving around them (world motion) as the viewpoint changes.

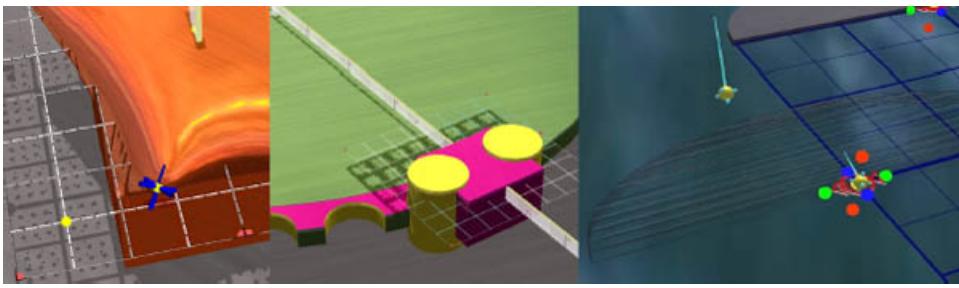


Figure 28.5 Jigs used for precision modeling. The blue 3D crosshairs in the left image represent the user’s hand. The user drags the lower left corner of the orange object to a grid point (left). The user cuts shapes out of a cylinder at 15° angles (center). The user precisely snaps a wireframe-viewed object onto a grid (right). (Courtesy of Digital ArtForms and Sixense)

Viewpoint Control Patterns include the Walking Pattern, Steering Pattern, 3D Multi-Touch Pattern, and Automated Pattern. Warning: Some implementations of these patterns may induce motion sickness and are not appropriate for users new to VR or those who are sensitive to scene motion. In many cases, sickness can be reduced by integrating the techniques with the suggestions in Part III .

28.3.1 Walking Pattern

Description

The **Walking Pattern** leverages motion of the feet to control the viewpoint. Walking within VR [[Steinicke et al. 2013](#)] includes everything from real walking to mimicking walking by moving the feet when seated.

When to Use

This pattern matches or mimics real-world locomotion and therefore provides a high degree of interaction fidelity. Walking enhances presence and ease of navigation [[Usoh et al. 1999](#)] as well as spatial orientation and movement understanding [[Chance et al. 1998](#)]. Real walking is ideal for navigating small to medium-size spaces, and such travel results in no motion sickness if implemented well.

Limitations

Walking is not appropriate when rapid or distant navigation is important. True walking across large distances requires a large tracked space, and for wired headsets cable tangling can pull on the headset and be a tripping hazard (Section [14.3](#)). A human spotter should closely watch walking users to help stabilize them if necessary. Fatigue can result with prolonged use, and walking distance is limited by the physical motion that users are willing to endure. Cable tangling is also an issue when using a wired system; often an assistant holds the wires and follows the user to prevent tangling and tripping.

Exemplar Interaction Techniques

Real walking. **Real walking** matches physical walking with motion in the virtual environment; the mapping from real to virtual is one-to-one. Because of high interaction fidelity, real walking is an ideal interface for many VR experiences. Real walking typically does not directly measure foot motion, but instead tracks the head. Real walking results in less motion sickness due to better matching visual and vestibular cues (although sickness can still result due to issues such as latency and system miscalibration). The motion of one’s feet in a self-embodied avatar can be estimated or the feet can be tracked to provide greater biomechanical symmetry. Unfortunately, real walking by itself limits travel in the virtual world to the physically tracked space.

Redirected walking. **Redirected walking** [[Razzaque et al. 2001](#)] is a technique that allows users to walk in a VR space larger than the physically tracked space. This is accomplished by rotation and translation gains that are different than the true motion of the user, directing the user away from the edges of tracked space (or physical obstacles). Ideally, the gains are below perceptible thresholds so that the user does not consciously realize he is being redirected.

Walking in place. Various forms of **walking in place** exist [[Wendt 2010](#)], but all consist of making physical walking motions (e.g., lifting the legs) while staying in the same physical spot but moving virtually. Walking in place works well when there is only a small tracked area and when safety is a primary concern. The safest form of walking in place is for the user to be seated. Theoretically users can walk in place for any distance. However, travel distances are limited by the physical effort the users are willing to make. Thus, walking in place works well for small and medium-size environments where only short durations of travel are required.

The human joystick. The **human joystick** [[McMahan et al. 2012](#)] utilizes the user’s position relative to a central zone to create a 2D vector that defines the horizontal direction and velocity of virtual travel. The user simply steps forward to control speed. The human joystick has the advantage that only a small amount of tracked space is required (albeit more than walking in place).

Treadmill walking and running. Various types of treadmills exist for simulating the physical act of walking and running (Section [3.2.5](#)). Although not as realistic as real walking, treadmills can be quite effective for controlling the viewpoint, providing a sense of self-motion, and walking an unlimited distance. Techniques should make sure foot direction movement is compliant with forward visual motion. Otherwise, treadmill

techniques that lack directional and temporal compliance (Section [25.2.5](#)) can be worse than no treadmill. Treadmills with safety harnesses are ideal, especially when physical running is required.

28.3.2 Steering Pattern

Description

The **Steering Pattern** is continuous control of viewpoint direction that does not involve movement of the feet. There is typically no need to control viewpoint pitch with controllers as is done with desktop systems since users can physically look up and down.

When to Use

Steering is appropriate for traveling great distances without the need for physical exertion. When exploring, viewpoint control techniques should allow continuous control, or at least the ability to interrupt a movement after it has begun. Such techniques should also require minimum cognitive load so the user can focus on spatial knowledge acquisition and information gathering. Steering works best when travel is constrained to some height above a surface, acceleration/deceleration can be minimized (Section [18.5](#)), and real-world stabilized cues can be provided (Sections [12.3.4](#) and [18.2](#)).

Limitations

Steering provides less biomechanical symmetry than the walking pattern. Many users report symptoms of motion sickness with steering. Virtual turning is more disorienting than physical turning.

Exemplar Interaction Techniques

Navigation by leaning. **Navigation by leaning** moves the user in the direction of the lean. The amount of lean typically maps to velocity. One advantage of this technique is no requirement for hand tracking. Motion sickness can be significant as velocity varies (i.e., acceleration).

Gaze-directed steering. **Gaze-directed steering** moves the user in the direction she is looking. Typically, the starting and stopping motion in the gaze direction is controlled by the user via a hand-held button or joystick. This is easy to understand and can work well for novices or for those accustomed to first-person video games where the forward direction is identical to the look direction. However, it can also be disorienting as any small head motion changes the direction of travel, and frustrating since users cannot look in one direction while traveling in a different direction.

Torso-directed steering. **Torso-directed steering** (also called chair-directed steering when the torso is not tracked), utilized when traveling over a terrain, separates the direction of travel from the way one is looking. This has more interaction fidelity than gaze-directed steering since in the real world one does not always walk in the direction the head is pointed. In the case when the torso or chair is not tracked, a general forward direction can be assumed. This technique can have more of a nauseating effect if the user does not have a mental model of what the forward direction is or when the entire body is turned when the torso or chair is not tracked. Visual cues can be provided to help users maintain a sense of the forward direction (Figure [18.2](#)).

One-handed flying. **One-handed flying** works by moving the user in the direction the finger or hand is pointing. Velocity can be determined by the horizontal distance of the hand from the head.

Two-handed flying. **Two-handed flying** works by moving the user in the direction determined by the vector between the two hands, and the speed is proportional to the distance between the hands [[Miné et al. 1997](#)]. A minimum hand separation is considered to be a “dead zone” where motion is stopped. This enables one to quickly stop motion by quickly bringing the hands together. Flying backward with two hands is more easily accomplished than one-handed flying (which requires an awkward hand or device rotation) by swapping the location of the hands.

Dual analog stick steering. **Dual analog stick steering** (also known as joysticks or analog pads) work surprisingly well for steering over a terrain (i.e., forward/back and left/right). In most cases, standard first-person game controls should be used where the left stick controls 2D translation (pushing up/down translates the body and viewpoint forward/backward, pushing left/right translates the body and viewpoint left/right) and the right stick controls left/right orientation (pushing left rotates the body and viewing direction to the left and pushing right rotates the body and viewing direction to the right). This mapping is surprisingly intuitive and is consistent with traditional first-person video games (i.e., gamers already understand how to use such controls so they have little learning curve).

Virtual rotations can be disorienting and sickness inducing for some people. Because of this, the designer might design the experience to have the content consistently in the forward direction so that no virtual rotation is required. Alternatively, if the system is wireless and the torso or chair is tracked, then there is no need for virtual rotations since the user can physically rotate 360°.

World-grounded steering devices. World-grounded input devices (Section [27.2.1](#)) such as flight sticks or steering wheels are often used to steer through a world. Such devices can be quite effective for viewpoint control due to the sense of actively controlling a physical device.

Virtual steering devices. Instead of using physical steering devices, virtual steering devices can be used. **Virtual steering devices** are visual representations of real-world steering devices (although they do not actually physically exist in the experience) that are used to navigate through the environment. For example, a virtual steering wheel can be used to control a virtual vehicle the user sits in. Virtual devices are more flexible than physical devices as they can be easily changed in software. Unfortunately, virtual devices are difficult to control due to having no proprioceptive force feedback (although some haptic feedback can be provided when using a hand-held controller with haptic capability).

28.3.3 3D Multi-Touch Pattern

Related Patterns

World-in-Miniature Pattern (Section 28.5.2) and Volume-Based Selection Pattern (Section 28.1.4).

Description

The **3D Multi-Touch Pattern** enables simultaneous modification of the position, orientation, and scale of the world with the use of two hands. Similar to 2D multi-touch on a touch screen, translation via 3D multi-touch is obtained by grabbing and moving space with one hand (monomotor interaction) or with both hands (synchronous bimanual interaction). One difference from 2D multi-touch is that one of the most common ways of using 3D multi-touch is to “walk” with the hands by alternating the grabbing of space with each hand (like pulling on a rope but with hands typically wider apart). Scaling of the world is accomplished by grabbing space with both hands and moving the hands apart or bringing them closer together. Rotation of the world is accomplished by grabbing space with both hands and rotating about a point (typically either about one hand or the midpoint between the hands). Translation, rotation, and scale can all be performed simultaneously with a single two-handed gesture.

When to Use

3D multi-touch works well for non-realistic interactions when creating assets, manipulating abstract data, viewing scientific datasets, or rapidly exploring large and small areas of interest from arbitrary viewpoints.

Limitations

3D multi-touch is not appropriate when the user is confined to the ground. 3D multi-touch can be challenging to implement as small nuances can affect the usability of the system. If not implemented well, 3D multi-touch can be frustrating to use. Even if implemented well, there can be a learning curve on the order of minutes for some users. Constraints, such as those that keep the user upright, limit scale, and/or disable rotations, can be added for novice users. When scaling is enabled and the display is monoscopic (or there are few depth cues), it can be difficult to distinguish between a small nearby object and a larger object that is further away. Therefore, visual cues that help the user create and maintain a mental model of the world and one’s place in it can be helpful.

Exemplar Interaction Techniques

Digital ArtForms’ Two-Handed Interface. Digital ArtForms has built a mature 3D multi-touch interface called THI (Two-Handed Interface) [Schultheis et al. 2012] based off of the work of Mapes and Moshell [1995] and Multigen-Paradigm’s SmartScene interface [Homan 1996] from the 1990s. Figure 28.6 shows a schematic for manipulating the viewpoint. Scale and rotation occur about the middle point between the two hands. Rotation of the world is accomplished by grabbing space with both hands and orbiting the hands about the midpoint between the hands, similar to grabbing a globe on both sides and turning it. Note this is not the same as attaching the world to a single hand as rotating with a single hand can be quite nauseating. Translation, rotation, and scale can all be performed simultaneously with a single two-handed gesture. In multi-user environments, other users’ avatars appear to grow or shrink as they scale themselves.

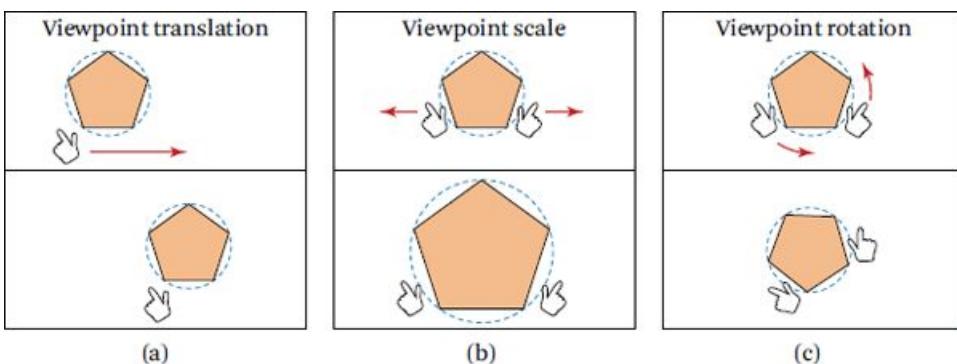


Figure 28.6 Translation, scale, and rotation using Digital ArtForms’ Two-Handed Interface. (From Mlyniec et al. [2011].)

Once learned, this implementation of treating the world as an object works well when navigation and selection/manipulation tasks are frequent and interspersed, since viewpoint control and object control are similar other than pushing a different button (i.e., only a single interaction metaphor needs to be learned and cognitive load is reduced by not having to switch between techniques). This gives users the ability to place the world and objects of interest into personal space at the most comfortable working pose via position, rotation, and scaling operations. Digital ArtForms calls this “posture and approach” (similar to what Bowman et al. [2004] call maneuvering). Posture and approach reduces gorilla arm (Section 14.1) and users have worked for hours without reports of fatigue [Jerald et al. 2013]. In addition, physical hand motions are non-repetitive by nature and are reported not to be subject to repetitive stress due to the lack of a physical planar constraint as is the case with a mouse.

The spindle. Building a mental model of the point being scaled and rotated about along with visualizing intent can be difficult for new users, especially when depth cues are absent. A “spindle” (Figure 28.7) consisting of geometry connecting the two hands (what Balakrishnan and

[Hinckley \[2000 \]](#) call visual integration), along with a visual indication of the center of rotation/scale dramatically helps users plan their actions and speeds the training process. Users simply place the center point of the spindle at the point they want to scale and rotate about, push a button in each hand, and “pull/scale” themselves toward it (or equivalently “pull/scale” the point and the world toward the user) while also optionally rotating about that point. In addition to visualizing the center of rotation/scale, the connecting geometry provides depth-occlusion cues that provide information of where the hands are relative to the geometry.

28.3.4 Automated Pattern

Description

The **Automated Pattern** passively changes the user’s viewpoint. Common methods of achieving this are by being seated on a moving vehicle controlled by the computer or by teleportation.

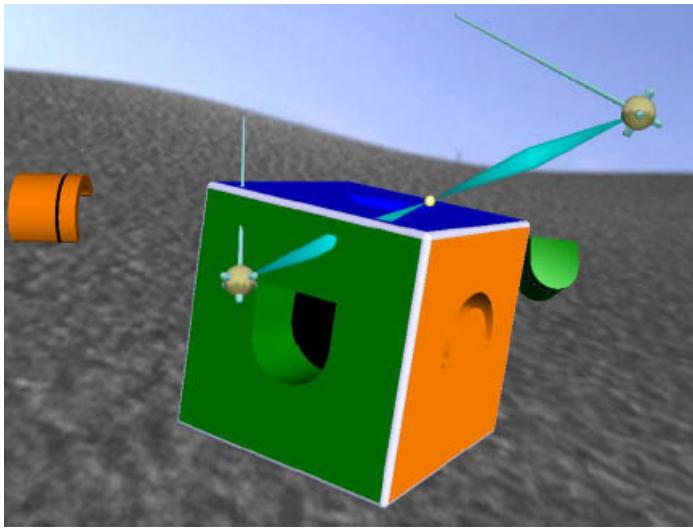


Figure 28.7 Two hand cursors and a spindle connecting those cursors. The yellow dot between the two cursors is the point that is rotated and scaled about. (From [Schultheis et al. \[2012 \]](#))

When to Use

Use when the user is playing the role of a passive observer as a passenger controlled by some other entity or when free exploration of the environment is not important or not possible (e.g., due to limitations of today’s cameras designed for immersive film).

Limitations

The passive nature of this technique can be disorienting and sometimes nauseating (depending on implementation). This pattern is not meant to be used to completely control the camera independent of user motion. VR applications should generally allow the look direction to be independent from the travel direction so the user can freely look around even if other viewpoint motion is occurring. Otherwise significant motion sickness will result.

Exemplar Interaction Techniques

Reducing motion sickness. Motion sickness can be significantly reduced with this technique by keeping travel speed and direction constant (i.e., keep velocity constant; Section [18.5](#)), providing world-stabilized cues (Section [18.2](#)), and creating a leading indicator (Section [18.4](#)) so users know what motions to expect.

Passive vehicles. **Passive vehicles** are virtual objects users can enter or step onto that transport the user along some path not controlled by the user. Passenger trains, cars, airplanes, elevators, escalators, and moving sidewalks are examples of passive vehicles.

Target-based travel and route planning. **Target-based travel** [[Bowman et al. 1998](#)] gives a user the ability to select a goal or location he wishes to travel to before being passively moved to that location. **Route planning** [[Bowman et al. 1999](#)] is the active specification of a path between the current location and the goal before being passively moved. Route planning can consist of drawing a path on a map or placing markers that the system uses to create a smooth path.

Teleportation. Teleportation is relocation to a new location without any motion. Teleportation is most appropriate when traveling large distances, between worlds, and/or when reducing motion sickness is a primary concern. Fading out and then fading in a scene is less startling than an instantaneous change. Unfortunately, straightforward teleportation comes at the cost of decreasing spatial orientation—users find it difficult to get their bearings when transported to a new location [[Bowman and Hodges 1997](#)].

28.4 Indirect Control Patterns

Indirect Control Patterns provide control through an intermediary to modify an object, the environment, or the system. Indirect control is more abstract than selection, manipulation, and viewpoint control. Indirect control is ideal when an obvious spatial mapping does not exist or it is difficult to directly manipulate an aspect of the environment. Examples include controlling the overall system, issuing commands, changing modes, and modifying non-spatial parameters.

Whereas the previously described techniques primarily describe both what should be done and how it is done, indirect control typically specifies only what should be done and the system determines how to do it. Because indirect control is not directly linked to that which it is controlling, signifiers such as the shape and size of controls, their visual representation and labeling, and apparent affordances of their underlying control structure are extremely important.

Indirect Control Patterns include the Widgets and Panels Pattern and Non-Spatial Control Pattern.

28.4.1 Widgets and Panels Pattern

Related Patterns

Hand Selection Pattern (Section [28.1.1](#)), Pointing Pattern (Section [28.1.2](#)), and Direct Hand Manipulation Pattern.

Description

The **Widgets and Panels Pattern** is the most common form of VR indirect control, and typically follows 2D desktop widget and panel/window metaphors. A **widget** is a geometric user interface element. A widget might only provide information to the user or might be directly interacted with by the user. The simplest widget is a single label that only provides information. Such a label can also act as a signifier for another widget (e.g., a label on a button). Many system controls are a 1D task and thus can be implemented with widgets such as pull-down menus, radio buttons, sliders, dials, and linear/rotary menus. **Panels** are container structures that multiple widgets and other panels can be placed upon. Placement of panels is important for being able to easily access them. For example, panels can be placed on an object, floating in the world, inside a vehicle, on the display, on the hand or input device, or somewhere near the body (e.g., a semicircular menu always surrounding the waist).

When to Use

Widgets and panels are useful for complex tasks where it is difficult to directly interact with an object. Widgets are normally activated via a Pointing Pattern (Section [28.1.2](#)) but can also be combined with other selection options (e.g., select a property for objects within some defined volume). Widgets can provide more accuracy than directly manipulating objects [[McMahan et al. 2014](#)]. Use gestalt concepts of perceptual organization (Section [20.4](#)) when designing a panel—use position, color, and shape to emphasize relationships between widgets. For example, put widgets with similar functions close together.

Limitations

Widgets and panels are not as obvious or intuitive as direct mappings and may take longer to learn. The placement of panels and widgets have a big impact on usability. If panels are not within personal space or attached to an appropriate reference frame (Section [26.3](#)), then the widgets can be difficult to use. If the widgets are too high, gorilla arm (Section [14.1](#)) can result for actions that take longer than a few seconds. If the widgets are in front of the body or head, then they can be annoying due to occluding the view (although this can be reduced by making the panel translucent).

If not oriented to face the user, information on the widget can be difficult to see. Large panels can also block the user's view.

Exemplar Interaction Techniques

2D desktop integration. Many panels are straightforward adaptations of windows from 2D desktop systems (Figure [28.8](#), left). An advantage of using desktop metaphors is their familiar interaction style and thus users have an instant understanding of how to use them. **2D desktop integration** brings existing 2D desktop applications into the environment via texture maps and mouse control with pointing. The system shown in Figure [3.5](#) also provides desktop metaphors such as double-clicking on a selected window header to minimize the window to a small cube (or double clicking the cube to make it large again). Bringing in such existing WIMP (Windows Icon Mouse Pointer) applications is typically not ideal from a 3D interaction perspective, but doing so does provide access to software that otherwise could only be accessed by exiting the virtual world. For example, an existing 2D calculator app could be available as a cube attached to the user's "belt." The user then simply selects and double clicks the cube to use the calculator tool without causing a break-in-presence.

Ring menus. A **ring menu** is a rotary 1D menu where a number of options are displayed concentrically about a center point [[Liang and Green 1994](#), [Shaw and Green 1994](#)]. Options are selected by rotating the wrist until the intended option rotates into the center position or a pointer rotates to the intended item (Figure [28.8](#), center). Ring menus can be useful but can cause wrist discomfort when large rotations are required. Non-isomorphic rotations (Section [28.2.1](#)) can be used to make small wrist rotations map to a larger menu rotation.

Pie menus. **Pie menus** (also known as marking menus) are circular menus with slice-shaped menu entries, where selection is based on direction, not distance. A disadvantage of pie menus is they take up more space than traditional menus (although using icons instead of text can help). Advantages of pie menus compared to traditional menus are that they are faster, more reliable with less error, and have equal distance for each option [[Callahan 1988](#)]. The most important advantage, however, may be that commonly used options are embedded into muscle memory as

usage increases. That is, pie menus are self-revealing gestures by showing users what they can do and directing how to do it. This helps novice users become experts in making directional gestures. For example, if a desired pie menu option is in the lower-right quadrant, then the user learns to initiate the pie menu, and then move the hand to the lower right. After extended usage, users can perform the task without the need to look at the pie menu or even for it to be visible. Some pie menu systems only display the pie menu after a delay so the menu does not show up and occlude the scene for fast expert users—known as “mark ahead” since the user marks the pie menu element before it appears [[Kurtenbach et al. 1993](#)].



Figure 28.8 Three examples of hand-held panels with various widgets. The left panel contains standard icons and radio buttons. The center panel contains buttons, a dial, and a rotary menu that can be used as both a ring menu and a pie menu. The right image contains a color cube that the user is selecting a color from. (Courtesy of Sixense and Digital ArtForms)

Hierarchical pie menus can be used to extend the number of options. For example, to change a color of an object to red, a user might (1) select the object to be modified, (2) press a button to bring up a property’s pie menu, (3) move the hand to the right to select “colors,” which brings up a “colors” pie menu, or (4) move the hand down to select the color red by releasing the button. If the user had to commonly change objects to the color red then she would quickly learn to simply move the hand to the right and then down after selecting the object and initiating the pie menu.

[Gebhardt et al. \[2013 \]](#) compared different VR pie menu selection methods and found pointing to take less time and to be more preferred by users compared to hand projection (i.e., translation) or wrist roll (twist) rotation.

Color cube. A **color cube** is a 3D space that users can select colors from. Figure 28.8 (right) shows a 3D color cube widget—the color selection puck can be moved with the hand in 2D about the planar surface while the planar surface can be moved in and out.

Finger menus. **Finger menus** consist of menu options attached to the fingers. A pinch gesture with the thumb touching a finger can be used to select different options. Once learned the user is not required to look at the menus; the thumb simply touches the appropriate finger. This prevents occlusion as well as decreases fatigue. The non-dominant hand can select a menu (up to four menus) and the dominant hand can then select one of four items within that menu.

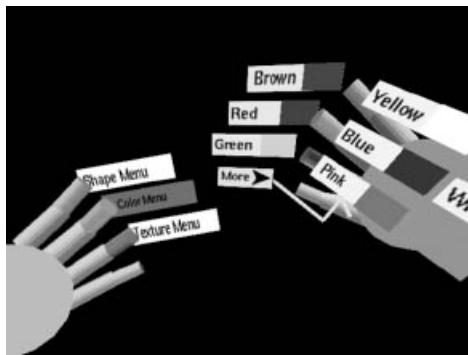


Figure 28.9 TULIP menus on seven fingers and the right palm. (From [Bowman and Wingrave \[2001 \]](#))

For complex applications where more options are required, a TULIP menu (Three-Up, Labels In Palm) [[Bowman and Wingrave 2001](#)] can be used. The dominant hand contains three menu options at a time and the pinky contains a “more” option. When the “more” option is selected, then the other three finger options are replaced with new options. By placing the upcoming options on the palm of the hand, users know what options will become available if they select the “more” option (Figure 28.9).

Above-the-head widgets and panels. **Above-the-head widgets and panels** are placed out of the way above the user, and accessed via reaching up and pulling down the widget or panel with the non-dominant hand. Once the panel is released, then it moves up to its former location out of view. The panel might be visible above, especially for new users, but after learning where the panels are located relative to the body, the panel might be made invisible since the user can use his sense of proprioception to know where the panels are without looking. [Miné et al. \[1997 \]](#) found users could easily select among three options above their field of view (up to the left, up in the middle, and up to the right).

Virtual hand-held panels. If a widget or panel is attached somewhere in the environment, then it can be difficult to find. If it is locked in screen space, then it can occlude the scene. One solution is to use **virtual hand-held panels** that have the advantage of always being available (as well as turned off) at the click of a button. Attaching the panel to the hand greatly diminishes many of the problems of world-spaced panels (e.g., panels that are difficult to read or get in the way of other objects can be reoriented and moved in an intuitive way without any cognitive effort). The panel should be attached to the non-dominant hand, and the panel is typically interacted with by pointing with the dominant hand (Figure 28.8). Such an interface provides a “double dexterity” where the panel can be brought to the pointer and the pointer can be brought to the panel.

Physical panels. Virtual panels offer no physical feedback, which can make it difficult to make precise movements. A **physical panel** is a real-world tracked surface that the user carries and interacts with via a tracked finger, object, or stylus. Using a physical panel can provide fast and accurate manipulation of widgets [[Stoakley et al. 1995](#), [Lindeman et al. 1999](#)] due to the surface acting as a physical constraint when touched. The disadvantage of a physical panel is that users can become fatigued from carrying it and it can be misplaced if set down. Providing a physical table or other location to set the panel on can help reduce this problem where the panel still travels with the user when virtually moving. Another option is to strap the panel to the forearm [[Wang and Lindeman 2014](#)]. Alternatively, the surface of the arm and/or hand can be used in place of a carried panel.

28.4.2 Non-Spatial Control Pattern

Related Patterns

Multimodal Pattern (Section [28.5.3](#)).

Description

The **Non-Spatial Control Pattern** provides global action performed through description instead of a spatial relationship. This pattern is most commonly implemented through speech or gestures (Section [26.4](#)).

When to Use

Use when options can be visually presented (e.g., gesture icons or text to speak) and appropriate feedback can be provided. This pattern is best used when there are a small number of options to choose from and when a button is available to push-to-talk or push-to-gesture. Use voice when moving the hands or the head would interrupt a task.

Limitations

Gestures and accents are highly variable from user to user and even for a single user. There is often a trade-off of accuracy and generality—the more gestures or words to be recognized then the less accurate the recognition rate (Section [26.4.2](#)). Defining each gesture or word to be based on its invariant properties and to be independent from others makes the task easier for both the system and the user. System recognition of voice can be problematic when many users are present or there is a lot of noise. For important commands, verification may be required and can be annoying to users.

Even for hypothetically perfectly working systems, providing too many options for users can be overwhelming and confusing. Typically, it is best to only recognize a few options that are simple and easy to remember.

Depending too heavily on gestures can cause fatigue, especially if the gestures must be made often and above the waist. Some locations are inappropriate for speaking (e.g., a library) and some users are uncomfortable speaking to a computer.

Exemplar Interaction Techniques

Voice menu hierarchies. **Voice menu hierarchies** [[Darken 1994](#)] are similar to traditional desktop menus where submenus are brought up after higher-level menu options are selected. Menu options should be visually shown to users so users explicitly know what options are available. See Section [26.4.2](#) for more information about speech recognition.

Gestures. Gestures (Section [26.4.1](#)) can work well for non-spatial commands. Gestures should be intuitive and easy to remember. For example, a thumbs-up to confirm, raising the index finger to select “option 1,” or raising the index and middle finger to select “option 2.” Visual signifiers showing the gesture options available should be presented to the user, especially for users who are learning the gestures. The system should always provide feedback to the user when a gesture has been recognized.

28.5 Compound Patterns

Compound Patterns combines two or more patterns into more complicated patterns. Compound Patterns include the Pointing Hand Pattern, World-in-Miniature Pattern, and Multimodal Pattern.

28.5.1 Pointing Hand Pattern

Related Patterns

Pointing Pattern (Section [28.1.2](#)), Direct Hand Manipulation Pattern (Section [28.2.1](#)), and Proxy Pattern (Section [28.2.2](#)).

Description

Hand selection (Section [28.1.1](#)) has limited reach. Pointing (Section [28.1.2](#)) can be used to select distant objects and does not require as much hand movement. However, pointing is often (depending on the task) not good for spatially manipulating objects because of the radial nature of pointing (i.e., positioning is done primarily by rotation about an arc around the user) [[Bowman et al. 2004](#)]. Thus, pointing is often better for

selection and a virtual hand is better for manipulation. The **Pointing Hand Pattern** combines the Pointing and Direct Hand Manipulation Patterns together so that far objects are first selected via pointing and then manipulated as if held in the hand. The user's real hand can also be thought of (and possibly rendered) as a proxy to the remote object.

When to Use

Use the Pointing Hand Pattern when objects are beyond the user's personal space.

Limitations

This pattern is typically not appropriate for applications requiring high interaction fidelity.

Exemplar Interaction Techniques

HOMER. The **HOMER technique** (Hand-centered Object Manipulation Extending Ray-casting) [[Bowman and Hodges 1997](#)] causes the hand to jump to the object after selection by pointing, enabling the user to directly position and rotate the object as if it were held in the hand. The scaled HOMER technique [[Wilkes and Bowman 2008](#)] scales object movement based on how fast the hand is moving (i.e., object translation is based on hand velocity). Fast hand motions enable gross manipulation whereas slow hand motions enable more precise manipulation, providing flexibility of object placement.

Extender grab. The **extender grab** [[Miné et al. 1997](#)] maps the object orientation to the user's hand orientation. Translations are scaled depending on the distance of the object from the user at the start of the grab (the further the object, the larger the scale factor).

Scaled world grab. A **scaled world grab** scales the user to be larger or the environment to be smaller so that the virtual hand, which was originally far from the selected object, can directly manipulate the object in personal space [[Miné et al. 1997](#)]. Because the scaling is about the midpoint between the eyes, the user often does not realize scaling has taken place. If the interpupillary distance is scaled in the same way, then stereoscopic cues will remain the same. Likewise, if the virtual hand is scaled appropriately, then the hand will not appear to change size. What is noticeable is head-motion parallax due to the same physical head movement mapping to a larger movement relative to the scaled-down environment.

28.5.2 World-in-Miniature Pattern

Related Patterns

Image-Plane Selection Pattern (Section [28.1.4](#)), Proxy Pattern (Section [28.2.2](#)), 3D Multi-Touch Pattern (Section [28.3.3](#)), and Automated Pattern (Section [28.3.4](#)).

Description

A **world-in-miniature** (WIM) is an interactive live 3D map—an exocentric miniature graphical representation of the virtual environment one is simultaneously immersed in [[Stoakley et al. 1995](#)]. An avatar or “doll” representing the self matches the user's movements giving an exocentric view of oneself in the world. A transparent viewing frustum extending out from the doll's head showing the direction the user is looking along with the user's field of view is also useful. When the user moves his smaller avatar, he also moves in the virtual environment. When the user moves a proxy object in the WIM, the object also moves in the surrounding virtual environment. Multiple WIMs can be created to view the world from different perspectives, the VR equivalent of 3D CAD Windowing System.

When to Use

WIMs work well as a way to provide situational awareness via an exocentric view of oneself and the surrounding environment. WIMs are also useful to quickly define user-defined proxies and to quickly move oneself.

Limitations

A straightforward implementation can cause confusion due to the focus being on the WIM, not on the full-scale virtual world.

A challenge with rotating the WIM is that it results in a lack of directional compliance (Section [25.2.5](#)); translating and rotating a proxy within the WIM can be confusing when looking at the WIM and larger surrounding world from different angles. Because of this challenge, the orientation of the WIM might be directly linked to the larger world orientation to prevent confusion (a forward-up map; Section [22.1.1](#)). However, this comes at the price of not being able to lock in a specific perspective (e.g., to keep an eye on a bridge as the user is performing a task at some other location) as the user reorients himself in the larger space. For advanced users, there should be an option to turn on/off the forward-up locked perspective.

Exemplar Interaction Techniques

Voodoo doll technique. The voodoo doll technique uses image-plane selection techniques (Section [28.1.3](#)) by having users temporarily create

miniature hand-held proxies of distant objects called **dolls** [[Pierce et al. 1999](#)]. Dolls are selected and used in pairs (with each doll representing a different object or set of objects in the scene). The doll in the non-dominant hand (typically representing multiple objects as a partial WIM) acts as a frame of reference, and the doll in the dominant hand (typically representing a single object) defines the position and orientation of its corresponding world object relative to the frame-of-reference doll. This provides the capability for users to quickly and easily position and orient objects relative to each other in the larger world. For example, a user can position a lamp on a table by first selecting the table with the non-dominant hand and then selecting the lamp with the dominant hand. Now, the user simply places the lamp doll on top of the table doll. The table does not move in the larger virtual world and the lamp is placed relative to that table.

Moving into one's own avatar. By moving one's own avatar in the WIM, the user's egocentric viewpoint can be changed (i.e., a proxy controlling one's own viewpoint). However, directly mapping the doll's orientation to the viewpoint can cause significant motion sickness and disorientation. To reduce sickness and confusion, the doll pose can be independent of the egocentric viewpoint; then when the doll icon is released or the user gives a command, the system automatically animates/navigates or teleports the user into the WIM via an automated viewpoint control technique (Section [28.3.4](#)) where the user "becomes" the doll [[Stoakley et al. 1995](#)]. This avoids the problem of shifting the user's cognitive focus back and forth from the map to the full-scale environment. That is, users think in terms of either looking at their doll in an exocentric manner or looking at the larger surrounding world in an egocentric manner, but not both simultaneously. In practice, users do not perceive a change in scale of either themselves or the WIM; they express a sense of going to the new location. Using multiple WIMs allows each WIM to act as a portal to a different, distant space.

Viewbox. The **viewbox** [[Mlyniec et al. 2011](#)] is a WIM that also uses the Volume-Based Selection Pattern (Section [28.1.4](#)) and 3D Multi-Touch Pattern (Section [28.3.3](#)). After capturing a portion of the virtual world via volume-based selection, the space within that box is referenced so that it acts as a real-time copy. The viewbox can then be selected and manipulated like any other object. The viewbox can be attached to the hand or body (e.g., near the torso acting as a belt tool). In addition, the user can reach inside the space and manipulate that space via 3D Multi-Touch (e.g., translate, rotate, or scale) in the same way that she can manipulate the larger surrounding space. Because the viewbox is a reference, any object manipulation that occurs in either space occurs in the other space (e.g., painting on a surface in one space results in painting in both spaces). Note due to the object being a reference, care must be taken to stop the recursive nature of the view, otherwise an infinite number of viewboxes within other viewboxes can result.

28.5.3 Multimodal Pattern

Description

Multimodal Patterns integrate different sensory/motor input modalities together (e.g., speech with pointing). Section [26.6](#) categorizes how different modalities can work together for interaction.

When to Use

Multimodal Patterns are appropriate when multiple facets of a task are required, reduction in input error is needed, or no single input modality can convey what is needed.

Limitations

Multimodal techniques can be difficult to implement due to the need to integrate multiple systems/technologies into a single coherent interface and the risk of multiple points of failure. Techniques can also be confusing to users if not implemented well. In order to keep interaction as simple as possible, a multimodal technique should not be used unless there is a good reason to do so.

Exemplar Interaction Techniques

Immersive “put-that-there.” The classic example of the Multimodal Pattern is a **put-that-there** interface [[Bolt 1980](#)] that uses a combination of the Pointing Pattern (to select the “that” and “there”) and the Non-Spatial Control Pattern via voice (to select the verb “put”). [Neely et al. \[2004\]](#) implemented an immersive form of a “put-that-there” style of region definition. With this technique, the user names and defines the vertices of a polygonal region on a terrain via pointing and speaking. An example is “Make Target Zone Apple from here (pointing gesture) to here (pointing gesture) . . . and to here (pointing gesture).” Figure [32.2](#) shows the architectural block diagram for this system.

For manipulating a single object, a “that-moves-there” interface (Section [26.6](#)) where the object is selected before specifying the action is often more efficient (Section [26.5](#)).

Automatic mode switching. One knows when the hand is within the center of vision and can effortlessly bring the hand into view when needed. Mode switching can take advantage of this. For example, an application might switch to an image-plane selection technique (Section [28.1.3](#)) when the hand is moved to the center of vision and to a pointing selection technique (Section [28.1.2](#)) when moved to the periphery (since the source of a ray pointer does not need to be seen).

Interaction: Design Guidelines

Just as there is no single tool in the real world that solves every problem, there is no single input device, concept, interaction pattern, or interaction technique that is best for all VR applications. Although it is generally better to use the same interaction metaphor across different types of tasks, that is not always possible or appropriate. Interaction designers should take into account the specific task when choosing, modifying, or creating new interactions.

29.1 Human-Centered Interaction (Chapter [25](#))

Intuitiveness (Section [25.1](#))

- Focus on making interfaces intuitive—that is, design interfaces that can quickly be understood, accurately predicted, and easily used.
- Use interaction metaphors (concepts that exploit specific knowledge that users already have of other domains) to help users quickly develop a mental model of how an interface works.
- Include within the virtual world everything that is needed to help users form a consistent conceptual model of how things work (e.g., within world tutorials). Users should not have to rely on any external explanation.

Norman's Principles of Interaction Design (Section [25.2](#))

- Practice human-centered design and follow well-known general principles to help users create simplified mental models of how the interactions work. These include consistent affordances, unambiguous signifiers, constraints to guide actions and ease interpretation, obvious and understandable mappings, and immediate and useful feedback.

Affordances (Section [25.2.1](#))

- Remember that an affordance is not a property of an object, but is a relationship between an object and a user. Affordances don't only depend on the object being afforded, but also depend on the user; affordances may be different for users with different capabilities.

Signifiers (Section [25.2.2](#))

- Make affordance perceivable through signifiers. A good signifier informs users what is possible before interacting.
- Make obvious to the user the state of the current interaction mode.

Constraints (Section [25.2.3](#))

- Where appropriate, add constraints to limit possible actions and to improve accuracy and efficiency.
- Use constraints to add realism (e.g., do not allow users to travel through walls).
- Do not assume real-world rules such as gravity are always appropriate. For example, hanging tools in the space around the user makes them easier to grab.
- Use signifiers wisely to prevent users from making the wrong assumptions about constraints. If a user doesn't know what to do, she is effectively constrained.
- Make constraints consistent so learning can be transferred across tasks.
- Consider allowing expert users to remove constraints for advanced interactions.

Feedback (Section [25.2.4](#))

- Use feedback substitution when haptics are not available. For example, use audio and highlighting to signify touching of objects.
- Do not overwhelm users with too much feedback.
- Consider putting information in front of the user in the torso reference frame near the waist rather than on a heads-up display in the head reference frame.
- If a heads-up display in the head reference frame must be used, only present minimal information on the display.
- Provide the capability to turn on/off (or make visible/invisible) widgets, tools, and interface cues.

Mappings (Section [25.2.5](#))

- To maximize performance and satisfaction, maintain directional compliance, nulling compliance, and temporal compliance.
- Focus first on creating mappings that have directional compliance (the direction of sensory feedback should match the direction of the interface device) so users can anticipate motion in response to their physical input.
- For fully direct interactions, maintain position compliance (i.e., make the virtual position of objects match the physical position of the device).
- If position compliance is not appropriate, use nulling compliance (the virtual object should return to its original location when the device returns to its original location).
- Use nulling compliance to take advantage of muscle memory.
- Choose absolute input devices over relative input devices in order to maintain compliance.
- If the results of an interaction cannot be immediately computed (i.e., a lack of temporal compliance), then provide some form of immediate feedback to inform the user the problem is being worked on.
- For non-spatial mappings, use commonly accepted metaphors (e.g., up is more, down is less).

Direct vs. Indirect Interaction (Section [25.3](#))

- Use tools to extend one's reach so users feel like they are directly interacting.
- Use direct, semi-direct, and indirect interactions where appropriate and not where not appropriate. Do not try to force everything to be direct.

The Cycle of Interaction (Section [25.4](#))

- To design or improve interactions, use Norman's seven stages of interaction to break up typically subconscious processes into explicit steps.
- Think about what stages are missing or what stages do not work well that make interaction difficult. Then add or modify signifiers, constraints, mappings, and feedback as appropriate for each stage.
- Use the seven stages of interaction with task analysis as a stepping stone toward implementation.

The Human Hands (Section [25.5](#))

- Support interactions with two hands where appropriate.
- Do not assume just because two hands are used for an interface that the interface will be better. Two-handed interfaces can be difficult to use if designed inappropriately.
- Feedback from users is even more essential for two-handed interaction than one-handed interaction.
- Have the non-dominant hand maintain the reference frame so the dominant hand can work precisely without having to work in a locked position.
- Design bimanual interactions to work together in a fluid manner, switching between symmetric and asymmetric modes as appropriate for the task.

29.2 VR Interaction Concepts (Chapter [26](#))

Interaction Fidelity (Section [26.1](#))

- Consider using realistic interactions for training applications, simulations, surgical applications, therapy, and human-factors evaluations.
- Consider using non-realistic interactions for increasing performance and minimizing fatigue.
- Use magical interactions to enhance the user experience, circumvent the limitations of the real world, and teach abstract concepts.
- Consider interaction metaphors as a source of inspiration. Unless realistic interaction is a primary goal, use intuitive and useful magical techniques.

Proprioceptive and Egocentric Interaction (Section [26.2](#))

- Exploit the one real object every user has—the human body.
- Use torso tracking in addition to head and hands tracking whenever possible to maximize proprioception. Alternatively, chair rotation can be tracked to estimate torso rotation.

- Place commonly used tools relative to the body to take advantage of muscle memory so users do not have to take visual attention away from that which is being worked on (tools do not even have to be within the user's field of view).
- Don't assume exocentric perspectives preclude egocentric perspectives. Take advantage of egocentric intuition even when designing an exocentric experience.

Reference Frames (Section [26.3](#))

- Provide the capability for users to think and interact in an exocentric virtual-world reference frame when the intent is for users to create content over a wide area, form a cognitive map of the environment, determine their own global position, or plan travel on a large scale.
- Be careful of placing direct interfaces in virtual-world reference frames as they can be awkward to reach without easy and precise navigation capability.
- Draw rest frames (e.g., an automobile interior, a cockpit, or non-realistic cues) in the real-world reference frame for users to feel stabilized in space and to reduce motion sickness.
- Provide a place for users to set physical devices when not needed and render those objects in the real-world reference frame so they can easily be seen and picked up.
- Place information, interfaces, and tools relative to the body by placing them in the torso reference frame.
- Allow advanced users to turn on/off or make invisible (but still usable) items in the torso, hand, and head reference frames.
- Place a visual representation of hand controllers in the torso reference frame for non-tracked controllers and in the hand reference frame for tracked hand-held controllers.
- Place signifiers in the hand reference frame pointing to buttons, analog sticks, and/or fingers so it is obvious what they do. Provide the capability for users to turn them on and off.
- Minimize cues in the head reference frame for anything other than a pointer if head tracking is used for input.

Speech and Gestures (Section [26.4](#))

- Use explicit visual signifiers (e.g., a list of words or hand icons) to portray what speech or gesture commands are available. Provide feedback by highlighting the selected option.
- Have users verify important commands to prevent major errors.
- Use a small number of words or gestures that are well defined, natural, easy for the user to remember, and easy for the system to recognize.
- When more than one person will be present in the same space, use push-to-talk and/or push-to-gesture to prevent accidental unintended commands.
- Use direct structural gestures for immediate system response.
- When users have their own unique system, use speaker-dependent recognition if they are willing to train the system and adaptive recognition if they are not.
- Reduce the potential for error by only allowing a subset of vocabulary to be recognized depending upon the context.

Modes and Flow (Section [26.5](#))

- When multiple interaction modes are used, make it clear to the user what the current mode is.
- Use object-action (or selection-manipulation) sequences over action-object sequences.
- Enable smooth and easy transition between selecting an object and manipulating or using that object.
- Minimize distractions to enhance the flow of interactions and allow for full attention to the primary task.
- Design interactions so users do not have to physically (whether the eyes, head, or hands) or cognitively move between tasks.
- Use lightweight mode switching, physical props, and multimodal techniques to help maintain the flow of interaction.

Multimodal Interaction (Section [26.6](#))

- Use a single specialized input modality when it is clear that single modality is best for the task and there is no reason to include other modalities. Do not add a modality just for the sake of adding a modality.

- Consider using equivalent input modalities when user preferences are strongly divided.
- Use redundant input modalities to reduce noise and ambiguous signals.
- Use concurrent input modalities to improve efficiency by enabling the user to perform two interactions simultaneously.
- Use complementarity input modalities for “put-that-there” or “that-moves-there” types of interfaces.
- Allow transfer of input modalities when one modality device is unreliable so users will not have to start over if there is a failure.

Beware of Sickness and Fatigue (Section [26.7](#))

- Be extra careful of viewpoint control techniques that can induce motion sickness. Study Part [III](#) and follow the guidelines in Chapter [19](#) to minimize adverse health effects.
- When motion sickness is a primary concern (e.g., for users new to VR or a general audience), only use one-to-one mapping of real head motion or teleportation.
- Avoid using and creating interactions that require the user to hold the hands up high or in front of the body for more than a few seconds at a time. Use devices that do not require line of sight so interactions can be performed comfortably in the lap or from the sides of the body.

Visual-Physical Conflict and Sensory Substitution (Section [26.8](#))

- Enforce physics constraints when hand penetration into objects is shallow so the hand does not pass through the object surface. Do not enforce physics constraints when penetration is deep; instead allow the hand to pass through virtual objects.
- Consider drawing two virtual hands for a single physical hand—one that penetrates objects and one that doesn’t.
- Use highlighting to convey an object is selectable when the hand is close to the object.
- Use audio to convey collisions.
- Use passive haptics or vibrotactile haptics whenever possible.
- If training transfer is not important, use ghosting to signify a new potential position until the user confirms placement.

29.3 Input Devices (Chapter [27](#))

Input Device Characteristics (Section [27.1](#))

- Match the interaction technique with the device and match the device with the interaction technique. Understand the different device characteristics and classes in order to determine what is best for the project.
- Use 6 DoF devices whenever possible and then decrease DoFs in software where appropriate.
- Choose input devices that work in a user’s entire personal space, do not require line of sight, are robust to various lighting conditions, and work for all hand orientations.
- Use buttons for tasks that are binary, when the action needs to occur often, when immediate response and reliability are required, when abstract interactions are appropriate, and when physical feedback of feeling the button pressed/released is important.
- Do not overwhelm users with too many buttons.
- Use bare-hand systems, gloves, and/or haptic devices that correspond to virtual objects when a high sense of realism and presence is important.

Classes of Hand Input Devices (Section [27.2](#))

- If not sure what to use or there is no strong preference, then start with tracked hand-held controllers. They are currently the best option for a majority of interactive VR experiences.
- For public location-based entertainment, consider building customized interfaces such as world-grounded devices that are optimized for the particular experience.
- Attach labels to virtual representations of controllers to signify what the controls do.
- Use tracked hand-held controllers when the user commonly holds virtual devices to enhance presence through physical touch.
- Don’t assume gloves are only for full hand tracking. Consider using pinch gloves that don’t require holding a physical controller but have the advantages of buttons via pressing the fingers together.

Classes of Non-hand Input Devices (Section [27.3](#))

- If eye tracking is available, do not overuse it. Instead use eye gaze for specialized tasks and in subtle ways (e.g., have virtual characters respond when looked at).
- When designing eye gaze interactions, maintain the natural function of the eyes, augment rather than replace, focus on interaction design, improve the interpretation of eye movements, choose appropriate tasks, use passive gaze over active gaze, and leverage gaze information for other interactions.
- Use microphones that are specially designed for speech recognition.

29.4 Interaction Patterns and Techniques (Chapter [28](#))

Selection Patterns (Section [28.1](#))

- When interactions do not need to be realistic, use the Pointing Pattern or the Image-Plane Selection Pattern.

Hand Selection Pattern (Section [28.1.1](#))

- Use the Hand Selection Pattern when interactions need to be realistic.
- For high interaction fidelity, use a realistic virtual hand to select objects.
- For mid-interaction fidelity, use hands without arms to reach just beyond personal space (i.e., near action space).
- Consider the go-go technique for personal space and midrange action space.

Pointing Pattern (Section [28.1.2](#))

- For more controlled pointing, consider using a precision mode.
- For selecting objects that are small, consider using pointing with object snapping.
- Do not use dwell selection unless there is no other good way to provide a signal or there is another good reason to do so.
- Do not use eye tracking for selecting objects unless there is a good reason to do so.

Image-Plane Selection Pattern (Section [28.1.3](#))

- For easy-to-use touch at a distance, use image-plane selection. However, do not use when the user must frequently select objects as gorilla arm can result.

Volume-Based Selection Pattern (Section [28.1.4](#))

- Use the Volume-Based Selection Pattern when selecting data/space that has no geometric surfaces.
- Be careful of requiring volume-based selection for novice users.

Manipulation Patterns (Section [28.2](#))

Direct Hand Manipulation Pattern (Section [28.2.1](#))

- Use the Direct Hand Manipulation Pattern unless there is a reason not to as it is more efficient and satisfying than other manipulation patterns.
- For high interaction fidelity, use a virtual hand for both selection and manipulation.
- Consider using non-isomorphic rotations to reduce clutching and to increase performance and precision.

Proxy Pattern (Section [28.2.2](#))

- Use the Proxy Pattern to intuitively manipulate remote objects or when the user scales himself or the world.
- Use tracked physical props for direct action-task correspondence. Such props result in an easy-to-use interface that does not require any training, facilitate natural two-handed interactions, and provide tactile feedback to the user.

3D Tool Pattern (Section [28.2.3](#))

- Use the 3D Tool Pattern to enhance the capability of the hands to manipulate objects.
- Use signifiers for object-attached tools to make it obvious how to use those tools to manipulate the object.
- To enable precise modeling and reduce complexity, use jigs that enable user-controlled constraints, snapping, and discrete manipulations.

Viewpoint Control Patterns (Section [28.3](#))

- Be especially careful of sickness and injury when choosing, designing, and implementing viewpoint control techniques, especially for users new to VR. See Part [III](#).

Walking Pattern (Section [28.3.1](#))

- Use the Walking Pattern when high biomechanical symmetry and high presence is desired, fatigue is not a concern, and safety measures to prevent tripping, physical collisions, and falling are covered.
- Use real walking when the physically tracked space is as large as or larger than the virtual walkable space, and both spatial understanding and minimizing sickness are important.
- Use redirected walking when the physically tracked space is smaller than the virtual walkable space.
- Use walking in place when the physically tracked space is small or safety is a concern.
- Use a treadmill when walking/running vast distances is required.

Steering Pattern (Section [28.3.2](#))

- Use the Steering Pattern when sickness is not a primary concern, interaction fidelity is not important, acceleration/deceleration can be minimized, and real-world stabilized cues can be provided.
- Make steering as simple as possible to minimize cognitive load so the user can focus on spatial knowledge acquisition and information gathering.
- Provide visual cues to help users know what direction is forward.
- Consider not using virtual rotations if torso/chair tracking is available and the system is wireless.
- If virtual rotations are required and the user can be constrained to the ground, use dual analog sticks.

3D Multi-Touch Pattern (Section [28.3.3](#))

- For applications not requiring high interaction fidelity, use the 3D Multi-Touch Pattern when creating assets, manipulating abstract data, viewing scientific datasets, or rapidly exploring large and small areas of interest from arbitrary viewpoints.
- Consider adding constraints (e.g., force uprightness, limit scale, and/or disable rotations) for novice users.
- Provide a visual indication of the center of rotation/scale.

Automated Pattern (Section [28.3.4](#))

- Use the Automated Pattern when free exploration of the environment is not desired or not possible.
- Use teleportation when traveling large distances, or between worlds, when efficiency is important, or when motion sickness must be minimized. Do not use if spatial orientation must be maintained.
- Use smooth transitions when maintaining spatial orientation is the primary concern.
- When passively moving the user, reduce motion sickness by keeping visual velocity as constant as possible, providing stable real-world reference cues (e.g., a cockpit), and/or providing a leading indicator.

Indirect Control Patterns (Section [28.4](#))

- Use an Indirect Control Pattern when spatial mappings are not appropriate or details of how something is done are not important to the user. Uses include controlling the overall system, issuing commands, changing modes, and changing non-spatial parameters.
- For indirect control, make signifiers obvious, such as the shape and size of controls, their visual representation and labeling, and apparent affordances of their underlying control structure.

Widgets and Panels Pattern (Section [28.4.1](#))

- Use the Widgets and Panels Pattern when it is difficult to directly interact with an object.
- Where appropriate, use well-known 2D interaction metaphors such as pull-down/pop-up menus, radio buttons, and checkboxes.
- Use gestalt concepts of perceptual organization when designing a panel—use position, color, and shape to emphasize relationships between widgets. For example, put widgets with similar functions close together.
- Place widgets and panels in a way that is easy for users to access (e.g., on the non-dominant hand or in the torso reference frame).
- For commonly used commands, use pie/marking menus to teach gestures and to embed those gestures into muscle memory.
- For pie/marking menus, use pointing over projection or roll.
- If pinch gestures are available, place menu options on the fingers.
- Consider placing panels or widgets above the head so users can pull them down when needed.
- Consider placing a panel in the non-dominant hand that can be turned on or off, and where widgets on the panel are controlled by the dominant hand.
- For 2D tasks that require precision, consider using a physical panel the user holds in the non-dominant hand or that is attached to the forearm

Non-Spatial Control Pattern (Section [28.4.2](#))

- Use the Non-Spatial Control Pattern for global action performed through description instead of a spatial relationship.
- Use real-world words and gestures that are intuitive and easy to remember. Keep the number of options small and simple.
- Provide signifiers (e.g., icons for gestures or a list of words for voice) to remind non-expert users what options are available.
- Always provide some form of feedback.
- When accuracy is more important than speed, verify commands. Make the confirmation process fast and do not require precision. Examples include clicking a physical button or saying “confirm”
- Use push-to-talk or push-to-gesture to prevent accidental unintended commands.
- Use voice control when moving the hands or the head would interrupt a task.
- Be careful of depending on voice recognition when multiple people are in the same physical environment or there is a lot of noise.

Compound Patterns (Section [28.5](#))

Pointing Hand Pattern (Section [28.5.1](#))

- When high interaction fidelity is not required, use the Pointing Hand Pattern to select objects at a distance but to manipulate them as if held in the hand.

World-in-Miniature Pattern (Section [28.5.2](#))

- Use the World-In-Miniature Pattern to provide situational awareness, to quickly define user-defined proxies, or to quickly move oneself.
- Consider using the equivalent of a forward-up map for the world-in-miniature so that the orientation of the map matches the orientation of the larger world. When appropriate, provide the capability to turn this feature off.
- To reduce sickness, do not directly map the user’s doll in the world-in-miniature to the user’s movement. Instead, animate/navigate or teleport the viewpoint into the doll when commanded by the user.

Multimodal Pattern (Section [28.5.3](#))

- Use the Multimodal Pattern when multiple facets of a task are required, when reduction in input error is needed, or when no single modality can convey what is needed.
- Only use a multimodal technique when there is a good reason to do so. When used, keep interactions as simple as possible.
- Consider using automatic mode switching when each technique is only usable in specific situations and those situations are clear to the user.

PART VI

ITERATIVE DESIGN

Design thinking takes a solution-focused approach to problem solving, working collaboratively to iterate an endless, shifting path toward perfection. It works toward product goals via specific ideation, prototyping, implementation, and learning steps to bring the appropriate solution to light .

—Jeff Gothelf and Josh Seiden [[2013](#)]

Up to this point, this book has provided a background on the mechanisms of perception, VR sickness, content creation, and interaction. However, such mechanisms are not entirely understood and very much depend on the goals and design of the VR project. Furthermore, assumptions about the real world or how we perceive and interact with it do not always hold in VR. One cannot simply look up the numbers when designing VR experiences or get the experience perfect on the first attempt. VR design is also very different than 2D desktop or mobile application design, and there is not nearly as much known or standardized about VR. VR design must be iteratively created based off of continual redesign, prototyping, and feedback from real users. In fact, the design of many VR experiences is due to “accidental discoveries” that were not initially intended (Andrew Robinson and Sigurdur Gunnarsson, personal communication, May 11, 2015). This chapter focuses on iterative design concepts to help the team move toward ideal experiences as quickly and efficiently as possible.

With new development platforms such as Unity and other tools, VR experiences can be created in a fraction of the time they could be created compared to just a few years ago. Experienced designers can now literally create a simple VR experience in hours, without any programming experience required. Some modifications can be as easy as a click of a button, and some assets and behaviors built by others can be integrated with basic programming skills. Because of how quickly simple VR experiences can be built and modified, prototyping basic individual components can often be done in a day or less. However, to go beyond the basics, designers should either be programmers themselves, learn basic programming skills for simple prototyping and testing, or work directly in pairs with programmers. Note that this is not to say professional programmers are not required for creating compelling VR experiences. Integrating advanced/innovative features and behaviors, understanding software architecture, and writing clean code takes time to do properly and are essential for creating efficient and maintainable code.

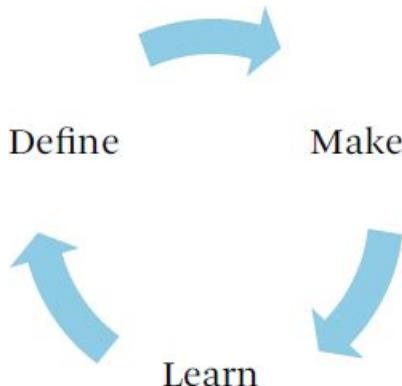


Figure VI.1 High-level iterative process common to all VR development.

The Define-Make-Learn Cycle

Although a single overall detailed process is not appropriate for VR design, there is general agreement that **iterative design** consists of defining the project, building prototypes, learning from users, and continually improving upon previous ideas. Careful consideration of these stages is critical to optimizing user experience and comfortable VR.

Part [VI](#) discusses the philosophy of iterative design and then the three iterative stages ([Figure VI.1](#)) as summarized below.

1. **The Define Stage.** This stage attempts to answer the question “What do we make?” and includes everything from the high-level vision to listing requirements.
2. **The Make Stage.** This stage answers the question “How do we make it?” and then proceeds to make it.
3. **The Learn Stage.** This stage answers the question “What works and what does not work?” The answers are fed back into the Define Stage to refine what is to be made.

Although conceptually these stages are performed in sequence, they are tightly interwoven and often occur in parallel. Within the three stages, various smaller process pieces are described that can be fit into the larger iterative process as needed. These individual pieces may or may not be used depending on the project and what part of the life cycle the process is in. The stages are often not consistent even across iterations of the same project. Thus, it would both limit and overcomplicate the design process to try to formalize a sequence of VR design processes, or even to formalize each individual process piece. Instead, basic overviews of the process concepts are described within each of the three primary stages. The specifics will depend on the project goals and team preference.

Overview of Chapters

Part VI consists of five chapters that describe the overall iterative process as well as more detailed specific processes as they apply to creating VR experiences.

Chapter 30, Philosophy of Iterative Design, first provides a review of overarching concepts essential to VR design. Iterative design as it applies to VR is especially dependent on art, science, human-centered design, iteration, the specifics of the project, and the team creating the experience.

Chapter 31, The Define Stage, discusses how VR application ideas are created and refined. Fifteen ways of defining different aspects of the project are described in this chapter.

Chapter 32, The Make Stage, discusses the actual implementation of the idea into initial tangible prototypes and then into more polished experiences that are delivered to end users.

Chapter 33, The Learn Stage, describes core research concepts and explains how to obtain qualitative feedback from representative users, collect more objective/quantitative data, improve the design through constructivist approaches, and formally test/compare different implementations.

Chapter 34, Iterative Design: Design Guidelines, summarizes the iterative process and lists a number of actionable guidelines to help VR creators move toward engaging experiences.

Philosophy of Iterative Design

Design has multiple meanings yet all are intimately related. **Design** is a created object, a process, and an action. Quality design focuses on utility, effectiveness, efficiency, elegance, delight, and the conveying of meaning [[Brooks 2010](#)]. For the purposes of this book, design entails the entire creation of a VR experience from information gathering and goal setting to delivery and product enhancement/support.

30.1 VR Is Both an Art and a Science

VR is an art because we have to be prolific in creating new experiences based off of common sense, rules of thumb, cultural metaphors, creative outside-the-box thinking, etc. VR is an art also because creating new worlds enables us to throw away many of the rules of the real world; VR requires innovation. Creating a quality VR experience is not just about optimizing a set of algorithms. It is more imperative for the humans behind the creation to decide on the right things to design instead of perfectly designing the wrong things. VR designers seek to find what is useful, whether realistic or magical.

VR is a science because our intuition of what will work often doesn't work in practice. We have to get users to test our ideas, collect data, and analyze that data to help determine how to improve the experience. As in programming, **iterative design** consists of quickly testing many implementations, continually improving upon previous ideas. The faster we can obtain quality answers to meaningful questions, the faster we can iterate toward better and better experiences.

30.2 Human-Centered Design

The measure of progress for a VR project is the user experience. If the experience is improved, then progress is being made. Traditional software development measures, such as the number of lines of code, are very different from measures that should be used for VR progress. Such measures are largely useless for VR. A well-working interaction technique that takes a couple of hours to integrate is worth far more than tens of thousands of lines of code that took months of work but doesn't feel right. A brilliantly engineered solution is irrelevant if it doesn't work for the human.

Bad VR insists users perform abnormally, to adapt themselves to the peculiar demands of the system. When users are not able to meet the inhuman expectations of the virtual environment, then there is a design problem. Human-centered design errors are often only discovered when those external to the project experience the creation. Obtaining feedback from real users is critical early in the project, for if human-centered design error is discovered near ship time, then the entire project may be deemed a failure instead of many small failures discovered and fixed in early iterations.

30.3 Continuous Discovery through Iteration

A core concept of VR design is that discovery should always be occurring. **Continuous discovery** is the ongoing process of engaging users during the design and development process [[Gotheff and Seiden 2013](#)]. The goal of discovery is to understand what users want to do, why they want to do it, and how they can best do it.

Knowing or rationalizing everything in advance for building complex systems results in projects becoming over-budget, overdue performance disasters [[Brooks 2010](#)]. This is not only because there are not-yet known answers to questions the team has at the beginning, but also because the team doesn't yet know what they don't know. Questions will lead to more questions that were not initially considered. Discovering what one did not formerly even consider is where new insights and breakthroughs in design most often occur.

Iteration is more important for developing VR systems than it is for developing non-VR systems [[Wingrave and LaViola 2010](#)]. The possibilities with VR extend beyond what is possible with reality, yet doing one thing badly can have severe consequences. Even if everything were known about VR, then the number of factors and combinations of factors that can go into a VR experience explodes exponentially. Thus, little can be known in advance and every project will be unique.

We can't afford to wait until the end of the project to see if everything works. (Hint: It won't.) Learning, change, and innovation are much less expensive at the beginning than at the end, and there must be many options available that enable fast adjustments as necessary. The question, then, is not if iterative design should be used but how to most quickly and effectively iterate. This is accomplished via rapid prototyping and getting feedback from experts and representative users as quickly and as often as possible. As advocated by *The Lean Startup* [[Ries 2011](#)], fail early and fail often. Each failure teaches about what to do better. Failures are essential for exploration, creativity, and innovation. If failure is not occurring often, then one is not innovating enough and breakthroughs are not likely to occur. It is fairly easy to create a safe VR experience where there is no risk of learning. But that is also the route to a dull, uninteresting experience.

Even if everything could be known at the start of the project, change happens over time. The reality of life, business, and technical advancement dictate that what was important yesterday may not be important today. When reality gets in the way of the plan, then that reality always wins. Another challenge that is especially true for VR projects is that when one part of the application changes, it rarely occurs in isolation. Always be on the lookout for changes in assumptions, technology, and business. Then change course as necessary through adaptive planning.

For those new to VR (this includes everyone on the team from business development to programmers), the first step of discovery is to step into the role of user and try a broad range of existing VR experiences, making notes of what works and what doesn't work. Then, new insights are discovered by prototyping new or modified ideas and getting feedback from users. In fact, this is more important than planning and pre-analysis.

There is more value in getting a bad prototype out than spending days thinking, considering, and debating. It is important to create a culture of accepting failure, especially early in a VR project, so that team members feel it is safe to experiment. Such experimentation encourages creativity and innovation and leads to effective VR. Feedback and measurement from real application beats opinions based on theory.

In the end, the success or failure of the VR experience isn't the team's decision—it's the consumers of the experience. The sooner the team can discover what gets results for users, then instead of iterating in an arbitrary direction, the team can iterate toward success.

30.4 There Is No One Way—Processes Are Project Dependent

In any field, no single design process is appropriate for every project. This is even truer for VR for multiple reasons. Even though VR research has occurred over multiple decades, the community is constantly learning/changing and there are not yet (and there may never be) standardized processes to follow. VR covers a broad range of industries and applications. For example, creating an assembly-line automotive training system is going to have a very different process than creating an immersive film. Even within the same project there are many factors that will determine what process is most appropriate at the time. Factors include the current size of the team, approaching deadlines, and how far along the project is. Because there are so many unknowns, and large variance for what is known, what is most appropriate must be determined on a project-by-project basis, and “appropriate” will change as the project iterates toward a solution.

Given the variety of VR project demands, why define processes in advance at all? There must be some way to bring order to chaos; otherwise, a final cohesive application can't be achieved. At some point, approval must be obtained to move ahead. Schedules and budgets cannot be overrun. Customers must be kept satisfied. Processes in some form are indeed required for these things to happen. However, they should not be used blindly. Understanding various processes, and the advantages and disadvantages of each, enables a team to selectively choose from a library of processes as appropriate and to integrate them together as needed.

Individual team members and communication between team members should have priority over sticking to a formal process, and responding to change should have priority over following a plan [[Larman 2004](#)]. This does not mean processes and plans do not matter. It does mean ongoing input from individuals, communication between the team, and appropriate change are more important. Good processes will support these things rather than hinder them.

Of course, a primary goal of processes is to help identify and prioritize matters of most importance, providing focus and clarity to the team. Good processes also provide for easy and swift exceptions—all rules can be broken when appropriate. The specifics of what processes fit a project are best chosen based on experience. There are no rules or algorithm that state, which process is most appropriate for a specific project although this book does provide some general guidelines. If it is not clear what specific processes to start with, then just pick some and go with them. Just like iterating upon the project, you will iterate upon the processes, improving upon them as you use and learn from them. Do not become attached to the processes and be willing to throw them out if and when they are not working.

30.5 Teams

VR by its nature is cross-disciplinary and communication between teammates is essential for VR development. Team communication is about externalizing one's work [[Gothelf and Seiden 2013](#)] to others through spoken conversation, sketches, whiteboards, foamcore boards, artifact walls, printouts, sticky notes, and of course prototype demos. All but spoken conversation helps to equalize team member input, regardless of whether someone is loud or quiet. It is very important to keep such forms of structural communication informal and raw as long as possible. More formal documentation should trail, not lead. Such communication should be easily modifiable with a way to easily determine what has been updated. Keep teams small to maximize communication. Individuals will likely have multiple roles. If the project is being developed as part of a larger organization, break groups into smaller teams.

A team effort often (but not always!) results in bigger and better results than an individual due to input from a range of opinions/viewpoints. However, design should not be done by committee. Collaborative design should be led by a single authoritative director (Section [20.3](#)) who listens well but acts as a final decision maker on high-level decisions (the one exception to this is for teams of only two individuals, similar to pair programming). This director needs to be completely trusted both by the team working with him as well as by those on the business/finance side. The director should clearly understand the conceptual design of the entire project. If it is not possible for the director to understand how everything is connected, then the design is overcomplicated and should be simplified.

The entire team should be on board with the overarching design philosophy. Not only should individuals be aware of what other team members are working on but also each individual should be actively involved with other's work to some degree. That is, everyone should co-create, not just critique. The entire team should work toward getting the design right through prototypes and feedback from users before making final decisions on polishing what has been built. This also prevents individuals from becoming attached to their creations, which results in less willingness to give up those creations.

Some team members might be subject-matter experts but have little experience with VR so they don't know what is possible or the limitation of today's technology. Other team members may be VR experts but have little understanding of the problem domain. Those with a traditional design background should understand challenges of VR implementation. Those with a programming background should be open to learning from other team members, external suggestions from those experienced in VR, other disciplines, and user feedback. If any individual on a team is close minded to ideas that are not his own and/or other disciplines, then that individual is going to be a hindrance to creating compelling VR experiences (much more so than any other discipline due to the cross-disciplinary nature of VR). Warn the team member to change his attitude, and if he is unwilling to do so then remove him from the project as soon as possible as if cancer has invaded the team (because like cancer, such attitudes can kill a VR project).

The Define Stage

The **Define Stage** of a project is the beginning but also continues until the end as more is discovered from the Make and Learn Stages. All parts of the Define Stage should be described from the client's or user's point of view and be able to be understood by anyone. That way multiple perspectives can be taken into account.

Be careful of “analysis paralysis”—trying to have every single detail figured out before pulling the trigger. Spending too much time at the beginning on the Define Stage can be detrimental to the project because what is defined may not be possible with today’s technology or may not ultimately be what customers want. In many cases, it is best to start with only defining a general overview of the project rather than getting into too much detail. In some cases, developers (e.g., solo indie developers) might even choose a “fire then aim” strategy of first starting with the Make Stage. In other cases, analysis and definition can be more important (e.g., command and control applications where the VR system must be integrated with existing systems and procedures). Knowing when to start implementing is largely a matter of experience and judgment of knowing when is enough. If you are not sure if you are ready to move on to the Make Stage, then it is better to err on moving forward. The Define Stage will always be waiting for your return to fill in more details as necessary.

This chapter discusses several Define Stage concepts. The concepts are listed in approximate order they might occur, but like everything else in iterative design, the order is best chosen based on project needs. Not all projects will use all of the concepts, although the more concepts that are eventually used and refined, the more likely the success of the project. Note for all of these concepts, it is not just important to define *what* the project is, but it is important to state *why* those decisions were made. By doing so, the team can better justify keeping, changing, or removing each element on future iterations.

31.1 The Vision

Pyramids, cathedrals, and rockets exist not because of geometry, theory of structures, or thermodynamics, but because they were first pictures—literally visions—in the minds of those who conceived them .

—Eugene Ferguson, *Engineering and the Mind's Eye* [[1994](#)]

Like any great engineering achievement, a VR project must be started by someone somewhere. What is the overall vision? What is trying to be achieved? Why is it being done? This is the chance to make it all up. With unlimited resources, what could be achieved?

The early vision is largely speculation. There is no way to know the final solution or even what all the concerns are until the project is completed. How can a project be defined when there are so many unknowns? Guess! Articulated guesses beat unspoken or vague models [[Brooks 2010](#)]. Guessing forces the team to think about the exposed assumptions and to think very carefully about the design. Guessing is the beginning of an iterative journey that will better define goals as more is learned. Both the formulation of the problem and its solution will evolve together, with interchange of information between the two [[Dorst and Cross 2001](#)].

Intelligent and informed speculation is based on the real world and its people. Refined speculation begins by talking with others as discussed in the next section.

31.2 Questions

At first thought, it seems having such creative freedom to make it all up would be easy to do. However, this is often the stage of the project that can be the most challenging. In fact, [Brooks \[2010 \]](#) states “the hardest part of design is deciding what to design.” Fortunately, VR projects do not exist in isolation. Understanding the background and context of how the project will fit in with the larger picture helps to define the project.

The most important time to get feedback from others is during the conceptualization of the idea. Innovative ideas, ones that create new interactions, experiences, and significantly better results, come about by reconsidering the goals and focusing on what people really want. The best way to determine what people really want is accomplished by talking with people and asking lots of questions. However, it's not as simple as asking what they want. Often, what people think they want is not what they actually want. As a consultant, the author has found one of the greatest services to provide is to help clients discover what they want. Don't expect people to straight out tell you what they want and do not expect your assumptions of what they want to be correct. Asking the right questions is essential.

The most important questions to ask at this stage begin with the word “why.” The statement “People don’t want to buy a drill, they want to buy a hole” does not go deep enough into what the desired goal actually is. Why does a client want to buy a hole? She wants to install bookshelves. Why does she want to install bookshelves? She wants a place to conveniently store books. Why does she want to conveniently store books? Perhaps she wants to store information in an organized manner so she can more conveniently access that information and she is frustrated with her existing bookshelf. The creative solution may be something completely different than a bookshelf. Similarly, consumers do not want to buy HMDs. They might want to effectively train for specific tasks, or they may want to bring more customers into their place of business. A VR demo might persuade potential customers to physically travel to the store to purchase merchandise that is largely unrelated to VR technology.

In addition to continuously asking “why” questions, here are some examples of useful high-level questions to understand background and context.

- What is the inspiration/motivation? Is it meant to replace some existing system (a real-world system or an older VR system)? Or is it meant to

create an entirely new experience?

- Who are the decision makers? Who are the principal stakeholders and intended beneficiaries?
- Where will the system be deployed? Will it be in a corporate office, at a theme park, or in people's homes?
- Is the VR project part of a larger vision? Is the project a brand-new initiative, or is it being fit into an existing project? How is the VR project related to other projects? Even if not directly a part of a larger vision or project, the relationship between them may still be important.
- Does the project play a minor or major role in the organization? What impact will it have on the organization?
- What are the characteristics of the cultural, social, and politic aspects of the target organization or community? Military settings are very different from entertainment settings.
- Who will the project affect other than the users?
- How many users will be using the system? How many locations?
- What is the planned life span once deployed? Is the expectation for this to be a single project or will there be further work (assuming success) after completion?
- What are the desired outcomes? How will success be measured?
- What are the resources and their limitations? What is the budget range and what is the time frame of when the project must be completed?

More detailed questions should also be asked as one begins to better understand the context. Talking with subject-matter experts is essential, especially for non-entertainment applications. One qualified opinion from a subject-matter expert is often more valuable than 100 uninformed opinions (although a wider uninformed audience can be useful as well since they may be your future customers). Respect experts' time by preparing specific questions that are relevant to their expertise.

In addition to asking questions, it is helpful to see for yourself. Physically travel to the site to meet with insiders so they can explain what will likely be an alien world for you. Ask various individuals there whether they are employees or customers, what they are doing, why they are doing it, and how it might be done better. Carefully observe and take notes about people's actions in addition to their words—their actions may indeed be more informative. If you can put yourself in their shoes, then you will understand. Ask questions not only to others but also ask questions to yourself. How do they currently perform tasks? If the goal is to better educate people about a topic, then how do people currently learn about that topic? What barriers do you see that are keeping them from learning more effectively?

31.3 Assessment and Feasibility

As one learns more about the goal of the project, it is useful to step back and assess if VR is the right solution. Whereas VR has the potential to cross all disciplines and be useful in a wide variety of applications, that does not mean VR is the ideal solution to every problem. VR is certainly not the ideal tool for every problem today as the technology is still far from being perfected.

Example questions in assessing and therefore determining the feasibility of a VR project include the following.

- What results must be achieved for the project to be considered a success?
- What functionality must be supported? Can non-VR technology support it better than today's VR technology?
- Will a VR solution fit within the real-world context such as the environment, organization, and culture?
- What are the competitive advantages to existing and/or non-VR solutions? What are the alternatives and the trade-offs?
- What is the expected profit, cost savings, and/or other added benefits?
- What are the hidden costs outside of project development? Examples include shipping or deployment costs, user-training costs, marketing and sales, and maintenance.
- Is the project possible given the budget, time, and other resources?

The answers to these questions are rarely binary. Often, a mix of the real world and VR is optimal (e.g., traditional education mixed with practice in a simulator).

31.4 High-Level Design Considerations

The user experience is the end of a chain of design considerations, and the context in which different design options are considered is important. It is useful to think about design from three different points of view [[Bolas 1989](#), [1992](#)].

Design with virtual environments focuses on the use of VR to help solve an existing problem or to create a new invention. Carefully

considering application needs can drive the design when VR is used as a design tool itself, for example, scientific data visualization or automotive design. If done well, VR can amplify human intelligence during the investigation and solving of real-world problems (e.g., enabling easier matching and recognition of complex patterns). With this approach, VR systems should complement, not replace, other tools and forms of media in order to maximize insight.

Design for virtual environments focuses on improving the hardware and software of VR systems themselves. With this approach, the designer should carefully consider the technology affordances used to provide the experience. For example, considering input device characteristics and classes (Chapter [27](#)) can help the designer choose, modify, or upgrade specific hardware. Considering different interaction patterns and techniques (Chapter [28](#)) can help to improve existing implementations and to develop new interaction metaphors.

Design of virtual environments focuses on the creation of completely synthetic environments—the virtual worlds. This approach works well for artistic mediums and entertainment applications. Part [IV](#) focuses on creation of content.

31.5 Objectives

Objectives are high-level formalized goals and expected outcomes that focus on benefits and business outcomes over features. Focusing on benefits/outcomes enables the team to work toward the vision and solve problems instead of implementing features that users may or may not care about. Features are usually more preferred by engineers as they are easier to implement but may not be what users care about. Yet engineers also do not like to be micromanaged. Thus it is best to let the engineers do what they do best—solve problems through the use of features they come up with that help to achieve the objectives. The team will gain insight into the value of features as they are being built and tested. If a feature ends up not moving the project toward the objectives, it can be changed, removed, or replaced.

Benefits that objectives describe often deal with time and/or cost savings, revenue generation, lower safety risks, increase in user productivity, etc. Note that objectives are different than requirements described in Section [31.15](#). Requirements are more specific and often more technical (e.g., latency and tracking requirements).

Quality objectives include what is described by the acronym **SMART**—Specific, Measurable, Attainable, Relevant, and Time-bound:

Specific. The objective should be clear and unambiguous; state exactly what is expected. The objective might also include specifically why it is important.

Measurable. The objective should be concrete so that progress toward the goal and ultimately success can be determined in an objective manner. This is essential for the Learn Stage (Chapter [33](#)) where measurement and testing occurs.

Achievable. The objective should somehow be feasible. The description of how is not important here, just the belief that it can be achieved in some manner.

Relevant. The objective should matter. This does not necessarily directly affect the end result; the objective might support other objectives that do affect the end result.

Time-bound. An objective should state a date of when it will be completed.

An example of an objective is “The VR training system will be deployed on January 1, 2016, and will result in a 30% increase in productivity as defined in Section X after three sessions of training.” The 30% productivity goal is a number that can be worked toward before delivery of the system through user testing.

A long list of objectives may be quite aggressive. Rarely can multiple aggressive goals be achieved in the first delivery nor should they. Prioritize objectives for the first delivery and save other objectives for later delivery. It is better to deliver 50% of the highest-priority objectives than to achieve 50% success on each individual objective.

31.6 Key Players

An important early step is to identify and enroll key players. A **key player** is someone who is essential to the success of the project. Key players might be stakeholders, partners, the client or sponsor, government agencies, customers, subject-matter experts, end users, marketing experts, usability experts, consultants, business developers, engineers, etc. It is vital that the core key players are in agreement with the vision, really care about the project, and are committed to achieving success. Identifying and enrolling the *right* key players can be a huge undertaking in itself. This might include pitching to investors, writing business proposals, assessing client needs, asking for referrals, recruiting others to join the team, etc.

Embracing key players often takes one of two paths.

Enroll others in the vision. Find key players who are inspired by the vision and believe it can be accomplished, and move on when the individuals are not right for the project. Don’t waste time on those who are not a good fit. This path is most common when creating an entertainment experience or when recruiting others to join the team. Note that just because an individual or small group originally creates the vision does not mean they should not seek input from enrolled key players.

Help to solve a need. Identify pain points and work toward finding a way to solve problems and satiate existing desires. This path is common in industry, for example to help reduce costs or find better ways to train employees. Demonstrating how you can help individuals (especially the decision makers) with their specific problems makes all the difference in converting skeptics to key players.

In both cases, be prepared to do a lot of pitching and proposing. Realize in advance that many people may not have an interest no matter how good of a fit you think they might be. Listen to what they have to say (even if they do not want to be involved they may have valuable input) and then move on. Do not waste time chasing such individuals.

31.7 Time and Costs

Accurate estimates are extremely difficult, even for those with extensive project experience. Early estimates should not be expected to be accurate. Because of this, it often makes sense to separate out initial assessment and feasibility from implementation when negotiating contracts. Contracts might also be written with milestones, with the client making additional payment as milestones are achieved and with the expectation that the contract might be renegotiated at each milestone as more is learned and estimates are refined. If a project is late, it is not a good idea to add additional team members as Brooks' Law states “adding manpower to a late software project makes it later” [[Brooks 1995](#)]. Because time, budget, and quality are often non-negotiable, the scope of the project may need to be scaled back when a project is taking longer than expected.

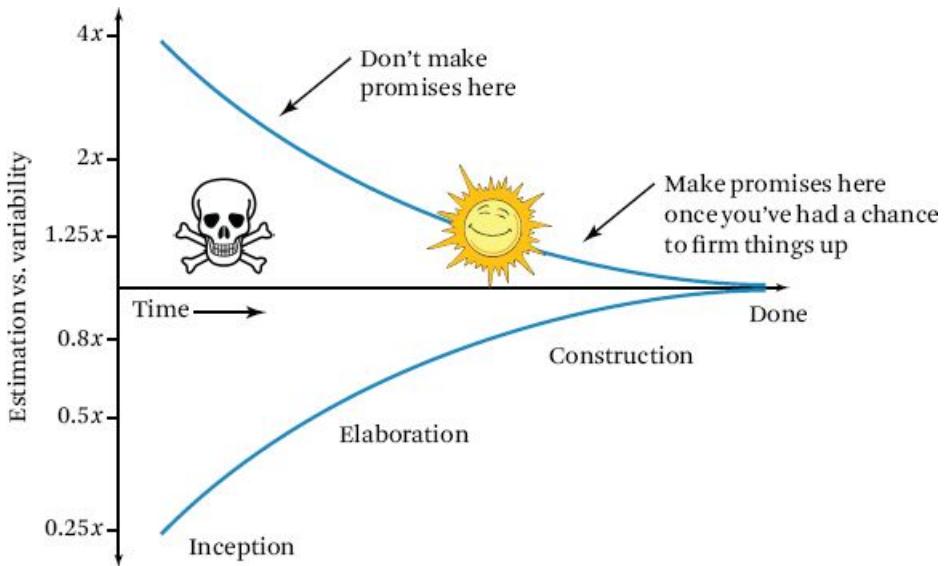


Figure 31.1 Estimating development is often incorrect at the beginning. It is not uncommon for estimates to be off by a factor of four.
(Based on [Rasmusson \[2010 \]](#))

Planning poker is a game of estimating development effort [[Cohn 2005](#)]. Although it is best done when planning for implementation, it can also be done at a higher level early in the project with the understanding the estimates will not be as refined since the project is not yet well defined. The team sits at a table, a specific task is stated, and all players write down an estimate of the time they think it will take to complete the task. After secretly writing the estimated time on the card, players face the card down on the table and all cards are flipped right side up. If all estimates are in the same general range, then the average is taken. Planning poker is not a voting system—three short estimates do not outvote one longer estimate. If there are outliers, then those outliers are discussed. It may be the case that the person who wrote the outlier considered something other team members had not.

Initial development estimates are often off by as much as 400% [[Rasmusson 2010](#)], and developers tend to overestimate their abilities even when they know they tend to overestimate. Estimates improve as implementation continues and more is learned about the project (Figure 31.1). One way to determine a better estimate is to track the ratio of how long tasks are taking relative to the initial estimate. Future task estimations can be improved by multiplying the tracked ratio by the initial task estimates. For example, if tasks estimated to take 5 days actually take 15 days, then the ratio is 3. Future tasks estimated to take 5 days will also likely take 15 days, and tasks estimated to take 10 days will likely take 30 days.

31.8 Risks

The intention of identifying risk is to bring awareness of any danger that might affect the project so that appropriate action can be taken to mitigate that risk. After brainstorming with the team on all possible risks, divide the listed risks into two groups: (1) risk the team has influence or control over and (2) risk the team has no influence or control over. Understanding risk that is outside the team’s control is important for determining if the project is worth pursuing. Once such risk is understood, the leadership must decide to fully commit (or end the project). Then the team can focus on minimizing controllable risk.

Risk is especially important to be aware of for VR projects because there are so many unknowns and technology is improving so quickly. Risk increases exponentially as project size and duration increases (Figure 31.2). A two-year project may quickly be put out of business by a new start-up that didn’t even exist at the beginning of the project. To minimize risk, projects should be kept as short as possible. This does not mean larger projects should not be undertaken. It does mean to break larger projects down into smaller subprojects. After successful completion of the smaller project, then new contracts and extensions can be made.

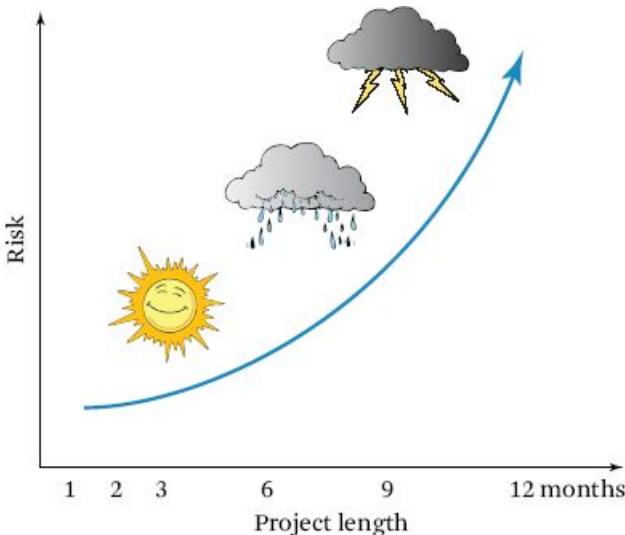


Figure 31.2 The risk of a VR project failure increases exponentially over time. (Based on [Rasmusson \[2010\]](#))

31.9 Assumptions

Assumptions are high-level declarations of what one or more members of the team believe to be true [[Gothelf and Seiden 2013](#)]. Everything created by humans begins with assumptions, whether the creators consciously realize those assumptions exist or not. VR projects are no different.

Explicitly looking for and declaring assumptions allows team members to begin a project from a common starting point. Get the team together and carefully go through the project description and dissect its assumptions. The team will likely find that what they assume to be shared assumptions are actually different assumptions. Be bold and precise in listing assumptions, even when not sure. Wrong explicit assumptions are much better than vague assumptions. Wrong assumptions can be tested; vague assumptions cannot [[Brooks 2010](#)].

Examples of assumptions are who the targeted users are, user needs/desires, what the greatest challenges are, what the important VR elements of the experience are, etc. The list of assumptions should be quite long. Unfortunately, it is not feasible to test all assumptions, so prioritize the assumptions with the riskiest assumptions listed first.

31.10 Project Constraints

All real-world and virtual-world projects have **constraints** that must be designed around. Project constraints can come from a wide range of sources. For example, a constraint might be limited hardware functionality (e.g., tracking may only be reliable within some volume), a maximum amount of accepted latency (e.g., 30 ms is low enough to do prediction fairly well), or budget/time constraints (limiting the complexity of implementation). A single individual should be responsible for tracking/controlling constraints, with transparency to the entire team.

Project constraints at first seem harmful to what can be done. Whereas it is true that for building a mediocre experience, it is easier to do a general-purpose design with few constraints (e.g., put the user in a world and have him fly around) than to do a special-purpose design [[Brooks 2010](#)]. However, for a high-quality experience, it is easier to do a special-purpose design because establishment of the problem has already started—it is more a task of discovering constraints. Discovering existing constraints provides clarity and focus resulting in higher-quality experiences that can be more effectively built. Constraints also provide a basis for feedback (i.e., constraints can help focus questions) and challenge the team, stimulating fresh creations.

31.10.1 Types of Project Constraints

Explicitly listing project constraints can be very helpful in narrowing the design space. When listing constraints, consider the following [[Brooks 2010](#)].

Real constraints are true impediments that cannot be changed. Examples include physical barriers, rules outside of the team's control, the number of buttons on a specific controller, and the tracked volume of a space.

Resource constraints are real-world limitations of supply. For any project, there is always at least one scarce resource that must be rationed or budgeted. Examples are dollars, deadlines, minimum hardware specs users are expected to own, end-to-end system latency, and battery life for mobile VR. After listing all possible resource constraints, prioritize them, as there will always be more to do. Some of the low-priority constraints might be moved to misperceived constraints (described below). Track resource constraints with transparency to the entire team, and have a single individual control firmly.

Obsolete constraints were once real constraints but are no longer valid. This can be due to a change in rules or due to the never-ending improvement of technology. Examples include better tracking, faster CPUs, more buttons on a controller, and the number of reliably detected gestures.

Misperceived constraints are perceived to be real but are not. They are often so embedded in our lives that we don't even realize we treat them

as constraints. VR is full of misperceived constraints because many of the rules of the real world do not apply to VR. The designer can choose to allow users to walk through walls or reach a greater distance than what can be done in the real world.

Indirect constraints are the side effects of real constraints. These constraints are not necessarily real constraints as indirect constraints are based on assumptions of how something is achieved. Indirect constraints are important to differentiate because if they are seen as real constraints, then solutions are cut off. The number of polygons for a scene is an indirect constraint. The true constraint is the frame rate as rendering polygons can be optimized.

Intentional artificial constraints are added by the designer to narrow the design and to enhance the user experience. Adding constraints to a VR experience can significantly improve interaction. Section [25.2.3](#) discusses the power of adding interaction constraints.

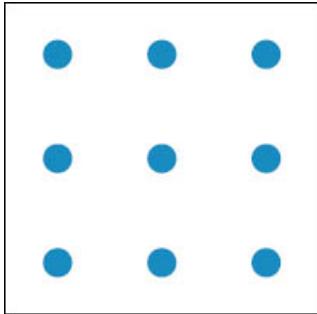


Figure 31.3 A design puzzle. Draw over all nine dots with four straight lines connected end-to-end. Figure [31.5](#) shows a solution.

31.10.2 A Constraints Puzzle

Take a look at Figure [31.3](#) and try to draw over all nine dots with no more than four straight lines connected end-to-end.

If you can't find a solution, then go through an exercise of explicitly listing the constraints in the points listed above. You may find a solution once you explicitly list all of the different types of constraints. If not, go back and think more carefully about any misperceived constraints. If you are able to do this (and you haven't seen the solution before), then congratulations! There are very few people who are able to solve this without first seeing one of the solutions. Did explicitly listing out the constraints (specifically misperceived constraints) help? A reason why most people can't solve this puzzle is because most people don't take the time to explicitly think about constraints.

Now try to draw over all nine dots with three straight lines connected end-to-end. (Hint: Think even further outside the box.) Figure [31.7](#) shows a solution.

Now try to draw over all nine dots with three straight lines, but this time the lines cannot extend beyond the box boundaries. If you think that is impossible, then go back to your list of constraints and update that which has changed. Figure [31.8](#) shows a solution.

Now try to figure out how you could draw over all nine dots with a single straight line. Note that you may be constrained by the tool you are using to solve the problem instead of a constraint inherent to the problem. List any constraints on tools that people normally use to solve such puzzles and then try again. Figure [34.1](#) shows a solution.

 Name	<ul style="list-style-type: none">• Job• Experience• Activities• Attitude• Competencies• Age
<ul style="list-style-type: none">• Problems• Pain points• Needs• Concerns• Fears• Desires	<ul style="list-style-type: none">• Knowledge of VR• Dream VR system• Vision of VR• VR hardware access• Budget for VR• Activities that fit VR

Figure 31.4 Persona template.

31.11 Personas

Attempting to design for the entire population is difficult as users vary widely in their ability to use VR systems, making generalizations difficult [[Wingrave et al. 2005](#), [Wingrave and LaViola 2010](#)]. As a result, targeting more specific users makes the design easier. Personas help do this.

Personas are models of the people who will be using the VR application. Getting clear on users by explicitly defining personas helps to prevent the

design from being driven by design/engineering convenience where users are expected to adapt to the system. Not only should the application be defined and designed around these personas, but representative users can be targeted to collect feedback for the Learn Stage (Chapter [33](#)).

Divide a notecard into four quadrants. Start by providing a simple sketch and name in the upper left quadrant. Add a basic description of the person in the upper right quadrant. Add different types of challenges the person has in the lower left quadrant. Then add information about how the person relates to VR in the lower right quadrant. Figure [31.4](#) shows an example template that might be used. Do this for 3–4 characters that represent the full range of your targeted users.

During later iteration cycles, personas should be validated and modified as more is learned about real users. If personas are especially important (e.g., for therapy applications), then data should be collected with interviews (Section [33.3.3](#)) and/or questionnaires (Section [33.3.4](#)).

31.12 User Stories

User stories emerged from agile development methods and are short concepts or descriptions of features customers would like to see [[Rasmussen 2010](#)]. They are often written on small index cards to remind the designer not to go into too much detail. We don't know yet if we're actually going to need or implement that feature. They are written from the user's point of view and should be written with the client and multiple team members.

User stories are written in the form “As a <type of user> I want <some goal> so that <some reason>.” This defines who, what, and why. User stories often turn into requirements and constraints. Break big story items into smaller, manageable stories. Go through the list and clean it up, remove duplicates, group like items together, and turn it into an action item or to-do list.

Ideally, user stories fit with the acronym **INVEST** created by Bill Wake [[Wake 2003](#)]: Independent, Negotiable, Valuable, Estimable, Small, and Testable.

Independent user stories are easier to develop for when they do not overlap with other user stories. This means the stories can be implemented in any order and the creation or modification of one story does not affect another story. This can be a challenge for VR because interactions often affect other interactions.

Negotiable user stories are high-level descriptions that are modifiable. VR components must be modifiable because what we think will work well does not always work well, and what works can only be determined by learning from users who actually try the implementations.

Valuable user stories are specifically focused on giving value to the user and in a language that is understandable by anyone. A brilliantly implemented framework has little value if it does not offer value to users.

Estimable user stories are understandable so that implementation can be estimated. If the story is too complex, then it should be broken down into easier-to-understand estimable stories.

Small user stories enable fast implementation. If a story is too big to be quickly implemented, it should be broken down into smaller stories.

Testable user stories are written so that a simple experiment can clearly determine if the implemented story is working or not.

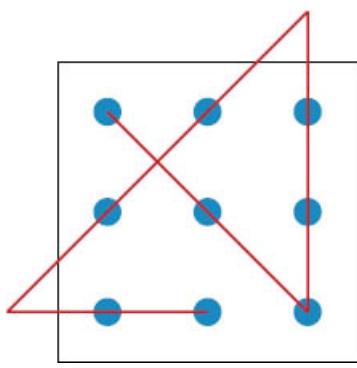


Figure 31.5 A solution to Figure 31.3. By realizing the surrounding box is a misperceived constraint, one can think outside of the box. Now try doing the same with three straight lines (solution shown in Figure [31.7](#)).

31.13 Storyboards

Storyboards are early visual forms of an experience. They are especially good at getting the point across to those only loosely involved with the project. Figure [31.6](#) shows an example of a storyboard.

Traditional software storyboards typically end up being screenshots more than interactions which are closer to a mock-up than an experience. Storyboards are more useful for VR because the user can be shown directly interacting with objects—experiences and interactions can be quickly sketched and conveyed without having to worry about screen layout details.

Storyboarding can, however, be more difficult for VR than for video storyboards because of the non-linear nature of VR. In many other cases there might be multiple connections between storyboard frames.

31.14 Scope

When defining the project, explicitly stating what will not be done can be just as helpful as stating what will be done [Rasmussen 2010]. This will set clear expectations for all involved and allow the team to focus on what is important (Table 31.1). Items defined to be in scope are what are nonnegotiable and must be completed. The team should explicitly not be concerned with items that are out of scope; they might or might not be included in later releases.

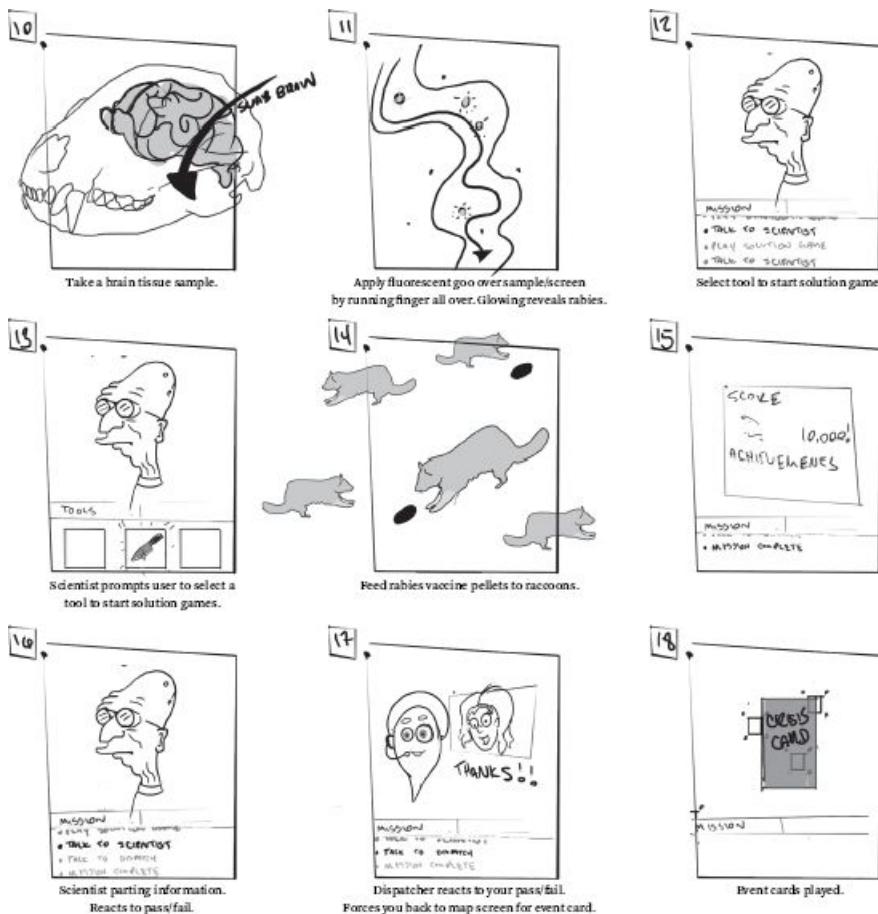


Figure 31.6 Part of an example storyboard for an educational game. (Courtesy of Geomedia)

Table 31.1 Defining what is in and out of scope makes it clear to all parties involved what will and will not be done.

In Scope

Bimanual tracking

Ray selection

Embodied avatar

Hand-held panel

Out of Scope

Tracking for standing and walking

Networked environment

Voice recognition

Fine-tuning of colors

Unresolved

Travel technique

What widgets/tools to include on the panel

31.15 Requirements

Requirements are statements that convey the expectations of the client and/or other key players, such as descriptions of features, capabilities, and quality. Each requirement is a single thing the application should do. Requirements come from other parts of the Define Stage, such as assumptions, constraints, and user stories. Like other parts of the Define Stage, the client and/or other key players should be actively involved in defining requirements. Requirements do not specify implementation.

Requirements aid with the following.

- Requirements communicate mutual understanding between the client and contractor, as well as any other parties involved. Requirements should be written in language that can be easily understood by everyone involved with the project.
- Requirements help to organize, understand, and decompose goals of the project into more explicit descriptions.
- Requirements act as input to the design specification (Section [32.2](#)).
- Requirements are used as a source for determining if a system is working as expected (Chapter [33](#)).

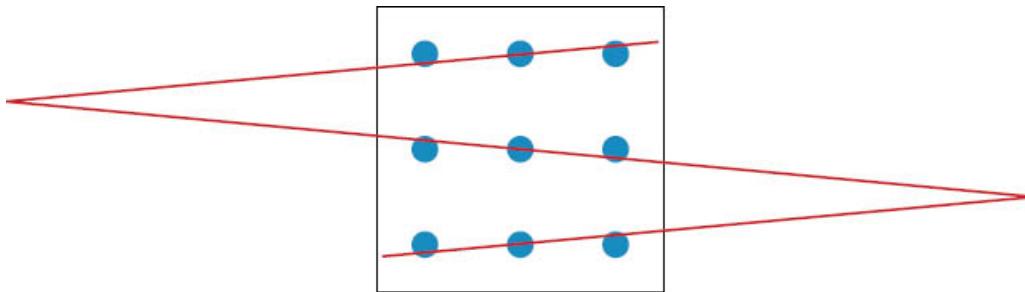


Figure 31.7 A solution to Figure 31.3 using only three lines. Now try to draw over all nine dots with three straight lines, but this time the lines cannot extend beyond the box boundaries (solution shown in Figure 31.8).

A requirements document should be as minimal as possible without compromising clarity. Long requirements documentation ends up rarely being read entirely by anyone other than the person writing the document, and that might not even occur due to different groups writing different sections. Not reading the document in its entirety can be a major problem due to key players not being clear on understanding the entire project.

Each individual requirement should be complete, verifiable, and concise, yet with room for innovation and change. Requirements may be changed and clarified throughout the project as initially some requirements may be ill-defined and others might become discovered as the project evolves. Not proceeding until all requirements are written down would mean never starting. The Agile Manifesto states, “Any attempt to formulate all possible requirements at the start of a project will fail and would cause considerable delays” [[Pahl et al. 2007](#)]. Work with the client and/or other key players when changes are necessary.

31.15.1 Quality Requirements

Quality requirements (also known as non-functional requirements) define the overall qualities or attributes of a system or application. Quality requirements can be thought of as a definition for quality of service or quality control. Quality requirements may place restrictions or constraints on the solution, such as usability, aesthetics, security, reliability, and maintainability. Common quality requirements include system requirements, task performance requirements, and usability requirements as described below.

System requirements describe parts of the system that are independent of the user, such as accuracy, precision, reliability, latency, and rendering time.

Accuracy is the quality or state of being correct (i.e., closeness to the truth). A system with consistent systematic error has bias and low accuracy. For example, distortion in a tracker system has low accuracy because if a user moves in one direction, the tracked point might unexpectedly and wrongly move the tracked point in a different direction.

Precision is the reproducibility and repeatability of getting the same result. A tracked tool that shakes or jitters in the hand due to bad tracking has low precision and results in difficulty selecting small objects or menu items.

Reliability is the extent to which some part of the system works consistently (also see Section [27.1.9](#)). It is often measured as a hit rate or failure rate. For example, a tracking system might be required to lose tracking no more than once per 10 minutes.

Latency is the time a system takes to respond to a user’s action (Chapter [15](#)).

Rendering time is the amount of time it takes for the system to render a single frame. A requirement might be that rendering time not exceed 15 ms for any single frame to ensure frames are not repeated.

Task performance requirements focus on the effectiveness of interactions. **Task performance** is the measured effectiveness of a task as performed by users, such as time to completion, performance accuracy, performance precision, and training transfer.

Time to completion is how fast a task can be completed. This is typically measured as the average time it takes for users to complete the task.

Performance accuracy is the resulting correctness of a user's intention. A manipulation or navigation task has decreased accuracy as the distance from the desired position or path increases.

Performance precision is the consistency of control that users are able to maintain. It can be thought of as the fine-grained control afforded by the technique [[McMahan et al. 2014](#)]. Travel is precise if it enables travel along narrow paths.

Training transfer is how effectively knowledge and skills obtained through the application transfer to the real world. Training transfer is influenced by many factors such as how well the stimuli and interactions match the real world [[Bliss et al. 2014](#)].

Usability requirements describe the quality of the application in terms of convenient and practicable use. Common usability requirements include ease of learning, ease of use, and comfort [[McMahan et al. 2014](#)].

Ease of learning is the ease with which a novice user can comprehend and begin to use an application or interaction technique. Ease of learning is often measured by the time it takes a novice to reach some level of performance, or by characterizing performance gains as use increases.

Ease of use is the simplicity of an application or technique from the user's perspective and relates to the amount of mental workload induced upon the user of the technique. Ease of use is typically measured through subjective self-reports, but measures of mental workload are also used.

Comfort is a state of physical ease and freedom from sickness, fatigue, and pain. Different input devices and interaction techniques can affect user comfort, and making small changes can have a big impact. User comfort is especially important for experiences that are longer than a few minutes. User comfort is typically self-reported.

31.15.2 Functional Requirements

A **functional requirement** specifies what some part of the system does or what the user can do and often includes a set of inputs, behaviors, and outputs. Functional requirements are typically more dependent on the specifics of the project than quality requirements. Detailed functional requirements often evolve out of task analysis (Section [32.1](#)) and use cases (Section [32.2.3](#)). Examples of functional requirements include the following.

- Computer-controlled characters will navigate to the designated position via the shortest path, where all portions of the path will have (1) a slope between -20° and 20° and (2) a width greater than 0.8 m.
- All objects marked as selectable will be able to be directly selected by the user via intersecting the hand geometry with the object and pushing the grab button.
- Users will not be able to navigate closer to geometry than 0.2 m. in the horizontal when that geometry is at a height between 0.2 m and 2.4 m above the travel plane.

31.15.3 Universal VR Requirements

Listed below are some requirements that should be considered for all fully immersive VR applications. These requirements should be achieved and maintained from the start of development as optimization can be a challenge late in the project (e.g., reducing scene complexity can require new art assets). The moment these requirements are not being met, the software should automatically and immediately recognize the problem and attempt to resolve the problem by reducing the scene and/or rendering complexity (e.g., transition to simpler lighting algorithms). The less complex settings should then be maintained instead of alternating back and forth as variable latency can cause sickness and prevent adaptation (Section [18.1](#)). If the requirements still cannot be met, then the scene should **fade out** and/or transition to a simpler scene or video see-through if available (assuming video see-through can achieve the requirements). The fading out of the scene during development will be frustrating to developers but will ensure quality as they will be motivated to fix the problem.

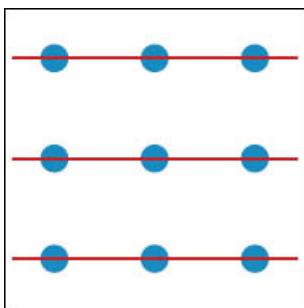


Figure 31.8 A solution to Figure [31.3](#) but with an obsolete constraint removed. The new problem statement did not say the lines had to be connected. Now try to draw over all nine dots with a single line (solution shown in Figure [34.1](#)).

End-to-end system delay will not fall below 30 ms. The scene will automatically fade out if this requirement cannot be maintained.

Minimum frame rate will be the refresh rate of the HMD (60–120 Hz depending on the HMD). The scene will automatically fade out if this requirement cannot be maintained.

The screen will fade out if and when head tracking is lost. The scene will automatically fade out if this requirement cannot be maintained.

Any camera/viewpoint motion not controlled by the user will not contain accelerations for periods longer than one second. Such motions should only rarely occur, if ever.

Input devices will maintain 99.99% or better reliability. Anything less is frustrating to users. Loss of tracking every 10,000 readings at 100 Hz results in lost tracking every 10 seconds.

The Make Stage

The **Make Stage** is where the specific design and implementation occurs. This stage is arguably the most important part of creating VR experiences. The other stages would only be dreams and theory without the Make Stage whereas at least something, albeit perhaps not good, can be used and experienced with only the Make Stage. The Make Stage is also often where most of the work occurs. Fortunately, the implementation of VR experiences is in many ways similar to creating other software once the experience is properly defined and feedback is obtained. The same logic, algorithms, engineering practices, and design patterns are still used. One primary difference is that there is more emphasis on working with hardware. Makers should not develop VR applications without full access to the VR hardware the system is designed for because the VR experience is so tightly integrated with that specific hardware.

The Make Stage often consists of primarily using existing tools, hardware, and code where the focus is on gluing the different pieces together so that they work together in harmony. This is the case for most VR projects, as development tools such as Unity and the Unreal Engine have proven to be very effective for building a wide variety of VR worlds. In some cases, the project may require implementing from the ground up. Examples of when that might be the better option is when a true real-time system is required, in-house code with specialized functionality already exists, or specialized/optimized algorithms are required (e.g., volumetric rendering). However, the default should be to use existing frameworks and tools unless there is a good reason not to do so. Resistance to change is no excuse for not using more effective technology. Developers will find the learning curve of using modern tools quite small relative to the benefits.

This chapter discusses task analysis, design specification, system considerations, simulation, networked environments, prototyping, final production, and delivery.

32.1 Task Analysis

Task analysis is analysis of how users accomplish or will accomplish tasks, including both physical actions and cognitive processes. Task analysis is done via insight gained through understanding the user, learning how the user currently performs tasks, and determining what the user wants to do. Task analysis provides organization and structure for description of user activities, which make it easier to describe how the activities fit together.

Task analysis processes have become complex, fragmented, difficult to perform, and difficult to understand and use [[Crystal and Ellington 2004](#)]. The goal of task analysis should be to organize a set of tasks so they can be *easily and clearly* communicated and thought about in a systematic manner. If the result of the task analysis complicates understanding through extensive documentation, then the analysis has failed. Task analysis should be kept simple without requiring learning specialized processes or symbols. Part of the reason to do a task analysis is to communicate how interaction works, and anyone should be able to understand the result without requiring specialized training in reading specific forms of task analysis diagrams.

32.1.1 When to Do Task Analysis

The first iteration of task analysis is done when trying to understand how a task is or will be performed. When the goal for the VR application is to replicate actions in the real world (e.g., training applications), task analysis should start with the actual tasks performed in the real world in order to understand what tasks must be carried out, to document proper descriptions of activities, and to look for how to improve the process.

Applications built without doing a full task analysis may have low validity relative to the real world due to the task being simplified in the virtual environment [[Bliss et al. 2014](#)]. In other cases, task analysis might be used in creating a new magic VR interaction technique that is very different than any real-world task.

Task analysis helps to define the design specification (Section [32.2](#)) and to realize early the implications of redesign. Task analysis results in more than just straightforward understanding of simple tasks; it helps to create understanding of the relationship between different tasks, how information flows, and how users affect task sequence by their decisions. This understanding can help to prioritize tasks and to determine what can be automated without user intervention. Without task analysis, designers might be forced to guess or interpret desired functionality that often leads to poor design. Evaluation plans and validation criteria also depend on task analysis.

Task analysis can be used for many purposes. One use is to understand, modify, and create new interaction techniques. This can be done by decomposing the techniques into subtask components, which enables comparison of subtask components rather than comparing holistic techniques. One can then replace subtask components with other components (whether already used elsewhere or new methods) from other techniques.

Task analysis is also used at later stages of development to determine how the current implementation deviates from the design/intended solution and what the consequences are. Like all aspects of VR development, task analysis should be flexible and iterative, allowing for modification during all but the final stages of production.

32.1.2 How to Do Task Analysis

There are many ways to do task analysis. Here steps are generalized into finding representative users, task elicitation, organizing and structuring, reviewing with representative users, and iteration.

Find Representative Users

Although in practice task analysis is initially done by the team, it is important to observe and interview representative users. Those matching the personas discussed in Section [31.11](#) should be sought out to make sure the analysis represents activities of the user population. Use more than one person to ensure the analysis is representative of the entire user population. This can be a time-consuming process, so outsourcing someone to find such individuals allows the team to focus on what they are good at.

Task Elicitation

Task elicitation is gathering information in the form of interviews, questionnaires, observation, and documentation [[Gabbard 2014](#)].

Interviewing (Section [33.3.3](#)) is verbally talking with users, domain experts, and visionary representatives—this provides insight into what users need and expect. Questionnaires (Section [33.4](#)) are generally used to help evaluate interfaces that are already in use or have some operational component. Observation is watching an expert perform a real-world task or a VR user trying a prototype (this resembles formative usability evaluation—see Section [33.3.6](#)). Documentation review identifies task characteristics as derived from technical specifications, existing components, or previous legacy systems.

To collect information efficiently, one should plan/structure the process. Focus on activities that are currently most relevant and start with an activity that begins with a rough description of what to do. Carefully thinking about the different stages of the cycle of interaction (Section [25.4](#)) is a great place to start. Ask “how” questions to break the task into subtasks for more detail (but don’t get bogged down with details—know when enough is enough). Ask “why” questions to get higher-level task descriptions and context. “Why” questions and asking what happens before and what happens after can also help to obtain sequential information. Use graphical depictions with those being talked with to collect higher-quality information.

Organize and Structure

The most common form of task analysis is **hierarchical task analysis** where tasks are decomposed into smaller subtasks until a sufficient level of detail is reached. Each node below a task in the hierarchy addresses a single subtask. Each subtask can be thought of as a question that must be answered by the designer, and the set of subtasks is the set of possible answers for that question. This is a good place to start as it works well for understanding the details of how users perform actions. High-level task descriptions go on the top with those tasks broken down into lower-level task descriptions. The sequence of tasks is ordered from left to right. Don’t be overly concerned about getting this perfect or thinking there is a right way to do it—like everything in iterative design, you will iterate toward something better. The specifics may depend on the project and you can use your own style, such as user activities being represented by boxes and system activities represented by circles and relationships between boxes/circles represented by arrows.

Hierarchical task analysis has its limitations as the hierarchical pattern limits what can be described with tasks. In addition to hierarchical diagrams, tables, flow charts, hierarchical decomposition, state transition diagrams, and notes can be used where appropriate. Task analysis might also include defining affordances and the corresponding signifiers, constraints, feedback, and mappings (Section [25.2](#)). For activities that are performed multiple times, generalize the activity to a pattern and give it a name. Then that type of activity can be represented by a single node in different locations. If the task deviates from the general pattern, give it a secondary name and/or a note of specific differences.

Review with Users

Once data has been organized and structured, one should review with the users who information was elicited to verify understanding.

Iteration

Task analysis provides the basis for design in terms of what users need to be able to do. This analysis feeds into other steps of the Make Stage, and these Make Stage steps along with lessons acquired from the Learn Stage will feed back into refining the task analysis. Like everything else in iterative design, changes to the task analysis will occur. Document the latest changes so it is obvious what changed. Different colors convey such change well.

32.2 Design Specification

Design specification describes the details of how an application currently is or will be put together and how it works. Design specification is the between stage of defining the project and implementation, and is often closely intertwined or simultaneous with prototyping. In fact, in many cases the person doing the design is also doing much of the prototyping.

Specifying the design isn’t just conducted to satisfy that which was created in the Define Stage, but elicits previously unknown assumptions, constraints, requirements, and details of task analysis. **Early wanderings** are the exploration of radically different designs early in the process before converging toward and committing to the final solution [[Brooks 2010](#)]. As new things are discovered, the corresponding defining documentation should be updated as appropriate.

Some of the most common tools for building VR applications are described in this section: sketches, block diagrams, use cases, classes, and software design patterns.

32.2.1 Sketches

A **sketch** is a rapid freehand drawing that is not intended as a finished work but is a preliminary exploration of ideas. Sketching is an art and very different from computer-generated renderings. Bill Buxton discusses what characteristics make good sketches in his book *Sketching User Experiences* [Buxton 2007]:

Quick and timely. Too much effort should not be put into creating a sketch and it should be able to be quickly created on demand.

Inexpensive and disposable. A sketch should be able to be made without concern of cost or care if it is selected for refinement.

Plentiful. Sketches should not exist in isolation. Many sketches should be made of the same or similar concepts so that different ideas can be explored.

Distinct gestural style. The style should convey a sense of openness and freedom so that those viewing it don't feel like it is in a final form that would be difficult to change. An example is the way edges typically do not perfectly line up that distinguishes a sketch from a computer rendering that is tight and precise.

Minimal detail. Detail should be kept to a minimum so that a viewer quickly gets the concept that is trying to be conveyed. The sketch should not imply answers to questions that are not being asked. Going beyond good enough is a negative for sketching, not a positive.

Appropriate degree of refinement. Detail should match the level of certainty in the designer's mind.

Suggestive and explorative. Good sketches do not inform but rather suggest, and result in discussion more than presentation.

Ambiguity. Sketches should not specify everything and should be able to be interpreted differently, with viewers seeing relationships in new ways, even by the one doing the sketching.

Figure 32.1 shows a sketch drawn by Andrew Robinson of CCP Games with all of these characteristics from the early phases of the VR game EVE Valkyrie.

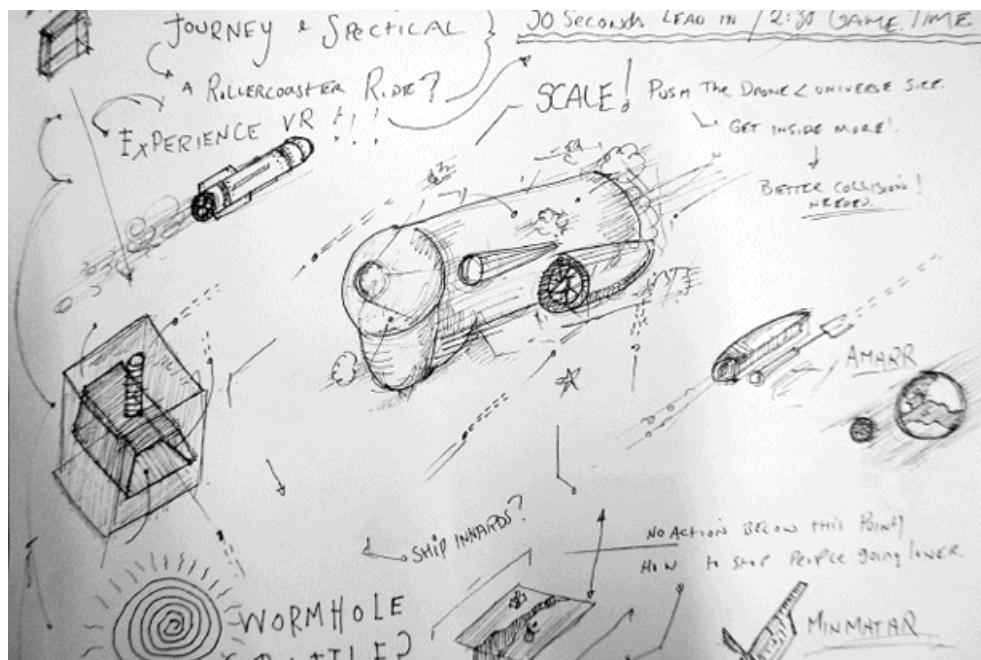


Figure 32.1 Example of an early sketch from the VR game EVE Valkyrie. (Courtesy of Andrew Robinson of CCP Games)

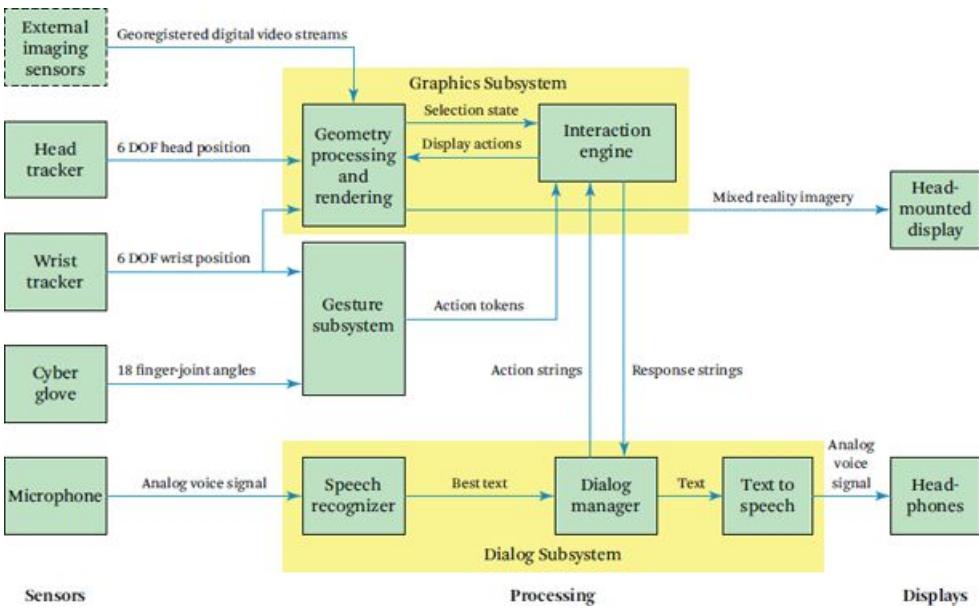


Figure 32.2 A block diagram for a multimodal VR system developed at HRL Laboratories. (Based on [Neely et al. \[2004\]](#))

32.2.2 Block Diagrams

Block diagrams are high-level diagrams showing interconnections between different system components. Boxes are used to denote components, and arrows connecting the components show the input and output between the components. Block diagrams are used to show the overall relationship of components without concern for the details. Figure 32.2 shows a block diagram for a multimodal VR system with various software and hardware components.

32.2.3 Use Cases

A **use case** is a set of steps that helps to define interactions between the user and the system in order to achieve a goal. Defining use cases helps developers identify, clarify, and organize interactions in a way that makes it easier to implement the interactions. As interaction complexity grows, the need for explicit use cases increases. Use cases often start from user stories (Section 31.12) but are more formal and detailed, enabling developers to get closer to implementation. Use cases can also result from task analysis (Section 32.1).

Use case scenarios are specific examples of interactions along a single path of a use case. Use cases have multiple scenarios, i.e., a use case is a collection of possible scenarios related to a particular goal.

There is no standard way of writing a use case. Some prefer visual diagrams whereas others prefer written text. Some form of the following should be included.

- Use case name
- Intended result
- Description
- Preconditions
- Primary scenario
- Alternate scenarios

Outlining the scenarios with hierarchical levels can be useful to provide both high-level and more detailed views. This also allows the filling in of details as more specific detail is needed. Colors or different font types can be used to see what needs to be implemented and what has already been implemented/tested, the differences between the primary path and alternative paths, or who is assigned to what steps.

Use cases do not need to exist in isolation. A use case might extend upon an already existing use. Or a large use case might be broken apart into multiple use cases if there are common steps that are used multiple times throughout the larger use case. These smaller use cases can also be included within other larger use cases where appropriate. This helps to make user interactions consistent across the entire experience. For example, a pointing selection technique might be a small use case that is part of a larger use case that requires many selections and manipulations. The same small pointing use case might also be used as part of an automated travel technique (select an object to travel toward and then the system somehow moves you to that location).

32.2.4 Classes

As the design specification evolves, it is necessary to get clearer on how the design will actually be implemented in code. This can be done by taking information from the previous steps and organizing that information into common structures and functionality into themes that can be described and implemented as classes. A **class** is a template for a set of data and methods. Data corresponds to nouns and properties from information collected in previous steps, and methods correspond to verbs and behaviors.

A program **object** is a specific instantiation of a class. A class exists by itself without having to be compiled into an executable program (e.g., classes exist even when the virtual environment does not yet exist) whereas an object is a representation of information organized in a manner defined by a class that is included in the executable program. An example of a program object is a perceivable virtual object in the virtual environment, such as a rock on the ground that can be picked up.

A real-world analogy of a class is a set of blueprints for a house. A blueprint is like a class that describes the house but is not a house itself. Construction workers instantiate the blueprints into a real house. The blueprints can be used to build an arbitrary number of houses. Likewise, classes define what can be instantiated into objects inside a virtual environment. An example of a class is a box class that can be instantiated as a specific box that exists in the virtual environment. The box class can be instantiated many times to create multiple box objects in the environment. The class might define what characteristics are possible, and the box object can take on any of those possible characteristics. For example, different box objects in the environment might have different colors.

Classes and objects do not have to represent physical things. They might also represent properties, behaviors, or interaction techniques. A simple example is a color class. Instantiated objects of the color class might be specific colors such as red, green, or blue. These color objects, along with other objects, could then be associated to each individual box to describe what the box looks like.

A **class diagram** describes classes, along with their properties and methods, and the relationship between classes. Class diagrams directly state what needs to be (or already is) implemented. Figure 32.3 shows an example of such a diagram consisting of a single class.

32.2.5 Software Design Patterns

A software **design pattern** is a general reusable conceptual solution that is used to solve commonly occurring software architecture problems. It is described from a system architect's and programmer's point of view, where implementation structure is described by relationships and interactions between classes and objects. Design patterns speed up development by using tested, proven concepts that have been useful for other developers, and provide a common language to communicate with other developers.

Some common patterns for VR are factories to instantiate objects (e.g., for the user to create many similar objects); adapters to connect different software libraries (e.g., to create a single way of supporting different HMDs); singletons to ensure only one object of a type exists (e.g., a single scene graph or hierarchical structure of the world); decorators to dynamically add behaviors to objects (e.g., adding a sound to an object that previously did not support sound); and observers so objects can be informed when some event occurs (e.g., callbacks so some part of the system knows when the user has pushed a button).

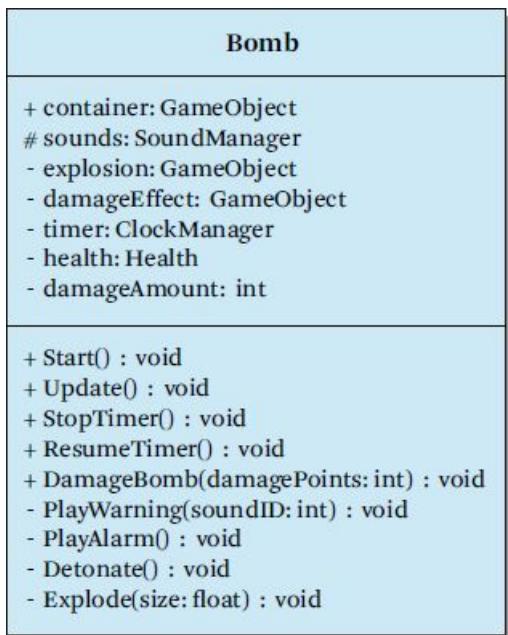


Figure 32.3 An example class diagram.

32.3 System Considerations

32.3.1 System Trade-Offs and Making Decisions

Considering the different trade-offs of a project leads to a new understanding of the intricately interlocked interplay of factors [Brooks 2010] and how they might be implemented. Clarity of understanding trade-offs helps the team to choose hardware, interaction techniques, software design,

and where to focus development. Such decisions depend upon many factors, such as hardware affordability/availability, time available to implement, interaction fidelity, representative users' susceptibility to sickness, etc. Table 32.1 shows some of the more common decisions that must be made when designing and implementing a VR experience. Most of these system decisions should be made early in the project. Even if it turns out the decision was wrong, then it will quickly become apparent and can be changed accordingly at an early iteration.

Table 32.1 Some common VR decisions and example choices.

Decision	Example Choices
Hand input hardware	None, Leap Motion, Sixense STEM
Number of participants	Single user, multiplayer, massively multiplayer
Viewpoint control pattern(s)	Walking, steering, world-in-miniature, 3D multi-touch, automated
Selection pattern(s)	Hand selection, pointing, image-plane, volume-based
Manipulation pattern(s)	Direct hand manipulation, proxy
Realism	Real-world capture, cartoon world
Vection	None, short periods of self-motion, linear motion only, active, passive
Intensity	Relaxing, heart-thumping
Sensory cues	Visual, auditory, haptics, motion platform
Posture	Sitting, standing, walking around

In some cases it is appropriate to support multiple options. Be careful of trying to support too many options as supporting everything by implementing for the “average” option results in no single optimal experience. Instead, implement different metaphors and interactions that most appropriately fit each option. It is generally better to start by choosing a single option and optimize for it before adding secondary options, features, and support.

32.3.2 Support of Different Hardware

Although some hardware has similar characteristics, other hardware can have very different characteristics (Chapter 27). Supporting different hardware that is in the same class can enable a wider audience since users may not all have the same access to the same hardware. Fortunately, in some cases the specific hardware only slightly changes the experience. For example, the Sony PlayStation Move is the same class of input device as the Sixense STEM Controller (both are 6 DoF tracked hand-held controllers). Supporting both will require additional code but the core interaction techniques can remain the same. Just because hardware is similar, do not assume testing on one piece of hardware will apply to all hardware in that same class; test across the different hardware and optimize as needed.

It can be dangerous to support different hardware classes for a single experience. Supporting different input device classes as being interchangeable is difficult and most often results in the experience being optimized for none of them. Those experiences that take full advantage of and are dependent upon unique characteristics of some input device class should not support other input device classes. For example, the Sony Playstation Move and Sixense STEM are very different from Microsoft Kinect or Leap Motion. The exception is for special cases when the experience is designed to take advantage of multiple input device classes in a hybrid system where all users are expected to have access to and use all hardware.

If different hardware classes must be supported, then the core interaction techniques should be independently optimized for each in order to take advantage of their unique characteristics. A tracked hand-held controller will have different interactions and experiences than a bare-hands system.

32.3.3 Frame Rate and Latency

A frame rate that maintains at least the refresh rate of the HMD should be obtained from the beginning and maintained throughout the project (Section 31.15.3). Otherwise it will be more difficult to optimize at a later time (e.g., lower polygon-count assets may need to be re-created). Fortunately, assuming scene complexity is reasonable, today’s hardware makes this relatively easy to achieve. Frame rate should be carefully

observed as new assets are added and code complexity increases. Even occasional drops in frame rate can be uncomfortable to users. Consistent latency is as important as low latency (Section [15.2](#)).

Some rendering algorithms/game engines perform multiple passes to add special effects. In some cases this can add one or more additional frames of latency even while a high frame rate is achieved. VR creators should not rely only on frame rate and should measure end-to-end latency with a latency meter (Section [15.5.1](#)). As discussed in Section [18.7](#), prediction and warping can be used to reduce some of the negative effects of latency, but this only works well for latencies within a 30 ms range.

32.3.4 Sickness Guidelines

Developers should respect the adverse effects of VR (Part [III](#)) even if they are not prone to VR sickness themselves. Waiting for feedback from the Learn Stage can dramatically slow down the iterative process and might require complete redesign/reimplementation if problems are not taken care of while implementing. It is as important for programmers to understand adverse health effects and how to mitigate them as it is for anyone on the team.

32.3.5 Calibration

Proper calibration of the system is essential and tools should be created for both developers and users to easily and quickly calibrate (if not automated). Examples of calibration include interpupillary distance, lens distortion-correction parameters, and tracker-to-eye offset. If these settings are not correct, scene motion will occur while rotating the head, which will result in motion sickness. See [Holloway \[1997\]](#) for a detailed discussion of various sources of HMD calibration error and their effects.

32.4 Simulation

Simulation can be more challenging for VR than it is in more traditional applications. This section briefly discusses some of these challenges and some ways to solve them.

32.4.1 Separate Simulation from Rendering

Simulations should be executed asynchronously from rendering [[Bryson and Johan 1996](#), [Taylor et al. 2010](#)]. Slow updates of parts of the scene are okay as long as rendering resulting from head motion occurs at the HMD refresh rate and under 30 ms of latency. For example, a scientific simulation of colliding galaxies might execute remotely on a supercomputer with updates coming in only once per second. This slow dynamic updating of the data will be obvious to the user, but the user will still be able to view the static aspects of the data (e.g., the user can move his head into the non-moving data) in real time.

For realistic physics simulations, it is usually required that the simulation be calculated at a fast update rate. This is especially true if the physics simulation is being used to render haptic forces (Section [3.2.3](#)) as haptics updated at a rate less than 1,000 Hz can result in objects not feeling solid. Higher rates are even better at making objects feel even more solid [[Salisbury et al. 2004](#)].

32.4.2 Fighting Physics

Mixing human interaction with physics simulation can be extremely challenging in VR because the simulation of objects “fights” with where the physical hand is. This is due to the results of the physics simulation contradicting where the hand actually is. This can result in the object appearing to **jitter** (moving quickly back and forth) because the human input and the simulation “fight” or go back and forth between the calculated pose and the actual hand pose. The simple solution is to not fight. That is do not try to have two separate policies simultaneously determine where an object is.

The most common solution when an object is picked up by a user is to stop applying physics simulation to that object. The object may still apply forces to other non-static objects (e.g., hitting a ball with a baseball bat). This lack of physics being applied to the hand-held object results in the object moving through static objects (e.g., a wall or large desk) since those static objects don’t move and don’t apply forces back to the hand-held object (although sensory substitution such as a vibrating controller can be applied as discussed in Section [26.8](#)). Although not realistic, this approach is often less presence-breaking than the alternative of jitter and unrealistic physics.

The other option is to ignore hand position when a hand-held object penetrates another object. This works well when penetration is shallow but not for deeper penetrations (Section [26.8](#)). Unfortunately, penetration depth typically cannot be controlled (i.e., most VR systems do not provide physical constraints) so this is typically not a good option.

32.4.3 Jittering Objects

Physics simulation even without user intervention can cause objects to jitter. For example, an object will sometimes appear to jitter on the ground even when the user is not touching it. This can occur due to different reasons such as round-off error/numerical approximation or using linear estimations for non-linear behavior. For example, the simulation might overestimate the distance an object has fallen resulting in the object penetrating the ground. The simulation then applies a force to move the ball back up just above the ground. But then gravity takes over and the object falls just under the surface and the cycle continues. A simple solution can be implemented by clamping object motion to zero when that motion reaches some minimum value. This unfortunately causes objects to sometimes appear to suddenly stop when the minimum value is reached.

Allowing a small amount of penetration without causing a collision response can also help. This, however, can result in objects appearing to penetrate other objects if the value is too large. More elegant solutions can be implemented, such as applying a stronger damping force when the motion falls below some threshold.

Solving for rag-doll character motions with purely physically based simulation can also result in instability that appears as jitter. Thus, if a physically-based rag doll is used, the simulation might not control the character's bones directly but be a hybrid solution of the physics simulation feeding into code that simplifies the simulation results, appearing as smoother believable motions.

32.4.4 Flying Objects

It is also often the case that mixing human input and simulation will result in extremely large forces causing objects to fly quickly off into the distance. This can happen when the physical hand pushes an object past the surface of some other object and then releases the object. The physics simulation then takes over and overcompensates by applying a large force to move the object instantaneously toward the surface. If the hand has penetrated the object too far, then the force will be large resulting in the object flying off into the distance in an unrealistic manner. Specialized code might be required to detect such occurrences and then handle the situation differently than using the standard physics formulas. For example, if the object is released inside an object then the object might first be moved toward the closest surface before letting the physics engine take over.

Similarly, simulation instability can result when multiple physically simulated objects are bound closely together. The instability can exacerbate quickly and cause objects to fly off into space. Such scenarios can be quite chaotic, and difficult to automatically detect and correct (David Collodi, personal communication, May 4, 2015). Thus, the designer should avoid having multiple physically simulated objects interacting whenever possible, especially if the objects are in an enclosed space or are tightly constrained against one another (e.g., a stack of blocks in a small pit or multiple boxes joined together in a circular fashion).

32.5 Networked Environments

A networked VR system where multiple users share the same virtual space has enormous challenges beyond the challenges of a single-user system. Essentially, a collaborative VR system is a distributed real-time database with multiple users modifying it in real time with the expectation that the database is always the same for all users [[Delaney et al. 2006a](#)].

32.5.1 Ideals of Networked Environments

Network consistency is the ideal goal that, at any point in time, all users should perceive the same shared information at the same time [[Gautier et al. 1999](#)]. Unfortunately, perfect consistency is impossible to achieve due to technical challenges such as network latency, packet loss, and difficulty in maintaining true quality of service. However, by understanding the challenges and violations of network consistency, we can work toward the ideal of creating higher-quality shared experiences.

Network consistency can be broken down into synchronization, causality, and concurrency [[Delaney et al. 2006a](#)].

- **Synchronization** is the maintenance of consistent entity state and timing of events for all users. Ideally, all clocks are synchronized.
- **Causality** (also known as ordering) maintains consistent ordering of events for all users. Ideally, all events are executed on each computer in the true order they occur no matter which computer they occurred on.
- **Concurrency** is the simultaneous execution of events by different users on the same entities. Object ownership/control must be resolved. Ideally, there are no conflicts over shared objects.

Violations of any of the three challenges listed above can lead to inconsistent states of the world between users and breaks-in-presence. Problems specific to networked environments include the following.

- **Divergence** is the temporal-spatial state of an entity being different for different users. Divergent objects result in users interacting with those objects inconsistently, sometimes leading to inconsistent behavior causing object states to diverge even further. An example is simulation divergence that occurs when physics simulations are executed on different computers.
- **Causality violations** are events that are out of order so that effects appear to occur before cause. A ball appearing to bounce before a remote user appears to dislodge the ball is a causality violation.
- **Expectation violations** (also known as intention violations) are resulting effects from concurrent events that are different than the expected or intended effects. An expectation violation occurs when a remote user changes the color of an object at approximately the same time as the local users change the color of the same object. The local user expects it to turn one color but it changes to some other color.

In addition to being consistent, networked VR systems should also be easily usable and believable. Responsiveness and perceived continuity are two primary contributors of the user experience as it relates to building networked systems.

- **Responsiveness** is the time taken for the system to register and respond to a user action. Synchronization, causality, and concurrency could more easily be achieved by adding a delay for all users but would result in low responsiveness. Ideally, users can interact with all networked objects as if they were local without having to wait for confirmation from other computers.

- **Perceived continuity** is the perception that all entities behave in a believable manner without visible jitter or jumps in position and that audio sounds smooth.

32.5.2 Message Protocols

Communication between computers occurs through packets. A packet is a formatted piece of data that travels over a computer network. The type of packets used to convey information can have a big impact on performance and synchronization.

UDP (user datagram protocol) is a minimal connectionless transmission model that operates on a best-effort basis. Packets are simply sent out to a destination without any guarantee of delivery, ordering, or duplication. Packets might not be received, for example, if the receiving computer is overloaded with too many incoming packets. UDP should be used when low latency is a priority, updates occur often, and each update is not essential (e.g., when the state of the world will get updated soon enough if a recent packet was not received). Examples of when to use UDP are for continually updated character position and audio.

TCP (transport control protocol) is a bidirectional reliable ordered byte-stream model that comes at the cost of additional latency. TCP works by the receiving computer acknowledging with the sending computer that the packet was successfully received. Information is also included in each packet that guarantees the order that packets were received will match the order in which they were sent. TCP should be used when state information is only sent once (or occasionally) to ensure the receiving computer can update its state of the world to match the sender's. An example of when to use TCP is when some one-time event has occurred and the receiving computer needs to update its state of the world based on that one-time event.

Multicast is a one-to-many or many-to-many distribution of group communication where information is simultaneously sent to a group of addresses instead of a single address at a time. Clients simply subscribe to the multicast channel and then receive updates until they unsubscribe. Multicast works well in theory, but unfortunately there are technical challenges that often keep multicast from working well, and it can cause network overload depending on how it is implemented on network hardware (that is often out of control of the developer). True network multicasting (also known as IP multicasting) is ideal, but not all commercial routers support it (or it is disabled) so an application layer on top of the network is often built to simulate true multicasting. Unfortunately, the application layering approach is not as efficient. When possible, network multicasting should be used over application layering multicasting.

32.5.3 Network Architectures

There are many ways of connecting virtual worlds. However, **network architectures** are generally described as peer-to-peer, client-server, or hybrid architectures.

Peer-to-peer architectures transmit information directly between individual computers. Each computer maintains its own state of the world and ideally matches all other computer states of the world. In a true peer-to-peer architecture, all peers have equal roles and responsibilities. The greatest advantage of peer-to-peer architectures for VR is fast responsiveness. One of the challenges of peer-to-peer architectures is discovering virtually local peers so that users can see each other. Thus, hybrid architectures are often used for networked virtual environments where some centralized server with global knowledge connects users.

Client-server architectures consist of each client first communicating with a server and then the server distributing information out to the clients. Fully **authoritative servers** control all the world state, simulation, and processing of input from clients. This makes all the users' worlds consistent at the cost of responsiveness and continuity (although this can be improved by the local system giving the illusion of responsiveness and continuity —see Section 32.5.4). As demonstrated by Valve and others, authoritative client-server models can work well for a limited number of users [[Valve 2015](#)]. To scale to larger persistent worlds, multiple servers should be used to reduce computational and bandwidth requirements, as well as to provide redundancy if the server crashes.

Some network architectures do not clearly fall within either of the above architectures. **Hybrid architectures** use elements of peer-to-peer and client-server architectures.

Non-authoritative servers technically follow the client-server model but behave in many ways like a peer-to-peer model because the server only relays messages between clients (i.e., the server does not modify the messages or perform any extra processing of those messages). Primary advantages are ease of implementation and knowledge of all other users. Disadvantages are that systems can become out of sync and clients can cheat as they are not governed by an authoritative system.

Super-peers are clients that also act as servers for regions of virtual space and can be changed between users (e.g., when a user logs off). The greatest advantage of super-peers over pure client-server architectures is increased heterogeneity of resources (e.g., bandwidth, processing power) across peers. However, super-peer networks can be challenging to implement well.

Other hybrid architectures use peer-to-peer for transferring some data (e.g., voice) while also using a server to secure important data and to maintain a persistent world. Figure 32.4 shows an example of a VR system for distributed design review that uses a mix of a server (HIVE Server; [Howard et al. 1998](#)), audio multicast, and a peer-to-peer distributed persistent memory system (CAVERNSoft; [Leigh et al. 1997](#)).

32.5.4 Determinism and Local Estimation

Since bandwidth limits the rate at which packets can be sent and received, a straightforward implementation can result in remotely controlled

entities appearing to update in discrete steps—i.e., they will appear to be frozen for moments of time with frequent teleportation to new locations as new packets are received. Such discontinuities can be reduced depending if the actions are deterministic, nondeterministic, or partially deterministic. Nondeterministic actions occur through user interaction, such as when a user grabs an object, and remote nondeterministic actions can only be known by receiving a packet from the computer on which the action took place. Deterministic actions should be computed locally, and there is no need to send out or receive packets in such cases.

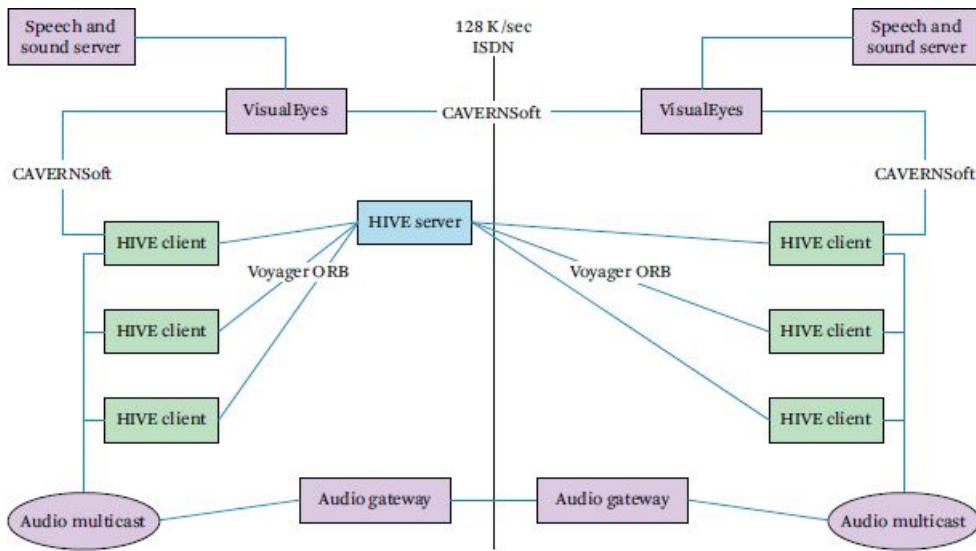


Figure 32.4 Example of a network architecture with a centralized server and audio separated from other data. (Based on [Daily et al. \[2000\]](#))

Partially deterministic actions of remote entities (e.g., a user is steering in a general direction) can be estimated every rendered frame in order to create perceived continuity. **Dead reckoning** or extrapolation estimates where a dynamic entity is located based on its state defined by information from its most recently received packet. As a new packet is received the state is updated with the new true information. If the estimated state and updated state diverge by too much, then the entity will appear to instantaneously teleport to the new location. Such discontinuities can be reduced by interpolating between the estimated position and the new position defined by the new packet. Valve’s Source game engine uses both extrapolation and interpolation to provide a fast and smooth gaming experience [[Valve 2015](#)]. Although there is a tradeoff of smoothness versus entity latency, this form of latency does not cause sickness as the local user’s head tracking and frame rate are independent of other users.

32.5.5 Reducing Network Traffic

Reducing network traffic is often a necessity to optimize networked environments no matter what architecture or communication protocol is used. This is especially true when scaling up to a large number of users. In addition to general techniques such as data compression, sending out packets only when information is needed can dramatically reduce network traffic [[Delaney et al. 2006b](#)].

One way to reduce network traffic is to compute the divergence between the local true state and the remote estimated state (e.g., entity position estimated by dead reckoning). For example, divergence will occur when a user starts/stops moving or steers in a different direction. **Divergence filtering** only sends out packets when the divergence of an entity reaches some threshold.

Relevance filtering only sends out a subset of information to each individual computer as a function of some criterion. A common way to do this is for individual servers to control some virtual space (e.g., a grid structure consisting of cells with an individual server controlling each cell) so that packets are only sent to nearby users. Server overloading caused by a large number of users in a small area can be a problem. This can be solved by using a dynamic grid/cell size that varies cell size depending on the number of users in an area (e.g., [Hu et al. 2014](#)).

Animations can also be used to reduce bandwidth requirements. For example, animating a user’s legs is often more appropriate when a user moves instead of updating the actual joints of the avatar every frame based on incoming network packets (and legs are often animated anyway even in a single-user system). Gesture recognition can be used to first interpret a user’s signal (e.g., a hand wave) so a high-level signal can be sent out and then reproduced on each client.

Audio often takes an order of magnitude or more bandwidth than other updates [[Daily et al. 2000](#)] and is especially a problem when many users attempt to speak simultaneously (Jesse Joudrey, personal communication, April 20, 2015). Stress tests can quickly be conducted by getting many users to speak simultaneously through the same server. Supporting spatialized audio requires each user’s audio stream to remain independent, so combining all audio into a single stream on a server, as is often done with non-immersive communication, is not an option. If spatialized audio is used, it is better to send the audio using peer-to-peer rather than all audio going through a server. If feasible, multicasting should be used so that users automatically subscribe and unsubscribe from the audio streams of virtual neighbors. Relevance filtering can be used to only send audio to nearby users. Less bandwidth, and thus lower-quality audio, can also be allocated to users who are further away or not in the general forward-looking direction [[Fann et al. 2011](#)]. Depending on the level of realism intended, the application might also provide the option to subscribe to teammates or important announcements regardless of distance.

32.5.6 Simultaneous Interactions

Two or more users simultaneously interacting with the same object is extremely difficult to do in networked environments and should be avoided if at all possible. Only one user at a time should own an object so that no other users can interact with it. This can be done through a token. A **token** is a shared data structure that can only be owned by a single user at a time. Before a user interacts with an object he must first request the token to make sure another user does not own it. This can make interaction difficult as there will be a delayed response when attempting to interact with an object. One way to reduce this latency is to request the key when the user approaches an object but before reaching for or selecting the object [[Roberts and Sharkey 1997](#)].

32.5.7 Networking Physics

Networked physics simulations are even a greater challenge than physics simulation on a single computer (Section [32.3](#)) because multiple representations of the same action occur at different physical locations in time and space. Theoretically, the same results should occur on different computers using the same equations and code. However, slight differences between clients (e.g., round-off error or differences in timing) cause the physics simulation to diverge over time. If a ball rolls in one direction on one computer it may roll in a different direction on a different computer, and eventually the ball may be in an entirely different part of the environment. Because of this, an authoritative server or single computer should own the simulation. Other computers can estimate the simulation, but the owning computer should send out regular updates to correct the current state, as described in Section [32.5.4](#).

32.6 Prototypes

A **prototype** is a simplistic implementation of what is trying to be accomplished without being overly concerned with aesthetics or perfection. Prototypes are informing because they enable the team to observe and measure what users do, not just what they say. Prototypes win arguments and measurement using these prototypes trumps opinion.

A **minimal prototype** is built with the least amount of work necessary to receive meaningful feedback. Temporary tunnel vision is okay when trying to find an answer to a question about a specific task or small portion of an experience. At early stages, the team shouldn't bother debating the details. Instead focus on getting *something* working as quickly as possible and then modify and add additional functionality/features on top of that or start from scratch if necessary after learning from the basic prototype.

Have a clear goal for each prototype. Only build primitive functionality that is essential to achieving the goal and reject elements not essential to that goal. Non-critical details can be added and refined later. Give up on trying to look good and feeling that a prototype is ugly, unfinished, or not ready. Expect that it won't be done well the first time or the fifth time and to be confronted by many unforeseen difficulties for what at first may have seemed to be a relatively straightforward goal. Discovering such difficulties is what prototypes are for, and it is best to find such problems as soon as possible. The sooner the bad ideas can be rejected and the best basic concepts and interaction techniques found, the sooner limited resources can be utilized for creating quality experiences.

Building prototypes quickly with minimal effort also adds the additional psychological benefit of not caring if the prototype is thrown away or not. Fast iteration and many prototypes result in an attitude of not being committed to one thing so developers don't feel like they are killing their babies.

Prototypes can vary significantly depending on what is trying to be accomplished. For example, a question about whether a certain aspect of a new interaction technique will work or not is very different than determining if the general market likes an overall concept.

32.6.1 Forms of Prototypes

A **real-world prototype** does not use any digital technology whatsoever. Instead of using a VR system, team members or users act out roles. This might include physical props or real-world tools such as laser pointers. Advantages are that such prototypes can often be created and tested spontaneously. Some disadvantages are lack of controlling conditions, difficulty in capturing quantitative data, mismatch of simulated actions to VR actions, and limitation of feedback to high-level structure/logic.

A **Wizard of Oz prototype** is a basic working VR application, but a human "wizard" behind a curtain (or on the other side of the HMD) controls the response of the system in place of software. The wizard typically enters commands on a keyboard or controller after the user verbally states his intention (e.g., to state where to travel to or to simulate a voice recognition system). When done well, this sort of prototype can be surprisingly compelling and the user may not even realize a human is controlling the system. Although collecting data for a specific implementation cannot be done with this type of prototype, high-level feedback can be obtained.

Programmer prototypes are prototypes created and evaluated by the programmer or team of programmers. Programmers continually immerse themselves in their own systems to quickly check modifications to their code. This is how most programmers naturally operate, sometimes conducting hundreds of mini experiments (e.g., change a variable value or logic statement to see if it works as expected) in a day.

Team prototypes are built for those on the team not directly implementing the application. Feedback from others is extremely valuable as it is often difficult to evaluate the overall experience when working closely on a problem. This can also reduce groupthink and having the belief that your VR experience is the best in the world when it is not. Team prototypes are used for quality assurance testing. Note team members providing feedback on the prototype might include those external to the core team such as when performing expert evaluations (Section [33.3.6](#)) via consultants. Other teams from other projects that understand the concepts of prototypes also work well.

Stakeholder prototypes are semi-polished prototypes that are taken more seriously and typically focus on the overall experience. Often

stakeholders are less familiar with VR than they like to admit. They expect a higher level of fidelity that is closer to the real product, which helps them to more fully understand what the final product will be like. Most stakeholders, however, want to see progress, so don't wait until the product is finalized. Their feedback about market demand, understanding business needs, competition, etc. can be extremely valuable.

Representative user prototypes are prototypes designed for feedback that should be built and shown to users as soon as possible. Users that have never experienced VR before are ideal for testing for VR sickness since they have not had the chance to adapt. These prototypes are primarily used for collecting data, as described in Chapter [33](#). The focus should be on building a prototype that best collects data for what is being targeted.

Marketing prototypes are built to attract positive attention to the company/project and are most often shown at meetups, conferences, trade shows, or made publicly available via download. They may not convey the entire experience but should be polished. A secondary advantage of these prototypes outside of marketing is that a lot of users can provide feedback in a short amount of time.

32.7 Final Production

Final production starts once the design and features have been finalized so that the team can focus on polishing the deliverable. When final production begins, it is time to stop exploring possibilities and stop adding more features. Save those ideas for a later deliverable. One of the great challenges of developing VR applications is that when stakeholders experience a solid demo, they get excited and start imagining possibilities and requesting new features. Whereas the team should always be open to input, they should also be aware and make it clear that there is a limit of what is possible within the time and budget constraints. Some suggestions might be easy to add (e.g., changing the color of an object), but others add risk as modifying a seemingly simple part of the system can affect other parts. Adding features also takes away from other aspects of the project. The team should make it clear that if stakeholders want additional features, there are trade-offs involved and the new features may come at the cost of delaying other features.

32.8 Delivery

Deployment ranges from the extremes of making the application continuously available online with weekly updates to multiple team members traveling to a site (or multiple sites) for a full system installation.

32.8.1 Demos

Demos (the showing of a prototype or a more fully produced experience) are the lifeblood of VR. After all, most people are not interested in a conglomeration of documentation, plans, and reports—they want to experience the results. They are also not interested in excuses of how it was working at the last conference or how something is broken. It is the VR experience that gives confidence to others that progress is being made and the team is creating something of value.

Conferences, meetups, and other events give the team their time to show off their creation. Scheduling demos is also a great way to create real accountability to get things done and get them done well. In the worst case, the team will be significantly motivated to do better next time after a demo turns out to be a disaster.

Before traveling, set the demo up in a room different from where equipment is normally located to make sure no equipment is missed when packing. If the equipment is beyond the basics, then take the packed material to yet a different room and set up again to ensure everything that is needed was indeed packed. Then pack two backups of everything. If at all possible, set up in the space where the demo will take place at least one day in advance to make sure there will be no surprises. On the day of the demo, arrive early and confirm everything is working.

Being prepared for in-house demos is also essential. With all the excitement of what is possible with VR, you never know when a stakeholder or celebrity is going to stop by wanting to see what you have. Note this does not mean anyone wanting a demo should be able to walk in any time of day to test the system—it simply means to be prepared when the important people show up. The team's time is valuable and demos can be distracting, so demos should be given sparingly unless of direct value to the project. A single individual should be responsible for maintaining and updating onsite demos and equipment so that when someone important shows up the system is more than barely working. The same individual might be responsible for keeping a schedule of demo events, or that might be some other individual.

32.8.2 Onsite Installation

Full project delivery often includes **onsite installation**, where one or more members of the team travel to the customer's site to set up the system and application. In such cases, do not assume installation will go smoothly. Debugging, calibration, development, and extra electrical/gaffer tape should be expected for complicated installation with multiple pieces of hardware. Issues might include electromagnetic interference and incompatible connections/cables with a client's system. In addition, training of staff may be required to use and maintain the system. What is obvious to you may not be obvious to them. Support and updates might also be included as part of a contract.

32.8.3 Continuous Delivery

The opposite of onsite installation is **continuous online delivery**, where online updates are provided often. Agile methods recommend delivering a new executable that goes out to customers on a weekly basis. This has value in forcing the team to focus on what's important and to be accountable, and it provides plenty of opportunity for data collection (A/B testing of changing one thing for a subset of customers is most common in this situation—see Section [33.4.1](#)).

The Learn Stage

All life is an experiment. The more experiments you make the better.

—Ralph Waldo Emerson

The **Learn Stage** is all about continuous discovery of what works well and what doesn't work so well. The pursuit of learning about one's creations and how others experience them is more essential for VR than for any other technology.

Learning about what works in VR and how to improve upon one's specific application can take many forms. Learning might be as simple as changing values in code and immediately seeing the results, which is not uncommon for programmers to do hundreds of time in a single day. Or learning might take months to conduct due to formal experiment design, development, testing hundreds of users, and analysis. Although not as rapid as code testing, this section leans toward fast feedback, data collection, and experimentation. This enables the team to learn quickly how well or not ideas meet goals, and to correct course immediately as necessary. It is also highly recommended to utilize VR experts, subject-matter experts, usability experts, experiment-design experts, and statisticians to ensure you are doing the right things to maximize learning. The earlier problems are found, the less expensive they are to fix. Finding a problem near final deployment can result in failure for the entire project.

Understanding the learning/research process not only enables one to conduct his own research but also enables one to understand the research of others, where it might apply to one's own needs or project, and to intelligently question claims instead of blindly accepting them.

Like the other stages of iterative design, the Learn Stage will evolve over the course of the project. Initial learning will consist of informally demonstrating the system to teammates to get general feedback. Eventually, more sophisticated methods of collecting and analyzing data might be employed that very well may go beyond what is discussed here. Readers are encouraged to use only the concepts that most directly apply to their project; all projects will not use all concepts. However, readers will benefit by being aware of all of them.

33.1 Communication and Attitude

Effective communication is essential for learning. This includes communication between team members as well as communication with users. Effective learning and communication is heavily dependent on attitude of how one seeks out and reacts to constructive criticism. Those who have problems with receiving criticism about their work will not make it far with VR. A positive mind-set and attitude about continual improvement is essential, and often the greatest breakthroughs occur through breakdowns that result from others pointing out issues that are not obvious to those creating the experience.

When demonstrating systems and asking for feedback, think positively for yourself, your team members, and users even when failure occurs. **Failure** is a learning experience; do not fear it. Of course the goal is success, but often we don't really know why we succeeded. With an appropriate attitude toward failure, it is often possible to figure out why, to ensure that the problem will not happen again.

When communicating with users, the following points are essential in order to maximize learning.

- Do not blame/belittle users or their opinions/interactions with the VR application. Doing so shuts them down emotionally where they will not effectively provide feedback.
- Actively investigate difficulties to determine how the project can improve.
- Assume what others are doing is partially correct and then provide suggestions that enable them to correct and move on.
- When someone points out a problem you are already aware of, thank them for noticing and encourage them to continue looking for problems.
- Instead of using words like failure, talk about learning.

As an additional benefit of communicating effectively with users, those giving feedback will also feel good about themselves and have a sense of contribution and ownership, which can turn them into fans/evangelists when the product becomes publicly available. Such community involvement is essential for crowdfunding campaigns (e.g., Kickstarter).

33.1.1 VR Creators Are Unique Users

A common mistake VR creators, especially programmers, make when creating VR experiences is that they assume what works for themselves will work for everyone else. Unfortunately, this is especially not the case for VR. Programmers often use the system in very specific ways without trying all unanticipated actions. Programmers also may have adapted to sensory incongruencies where they do not get sick as non-adapted users do.

Programmers don't typically believe strongly that users' opinions don't matter—in fact, they are often aware they should be collecting feedback but see it as an inconvenience to regularly communicate with users; it is just not a priority. Most programmers do enjoy showing off their work occasionally, but such demos are not done frequently enough, are not done in a manner that results in quality data collection, and do not result in changes because opinions are not taken seriously enough.

Programmers' time is extremely valuable, and it is true programmers should not be constantly bothered with giving demos and collecting data. Thus, the answer is that other team members should be used to collect data from users. However, it should be the case that programmers do at least occasionally participate in data collection, so that they take the feedback more seriously. There is greater resistance to seeing or hearing a report stating what one has created doesn't work well, compared to observing in person how it doesn't work well.

33.2 Research Concepts

Research takes many forms and has many definitions. One definition of **research** is a systematic method of gaining new information and a persistent effort to think straight [[Gliner et al. 2009](#)].

How many VR teams conduct research with the intent to truly understand what works for representative users? That is, how many VR professionals collect external feedback apart from giving the standard VR demo in the office, meetup, or conference? Unfortunately, very few. This is a reason why there are normally so many unknowns in creating an engaging VR experience, leaving the success of the project to chance instead of finding and pursuing the optimal path toward success.

This section covers background concepts that are useful for designing, conducting, and interpreting research in an effective manner.

33.2.1 Data Collection

Data collection enables the effectiveness of VR aspects to be specified, quantified, and compared. Data collection often involves a combination of both automated data capture and observation by human raters. Some commonly collected data for VR include time to completion, accuracy, achievements/accomplishments, frequency of occurrence, resources used, space/distance, errors, body/tracker motion, latency, breaks-in-presence, collisions with walls and other geometry, physiological measures, preferences, etc. Clearly, there are many measures that can be useful for VR. The team should collect different types of data rather than relying on a single measure.

Data can be divided into qualitative or quantitative data. Both are important as they each provide unique insight about a design's strength and weaknesses.

Quantitative data (also known as numerical data) is information about quantity—information that can be directly measured and expressed numerically. Quantitative data is best gathered with tools (e.g., a questionnaire, a simple physical measuring device, a computer, or body trackers) that require relatively little training and result in reliable information. Quantitative data analysis involves various methods for summarizing, comparing, and assigning meaning to data, which are usually numeric and which usually involve calculation of statistical measures. Typically, experimental approaches (Section [33.4](#)) focus on collecting quantitative data.

Qualitative data is more subjective—it is biased and can be interpreted differently by different people. Such data includes interpretations, feelings, attitudes, and opinions. The data is often collected through words (whether spoken or written) and/or observation. Qualitative data is not directly measured with numbers but, in addition to being summarized and interpreted, can be categorized and coded. Constructivist approaches (Section [33.3](#)) rely heavily upon qualitative data.

Both quantitative and qualitative approaches are useful for developing better VR experiences. In fact, both types of data are often collected from the same individuals. Qualitative data is often gathered first to get a high-level understanding of what might be later studied more objectively with quantitative data. Even for teams focusing only on qualitative data, it is useful to have a basic understanding of the core concepts of quantitative research, which can help in understanding when qualitative data and its conclusions might be biased.

33.2.2 Reliability

Reliability is the extent to which an experiment, test, or measure consistently yields the same result under similar conditions. Reliability is important because if a result is only found once, then it may have happened by chance. If similar results occur with different users, sessions, times, experimenters, and luck, then we can be highly confident there must be some consistent characteristic of what is being tested.

Measures are never perfectly reliable. Two factors that affect reliability are stable characteristics and unstable characteristics [[Murphy and Davidshofer 2005](#)]. **Stable characteristics** are factors that improve reliability. Ideally, the thing being measured will be stable. For example, an interaction technique is stable if different users have a consistent level of performance on a task when using that technique. **Unstable characteristics** are factors that detract from reliability. These include a user's state (health, fatigue, motivation, emotional state, clarity and comprehension of instructions, fluctuations of memory/attention, personality, etc.) and device/environment properties (operating temperature, precision, reading errors, etc).

An **observed score** is a value obtained from a single measurement and is a function of both stable characteristics and unstable characteristics. The **true score** is the value that would be obtained if the average was taken from an infinite number of measurements. The true score can never be exactly known, although a large number of measurements provide a good estimate. When testing participants repeatedly to measure learning effects, for example, it is important to have high reliability so that it is more likely the observed score was a result from learning versus occurring by chance.

Reliability values are often expressed as a correlation coefficient [[Gliner et al. 2009](#)] that takes a value between -1 and 1, as explained in Section [33.5.2](#).

33.2.3 Validity

A reliable measure may consistently measure something, but it might be measuring the wrong thing. **Validity** is the extent to which a concept, measurement, or conclusion is well founded and corresponds to real application. Validity is important to understand when conducting research to make sure one is actually measuring and comparing what one thinks they are measuring and comparing, conclusions are legitimate, and the results apply to the domain of interest.

Overall, validity is a function of many factors and researchers further break validity down into more specific types of validity. Unfortunately, researchers cannot agree on how to organize and name the different types of validity. Below general validity is broken down into face validity, construct validity, internal validity, statistical conclusion validity, and external validity.

Face Validity

Face validity is the general intuitive and subjective impression that a concept, measure, or conclusion seems legitimate. Face validity is one of the first things researchers consider when collecting data.

Construct Validity

Construct validity is the degree to which a measurement actually measures the conceptual variable (the construct) being assessed. A measure has high construct validity if it captures the hypothetical quality that it claims to be measuring. More specifically, a measure with high construct validity adequately covers all aspects of the construct; correlates with other measures of the same construct; is not affected by variables not related to the construct; and accurately predicts concurrent and future measurements, behavior, or performance. A common example of construct validity is the ability of an assessment tool to differentiate between experts and novices performing a given task.

The goal of collecting data may be to measure user performance, user preference, some aspect of a generalized interaction technique, or a specific implementation for analyzing usability issues. At early stages of testing, bugs and imperfections will be present. Because of system imperfection, the researcher must be careful what is being measured. In many cases, the specific implementation is not what is of interest, because problems are known that will be fixed. Rather, questions are more often about general user preferences, performance, and perception that are unrelated to bugs already on the maker's task list to be fixed. If a system artifact has been measured when that is not the intention, then a violation of construct validity has occurred. Researchers must be careful of concluding that a concept is not effective in general just because it does not work in its current implementation.

Internal Validity

Internal validity is the degree of confidence that a relationship is causal. Internal validity depends on the strength or soundness of the experiment design and influences whether one can conclude that the independent variable or intervention caused the dependent variable to change. When the effects of the independent variable cannot be separated from the possible effects of extraneous circumstances, the effects are said to be confounded (Section [33.4.1](#)). Internal validity can be threatened or compromised by many confounds including the following.

History is an extraneous environmental event that occurs when something other than the manipulated variable happens between the pre-test and the post-test. An example is if data collection is conducted on different days—some world event may have occurred that altered opinion or motivation for performing well.

Maturation is personal change in participants over time. The changes might be due to changes that are not relevant to that being tested. For example, participants may grow tired, become sick, or become resistant to VR sickness.

Instrumentation occurs due to change in the measurement tool or observers' rating behavior. For example, a tool for measuring reaction time or physiological measures could go out of calibration. Observers that rate participants change their criteria over time or, worse, different observers are used. Calibration, providing guidelines, and training observers are important for reducing instrumentation bias. The room setting, hardware setup, and software should not change between sessions.

Selection bias occurs due to non-random assignment or self-selection of participants to groups. Selection bias is best solved by randomly assigning participants to groups. If participants self-select themselves into groups, then bias can be reduced, but not removed, by not letting them know of the differing conditions. Fans volunteering to test a system will certainly be biased toward liking a VR experience more so than the general targeted population.

Attrition (also known as mortality) is the dropout of participants. This is a problem for validity when the attrition for one group is different from another group. This is a major problem when participants drop out due to VR sickness. If one condition causes one group to have more sickness, then that group will lose more participants. Those that do not drop out will likely have a higher tolerance for sickness, and thus the results may be due to some characteristic that correlates to tolerance of sickness rather than the factor being studied.

Retesting bias often occurs when the data is collected from the same individual two or more times. Participants are likely to improve/learn on the first attempt due to practice that carries over into later tests. Results may be due to this practice rather than due to the manipulated variable.

Statistical regression occurs when participants are selected in advance based on high or low scores. Many of those who scored low or high scored that way due to chance. On subsequent testing, scores tend to average out due to the initial scores being different from their true average score. There is no way to determine if the changes occurred due to statistical regression or due to the factor being tested. Selection of participants should not be based off of previous scores unless multiple tests showed consistent and reliable results.

Demand characteristics are cues that enable participants to guess the hypothesis. Participants might modify their behavior (consciously or subconsciously) to look good, appear politically correct, or make the experimenter look good or bad.

Placebo effects occur when a participant's *expectation* that the experiment causes a result instead of the experimental condition itself causing the result. For example, pre-tests with the Kennedy Simulator Sickness Questionnaire (Section [16.1](#)) have been found to cause heightened sickness as reported in post-tests [[Young et al. 2007](#)]. Or, if the participant believes that one interaction technique is better than another technique, then it may be the result occurs to the belief rather than the mechanics of the interaction.

Experimenter bias occurs when the experimenter treats participants differently or behaves differently. This might be explicit bias due to the experimenter encouraging the participant to perform better under one condition, but is more often unintentional. The experimenter may intend not to be biased but may not even realize his own body language and tone of voice convey a preference for one condition. Ways to minimize experimenter bias are to make the experimenter blind to the conditions (e.g., randomize conditions and do not allow the experimenter to see what the participant sees), use naive experimenters (i.e., hire an assistant who would not know the difference of conditions even if he saw the conditions), and/or minimize the experimenter's time with the participant (e.g., have the instructions given and data collected by the computer).

Statistical Conclusion Validity

Statistical conclusion validity is the degree to which conclusions about data are statistically correct. Some common threats to statistical conclusion validity include the following.

A **false positive** is a finding that occurs by chance when in fact no difference actually exist. (e.g., there is a chance that flipping an unweighted coin lands heads-up several times in a row). If the experiment is conducted again, then the difference will not likely be found.

Low statistical power most commonly occurs due to not having a large enough sample size. **Statistical power** is the likelihood that a study will detect an effect when there truly is an effect. A common mistake researchers make is to claim there is no difference between conditions, when in actuality there may be a difference but the researchers did not find the difference due to not having enough statistical power. In fact, an experiment can never prove two things are truly identical (an experimental result shows they are likely different with some probability), although techniques do exist that can show two things are close within some defined range of values.

Violated assumptions of data are wrongly assumed properties of data. Different statistical tests have different assumptions. For example, many tests assume data is normally distributed (Section [33.5.2](#)). If these assumptions are not valid, then the experiment may lead to wrong conclusions.

Fishing (also known as data mining) is the searching of data via many different hypotheses. If a difference is found when many variables are explored, the findings may be due to random chance. For example, if 100 different types of coins are flipped 10 times each, then one or more of those coin types may land heads up 10 out of 10 times just by chance—not necessarily because that coin type is weighted. Different statistical corrections can be used to compensate for the increased likelihood of finding a difference in multiple tests.

External Validity

External validity is the degree to which the results of a study can be generalized to other settings, other users, other times, and other designs/implementations. The results that one interaction technique is better than some other technique might only apply to the specific configuration of the lab, the type of users, or the specifics of the VR system (e.g., field of view, degrees of freedom, latency, software configurations). Some VR research results from the 1990s do not necessarily apply to today due to different users (e.g., researchers versus consumers) and due to changes in technology. At that time, latency of 100 ms or more was regarded as acceptable. Today, such numbers are completely unacceptable. Today's researchers have to use their best judgment on a case-by-case basis to decide if they want to take into account results from a previous time or reconduct some of those experiments.

If something changes in the design or implementation of the system, then the previously reliable findings may no longer apply. For example, one interaction technique might be superior to another interaction technique for the system being tested but not necessarily for a system with different hardware. Ideally, experimental results will be robust to different forms of hardware, different versions of design and software, and across some range of conditions. In many cases, experiments should be conducted again when hardware has changed or significant design changes have occurred. Similar results will give confidence that the design is more robust, possibly even to specific conditions not yet tested.

Of course different techniques will never generalize to all situations and all hardware. Different input devices can have very different characteristics as discussed in Chapter [27](#). Minimal computer requirements (e.g., CPU speed and graphics card capabilities) should also be determined based off of testing. This is much more important than traditional desktop and mobile systems due to risk of sickness (Part [III](#)). Even for results that have high external validity, VR applications should never be shipped that have not been extensively tested with the targeted hardware.

33.2.4 Sensitivity

Measures may be consistent (reliable) and truly reflect the construct of interest (validity) but may lack sufficient sensitivity to be of use. **Sensitivity** is the capability of a measure to adequately discriminate between two things; it is the ability of an experimental method to accurately detect an effect, when one does exist. Is the measure capable of distinguishing between multiple levels of the independent variable? If not, then find a way to increase sensitivity.

33.3 Constructivist Approaches

Constructivist approaches construct understanding, meaning, knowledge, and ideas through experience and reflection upon those experiences rather than trying to measure absolute or objective truths about the world. The approach focuses more on qualitative data and emphasizes the integrated whole and the context that data is collected in. Research questions using this approach are often open-ended but structured enough to be useful for summary and analysis.

This section describes various methods of collecting data in order to better understand VR experiences and to improve upon those experiences.

33.3.1 Mini-Retrospectives

Mini-retrospectives are short focused discussions where the team discusses what is going well and what needs improvement. Such retrospectives should be kept short but performed often. It is much better to have weekly 30 minute mini-retrospectives rather than monthly half-day retrospectives.

Retrospectives should be constructive. Jonathan Rasmusson states in *The Agile Samurai* [[Rasmusson 2010](#)] the retrospective prime directive:

Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand. In other words it's not a witch hunt.

Ideally, mini-retrospectives will result in themes for future iterations and areas to track that the team wants to improve upon.

33.3.2 Demos

Demos are the most common form of acquiring feedback from users. Demos are a great start for getting a general feel of others' interest. Demos should be given often to stay in close communication with the intended audience, to understand real users, to receive fresh ideas, and to market the project. Most importantly, demos give something for the team to work toward and to show progress is being made.

However, the team should distinguish between giving demos and collecting data. Whereas demos are easy to do and do provide some basic feedback, they are the least useful method of collecting data unless they are combined with more data-focused approaches. Most often the point of demos is to market the project. Because of this, data collected during demos is typically not quality data. This is due to many factors such as data collection not being a priority, users often not providing honest opinions (i.e., they are polite), feedback almost always being remembered rather than recorded, the chaos of many users, answering questions instead of collecting data, and bias of those giving the demo and reporting the results. When data collection really is a goal of a demo, then structured interviews and questionnaires can be used to collect less biased data.

The best learning from a demo occurs when a public demo goes horribly wrong in a big way. This type of learning can be extremely painful. But if the company/team can survive the disaster, then such a lesson usually ignites a team into taking massive action to build something of quality and to make sure to do things well next time so they will not be humiliated again.

33.3.3 Interviews

Interviews are a series of open-ended questions asked by a real person and are more flexible and easier for the user than questionnaires. An interview results in the best data when conducted immediately after the person being interviewed experienced the VR application. If the interview cannot be performed in person, then it can be performed by telephone or (worse) in text form.

Interviews are different from testimonials. The intention of a testimonial is to produce marketing material showing that real people like something. The intention of an interview is to learn. Interview guidelines should be created in advance that are designed to keep the interview on track and to reduce bias.

Appendix [B](#) provides an example of a simple interview guideline document where the intention is to generally improve a VR experience. When the intent is to improve upon more specific aspects of a VR experience, then more specific questions should be included.

Listed below are some tips for getting quality information out of interviewees.

Establish rapport. Skilled interviewers get to know and quickly become friends with the interviewees before asking questions specific to the intent. Differences in personal style, ethnicity, gender, or age may reduce rapport resulting in less truthful answers. For example, a professor or other authority figure conducting the interview may result in some interviewees not answering with their true thoughts. Attempt to match the interviewer with the intended audience based on the personas created, as described in Section [31.11](#).

Perform in a natural setting. Conducting interviews in unnatural or intimidating settings such as a lab can result in little response from interviewees. Many game labs have rooms set up as living rooms. The team can also travel to the site of the interviewee instead of having the interviewee travel to the site of the interviewers.

Time. Don't spend more than 30 minutes per demo/interview maximum in a single session. Set time limits by scheduling interviews back to back. It is better to interview multiple individuals than to interview one individual in depth. Individuals can be followed up with if they have more useful things to say.

33.3.4 Questionnaires

Questionnaires are written questions that participants are asked to respond to in writing or on a computer. Questionnaires are easier to administer and provide for more private responses. Note questionnaires are different than surveys as the intent for surveys is to make inferences about an entire population, which requires careful sampling procedures. Surveys are typically not appropriate for evaluation of VR applications.

Asking for background information can be useful for determining if participants fit the target population that resembles personas created in the Define Stage (Section [31.11](#)). Conversely, such information can be used to better define personas if participants are self-selected. The information can also be used to make sure a breadth of users is providing feedback, to match participants when comparing performance between participants, and to look for correlations with performance.

The Kennedy Simulator Sickness Questionnaire as described in Section [16.1](#) is an example of a questionnaire commonly used for VR studies.

Close-ended questions provide answers that are selected by circling or checking.

Likert scales contain statements about a particular topic, and participants indicate their level of agreement from strongly disagree to strongly agree. The options are balanced so there are an equal number of positive and negative positions.

Partially open-ended questions provide multiple answers that participants can circle or check, but also provide an option to fill in a different answer or additional information if they don't feel the listed options are appropriate.

Open-ended questions are extremely useful for obtaining more qualitative data and information about the experience that is not expected. However, some participants must be given extra encouragement to fill out such questions as it takes more cognitive effort than simply circling an option or checking a box.

An example questionnaire, used in evaluation of Sixense's MakeVR application [[Jerald et al. 2013](#)], is included in Appendix [A](#). The questionnaire includes the Kennedy Simulator Sickness Questionnaire, a Likert scale, a background/experience questionnaire, and open-ended questions.

33.3.5 Focus Groups

Focus groups are similar to interviews but occur in group settings. Group settings are more efficient than individual interviews and can stimulate thinking as participants build off of each other's ideas. Traditionally, focus groups have been used to determine how customers will perceive and react to new products or to gather opinions for improving political campaigns. For VR, participants typically use a prototype demo to base their feedback on. The goal is to stimulate thinking and elicit ideas aimed at improving on existing approaches and creating new experiences and interactions. Focus groups are extremely useful in the early phases of VR design as the feedback can be used to explore new concepts, better refine questions, and improve upon data collection via more structured approaches.

An excellent example of an effective focus group was conducted by Digital ArtForms as part of the National Institute of Health grant "Motion-Controlled Gaming for Neuroscience Education" [[Mlyniec 2013](#)]. Two focus groups consisted of two separate classrooms of fifth graders. After subject-matter expert Ann M. Peiffer, PhD, of Wake Forest School of Medicine described the basic concepts of stroke and its causes to the class, each child individually put together a virtual 3D puzzle of a brain that consisted of the different brain lobes. As they grabbed a virtual brain lobe with tracked hand-held controllers, audio and video embedded within the game explained what that lobe controlled. In the next level, a virtual character complained about and acted out a stroke symptom. The child then "cured" the character by injecting "medicine" into the appropriate lobe. Time constraints and scores were awarded to motivate students to complete the task.

In addition to collecting data via pre- and post-tests to determine learning effectiveness, the team asked the kids how the game might be improved and solicited ideas for future games. The children came up with the concept of zombies with symptoms caused by stroke. As the zombies would come at the player, the student would have to quickly determine what a zombie's deficit was and then reach for the correct brain lobe to throw at the zombie, feeding and curing it with the appropriate "medicine" before it could reach the player. This game concept never would have been conceived without input from the children. Digital ArtForms is now developing a VR title as a result of the focus group.

33.3.6 Expert Evaluations

Expert evaluations are systematic approaches conducted by experts that identify usability problems from the user's perspective with the intent to improve the user experience. When done properly, expert evaluations are the most efficient method of improving upon the usability of the system.

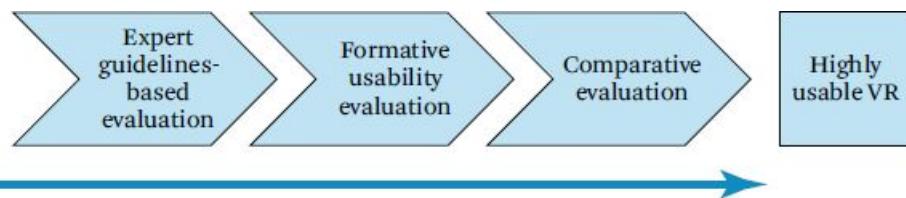


Figure 33.1 Progression of expert evaluation methods.

This section gives an overview of the methods that VR usability experts have found to be most efficient and effective for collecting quality data, enabling fast iteration toward ideal solutions. This is best done with different types of evaluations in succession as shown in Figure [33.1](#). Each evaluation component generates information that feeds into the next evaluation component, resulting in an efficient and cost-effective strategy for

designing, assessing, and improving VR experiences [[Gabbard 2014](#)]. These methods are typically used for multiple aspects of the system and performed at different phases of the project.

Expert Guidelines-Based Evaluation

Expert guidelines-based evaluation (also known as heuristic evaluation) identifies potential usability problems by comparing interactions (either existing or evolving) to established guidelines [[Gabbard 2014](#)]. The identified problems are then used for recommendations for improving the design. This should be done early in the development cycle so issues can be corrected before affecting other aspects of the design.

Ideally, multiple VR usability experts perform independent evaluations of a prototype. Multiple evaluators result in quality and quantity of data that outweigh the cost of those evaluators. As this is for general VR usability most efficiently evaluated by experts, no representative users are involved in this type of evaluation. Evaluations should give a score of each issue's severity and describe why it is a problem. After each expert independently evaluates the application, results are combined and ranked to prioritize fixing critical usability problems and to design subsequent formative evaluations.

Unfortunately, traditional 2D interface guidelines are not appropriate for VR, and no well-established guidelines specific to VR yet exist. Although not intended specifically for this type of evaluation, many of the guidelines in this book from Part III and Part V can be used. Other guidelines include the Oculus Best Practices Guide [[Oculus Best Practices 2015](#)], Joseph Gabbard's taxonomy of usability characteristics [[Gabbard 1997](#)], and NextGen Interactions' internal documentation.

Formative Usability Evaluation

After expert guidelines-based evaluation has revealed and solved as many usability problems as possible, it is time to move on to formative usability evaluation. **Formative usability evaluation** diagnoses problems by gathering critical empirical evidence from users interacting with an application during formative and evolving stages of design. The goal is to assess, refine, and improve usability, learning, performance, and exploration through observation of representative users [[Gabbard 2014](#), [Hartson and Pyla 2012](#)]. This improves the application by iteratively identifying problems throughout the design/development process so that the team can continually refine tasks and fine-tune interactions. The method should be conducted by VR usability experts as it relies heavily on a solid understanding of VR interaction that is dependent on context (i.e., it consists of more than a predefined list of items to evaluate). If done well, such evaluation can be quite efficient and effective for improving the usability of VR interactions.

Figure 33.2 shows the formative usability evaluation cycle. User task scenarios (which can be taken from task analysis—see Section 32.1) are first created to exploit and explore all identified tasks, information, and workflows. As users interact with the system, they “think aloud” by explicitly verbalizing their actions, thoughts, actions, and goals [[Hartson and Pyla 2012](#)]. Evaluators collect qualitative and quantitative data in order to identify problems as well as strengths. This data is then analyzed and summarized with suggestions for improvement and emphasis on keeping features that work well. This information can also be used to improve user task scenarios for the next iteration of the formative usability evaluation.

One of the most useful results of evaluation is a list of critical incidents [[Hartson and Pyla 2012](#)]. A **critical incident** is an event that has a significant impact, either positive or negative, on the user's task performance and/or satisfaction. This can include system crashes/errors, inability to complete a task, user confusion/disorientation, loss of tracking, sudden onset of discomfort, etc. Such critical incidents can dramatically impede usability and can have an effect on perception of application quality, usefulness, and reputation. Thus, identifying and correcting these problems as early as possible is essential to creating a quality experience.

Other data to collect includes accuracy and precision, task time to completion, number of errors, performance, and accomplishment of learning objectives. Such data can be measured through a scoring system visible to users, designed both to increase motivation and to target data collection.

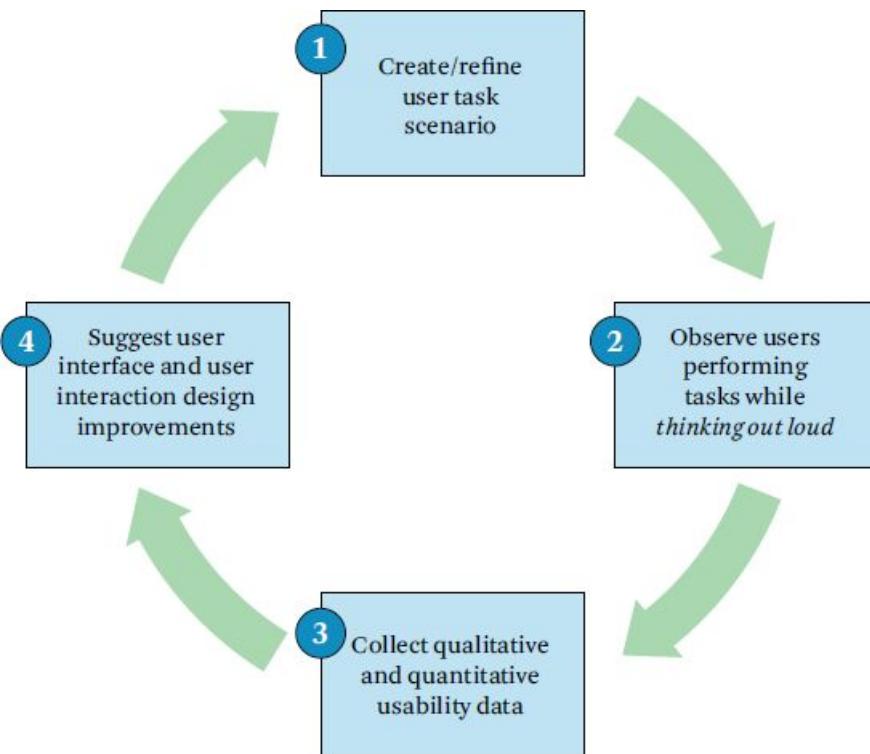


Figure 33.2 Formative usability evaluation cycle. (Adapted from [Gabbard \[2014\]](#))

During the final stages of formative evaluation, evaluators should only observe and not suggest how to interact with the system. This almost always uncovers problems of learning a system when no human is present to teach how to interact. Most users of the final experience will not have someone available to explain how to interact. Thus signifiers, instructions, and tutorials often result from this evaluation.

Comparative Evaluation

Comparative evaluation (also known as summative evaluation) is a comparison of two or more well-formed complete or near-complete systems, applications, methods, or interaction techniques to determine which is more useful and/or more cost-effective. Comparative evaluation requires a consistent set of user tasks (borrowed or refined from a formative usability evaluation) that can be used to collect quantitative data to compare across conditions.

Evaluating multiple independent variables can help to determine the strengths and weaknesses of different techniques. Evaluators compare the best of a few refined designs by having representative users perform tasks to determine which of the designs is best suited for integration within a larger project or for delivery to customers. Comparative evaluation is also used to summarize how a new system compares to a previously used system (e.g., does a VR training system result in more productivity than a traditional training system?). However, be careful of threats to validity (Section [33.2.3](#)) that may result in erroneous conclusions.

33.3.7 After Action Reviews

An **after action review** is a user debriefing with an emphasis on the user's specific actions. A member of the team discusses with the user his actions in order to determine what happened, why it happened, and how it could be done better. It is useful to watch and discuss real-world video of the user performing tasks along with a first-person perspective, which can easily be accomplished by using screen/video capture technology recorded during the VR experience. If the software can support it, then observing the action from a third-person viewpoint that can be interactively controlled also enables looking at the action from different perspectives.

33.4 The Scientific Method

The **scientific method** is an ongoing iterative process based on observation and experimentation. It typically begins with observations that lead to questions. These questions are then turned into testable predictions called hypotheses. These hypotheses can be tested in a variety of ways, including making further observations. The strongest tests come from carefully controlled and replicated experiments that gather empirical data. In order to gain confidence in a test result, the result should be replicated multiple times. The results of experiments more often lead to more questions than answers. Depending on the results of the tests, the hypothesis may require refinement, alteration, expansion, or rejection. This cycle of predict, test, and refine can repeat many times.

An **experiment** is a systematic investigation to answer a question and to gain new understanding. In most cases experimentation for the sake of creating VR applications is less formal than the extensive research methods performed by researchers more interested in scientific inquiry than creating experiences. Rigorous scientific/academic experiments are often overkill for those creating VR applications, and such formal experiments are not the focus of this book. However, understanding basic concepts of formal experiment can be useful for applied VR as such understanding can help design more basic informal experiments. Even if the perfect experiment cannot be conducted (and it rarely is perfect, even with months of

planning), at least the team is aware of some of the pitfalls that might occur. Understanding formal experimental approaches also helps to comprehend research papers and interpret/explain results.

33.4.1 Overview of Experimental Design

This section describes the basics of a single iteration of the scientific method.

Explore the Problem

The first step of exploring a problem is to obtain an understanding of what is to be studied. This can be done by learning from others (via trying their applications, reading research reports/papers, and talking with them); by trying oneself and observing others interacting with existing VR applications; by the constructivist approaches previously discussed; and by the various concepts discussed in the Define and Make Stages.

Formulate Questions

Once familiar with the overall problem, start by asking and answering the following questions:

- What specifically is the team trying to learn?
- What are the signals that provide feedback if something is working or not?
- What is the fastest and most effective way to get these questions answered?

State the Hypothesis

Answering the above questions, along with documentation from previous processes such as the listed assumptions and requirements, will lead to more specific questions that can be turned into hypotheses. A **hypothesis** is a predictive statement about the relationship between two variables that is testable and falsifiable. An example of a hypothesis is, “Interaction technique X results in less time to complete task Y than interaction technique Z.”

Until the team has gained extensive experience with basic experimental designs (or the team already contains a senior researcher with experience in conducting experiments), experiments should only test a single hypothesis at a time. Adding more complexity adds significant risk by potentially making wrong assumptions, selecting the wrong statistical tests, having to add many more participants/sessions, misinterpreting results, etc.

Determine the Variables

To conduct an experiment, different variables must first be precisely defined.

The **independent variable** is the input that is varied or manipulated by the experimenter. An example of an independent variable is the selection interaction technique used (e.g., object snapping versus precision mode pointing described in Section [28.1.2](#)).

The **dependent variable** is the output or response that is measured. The dependent variable values resulting from the manipulation of the independent variable are what will be statistically compared to determine if there is a difference. Examples of dependent variables are time to completion for a task, reaction times, and navigation performance.

Confounding factors are variables other than the independent variable that may affect the dependent variable and can lead to a distortion of the relationship between the independent and dependent variables. For example, if one group of participants is studied in the morning and another group in the evening, then one group may perform differently due to being more tired than the other group, not necessarily because the thing being studied causes different performance.

Being aware of and considering threats to internal validity (Section [33.2.3](#)) can help to find potential confounding factors. Once such confounding factors are determined to be a problem, they can often be removed by setting those factors to be constant. A control variable (also known as a constant variable) is kept constant in an experiment in order to keep that variable from affecting the dependent variable.

Decide on Within Subjects or Between Subjects

A decision that must be made for all experiments is whether to design for within subjects or between subjects. Both have their advantages and disadvantages.

A **within-subjects design** (also called repeated measures) has every participant experience each condition. Within-subjects designs have the advantage of fewer subjects being required due to each participant experiencing all conditions and less variability within individuals. This results in more efficient data collection due to spending less time on recruiting, scheduling, training, etc. The disadvantage of within-subjects design is that differences between conditions may be due to carryover effects instead of due to what is intended to be measured. **Carryover effects** occur when experiencing one condition causes retesting bias in a later condition. Examples are learning/training, fatigue, sickness, etc. One way to reduce bias from carryover effects is to counterbalance the order that participants experience the different conditions (e.g., half of the participants first experience condition A then B, half of the participants first experience B then A).

A **between-subjects design** (also known as A/B testing when only two variables are being compared) has each participant only experience a single condition. Advantages are short times for participants, low dropout rate, and no carryover effects. The main disadvantage is the requirement to have a larger number of participants due to different participants needed for different conditions and larger variability between individuals.

Conduct Pilot Studies

A full experiment can be quite costly. If the experiment contains flaws or wrong assumptions, then the entire study may be invalidated. A **pilot study** is a smallscale preliminary experiment that acts as a test run for a more full-scale experiment. Such preliminary studies are used to help determine feasibility; discover unexpected challenges; improve upon the experiment design; reduce time and cost; and estimate statistical power, effect size, and sample size.

Conduct the Experiment

Once researchers are confident the experimental design is solid, then they proceed to full data collection from all participants. Note that experimenters should not change conditions once data collection has started as this can cause confounding factors and nullify the results of the experiment.

Analyze the Data, Draw Conclusions, and Iterate

After all data has been collected, then statistical analysis (Section [33.5.2](#)) as defined in the experimental design is performed. If the design has been precisely defined, then there should be no ambiguity if a result was found or not. However, the experimenters likely discovered ways the experiment could be improved upon and gained insight for new questions to further explore and experiment with. The process continues in an iterative manner.

33.4.2 True Experiments vs. Quasi-Experiments

True experiments use random assignment of participants to groups in order to remove major threats to internal validity. Because of the random assignments, the only difference between the two groups of participants will be due to chance. True experiments are optimal for determining cause-effect relationships.

A **quasi-experiment** lacks true random assignment of participants to conditions. An example of not randomly assigning participants to groups is having participants self-select which group they will belong to (even if they don't know the difference of conditions between the two groups). Because of this non-random assignment, quasi-experiments are more likely to contain confounding factors. As a result it is more difficult to argue a cause-effect relationship. Quasi-experiments do have the advantage of being easier to set up than a true experiment when it is difficult to randomly assign participants.

33.5 Data Analysis

33.5.1 Making Sense of the Data

Data interpretation is not always obvious. Data collection almost always leads to further questions. In some cases, data will be contradictory due to differences in conditions, users, and the way the data is collected.

Look for Patterns

Single measurements rarely give the entire picture. Interpreting data from a single user is risky as variance between users can be large. Instead look for patterns over multiple users. Be careful of assuming that finding a pattern is the truth as fishing for patterns can lead to false conclusions (Section [33.2.3](#)).

Pay Attention to Outliers

An **outlier** is an observation that is atypical from other observations. Don't automatically disregard outliers without trying to understand first why the outlier occurred. Did the outlier occur because there was an error in the system? If so, this could be the most important signal telling you to fix the error! Sometimes these outliers can provide more insight than more typical data.

Verify through Different Conditions and Signals

Realize the results of an individual experiment are rarely the generalized truth (Section [33.2.3](#)). As the design, implementation, and hardware changes, consideration should be given about how such changes may change the experiment results. If the experiment was for a very specific situation, then vary the conditions and run the experiment again.

33.5.2 Statistical Concepts

Whereas this book does not go into the details of statistical analysis, this section does cover the basics so all team members can have a basic understanding of statistics; have a common language to speak; spot wrong assumptions; reduce the odds of incorrectly summarizing or interpreting

data; and better understand VR research of others and their technical papers. At least one member of the team should be good at statistical analysis. This can be a professional statistician but is more often a programmer or psychologist that is good at math and has experience using statistical software tools.

Measurement Types

Variables have different measurement scale types, and interpreting data can depend on the type properties. Variable scale types are divided into categorical, ordinal, interval, and ratio variables.

Categorical variables (also known as nominal variables) are the most basic form of measurement where each possible value takes on a mutually exclusive label or name. Nominal variables do not have an implied order or value. An example nominal variable is gender (male or female).

Ordinal variables are ordered ranks of mutually exclusive categories, but intervals between the ranks are not equal. An example of an ordinal variable is obtained by asking the question, “How many times have you used VR? A) Never, B) 1–10 times, C) 11–100 times, or D) over 100 times.”

Interval variables are ordered and equally spaced; the difference between values is meaningful. Interval variables do not have an inherent absolute zero. Temperature in Fahrenheit or Celsius and time of day are examples of interval variables.

Ratio variables are like interval variables but also have an inherent absolute zero. This enables meaningful fractions or ratios. Examples of ratio variables are number of users, number of breaks-in-presence, and time to completion.

Table 33.1 shows the ways values can be mathematically manipulated and statistically summarized depending on what measurement type is used. Be careful not to perform inappropriate calculations on data that are not ratio variables. For example, a common mistake is to multiply and divide interval data or to add/subtract ordinal data.

Table 33.1 Appropriate statistical calculations depend on the type of variable being measured. Check marks indicate it is appropriate to use the mathematical calculation with the measurement type.

	Categorical	Ordinal	Interval	Ratio
Counts and percentage	✓	✓	✓	✓
Median, mode, interquartile range	✓	✓	✓	
Addition & subtraction		✓	✓	
Mean & standard deviation		✓	✓	
Multiply & divide			✓	

Descriptive Statistics

Descriptive statistics summarize the main features of a dataset. Described below are some of the most common and useful concepts of descriptive statistics.

Averages. An **average** represents the middle of a dataset that the data tends toward. The average can take on different values depending on how it is computed. The three types of averages are the mean, median, and mode.

The **mean** takes into account and weighs all values of a dataset equally. The mean is the most common form of the average. Means can be overly influenced by an extremely small or large value that rarely occurs (i.e., an outlier). The mean is not appropriate when the data is categorical or ordinal.

The **median** is the middle value in an ordered dataset. The median is often more appropriate than the mean when data is skewed toward higher or lower values or there are one or more outliers. For example, the mean of the values (1, 1, 2, 3, 100) is 21.4 whereas the median is 2. In this case, the median 2 might be considered the more appropriate average of representing the central tendency of the data. The median is also most appropriate when the data is ordinal.

The **mode** is the most frequently occurring value in a dataset. The mode is most appropriate when the data is categorical.

Distribution of data. A **histogram** is a graphical representation of a distribution of data. Figure 33.3 shows an example histogram where participants responded by choosing from an option of seven discrete responses. In the case that measurement is not discrete, values can first be placed into bins (a series of intervals) and then the counts within each bin are displayed.

For some data, the histogram approaches what is called a normal distribution as the number of samples increases (Figure 33.4). A **normal distribution** (also known as a bell curve or Gaussian function) varies in width and height but is well defined mathematically and has useful properties for data analysis. Many statistical tests assume data is normally distributed.

In any experiment, there will always be variations in the collected data. The **spread** (also called variability) of a dataset signifies how dispersed the data is. Measures of spread include the range, the interquartile range, the mean deviation, the variance, and the standard deviation. All of these

measures are always zero or greater, where a value of zero signifies all values within the dataset are identical. The larger the spread, the more the data is spread out.

The **range** is simply the minimum and maximum values of a dataset. The **interquartile range** is the range of the middle 50% of the data. The interquartile range is often preferred to the total range since it automatically removes outliers and gives a better idea of where most of the data is located.

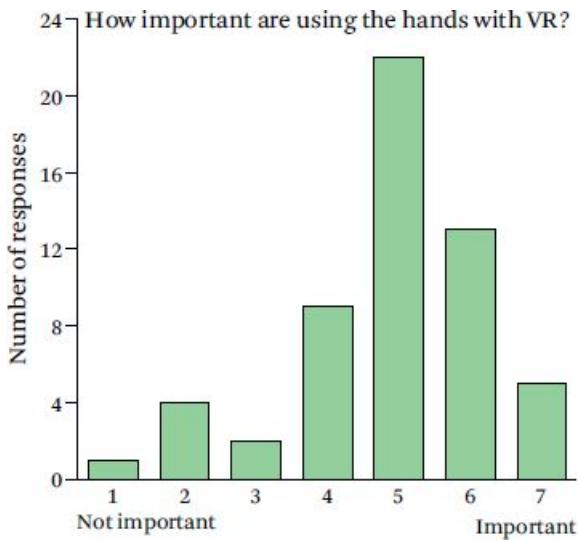


Figure 33.3 A histogram of responses from a questionnaire to VR experts inquiring about the importance of using the hands with VR.
(Courtesy of NextGen Interactions)

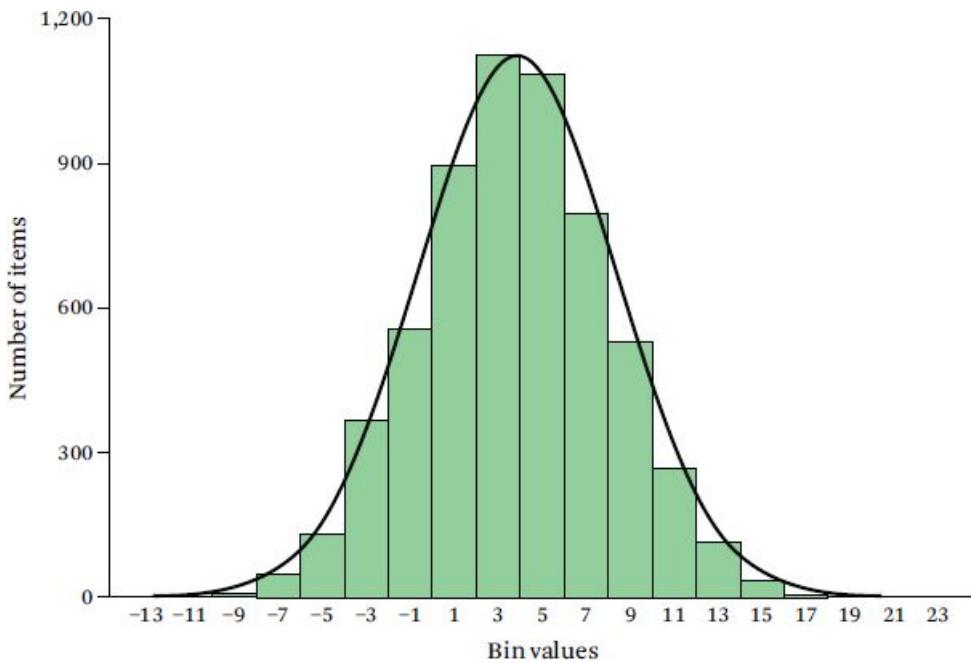


Figure 33.4 A dataset with an approximately normal distribution of data.

The mean deviation (also known as the mean absolute deviation) is how far values are, on average, from the overall dataset mean. The variance is similar, but is the mean squared distance from the mean. The standard deviation is the square root of the variance and is the most common measure of spread. The standard deviation is useful for several reasons, one being that it is in units of the original data (like the mean deviation). Another benefit is, assuming the data is normally distributed, plus or minus one standard deviation from the mean contains 68% of the data and plus or minus two standard deviations contain 95% of the data. For example, a VR training task might have taken on average 47 seconds to complete with a standard deviation of plus or minus 8 seconds, i.e., 68% of users completed the task between 39 and 55 seconds and 95% of users completed the task between 31 and 63 seconds.

Correlation

A correlation is the degree to which two or more attributes or measurements tend to vary together. Correlation values range between -1 and 1 . A correlation equal to zero means no correlation exists. A correlation of 1 is a perfect positive correlation—as one value goes up the other value always goes up. Likewise, a negative correlation exists if as one value goes up the other value always goes down. Values between -1 and 1 signify that as one value changes the other value tends to change, but not always. It is important to note that correlation is not sufficient for proving causation. There could be other reasons the data correlates.

Statistical and Practical Significance

Statistical significance states that with some confidence (a 95% confidence level is normally used) the statistical result of an experiment did not occur by chance and thus there is some underlying cause for the result. If a coin was flipped 10 times and it came up heads every time, we can be fairly certain there was some event or property of the coin that caused it to turn up heads. However, there is a possibility this occurred by chance no matter how many times the coin was flipped and it came up heads. The p-value of an experiment represents the probability that a result occurred by chance rather than because there is something inherently truthful about the result. A p-value less than 0.05 (corresponding to a 95% confidence level) is typically considered to be statistically significant.

Statistical significance does not necessarily mean the result is important enough to influence the next iteration of the project. Practical significance (also called clinical significance) implies the results are important enough that the results matter in practice. For example, time to completion for a VR task might be statistically shorter for some condition by 0.1 seconds. However, if the average time to completion is 10 minutes, then the 0.1-second time improvement is probably not of practical significance and not worth being concerned about. If the 0.1 second improvement is a reduction in system latency from 0.15 second, then it is certainly of practical significance.

Iterative Design: Design Guidelines

Designing iteratively is more important for VR than for any other medium due to there being so many unknowns. Do not expect to find or create a single overall detailed VR design process that works across all types of VR projects. Instead, learn the different processes to know when they are most appropriate to use. Focus on continuous iteration of defining the project, making the application, and learning from users.

34.1 Philosophy of Iterative Design (Chapter [30](#))

Human-Centered Design ([Section 30.2](#))

- Focus on the user experience.
- Give up traditional measures of software development such as lines of code.

Continuous Discovery through Iteration ([Section 30.3](#))

- Give up on knowing and rationalizing everything from the start.
- Collect feedback as quickly and as often as possible from experts and representative users.
- Fail early and fail often in order to learn as quickly and as much as possible.
- If failure is not occurring often in early iterations, then attempt to innovate more.
- Always be on the lookout for changes in assumptions, technology, and business. Then change course as necessary.
- Step into the role of the user and try a broad range of VR experiences.
- There is more value in getting a bad prototype working than spending days thinking, considering, and debating.
- Create a culture of accepting failure, especially early in the project to create a feeling that it is safe to experiment.
- Regard feedback and measurement from real application above opinion based on theory.

There Is No One Way—Processes Are Project Dependent ([Section 30.4](#))

- Prioritize team members and communication between team members over sticking to a formal process.
- Prioritize responding to change over following a plan.
- If it is not clear what processes to go with, then just pick some and go with them. Then iterate on improving them or change the processes.
- Do not become attached to processes. Be willing to throw them away if they are not working.

Teams ([Section 30.5](#))

- Keep teams small to maximize communication.
- If the team starts to become large, then break into smaller teams.
- Make sure everyone is on board with the team philosophy.
- All team members should actively co-create, not just critique.
- Create a culture of learning from other team members, external suggestions from those experienced with VR, other disciplines, and user feedback.
- If any team member is close minded to ideas that are not his own, warn him and if he doesn't change remove him from the project as soon as possible.
- Collect input from a wide range of opinions and viewpoints but do not make decisions by committee.
- Assign an authoritative director who listens well but acts as a final decision maker on high-level items.
- If it is too difficult for a single person (the director) to understand how everything is connected, then the design is overcomplicated. Simplify the design.

34.2 The Define Stage (Chapter [31](#))

- Do not spend too much time on the Define Stage in the first iteration.
- Be careful of analysis paralysis of trying to figure everything out before pulling the trigger.
- If not sure about moving on to the Make Stage, then move on to the Make Stage. Return to the Define Stage at a later time.
- In addition to documenting *what* decisions are made, document *why* decisions are made.

The Vision (Section [31.1](#))

- If some aspect of the project is unknown then guess. Articulated guesses beat unspoken or vague models.

Questions (Section [31.2](#))

- Get input from multiple individuals including all teammates, stakeholders, representative users, other teams, business developers, marketers, and expert consultants.
- Don't expect others to straight out tell you what they want. Help them discover what they want by asking questions.
- Ask questions that start with the word "why."
- Remember customers don't want an HMD, they want an experience and/or results.
- Understand more than just the VR aspect of the project. Understand the background and context of the project.

Assessment and Feasibility (Section [31.3](#))

- Do not assume VR is the right solution for every problem.
- Consider having the project consist of a mix of both VR and the real world.

High-Level Design Considerations (Section [31.4](#))

- Decide if you are focusing on design *with*, *for*, or *of* virtual environments.

Objectives (Section [31.5](#))

- Focus on benefits and outcomes instead of features.
- When creating objectives, use the SMART acronym—Specific, Measurable, Attainable, Relevant, and Time-bound.

Key Players (Section [31.6](#))

- All key players should be in agreement with the vision, really care about the project, and be committed to its success.
- Understand that many people may not have an interest no matter how good of a fit you think they might be. Listen to what they have to say and move on. Do not waste time chasing such individuals.

Time and Costs (Section [31.7](#))

- When negotiating contracts, separate out initial assessment and feasibility from implementation.
- Consider using milestones where additional payment is only received after achieving agreed-upon milestones.
- Remember Brooks' Law—adding manpower to a late software project makes it later.
- Play planning poker to estimate development effort.
- Understand developers tend to overestimate their abilities even when they know they overestimate their abilities.

Risks (Section [31.8](#))

- Explicitly identify risk to bring awareness of any danger that might affect the project so that appropriate action can be taken to mitigate that risk.
- Focus on minimizing controllable risk.

- To reduce risk, make projects as short as possible. If a project is large, break it down into smaller subprojects.

Assumptions (Section [31.9](#))

- Explicitly look for and declare assumptions so all team members begin from a common starting point.
- Be bold and precise in listing assumptions even when not sure. Wrong assumptions can be tested, vague assumptions cannot.
- Prioritize assumptions to test the riskiest assumptions first.

Project Constraints (Section [31.10](#))

- Make a single individual responsible for tracking and controlling constraints, with transparency to the entire team.
- Explicitly list constraints to narrow the design space.

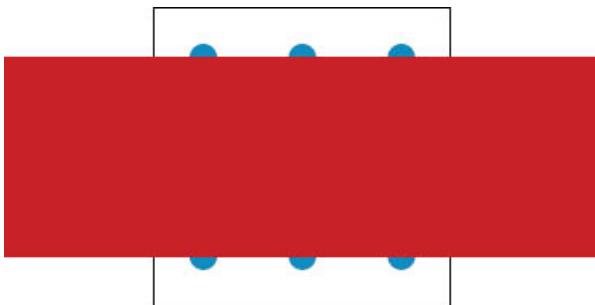


Figure 34.1 A solution to Figure 31.8 of using only a single line to cover all nine dots. The problem never stated a limit to the width of the line.

- Divide constraints into real constraints, resource constraints, obsolete constraints, misperceived constraints, indirect constraints, and intentional artificial constraints.
- Think outside the box by listing all misperceived constraints. Figure 34.1 shows a solution to Figure 31.8.

Personas (Section [31.11](#))

- Making generalizations to all people makes building a VR experience difficult. Target personas to make the design easier.
- Validate and modify personas on later iterations as more is learned about those actually using the system
- For applications where personas are especially important (e.g., therapy applications), collect data with interviews and questionnaires.

User Stories (Section [31.12](#))

- Write user stories in the form “As a <type of user> I want <some goal> so that <some reason>” to define who, what, and why.
- Create user stories that consist of the INVEST acronym: Independent, Negotiable, Valuable, Estimable, Small, and Testable.

Storyboards (Section [31.13](#))

- Show users directly interacting in storyboards without worrying about screen layout details since the screen perceptually goes away with fully immersive VR.
- Don’t limit storyboards to be linear. Storyboard frames might be connected in multiple ways.

Scope (Section [31.14](#))

- In addition to stating what will be done, explicitly state what will not be done so that the team can focus on what is important.

Requirements (Section [31.15](#))

- Requirements convey expectations of the customer; have the client and/or other key players actively involved in defining requirements.
- Requirements should be written in language that can be easily understood by all parties involved.
- Each requirement should be a single thing the application should do.

- Each requirement should be complete, verifiable, and concise, yet with room for innovation and change.
- Any attempt to formulate all possible requirements at the start of a project will fail and would cause considerable delays.
- Work with the client and/or partners when changes are necessary.
- Include universal requirements such as fading the screen out when latency rises above some threshold or when tracking is lost.

34.3 The Make Stage (Chapter [32](#))

- Makers should not develop VR applications without full access to VR hardware.
- Use existing frameworks and tools unless there is a good reason not to do so.

Task Analysis (Section [32.1](#))

- Task analysis provides organization and structure for description of user activities, which make it easier to describe how activities fit together.
- Use task analysis to understand what tasks must be carried out, to document descriptions of activities, and to look for how to improve processes.
- When the goal is to replicate actions in the real world (e.g., training applications), start with the real-world task instead of the not-yet-created VR task.
- Find representative users by looking for users who match previously created personas.
- Interview users, domain experts, and visionary representatives to gain insight into what users need and expect.
- Administrate questionnaires, observe experts in action, review existing documentation, and observe VR users trying prototypes.
- Carefully think about each stage of the cycle of interaction.
- Ask ‘how’ questions to break tasks into subtasks.
- Ask ‘why’ questions to get high-level task descriptions and context.
- Ask what happens before and what happens after to obtain sequential information.
- Use graphical depictions when talking with those whom information is being extracted from in order to collect higher-quality information.
- Don’t be fooled into thinking there is a perfect way to do task analysis.
- After organizing and structuring the task, review with those users whose information was solicited to verify understanding.

Design Specification (Section [32.2](#))

- In the early stages of specifying the design, explore different designs before converging and committing to a final solution.

Sketches (Section [32.2.1](#))

- Good sketches are quick and timely, inexpensive and disposable, plentiful, suggestive and explorative, and partially ambiguous. Sketches also have a distinct gestural style, have an appropriate degree of refinement, and contain minimal detail.

Block Diagrams (Section [32.2.2](#))

- Use block diagrams to show the overall relationship of components without concern for the details.

Use Cases (Section [32.2.3](#))

- Define use cases to help identify, clarify, and organize interactions in a way that makes them easier to implement.
- Outline use cases with hierarchical levels to provide both high-level and more detailed views. Fill in details as needed.
- Use color and/or different font types to easily see what needs to be implemented, what has already been implemented/tested, differences between paths, and/or who is assigned to what steps.

Classes (Section [32.2.3](#))

- Organize information with common structure and functionality from earlier steps into themes and then turn those themes into classes.
- Use class diagrams to state what developers need to implement.

Software Design Patterns (Section [32.2.5](#))

- Reuse software design patterns to solve commonly occurring software architecture problems.

System Considerations (Section [32.3](#))

System Trade-Offs and Making Decisions (Section [32.3.1](#))

- Don't try to support everything by implementing for the “average” option as that will result in no single optimal experience. Instead implement different metaphors and interactions that appropriately fit each option.
- Start by supporting single options and optimize them before adding secondary options.

Support of Different Hardware (Section [32.3.2](#))

- Attempting to support different hardware classes most often results in the experience being optimized for none of them.
- If different hardware classes must be supported, then the core interaction techniques should be independently optimized for each to take advantage of their unique characteristics.

Frame Rate and Latency (Section [32.3.3](#))

- Maintain a frame rate from the beginning that matches or exceeds the refresh rate of the HMD.
- Carefully observe the frame rate as new assets are added and code complexity increases. Even occasional frame drops can be uncomfortable for users.
- Do not only rely on frame rate to determine latency. Measure end-to-end latency using a latency meter.

Sickness Guidelines (Section [32.3.4](#))

- It is as important for programmers to respect adverse health effects as it is for anyone on the team even if they are not prone to sickness themselves.
- Don't wait for the Learn Stage to fix sickness-inducing problems.

Calibration (Section [32.3.4](#))

- Enable easy or automatic calibration—proper calibration is essential.

Simulation (Section [32.4](#))

Separate Simulation from Rendering (Section [32.4.1](#))

- Perform rendering asynchronously from simulation.
- Slow simulation updates are often acceptable as long as rendering from head motion occurs with low latency.
- For realistic physics simulations, use a fast update rate.

Fighting Physics (Section [32.4.2](#))

- The solution to fighting physics is to not fight. Do not attempt to have two separate policies simultaneously determine where an object is.
- Stop applying simulated forces to an object when the object is picked up by the user.

Jittering Objects (Section [32.4.3](#)) and Flying Objects (Section [32.4.4](#))

- Use physics “cheats” to reduce object jittering and objects flying off into space.
- Avoid having multiple physically simulated objects interacting whenever possible, especially if the objects are in an enclosed space or are tightly constrained against one another (e.g., a stack of blocks in a small pit or multiple boxes joined together in a circular fashion).

Networked Environments (Section [32.5](#))

Ideals of Networked Environments (Section [32.5.1](#))

- Focus on minimizing divergence, causality violations, and expectation violations while maximizing responsiveness and perceived continuity.

Message Protocols (Section [32.5.2](#))

- Use UDP when low latency is a priority, updates occur often, and each update is not essential. Examples of when to use UDP are when continually updating character position and audio.
- Use TCP when state information is only sent once (or occasionally) to ensure the receiving computer can update its state of the world to match the sender's state.

Network Architecture (Section [32.5.3](#))

- For fast responsiveness, use peer-to-peer networking.
- To ensure all users' worlds are consistent, use a fully authoritative server.
- Use hybrid architectures to take advantage of both peer-to-peer architectures and client-server architectures.

Determinism and Local Estimation (Section [32.5.4](#))

- For partially deterministic actions, use dead reckoning to estimate where remotely controlled entities are currently located.
- Use interpolation to reduce perceptual discontinuities where objects appear as jumping to new locations when new network packets are received.

Reducing Network Traffic (Section [32.5.5](#))

- To reduce network traffic, compute divergence of the local and true state and only send updates when divergence surpasses some threshold. Relevance filtering can also be used to only send relevant information to each computer.
- Conduct audio stress tests by having many users speak simultaneously.
- Use relevance filtering to only send audio to nearby users.
- If necessary, allocate less bandwidth to users who are further away or not in the general forward-looking direction.
- Use animations whenever possible instead of continuous updates.

Simultaneous Interactions (Section [32.5.6](#))

- Use an object token that can only be owned by a single player at a time to prevent users from simultaneously interacting with the same object.

Networking Physics (Section [32.5.7](#))

- When simulating physics across networks, have a single authoritative computer own the simulation (although other computers can estimate the simulation).

Prototypes (Section [32.6](#))

- Use prototypes to learn from what users do, not just from what they say.
- Begin with minimal prototypes, i.e., start with the least amount of work necessary to receive meaningful feedback.
- Focus on getting something working as quickly as possible and then modify and add additional functionality/features on to the minimal prototype or restart from scratch.
- Have a clear goal for each prototype.
- When building minimal prototypes, give up on trying to look good. Expect that it won't be done well on the first several attempts.

Forms of Prototypes (Section [32.6.1](#))

- As a first step, consider using real-world prototypes (e.g., physical props and no digital technology) with team members acting out roles.

- Use Wizard of Oz prototypes where an unseen team member enters commands on a workstation.
- Get representational user prototypes working as soon as possible and get feedback from as many naive users as possible.
- Focus on building the prototype that best collects data for what is being targeted.

Final Production (Section [32.7](#))

- When final production begins, stop exploring possibilities and stop adding more features. Save those ideas for a later deliverable.
- Be open to feedback from stakeholders, but make it clear that there is a limit to what is possible within time and budget constraints. There are trade-offs involved and adding new features may come at the cost of delaying other features.

Delivery (Section [32.8](#))

Demos (Section [32.8.1](#))

- Before traveling to give a demo, set the demo up in a room different from where equipment is normally located to make sure no equipment is missed when packing.
- Pack two backups of everything!
- Set up in the demo space the day before and confirm everything is working in the morning.
- Be prepared for in-house demos for when important people show up who can add value to the project. This does not mean anyone should be given a demo at any time; demos can take up valuable resources that could be devoted to the project.
- Have a single person be responsible for maintaining and updating in-house demos. Have a single person (possibly the demo maintainer) be responsible for scheduling demos.

Onsite Installation (Section [32.8.2](#))

- Do not expect installation for non-expert VR customers to go smoothly.
- For complicated installations with multiple pieces of hardware, be prepared to calibrate, modify source code, and use electrical/gaffer tape. Issues might include electromagnetic interference and incompatible connections/cables with a client's system.
- Offer a service to train staff in using and maintaining the system. What may be obvious to you may not be obvious to them.
- Offer support and updates as part of a contract.

Continuous Delivery (Section [32.8.3](#))

- When delivering online, consider offering builds on a weekly basis. This will force the team to be held accountable and to focus on what is important, as well as to provide plenty of opportunity for data collection.

34.4 The Learn Stage (Chapter [33](#))

- Utilize VR experts, subject-matter experts, usability experts, experiment-design experts, and statisticians to ensure you are doing the right things to maximize learning.
- Lean toward fast feedback, data collection, and experimentation. This enables the team to learn quickly how well or not ideas meet goals, and to correct course immediately.

Communication and Attitude (Section [33.1](#))

- Seek criticism as if the success of the project and your career depends on it. Because it does!
- Have a positive attitude toward constructive criticism.
- Think positively for yourself, team members, and users.
- Consider failure to be a learning experience and do not fear it.
- Do not blame/belittle users or their opinions/interactions.
- Actively investigate difficulties to determine how the project can improve.

- Assume what others are doing is partially correct and then provide suggestions that enable them to correct and move on.
- When someone points out a problem you are already aware of, thank them for noticing and encourage them to continue looking for problems.

VR Creators Are Unique Users (Section [33.1.1](#))

- Do not assume what works for you will work for everyone else. You likely won't try all cases and have grown immune to motion sickness.
- Programmers should at least occasionally participate in data collection so that they take feedback more seriously.

Research Concepts (Section [33.2](#))

Data Collection (Section [33.2.1](#))

- Do not get stuck in collecting a single type of data. There are many measures that can be used.
- First collect qualitative data to get a high-level understanding before collecting more objective quantitative data.

Reliability (Section [33.2.2](#))

- Never depend on a single measurement.
- Measure across different participants, sessions, times, and experiments to gain confidence that there is some consistent characteristic for that which is being studied.
- Remember the true score can never be exactly known, although a large number of measurements can provide a good estimate.

Validity (Section [33.2.3](#))

- Make sure you are measuring and comparing what you think you are measuring and comparing.
- Ideally, a measure should cover all aspects of the construct; correlate with other measures of the same construct; not be affected by variables not related to the construct; and accurately predict other measurement, behavior, and performance.
- A violation of construct validity has occurred when a system artifact has been measured when that is not the intention of the measurement.
- Do not conclude a concept is not effective in general just because it does not work in its current implementation.
- Reduce the chance of coming to incorrect conclusions by carefully considering threats to internal validity. For example, randomize assignment of conditions, keep the setup the same across data collection sessions, remove practice effects, remove any cues that enable participants to guess the hypothesis, and make the experimenter blind to experimental conditions.
- Be aware of threats to statistical conclusion validity to reduce the chances of making erroneous conclusions. For example, a statistical result may have happened by chance, not finding a result does not mean a result does not exist, assumptions of data may not be valid, and data mining increases the odds of finding something by chance.
- Do not assume a finding holds across different settings, other users, other times, and other designs/implementations.
- Collect data again after hardware has changed or significant design changes have occurred.

Sensitivity (Section [32.2.4](#))

- Increase the sensitivity to that which is being tested in order to increase the chance of finding an effect if it exists.

Constructivist Approaches (Section [33.3](#))

Mini-Retrospectives (Section [33.3.1](#))

- Conduct mini-retrospectives—short team discussions of what is going well and what needs improvement.
- It is better to have many short and consistent mini-retrospectives than occasional long retrospectives.
- Respect all team members and never make a retrospective a witch hunt.
- Focus on creating themes for future iterations and areas to track that the team wants to improve upon.

Demos (Section [33.3.2](#))

- Distinguish the difference between collecting user data (where the primary goal is to learn) and giving demos (where the primary goal is to market). Do not rely solely on demos for receiving feedback.

Interviews (Section [33.3.3](#))

- An interview results in the best data when given immediately after the person being interviewed experienced the VR application.
- Create interview guidelines in advance in order to keep the interview on track and to reduce bias.
- Attempt to match the interviewer with the intended audience based on previously created personas.
- Conduct interviews in comfortable and natural settings (e.g., a mock living room).
- Don't spend more than 30 minute per interview. Set time limits by scheduling interviews back to back. Individuals can be followed up with if they have more useful things to say.

Questionnaires (Section [33.3.4](#))

- Use questionnaires when ease of administration and privacy are important.
- Ask for background information to determine if participants fit the target population defined by previously created personas, to help better define personas, to make sure a breadth of users is providing feedback, to match participants when comparing between participants, and to look for correlations with performance.

Focus Groups (Section [33.3.5](#))

- Use focus groups in the early phases of VR design to explore new concepts, better refine questions, and improve upon data collection processes.
- Group settings are often more efficient than individual interviews and can stimulate thinking as participants build off of each other's ideas.

Expert Evaluations (Section [33.3.6](#))

- When done properly, expert evaluations are the most efficient and cost-effective method of improving upon the usability of a system and for iterating toward ideal solutions.
- Start early in the project with expert guidelines-based evaluations in order to correct issues before they affect other aspects of the design.
- Conduct formative usability evaluations during the formative and evolving stages of design in order to assess, refine, and improve usability, learning, performance, and exploration.
- Formative usability evaluation should be conducted by VR usability experts as it relies heavily on a solid understanding of VR interactions that vary depending on context (i.e., it consists of more than a predefined list of items to evaluate).
- Pay special attention to critical incidents and make it the highest priority to fix issues causing these incidents.
- Design a scoring system apparent to users in order to both increase motivation and target data collection.
- During the final stages of formative usability evaluation, only observe and do not suggest how to interact with the system.
- Use comparative evaluation to compare two or more well-formed complete or near-complete implementations. This is useful for choosing the best techniques for final integration and to test if a new system is better than a previous system (e.g., Does the new VR training system result in more productivity than traditional training?).

After Action Reviews (Section [33.3.7](#))

- Debrief users with an emphasis on the specific actions they took during the VR experience.
- Discuss what happened, why it happened, and how it could be done better.
- Discuss with the user the actions she was taking while watching a recording from both first-person and third-person views.

The Scientific Method (Section [33.4](#))

- Even if you do not conduct formal experiments, learn the basics in order to design more informal experiments, to comprehend research conducted by others, to be aware of some of the pitfalls that might occur in collecting data, and to properly interpret results.
- The first step of exploring a problem is to obtain an understanding of what is to be studied. Do this by learning from others (via trying their applications, reading research reports/papers, and talking with them); by trying oneself and observing others interacting with existing VR

applications; by constructivist approaches; and by the various concepts discussed in the Define and Make Stages.

- Once familiar with the overall problem, start by asking and answering high-level questions.
- Precisely state the hypothesis in a way that predicts the relationship between two variables.
- Be careful of confounding factors. Consider threats to internal validity to find confounding factors. Once found, set confounds to be constant with control variables.
- Use a within subjects design when only a small number of participants are available and carryover effects (e.g., learning, fatigue, sickness) are minimal and/or participants can be counterbalanced.
- Use a between subjects design when many participants are available, their availability is short, and carryover effects are a concern.
- Always conduct informal pilot studies before conducting a more formal and rigorous experiment in order to help determine feasibility; discover unexpected challenges; improve upon the experiment design; reduce time and cost; and estimate statistical power, effect size, and sample size.
- When random assignment is difficult, consider quasi-experiments instead of true experiments at the risk of adding confounding factors.

Data Analysis (Section [33.5](#))

Making Sense of the Data (Section [33.5.1](#))

- When first looking at data, start by looking for patterns but be careful of assuming a pattern is the truth as fishing for data can lead to false conclusions.
- Pay attention to outliers as an outlier can provide more insight than more typical data. Often an outlier is an important signal telling you to fix an error.
- The result of a single experiment is rarely the generalized truth. Do not base final decisions on a single experiment and consider how changes may affect results.

Statistical Concepts (Section [33.5.2](#))

- All team members should be familiar with basic statistical concepts so a common language is spoken, wrong assumptions and errors can be reduced, and research from others can be understood.
- At least one member of the team should be good at statistical analysis.
- Pay attention to measurement types and only perform appropriate calculations on each. For example, do not perform multiplication on interval data and do not add ordinal data.
- For averages, use the median when data is skewed to not take into account outliers or when data is ordinal. Use the mode when data is categorical.
- Visualize data with histograms to intuitively understand the spread of data.
- Understand the differences of statistical significance and practical significance. Statistical significance is not necessarily meaningful.

PART VII

THE FUTURE STARTS NOW

The ones who are crazy enough to think that they can change the world are the ones who do .

—Steve Jobs

This book has focused on providing a high-level overview of VR and dived into detail on some of the most important concepts. Even for those who have read the book in its entirety, it is only a starting point. There is much more detailed information in the provided references, and even those references are a small proportion of VR research that has occurred over the years. Then there is the general study of neuroscience, human physiology, human-computer interaction, human performance, human factors, etc. The list goes on. However, this in no way means that everything is understood. The field of VR and its many applications is completely wide open. We are in no danger of running out of new possibilities. And hopefully we never run out of those possibilities.

In the words of one of my former professors, Henry Fuchs, from his IEEE Virtual Reality keynote address, “We now have the opportunity to change the world—let’s not blow it!” (Fuchs, 2014). The future is here now and it is accessible by anyone, so don’t be left behind. It is up to you to use VR technology in whatever way you wish in order to help shape the future.

Part VII consists of two chapters that look into the future and provide some basic steps for how to get started in actually creating VR applications.

Chapter 35, The Present and Future State of VR, discusses where VR currently stands and where it may lead. Topics include the culture of VR, new generative languages, new indirect and direct ways of interacting, standards and open source, emerging hardware, and the convergence of AR with VR.

Chapter 36, Getting Started, is a short chapter explaining how to get started building VR applications. A schedule of tasks is provided that shows how anyone, even those with limited time, can quickly create new virtual worlds.

The Present and Future State of VR

In 2015, low-cost consumer VR technology is surpassing professional VR/HMD systems. A couple of years ago, even with an unlimited budget, you could not buy a system with the resolution, field of view, low latency, low weight, and overall quality that is now available at a price that anyone can afford. In addition, tools are accessible by anyone and are quite easy to use. This is resulting in the democratization of VR so that anyone can take part in defining where VR is going. VR is no longer a tool only for academics or corporate entities. The best VR technologies and experiences are just as likely to be created by a team of indie developers as by a Fortune 500 company.

Some of the challenges described in this chapter will open the door for the greatest opportunities over the next few years and beyond. This chapter describes some of the most exciting possibilities where VR is already going and what the future of VR might look like.

35.1 Selling VR to the Masses

Although VR is certainly an exciting industry to be in and many people know about it, the general population is not yet buying it. Instead, VR is being sold to innovators as a technical marvel. However, technology is not what is important for most people. In the future, the manner in which the story, emotions, and benefits are sold will need to be more compelling to reach a wider audience. How will this be done in a way that is more engaging to the masses so that they demand VR?

- Through entertainment experienced in a way no other technology can provide?
- Through networked worlds that ease and enhance social sharing?
- Through fulfilling people's needs and making life easier?
- Through enhancing quality of life via immersive health care and physical/mental exercise?
- Through cost savings and increased profitability?
- Through new entrepreneurship ventures that we can't yet even begin to fathom?

Focusing on any one of these is certainly a noble effort. Each will likely play a big part in changing many people's lives.

35.2 Culture of the VR Community

In some ways, VR culture is already forming. Consider the VR Meetup phenomenon.¹ In 2012, Karl Krantz, a self-proclaimed introvert, began to contact what he thought would be a small number of local VR professionals and enthusiasts to start a VR Meetup to share and discuss their creations. Today, Karl has evolved into somewhat of a VR celebrity. Not only has his local meetup expanded to over 2,000 members, but his model of allowing anyone to showcase their VR experiences at public events has been replicated throughout the world. Pockets of hundreds of VR enthusiasts in every major city gather to share their virtual worlds and discuss the latest VR trends on a regular basis. Non-VR meetups often struggle to have more than a few people attend their events, yet all one needs to do for a VR Meetup is to set up an account, find a location, set a date, send out some tweets and emails, and people will flock to the event. In the San Francisco Bay Area alone, there are so many different VR Meetups per month that it is difficult to keep up with all of them. Whereas the popular media has portrayed VR as being anti-social, these meetups have proven VR to be the social lubricant that gets normally introverted individuals passionately conversing.

Both Brendan Iribe, CEO of Oculus, and Mark Zuckerberg, CEO of Facebook, claim there will someday be a billion or more VR users [[Zuckerberg 2014](#), [Hollister 2014](#)]. In order for this to happen, an entire cultural shift is going to be required for the masses to accept such advanced mind-altering technology. Will this actually happen? It is certain to happen at some point as both humans and technology evolve. The question is, will that be 5 years from now or 50 years from now? In the short term, will it only be the fortunate few who care as is the case in Neal Stephenson's *Snow Crash* or will VR scale more quickly? Even if the Metaverse is not accessed by the hundreds of millions, various forms of VR will certainly add value to some smaller groups of us.

Regardless of who uses VR, cultural patterns of rules, hierarchies, and social constraints will certainly form within virtual worlds. Some of these will be coded into the experiences, but much of the social and cultural structure will evolve naturally just as it does in the real world. What actions will be considered rude and what actions will be considered polite? What will the penalty be for unacceptable behavior? Will the equivalent of government entities form that impose laws for the common good? Only time will tell, but these questions have important implications for the future of VR and should be considered when designing new worlds.

35.3 Communication

The book began with a discussion of how VR at its core is communication (Section [1.2](#)). This section describes how communication in VR will be taken to the next level beyond just interacting with objects as is done with most current VR applications.

35.3.1 A New Generative Language

Languages are not necessarily made of words but are any form of communication (e.g., body language). A **generative language** (also called future-based language), in contrast to a descriptive language, has “the power to create new futures, to craft vision, and to eliminate the blinders that are preventing people from seeing new possibilities” [[Zaffron and Logan 2009](#)]. In short, generative language transforms the way we perceive the world. VR can be thought of as a new generative language because we are creating new experiences that words cannot fully describe and can only be fully conveyed by actually experiencing it oneself. The language of VR will certainly bring about things that would never be created from traditional language alone, our past experiences, or our current understanding.

35.3.2 Symbolic Communication

Direct communication is not always the best way to interact within VR. Indirect **symbolic communication** is the use of abstract symbols (e.g., text) to represent objects, ideas, concepts, quantities, etc. [[Bowman et al. 2004](#)]. Consider how we use indirect symbolic communication in the real world. Symbolic communication enables us to convey information efficiently, precisely, and concisely; provides a way to clearly think, both inside our minds and in the physical world (e.g., paper and whiteboards); and allows structured data to persist throughout time.

In VR, symbolic output is quite easy to do and is used quite effectively. However, symbolic input is not so straightforward. Other than the widgets and panels interaction techniques described in Section [28.4.1](#) that work well for small numbers of options, symbolic input for more generalized input is rarely used. There are currently no obvious ways to symbolically input data in a way that is effective, efficient, and elegant.

Inventing new widgets can certainly help. Consider a touch wheel interface on a 2D tablet touch screen for entering dates. Such a widget works extremely well for touch screens but is not great for a mouse/keyboard interface. We need equivalent symbolic input techniques specifically for VR. And we need to go beyond that. Theoretically, speech, gestures, chord keyboards, and pinch input are ideal for symbolic input. However, there are multiple practical challenges to getting such theoretical concepts to work well within VR.

Speech recognition works well in controlled conditions. However, as many can attest to, talking to a voice prompt system via telephone can be extremely frustrating. Even if a system perfectly recognized the spoken word, there still remain significant challenges. In addition to semantic parsing and understanding, challenges include the context words are spoken in (consider the phrase ‘Put it to the left’), some people being resistant to using such systems due to lack of privacy, a perception of bothering others, and feeling awkward talking with a machine. It remains to be seen if speech recognition becomes more common within VR.

Gestures have similar challenges as voice recognition. In addition to error rates, fatigue can be an issue. Better tracking of the entire hand will certainly help so large dynamic gestures will not be required as is the case with Microsoft Kinect. It remains to be seen if reliability from camera-based systems is inherently a largely unsolvable physics problem due to line-of-sight challenges or if that can somehow be worked around. Gloves may become more common as gesture recognition improves and users accept the inherent encumbrance of wearing hardware. Attempts have been made with sign language recognition but have failed largely due to inaccurate and unreliable finger tracking. However, as gesture recognition continues to improve, sign language may indeed become a useful method of symbolic input. A place to start could be with numeric input since only a small number of gestures would need to be recognized and easy to learn since we already understand how to count with our fingers. The system must recognize more gestures than just the digits 0–9, such as to start numeric entry and delete an entry, but not many.

Many challenges have nothing to do with technical capability. The best design does not always win. Consider how smartphones first took input from a 12-digit matrix phone design that maps multiple presses on each digit to alphanumeric input (or word completion that attempts to guess what the user intended by mapping multiple presses to words in a dictionary). Today, smartphones still take symbolic input through a QWERTY keyboard touch interface that was originally designed for ten fingers but is now used with one or two fingers. Swipe pattern recognition as the finger stays on the screen makes it more efficient, but this interface is certainly not ideal. Like the smartphone industry, transition to better symbolic input within VR will take time and will evolve along a non-optimal path. Such a slow response to designing for the medium has already been proven to also be the case for VR, as most VR widgets are just replications of desktop widgets.

Touch can be extremely important for symbolic input. Haptics will help by conveying feedback such as pushing a key on a virtual keyboard. Buttons on chord keyboards inherently provide physical feedback from the feel of the button. Such capability can easily be integrated into tracked hand-held controllers. Pinch gloves also provide the sense of touch when two fingers touch. However, there is a steep learning curve to learning complex new input patterns. If there becomes enough value and demand for reliable symbolic input, then perhaps users will accept the pain of change (such as people were willing to learn to use typewriters).

35.3.3 Creating Empathy with Virtual Humans

Science fiction has predicted humans would live and interact with computer entities that resemble humans in natural ways, even to the point of developing an emotional relationship with those artificial entities. That is now already becoming a reality. Researchers from the University of Southern California Institute for Creative Technologies have developed such human-like entities complete with interactive social skills and connection/rapport-building capability. The system works by sensing the user’s emotional state and responding appropriately for psychotherapy applications (Figure [35.1](#)). Those who believed they were interacting with a computer-controlled virtual human versus a human-controlled virtual human reported a lower fear of self-disclosure, reported lower impression management (only disclosing positive information), displayed sadness more intensely, and were rated by observers as more willing to disclose [[Lucas et al. 2014](#)]. Although challenges certainly exist to integrate such a system within fully immersive VR (e.g., how do the sensors detect emotions conveyed on the face when half the face is covered by an HMD?), it is certainly only a matter of time when such capability is fully integrated into fully immersive VR. With fully immersive VR, virtual humans appear more life-size and real, rather than being confined to a screen, interacting with users as if they were real entities as envisioned by science fiction writers. It is unknown where this might eventually lead, but there is certainly potential to integrate artificial intelligent entities with VR that could result in helping real people overcome barriers that occur with traditional human-to-human communication.

35.3.4 Brain-to-Brain Communication

Researchers from Neuroelectrics and other institutions recently conducted a proof-of-concept experiment where individuals were able to directly convey information through brain-to-brain communication without intervention of motor or peripheral sensory systems (Figure 35.2) [Grau et al. 2014]. Binary streams of encoded words were transmitted between the remote minds of emitter and receiver participants, representing the realization of a human brain-to-brain interface. This was done by capturing voluntary motor imagery-controlled electroencephalographic (EEG) changes from one participant, encoding the information, and sending it from India to a remote participant in France through the Internet. The information was transformed to signals that conveyed to the receiving participant the conscious perception of phosphene (light flashes) through neuronavigated, robotized transcranial magnetic stimulation (TMS).

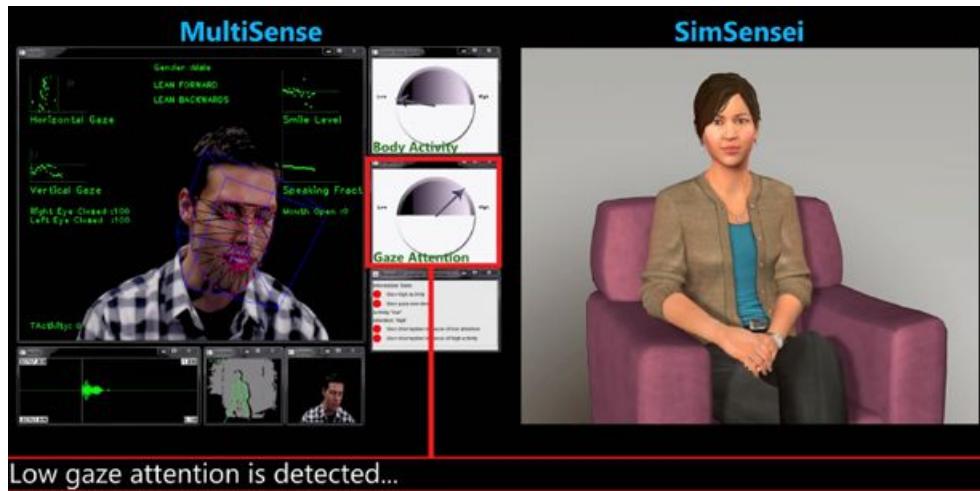


Figure 35.1 A computer-controlled virtual human (right) builds empathy with a user (left) by sensing his state and responding appropriately. (Courtesy of USC Institute for Creative Technologies. Principal Investigators: Albert (Skip) Rizzo and Louis-Philippe Morency)

The results provided a demonstration for the development of conscious brain-to-brain communication technologies. More fully developed implementations will open new research venues in cognitive, social, and clinical neuroscience and the scientific study of consciousness. Long-term possibilities include being able to modify one's sensory VR experience by thought alone. Such communication technologies will certainly have a profound impact on more than just VR technology.

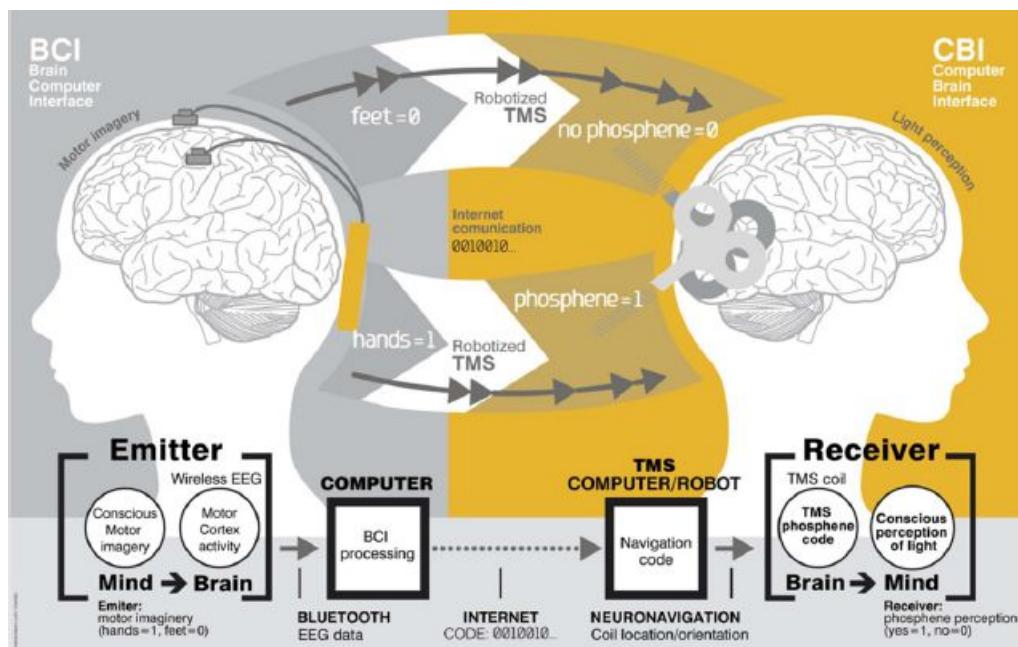


Figure 35.2 The brain-to-brain communication system as described in [Grau et al. \[2014 \]](#).

35.3.5 Full Neural Input and Output

Direct neural input through some standardized connection, as done in *The Matrix* trilogy, is certainly many years away. The transition will be very gradual, starting with neural input and output as done with Neuroelectrics described above.

Visual displays are already moving closer to the brain via virtual retinal displays where a laser draws images onto the retina [[Schowengerdt et al. 2003](#)] as done by the company Magic Leap. Visual displays will eventually move to contact lenses, starting with augmented reality, but eventually providing the capability to not only draw on the world but to completely block out the real world. The next step will be either a projector located

inside the eye or signals sent directly to the retina. In fact, multiple companies have already started to pursue direct retinal stimulation. Simple retinal stimulators providing a matrix display of 60 “pixels” have already been successfully implanted in blind patients, enabling them to see basic shapes, motions, and even letters/words in the real world (via mapping from a digital camera mounted on glasses to the implanted electrodes) [[da Cruz et al. 2013](#)].

Things become exceedingly complex beyond the retina as the parvo and magno cells transmitting signals to the superior colliculus and lateral geniculate nucleus (Section [8.1.1](#)) are not simply 2D representations of images. Ideas of how to transmit directly to the visual cortex is far beyond what can be predicted today, but it will likely require completely rethinking the concept of visual signals.

Virtual vestibular input would also be quite challenging to do in an accurate/precise way to be useful in reducing motion sickness. Artificial input to the vestibular system exists today, but in a very primitive form that adds to sickness and imbalance more than it helps. Regardless, if more control can be provided to the vestibular system, or more directly to the vestibular nuclei, then this may someday be feasible.

There is perhaps less reason for direct audio input as headphones can already be largely hidden from view inside the ear. However, if there is a need, this may be easier to do than other sensory input due to the more simplified nature of audio and lower bandwidth requirements as compared to vision.

Neural output to control objects within the virtual world and/or the real world is an entirely different story. There is already demand for such capability for those who are disabled. Prosthetic devices are already controlled by electromyography signals. One can easily imagine such technology could be extended to augmented and virtual reality. In fact, one only need to attend the Neurogaming Conference and Expo held every year in San Francisco to control a basic VR interface via thought alone.

35.4 Standards and Open Source

Some consider standards to be controversial. At one extreme there are those who say any and all standards hinder innovation whereas there are others who want to standardize everything. In truth, standards are simply a tool that, if used wisely, can simplify life for everyone, ease communication across groups, foster development across multiple platforms, and enable consumers to more objectively compare between competing products. Standards can also provide a catalyst for industry leaders to regularly get together, collaborate, and upgrade their findings as they progress—which can enhance innovation rather than hinder it. If standards blocked innovation, the automotive industry, the gaming industry, the Internet, and even VR would not be where it is now.

When defining standards, it’s a misnomer that they only refer to interoperability, which is often the main issue of contention within the industry. Interoperability is only part of the puzzle as there are other fundamental standards that need to be pursued. Standards can help with quality expectations, basic language, health and safety, and even defining common goals.

Precise and consistent terminology is one area where standards would be useful. For example, most people casually discuss field of view without specifying if that is the horizontal field of view or the diagonal field of view (among other factors). Similarly, latency is not well defined. Contrary to what many people believe, latency is more than simply the inverse of the frame rate or refresh rate (Section [15.4](#)). Furthermore, in a raster display, is latency from the time of some action until the time that the resulting first pixel in the upper-left corner in a frame (the start of the raster scan) is displayed or until the time the frame’s center pixel is displayed (a difference of 8 ms for a 60 Hz display)? Or what about the response time of the pixel? Depending on the technology used, pixel response can take several milliseconds, if not tens of milliseconds, to completely reach 100% of its intended intensity. Even if pixel response is immediate, what if the pixel persists for some time? Is latency up to the time the pixel initially appears or up to the average time that it is visible? Such detailed differences can make a big difference when discussing latency. Add these differences due to imprecise/unstandardized definitions together and the variability can easily be 20 ms.

These are some of the most basic elements of VR, and yet the industry players are far apart from one another, and this contributes to customer misunderstanding and unmet expectations. The importance of standards is just as much about interoperability between people and ideas as it is about compatibility and commonalities between vendors.

35.4.1 Open Source

Open source is a class of licensing that usually pertains to software. It works by allowing everyone to contribute to a given project that is free to use and transparent for all to see [[Schneider 2015](#)]. When something is published as open source, it immediately gives others the right to use the same code from that point on, though they can’t charge money for it. The idea can be rewritten from scratch and sold outside of the open source world, but modifications of existing code have to be transparent and free.

Open Source Interoperability

In the 1990s, Russ Taylor led the development of the VR Peripheral Network (VRPN), an open source device-independent and network-transparent system for communicating with VR devices [[Taylor et al. 2001a](#)]. VRPN has since become the most-often-used software library for connecting VR devices to VR applications. Russ claims one reason for its success is that “you can only standardize what nobody cares about” [[Taylor et al. 2001b](#)]. VR creators do care about having a standard way of connecting with VR devices where different devices can be used with the same application, but they do not care about the low-level detail of how that is accomplished.

OSVR (Open Source VR) is a collaboration that is owned and maintained by Razer and Sensics. It’s not a formal organization or non-profit structure but is instead a platform that is defined by signed licensing agreements between Razer and the participating vendors. OSVR is promoted as having over 250 commercial hardware developers, game studios, and research institutions involved.

Russ is now one of the primary developers for the OSVR platform that includes both open source software and hardware designs so that anyone can freely build their own or modify existing systems.

35.4.2 Platform-Specific/De Facto Standards

A **de facto standard** is a system or platform that has achieved a dominant position due to the sheer volume of people associated with it through public acceptance or market forces. For example, the Microsoft Xbox and PC use the DirectX de facto standard. Similarly, Valve has Steam OS. Advanced Micro Devices (AMD) has their own LiquidVR platform while Nvidia has GameWorks VR. They are platforms because their standards work with a range of compliant hardware, but they are not open standards.

35.4.3 Open Standards

Open standards are “standards made available to the general public and are developed (or approved) and maintained via a collaborative and consensus driven process” [[ITU-T 2015](#)].

What is key for an open standard is collaboration and agreement across a wide range of individuals with equal voting rights. Contributions can't be thrown in and taken out unless there is adequate group discussion and agreement. The Khronos Group has open source platforms, and the finalized spec is determined by the membership through votes. Without balance and equal voting share, huge innovations can get missed because the larger participants don't want to collaborate or they simply outgun the smaller players.

Unless all VR work moves toward a platform that a single vendor controls, the most effective open standards are backed by a formal non-profit organization, and everyone must be welcome to participate with equal rights. Unless this happens, the effort will fall apart and lose its credibility. Non-profit structures exist because they force developments to be open and transparent [[Mason 2015](#)].

Notable Open Standards Organizations

The Khronos Group is a good working example of a non-profit standards organization, and their work is free of royalty fees and licensing fees. OpenGL, OpenCL, WebGL, and more are all credited to the Khronos Group. To participate, members pay an annual fee, decisions are voted on, and the standards are implemented.

The Immersive Technology Alliance (ITA; <http://www.ita3d.com>) is a formal non-profit corporation that was originally founded in 2009 under a different name. Its executive director is Neil Schneider, who also created Meant to be Seen (MTSB; <http://mtbs3D.com>). MTSB is where the Oculus Rift was born and marks the spot where John Carmack and Palmer Luckey first met. The ITA exists to make immersive technology successful, and in addition to having its own working groups pertaining to standards and industry growth, it regularly collaborates with external organizations like SIGGRAPH, the Khronos Group, meetup organizations (e.g., SVVR), and more. They have also launched Immersed Access (<http://www.immersedaccess.com>), an NDA-backed private community for professionals to share and learn from one another in a safe environment that is closed off from the press. Immersed Access features unofficial discussion areas for OSVR, Oculus, Valve, and other platforms.

35.5 Hardware

New VR hardware developments are now occurring on a regular basis largely due to the access of 3D printing. HMDs are becoming lighter and with wider fields of view. Hybrid tracking is improving precision and accuracy for both the head and the hands. Magic Leap claims to be solving the accommodation-vergence conflict. Numerous companies are improving upon full body tracking. Eventually, exoskeletons may be built to provide better haptics and to complement human abilities with superhuman powers.

The leading VR companies (Oculus, Valve, Sony, and Sixense) are all now creating tracked hand-held controller options at low price points (at least compared to older professional systems). Such controllers are currently the best way to interact with a majority of fully immersive VR experiences (although such applications are still relatively rare due to the lack of user ownership of such devices). This is significant as not having hands in VR is equivalent to being paralyzed in the real world. As these types of hand input devices become more available, then developers will create better and more innovative methods of interacting. At some point most all VR experiences, other than completely passive experiences, will enable users to interact with their hands.

Even with 6 DoF hand input devices, current hand input has been described as a limiting “boxing glove” style interface where fingers are rarely tracked, and if fingers are tracked, they are rarely tracked accurately and never with 100% reliability. Tyndall is working in collaboration with TSSG and NextGen Interactions on commercializing their extremely accurate glove technology, previously used for surgical training. The goal is to provide a VR glove at a consumer price point that overcomes the significant challenges previous gloves have faced. Figure 35.3 shows a prototype of the glove.

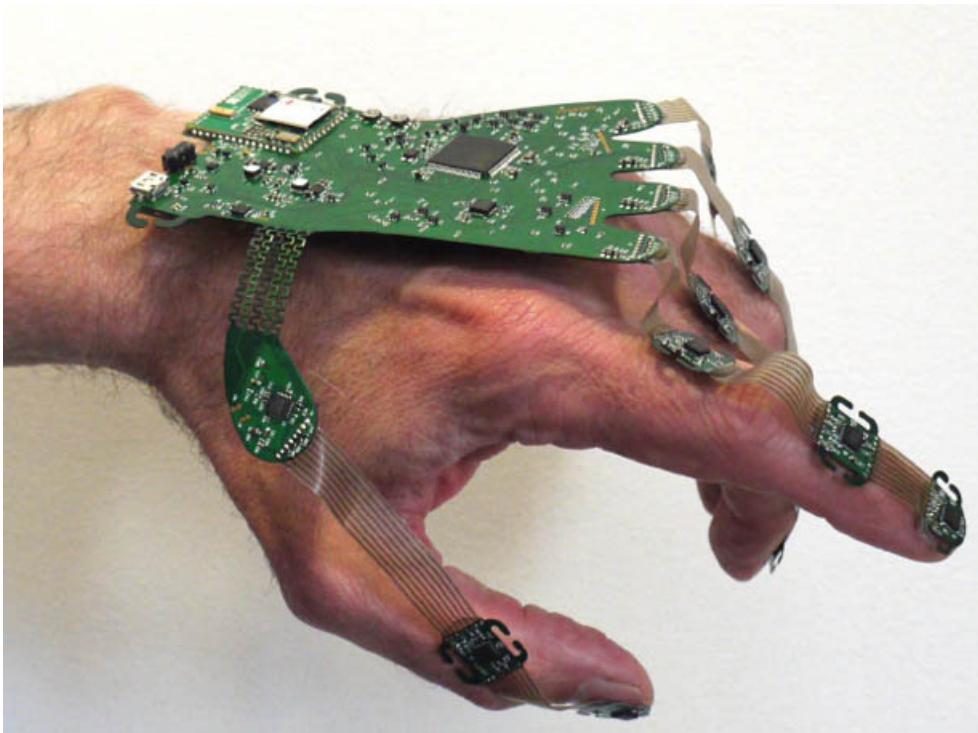


Figure 35.3 Early prototype of the Tyndall/TSSG VR glove. (Courtesy of Tyndall and TSSG)

35.6 The Convergence of AR and VR

Augmented reality and virtual reality have many differences, yet there are also many similarities and the two will likely converge. Although the experiences may still be very different (VR is transporting someone to a different world whereas augmented reality adds to the local world), the same hardware might be used for both. For example, cameras have been capturing the real world and bringing it into the non-see-through HMD VR experience since the 1990s (a form of augmented virtuality; see Section 3.1 and Figure 27.6). So the convergence of AR and VR is not new from a research perspective, but would be new as far as catching mainstream attention. Conversely, future optical-see-through HMDs will be able to make individual pixels opaque so digital imagery can completely occlude all or part of the real world.

[1](http://vr.meetup.com) . To find a local VR Meetup, go to <http://vr.meetup.com>.

Getting Started

The best way to predict the future is to create it .

—Alan Kay

VR technology is moving at lightning speed. It was only in 2011 when most people outside of academics, corporate research, and science fiction communities thought VR was an obsolete joke. Fast-forward to today where thousands of independent developers, start-ups, and Fortune 500 companies are creating VR experiences, positioning themselves as VR leaders. The message is clear: If a company wants to be competitive there is no time to waste on writing the perfect project plan lest they risk being left behind. It's a new world, one where the prototype should have been completed last week and informal feedback from an initial demo should have happened yesterday. Yes, spend some time on the Define Stage of the project. But spend a couple of days versus a month on the initial plan before building something. Or better yet jump straight to the Make Stage. Having no software development experience is no excuse—basic prototypes can be built with today's tools by pointing and clicking with a mouse. The content need not be novel—this is just for you to get started. Experiment by looking around, then modify and experiment with a few things. Then move back to the Define Stage by sketching some ideas out on paper. After all, if a teenager with no budget can build a basic VR experience that his friends and family are impressed by, then an adult committed to changing the future can do the same. Anyone who can't put together a basic plan, spend a few hundred dollars on hardware, create a prototype experience (or better yet, do so by working with a colleague or friend), and demonstrate that prototype to some acquaintances for feedback is not yet serious about positioning themselves as a leader of the VR movement. For those of you who are ready, **welcome to the new reality!**

Here is a very reasonable schedule for someone who can build VR experiences *part-time in little as a few hours per week* . If you have somehow managed to arrange your life or convinced your boss to do this full-time, then the limiting factor will be the arrival of the hardware. If you have experience in software development, then you can easily condense this schedule to under a week. So stop making excuses and **get to work!**

- Week 1

- Find and attend a local VR Meetup (<http://vr.meetup.com>). Try some demos and talk to as many people as possible. Get to know those who are also serious about developing VR experiences. Collect their contact information.

- Order an HMD.

- Week 2

- While waiting for the hardware to arrive, download Unity (it is free for individuals or companies making under \$100K/year) or your development tool of choice.

- Search online for some non-VR tutorials and start working through them in order to become comfortable with the core tools (even if you don't plan on being a developer, you should know the basics).

- Week 3

- Once the HMD arrives, follow the directions that came with the HMD. Try the manufacturer's basic demos.

- Find and download some different VR experiences. Note what you like and don't like.

- Week 4

- Using Unity or your tool of choice, add a texture-mapped plane, a cube floating in space, and a light source (I have performed the initial Define Stage for you!).

- Build the application, put on the HMD, and look around.

- Take off the HMD and modify the scene. Build and put on the HMD, noticing how things have changed.

- Congratulations, you have performed the most basic form of the define-make-experiment iteration in perhaps a couple of hours. It doesn't matter how bad it is at this point, you are learning fast.

- Now iterate some more!

- Week 5

- Write a basic concept for your first project based on what you liked and didn't like about the experiences you tried and your first scene that you built. Be creative.

- Don't worry about details such as requirements, constraints, and task analysis yet. That can come later.

- Week 6

- Build some low-level functionality for your basic concept. Don't worry about the art yet. Remember this is a prototype to test your ideas.
- Show to friends, family, and people you met at the meetup. Get their feedback of what works and what doesn't work.
- Week 7 and beyond
- Iterate. Iterate. Iterate. For each iteration, expand upon each stage or start completely over at any time. Then repeat Week 7.

Congratulations—you are now a VR creator and a contributor to the VR revolution! This is not the end, for it is only the beginning.

APPENDIX A

Example Questionnaire

This appendix contains an example questionnaire from NextGen Interactions used for evaluation of MakeVR (formerly Studio-In Motion) in collaboration with Digital Art-Forms and Sixense [[Jerald et al. 2013](#)]. The questionnaire was distributed to users after they constructed a castle within the application that took approximately two hours per user. The questionnaire includes a Kennedy Simulator Sickness Questionnaire (Section 16.1), a Likert Scale (Section 33.3.4), previous experience, and an open-ended questionnaire inquiring about what worked and what could be improved.

Studio In-Motion Questionnaire Page		1 of 4		
Simulator Sickness Questionnaire				
Do you feel that you are in the same state good health as when you started the <input type="checkbox"/> Yes <input type="checkbox"/> No experiment? If you answered no please explain briefly in the space provided below:				
For each of the following conditions, please circle how you are feeling right now, on the scale of <i>none</i> through <i>severe</i> .				
1. General discomfort	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
2. Fatigue (weariness or exhaustion of the body)	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
3. Headache	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
4. Eye strain (weariness or soreness of the eyes)	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
5. Difficulty focusing	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
6. Increased salivation	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
7. Sweating	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
8. Nausea (stomach distress)	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
9. Difficulty concentrating	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
10. Fullness of head (sinus pressure)	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
11. Blurred vision	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
12. Dizzy (with eyes open)	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
13. Dizzy (with eyes closed)	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
14. Vertigo (surroundings seem to swirl)	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
15. Stomach awareness (just a short feeling of nausea)	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
16. Burping	<i>none</i>	<i>slight</i>	<i>moderate</i>	<i>severe</i>
If you expressed slight, moderate, or severe on any of the questions above, please state if you felt that way before using the system and if so, explain how you felt worse after using the system.				
You should not drive an automobile for at least one half-hour after the end of the experiment.				

Likert Scale

Mark a single box (*strongly disagree, disagree, undecided, agree, or strongly agree*) for each of questions below.

	<i>strongly disagree</i>	<i>disagree</i>	<i>undecided</i>	<i>agree</i>	<i>strongly agree</i>
I found the interface easy to learn.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Once I learned the interface, I found the navigation to be easy and intuitive to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Once I learned the interface, I found object manipulation easy and intuitive to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Once I learned the interface, I found that I could focus on being creative instead of on the technicalities of the interface.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I prefer the new interface to a mouse and keyboard interface.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I was able to create the objects/scenes that I wanted.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I would use the system myself in my home.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I would recommend the system to friends.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Overall, I enjoyed the experience.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Demographic Information

1. Age and gender

Age _____

Gender _____

Studio In-Motion Questionnaire[Page 3 of 4](#)**Background/Experience**

For each question, please put a check next to your answer.

1. How often do you use a computer in your average week?

I use computers less than . . .

 1. 1 hour 2. 2 hours 3. 5 hours 4. 10 hours 5. 20 hours 6. 40 hours 7. more than 40 hours

2. Over the past two years, what is the most you have played video games in a single week? Select one.

I played less than . . .

 1. 1 hour 2. 2 hours 3. 5 hours 4. 10 hours 5. 20 hours

6. 40 hours

7. more than 40 hours

3. How many times have you used systems where you did not use your hands in free 3D space to control the computer application (e.g., not using a keyboard, mouse, or touchscreen).

Before today I used free 3D space interfaces . . .

1. Never

2. 1 time

3. 2 times

4. 5–10 times

5. 11–20 times

6. 20–100 times

7. nearly every day at school, at work, or for entertainment

4. How much experience do you have with CAD (Computer Aided Design) software (e.g., Maya, 3D Studio, Sketchup)

Before today I used CAD . . .

1. Never

2. 1 time

3. 2 times

4. 5–10 times

5. 11–20 times

6. 20–100 times

7. nearly every day at school, at work, or for entertainment

Studio In-Motion Questionnaire

[Page 4 of 4](#)

Open-Ended Questions

1. What did you most like about the system?

2. How long did it take you to get good at using the system?

3. Were you able to create what you wanted to create?

4. Could you have created a more interesting scene given more time?

5. What did you dislike about the system?

6. What suggestions or ideas do you have for improving the system?

7. Did you feel tired from using the systems? If so, in what areas of your body did you feel fatigue?

8. What do you think consumers would pay for this system?

9. Any other comments?

APPENDIX B

Example Interview Guidelines

This appendix provides an example of an interview guidelines document for use in collecting qualitative feedback from users after they experience a VR application. (Document courtesy of NextGen Interactions.)

Interview Question Guidelines



Interview prep. Turn on the video camera and mic. Test to make sure video and sound is recording properly.

Written questionnaire. Expand on the answers provided in the exit questionnaire.

General comments. Open-ended questions of free thought and discussion.

Health. How do they feel? If not well, is there any insight when it started to occur or what caused it?

Presentation. Any comments on the audio, visual, or other cues? Did anything not look, sound, or feel good?

Engagement and presence. Did they feel engaged in the virtual world? Did they feel like they were looking at/observing the experience or did they feel like they were a part of the experience? Were there any specific problems or events that broke the illusion?

Understanding and intuitiveness. Was the experience understandable? Was it obvious what they were expected to do? What specifically was difficult to understand?

Time usage. Did they want to use the system more? Or was the session too long or too tiring?

Ease of use and difficulties. Were the tasks easy or difficult to interact with? Any comments on the difficulty of using the system? What specifically could be improved upon?

Missing features and ideas. What would they have liked to experience that was not available? What would be some useful things to include in a future version?

Other comment and suggestions. Any other comments or suggestions to make the system and experience better?

Thank them for participating. Ask them if they would like to participate in future events and/or provide further feedback. Collect contact information.

Glossary

2D desktop integration. A form of the Widgets and Panels Pattern where existing 2D desktop applications are brought into the environment via texture maps and mouse control with pointing. (Section [28.4.1](#))

2D warp. Pixel selection from or reprojection of a reference image according to newly desired viewing parameters. Can be used to reduce effective latency (also known as time warping). (Section [18.7.2](#))

3D Multi-Touch Pattern. A viewpoint control pattern that enables simultaneous modification of the position, orientation, and scale of the world. (Section [28.3.3](#))

3D Tool Pattern. A manipulation pattern that enables users to directly manipulate an intermediary 3D tool with their hands that in turn directly manipulates some object in the world. (Section [28.2.3](#))

above-the-head widgets and panels. A form of the Widgets and Panels Pattern accessed via reaching up and pulling down the widget or panel with the non-dominant hand. (Section [28.4.1](#))

absolute input device. An input device that senses pose relative to a constant reference independent of past measurements and is nulling compliant. (Section [27.1.3](#))

accommodation. The mechanism by which the eye alters its optical power to hold objects at different distances into focus on the retina. (Section [9.1.3](#))

accommodation-vergence conflict. A sensory conflict that occurs due to the relationship between accommodation and vergence not being consistent with what occurs in the real world. Can lead to VR sickness. (Section [13.1](#))

accuracy. The quality or state of being correct (i.e., closeness to the truth). (Section [31.15.1](#))

action space. The space of public action where one can move relatively quickly, speak to others, and toss objects (from about two meters to 20 meters). (Section [9.1.2](#))

action-intended distance cues. Psychological factors of future actions that influence distance perception. (Section [9.1.3](#))

active haptics. Artificial forces that are dynamically controlled by a computer. (Section [3.2.3](#))

active motion platform. A motion platform controlled by the computer simulation. (Section [3.2.4](#))

active readaptation. The use of real-life targeted activities aimed at recalibrating the sensory systems in order to reduce VR aftereffects. (Section [13.4.1](#))

active touch. The physical exploration of an object, usually with the fingers and hands. (Section [8.3.3](#))

adverse health effect. Any problem caused by a reality system or application that degrades a user's health such as nausea, eye strain, headache, vertigo, physical injury, and transmitted disease. (Part [III](#))

aerial perspective. A pictorial depth cue where objects with more contrast appear closer than duller objects due to scattering of light particles in the atmosphere. Also known as atmospheric perspective. (Section [9.1.3](#))

afference. Nerve impulses that travel from sensory receptors inward towards the central nervous system. (Section [7.4](#))

affordance. Possible actions and how something can be interacted with by a user. A relationship between the capabilities of a user and the properties of a thing. (Section [25.2.1](#))

after action review. A user debriefing with an emphasis on the user's specific actions in order to determine what happened, why it happened, and how it could be done better. (Section [33.3.7](#))

aliasing. An artifact that occurs due to approximating data with discrete sampling. (Section [21.4](#))

ambient sound effects. Subtle surround sound that can significantly add to realism and presence. (Section [21.3](#))

apparent motion. The perception of visual movement that results from appropriately displaced stimuli in time and space even though nothing is actually moving. (Section [9.3.6](#))

application. The non-rendering aspects of the virtual world including updating dynamic geometry, user interaction, physics simulation, etc. (Section [3.2](#))

application delay. The time from when tracking data is received until the time data is passed onto the rendering stage. (Section [15.4.2](#))

apprehension. The actual subjective experience that is consciously available and can be described. Can be divided into object properties and situational properties. (Section [10.1](#))

assumptions. High-level declarations of what one or more members of the team believe to be true. (Section [31.9](#))

attention. The process of taking notice or concentrating on some entity while ignoring other perceivable information. (Section [10.3](#))

attention map. A visual image that provides useful information of what users actually look at. (Section [10.3.2](#))

attentional capture. A sudden involuntary shift of attention due to salience. (Section [10.3.2](#))

attentional gaze. A metaphor for how attention is drawn to a particular location or thing in a scene. (Section [10.3.2](#))

attitudes. Values and belief systems about specific subjects. (Section [7.9.3](#))

attrition. A threat to internal validity resulting from participants dropping out of the study. Also known as mortality. (Section [33.2.3](#))

augmented reality (AR). The addition of computer-generated sensory cues on to the already existing real world. (Section [3.1](#))

augmented virtuality. The result of capturing real-world content and bringing it into VR. (Section [3.1](#))

auralization. The rendering of sound to simulate reflections and binaural differences between the ears. (Section [21.3](#))

authoritative server. A network server that controls all the world state, simulation, and processing of input for all clients. (Section [32.5.3](#))

autokinetic effect. The illusory movement of a single stable point-light that occurs when no other visual cues are present. (Section [6.2.7](#))

Automated Pattern. A viewpoint control pattern where the viewpoint is controlled by something other than the user. (Section [28.3.4](#))

avatar. A character that is a virtual representation of a real user. (Section [22.5](#))

average. A value representing the middle of a dataset that the data tends toward. Can more specifically be the mean, median, or mode. (Section [33.5.2](#))

back projection. A form of feedback from higher-level areas of the brain based on previous information that has already been processed (i.e., top-down processing). (Section [8.1.1](#))

background. Scenery in the periphery of the scene located in far vista space. (Section [21.1](#))

Bare-hand input devices. A class of input devices that work via sensors aimed at the hands (mounted in the world or on the HMD). (Section [27.2.5](#))

behavioral processing. Learned skills and intuitive interactions triggered by situations that match stored neural patterns, and largely subconscious. (Section [7.7.2](#))

beliefs. Convictions about what is true or real in and about the world. (Section [7.9.3](#))

between-subjects design. An experiment that has each participant only experience a single condition. Also known as A/B testing when only two variables are being compared. (Section [33.4.1](#))

bimanual asymmetric interaction. Bimanual interaction where each hand performs different actions in a coordinated way to accomplish a task. (Section [25.5.1](#))

bimanual interaction. Two-handed interaction that can be classified as symmetric or asymmetric. (Section [25.5.1](#))

bimanual symmetric interaction. Bimanual interaction where each hand performs identical actions. Can be classified as synchronous or asynchronous. (Section [25.5.1](#))

binaural cue. Two different audio cues, one for each ear, that help to determine the position of sounds. Also known as stereophonic cues. (Section [8.2.2](#))

binding. The process by which stimuli are combined to create our conscious perception of a coherent object. (Section [7.2.1](#))

binocular disparity. The difference in image location of a stimulus seen by the left and right eyes resulting from the eyes' horizontal separation and the stimulus distance. (Section [9.1.3](#))

binocular display. A display with a different image for each eye, providing a sense of stereopsis. (Section [9.1.3](#))

binocular-occlusion conflict. A sensory conflict that occurs due to occlusion cues not matching binocular cues. Can lead to VR sickness. (Section [13.2](#))

biocular display. A display with two identical images, one for each eye. (Section [9.1.3](#))

biological clock. Body mechanisms that act in periodic manners, with each period serving as a tick of the clock, which enable us to sense the

passing of time. (Section [9.2.3](#))

biological motion perception. The ability to detect human motion. (Section [9.3.9](#))

biomechanical symmetry. The degree to which physical body movements for a virtual interaction correspond to the body movements of the equivalent real-world task. (Section [26.1](#))

blind spot. An area on the retina where blood vessels leave the eye and there are no light-sensitive receptors resulting in a small area of the eye being blind (although we rarely perceive that blindness). (Section [6.2.3](#))

block diagram. High-level diagrams showing interconnections between different system components. (Section [32.2.2](#))

bottom-up processing. Processing that is based on proximal stimuli that provides the starting point for perception. Also known as data-based processing. (Section [7.3](#))

breadcrumbs. Markers that are dropped often by the user when traveling through an environment. (Section [21.5](#))

break-in-presence. A moment when the illusion generated by a virtual environment breaks down and the user finds himself where he truly is—in the real world wearing an HMD. (Section [4.2](#))

brightness. The apparent intensity of light that illuminates a region of the visual field. (Section [8.1.2](#))

button. An input device that controls one degree of freedom via pushing with a finger. Typically takes one or two states although some buttons can take analog values (i.e., analog triggers). (Section [27.1.6](#))

Call of Duty Syndrome. The mental model that one's view direction, weapon/arm, and forward direction always point in the same direction as is the case for typical non-VR first-person shooter games. When this expectation does not match a VR experience then sickness can result. (Section [17.2](#))

caricature. A representation of a person or thing in which certain striking characteristics are exaggerated and less important features are omitted or simplified in order to create a comic or grotesque effect. (Section [22.5](#))

carryover effect. An effect that occurs when experiencing one condition causes retesting bias in a later condition. (Section [33.4.1](#))

categorical variable. The most basic form of measurement where each possible value takes on a mutually exclusive label or name. Also known as a nominal variable. (Section [33.5.2](#))

causality (networking). Consistent ordering of events for all users. Also known as ordering. (Section [32.5.1](#))

causality violation. Events that are out of order so that effects appear to occur before cause. (Section [32.5.1](#))

CAVE. A reality system where the user interacts in a physical room surrounded by stereoscopic perspective-correct images displayed on the floor and walls. CAVE is an acronym for CAVE Automatic Virtual Environment. (Sections [3.2.1](#), [6.2](#))

change blindness. The failure to notice a change of an item on a display from one moment to the next. (Section [10.3.1](#))

change blindness blindness. The lack of awareness of change blindness and the belief that one is good at detecting changes when in fact one is not. (Section [10.3.1](#))

change deafness. A physical change in an auditory stimulus that goes unnoticed by a listener. (Section [10.3.1](#))

channel. A constricted route, often used in VR and video games (e.g., car-racing games) that give users the feeling of a relatively open environment when it is not very open at all. (Section [21.5](#))

choice blindness. The failure to notice that a result is not what the person previously chose. (Section [10.3.1](#))

circadian rhythm. A biologically recurring natural 24-hour oscillation. (Section [9.2.3](#))

class (software). A program template for a set of data and methods. (Section [32.2.4](#))

class diagram. A description of classes, along with their properties and methods, and the relationship between classes. (Section [32.2.4](#))

client-server architecture. A network architecture where each client communicates with a server and then the server distributes information out to the clients. (Section [32.5.3](#))

close-ended question. A question with multiple answers that can be circled or checked. (Section [33.3.4](#))

clutching. The releasing and regrasping of an object in order to complete a task due to not being able to complete it in a single motion. (Section [27.1](#))

cocktail party effect. The ability to focus one's auditory attention on a particular conversation while filtering out many other conversations in the

same room. (Section [10.3.1](#))

cognitive clock. The inference of time based on mental processes that occur during an interval . . (Section [9.2.3](#))

color constancy. The perception for the colors of familiar objects to remain relatively constant even under changing illumination. (Section [10.1.4](#))

color cube. A form of the Widgets and Panels Pattern that consists of a 3D space that users can select colors from. (Section [28.4.1](#))

color vision. The ability to discriminate between stimuli of equal luminance on the basis of wavelength alone. (Section [8.1.3](#))

comfort. A state of physical ease and freedom from sickness, fatigue, and pain. (Section [31.15.1](#))

communication. The transfer of energy between two entities, even if just the cause and effect of one object colliding with another object. (Section [1.2](#))

comparative evaluation. A comparison of two or more well-formed complete or near-complete systems, applications, methods, or interaction techniques to determine which is more useful and/or more cost-effective. Also known as summative evaluation. (Section [33.3.6](#))

compass. A personal wayfinding aid that helps give a user a sense of exocentric direction. (Section [22.1.1](#))

complementarity input modalities. The merging of different types of input into a single command. (Section [26.6](#))

compliance. The matching of sensory feedback with input devices across time (temporal compliance) and space (spatial compliance). (Section [25.2.5](#))

compound patterns. A set of interaction patterns that combine two or more patterns into more complicated patterns. Includes the Pointing Hand Pattern, World-in-Miniature Pattern, and Multimodal Pattern. (Section [28.5](#))

conceptual integrity. Coherence, consistency, and sometimes uniformity of style. (Section [20.3](#))

concurrency (networking). The simultaneous execution of events by different users on the same entities. (Section [32.5.1](#))

concurrent input modalities. The option to issue different commands via two or more types of input simultaneously. (Section [26.6](#))

cone-casting flashlight. A form of the Volume-Based Selection Pattern that uses hand pointing but uses a cone for selection instead of a ray. (Section [28.1.4](#))

cones. Receptors on the first layer of the retina that are responsible for vision during high levels of illumination, color vision, and detailed vision. (Section [8.1.1](#))

confounding factors. Variables other than the independent variable that may affect the dependent variable and can lead to a distortion of the relationship between the independent and dependent variables. (Section [33.4.1](#))

conjunction search. Looking for particular combinations of features. (Section [10.3.2](#))

conscious. The mind's totality of sensations, perceptions, ideas, attitudes, and feelings that we are aware of at any given time. (Section [7.6](#))

constraints (interaction). Limitations of actions and behaviors. (Section [25.2.3](#))

constraints (project). Limitations that a project must be designed around. Types of project constraints include real constraints, resource constraints, obsolete constraints, misperceived constraints, indirect constraints, and intentional artificial constraints. (Section [31.10](#))

construct validity. The degree to which a measurement actually measures the conceptual variable (the construct) being assessed. (Section [33.2.3](#))

constructivist approach. Construction of understanding, meaning, knowledge, and ideas through experience and reflection upon those experiences rather than trying to measure absolute or objective truths about the world. (Section [33.3](#))

contextual geometry. The context of the environment, typically in action space, that a user finds himself in. Contains no affordances. (Section [21.1](#))

continuity error. Changes between shots in film; many viewers fail to notice significant changes. (Section [10.3.1](#))

continuous delivery. Updated executables that are provided online as often as once a week. (Section [32.8.3](#))

continuous discovery. The ongoing process of engaging users during the design and development process in order to understand what users want to do, why they want to do it, and how they can best do it. (Section [30.3](#))

control symmetry. The degree of control a user has for an interaction as compared to the equivalent real-world task. (Section [26.1](#))

control variable. A variables that is kept constant in an experiment in order to keep that variable from affecting the dependent variable. (Section [33.4.1](#))

control/display (C/D) ratio. The scaling ratio of the non-isomorphic rotational mapping of the hand to an object or pointer. (Section [28.1.2](#))

convergence. The rotation of the eyes inward towards each other in order to look closer. (Sections [8.1.5](#), [9.1.3](#))

core experience. The consistent and essential moment-to-moment activity of users making meaningful choices resulting in meaningful feedback. (Section [20.2](#))

correlation. The degree to which two or more attributes or measurements tend to vary together. (Section [33.5.2](#))

covert orienting. The act of mentally shifting one's attention that does not require changing the sensory receptors. (Section [10.3.2](#))

critical incident. An event that has a significant impact, either positive or negative, on the user's task performance and/or satisfaction. (Section [33.3.6](#))

cubic environment map. Rendering of a scene onto six sides of a large cube. (Section [18.7.2](#))

cybersickness. Visually induced motion sickness resulting from immersion in a computer-generated virtual world. (Part [III](#))

cycle of interaction. An iterative process of forming a goal, executing the action, and evaluating the results. (Section [25.4](#))

cyclopean eye. A hypothetical position in the head that serves as a reference point for the determination of a head-centric straight-ahead. Also see head reference frame. (Section [26.3.5](#))

dark adaptation. The increase in one's sensitivity to light in dark conditions. (Section [10.2.1](#))

de facto standard. A system or platform that has achieved a dominant position, due to the sheer volume of people associated with it, through public acceptance or market forces. (Section [35.4.2](#))

dead reckoning. The estimated location of a remotely controlled entity based on extrapolation of its most recently received packet. (Section [32.5.4](#))

decisions. Conclusions or resolutions reached after reflection and consideration. (Section [7.9.3](#))

Define Stage. The portion of the iterative design process where ideas are initially created and refined as more is discovered from the Make and Learn Stages. (Part [VI](#), Chapter [31](#))

degrees of freedom (DoF). The number of independent dimensions available for the motion of an entity or that an input device can manipulate. (Sections [25.2.3](#), [27.1.2](#))

delay compensation technique. Techniques to reduce effective latency. Examples include head-motion prediction and post-rendering latency reduction. (Section [18.7](#))

deletion. The omission of certain aspects of incoming sensory information by selectively paying attention to only parts of the world. (Section [7.9.3](#))

demand characteristic. A threat to internal validity resulting from cues that enable participants to guess the hypothesis. (Section [33.2.3](#))

demo. The showing of a prototype or a more fully produced experience. (Section [32.8.1](#))

dependent variable . . . The output or response that is measured during an experiment. (Section [33.4.1](#))

depth cues. Clues that help us perceive egocentric depth or distance. Can be classified as pictorial depth cues, motion depth cues, binocular disparity, and oculomotor depth cues. (Section [9.1.3](#))

descriptive statistics. Summative values that describe the main features of a dataset. (Section [33.5.2](#))

design. A created object, a process, and an action. The entire creation of a VR experience from information gathering and goal setting to delivery and product enhancement/support. (Chapter [30](#))

design pattern (software). A general reusable conceptual solution that is used to solve commonly occurring software architecture problems. (Section [32.2.5](#))

design specification. Details of how an application currently is or will be put together and how it works. (Section [32.2](#))

detection acuity. The smallest stimulus that one can detect (as small as 0.5 arc sec) in an otherwise empty field and represents the absolute threshold of vision. Also known as visible acuity. (Section [8.1.4](#))

direct communication. The direct transfer of energy between two entities with no intermediary and no interpretation attached. Consists of

structural communication and visceral communication. (Section [1.2.1](#))

direct gesture. A gesture that is immediate and structural in nature that conveys spatial information and can be interpreted and responded to by the system as soon as the gesture starts. (Section [26.4.1](#))

Direct Hand Manipulation Pattern. A manipulation pattern that corresponds to the way we manipulate objects with our hands in the real world. The object attached to the hand moves along with the hand until released. (Section [28.2.1](#))

direct interaction. An impression of direct involvement with an object rather than of communicating with an intermediary. (Section [25.3](#))

directional compliance. Sensory feedback that matches the rotation and directional movement of an input device. (Section [25.2.5](#))

director. The lead designer of a project who primarily acts as an agent to serve the users of the experience, but also serves to fulfill the project goals and to direct the team in focusing on the right things. (Section [20.3](#))

discoverability. The exploration of what something does, how it works, and what operations are possible. (Section [25.2](#))

displacement ratio. The ratio of the angle of environmental displacement to the angle of head rotation. (Section [10.1.3](#))

display delay. The time from when a signal leaves the graphics card to the time a pixel changes to some percentage of the intended intensity defined by the graphics card output. (Section [15.4.4](#))

distal stimuli. Actual objects and events out in the world (objective reality). (Section [7.1](#))

distortion (psychological). The modification of incoming sensory information, enabling one to take into account the context of the environment and previous experiences. (Section [7.9.3](#))

divergence. The rotation of the eyes outwards away from each other in order to look further into the distance. (Sections [8.1.5](#), [9.1.3](#))

divergence (networking). The temporal-spatial state of an entity being different for different users. (Section [32.5.1](#))

divergence filtering. A technique to reduce network traffic by only sending out packets when the divergence of an entity reaches some threshold. (Section [32.5.5](#))

doll. A miniature hand-held proxy of an object or set of objects. (Section [28.5.2](#))

dominant eye. The eye used for sighting tasks. (Section [9.1.1](#))

dominant hand. The user's preferred hand for performing fine motor skills with bimanual asymmetric interaction. (Section [25.5.1](#))

dorsal pathway. A neural path from the LGN to the parietal lobe, which is responsible for determining an object's location (as well as other responsibilities). Also known as the "where," "how," or "action" pathway. (Section [8.1.1](#))

double buffer. A rendering system that has a buffer the display processor draws to and a second buffer that feeds data to the display. (Section [15.4.4](#))

dual adaptation. The perceptual adaptation to two or more mutually conflicting sensory environments that occurs after frequently alternating between those conflicting environments. (Section [10.2.2](#))

dual analog stick steering. A form of the Steering Pattern that utilizes standard first person shooter controls to navigate over a terrain. (Section [28.3.2](#))

dwell selection. Selection by holding a pointer on an object for some defined period of time. (Section [28.1.2](#))

early wandering. The exploration of radically different designs early in the process before converging toward and committing to the final solution. (Section [32.2](#))

ease of learning. The ease with which a novice user can comprehend and begin to use an application or interaction technique. (Section [31.15.1](#))

ease of use. The simplicity of an application or interaction technique from the user's perspective. The amount of mental workload induced upon the user. (Section [31.15.1](#))

edge. Boundaries between regions that prevent or deter travel. (Section [21.5](#))

efference. Nerve impulses that travel from the central nervous system outward toward effectors such as muscles. (Section [7.4](#))

efference copy. Signals equal to efference that travel to an area of the brain that predicts afference, enabling the central nervous system to initiate responses before sensory feedback occurs. (Section [7.4](#))

egocentric interaction. Interaction from within an environment where the user has a first person view. (Section [26.2.1](#))

egocentric judgment. The sense of where (direction and distance) a cue is relative to the observer. Also known as subject-relative judgment. (Section [9.1.1](#))

emotional processing. The affective aspect of consciousness that powerfully processes data, resulting in physiological and psychological visceral and behavioral responses. (Section [7.7.4](#))

equivalent input modalities. The option for users to choose which type of input to use, even though the result would be the same across modalities. (Section [26.6](#))

evaluation. The judgment portion of the cycle of interaction about achieving a goal, making adjustments, or creating new goals. Consists of perceiving, interpreting, and comparing. (Section [25.4](#))

event. A segment of time at a particular location that is perceived to have a beginning and an end or a series of perceptual moments that unfolds over time. (Section [9.2.1](#))

evolutionary theory. A motion sickness theory that states it is critical to our survival to properly perceive our body's motion and the motion of the world around us. Also known as the poison theory. (Section [12.3.2](#))

execution. The feedforward portion of the cycle of interaction that bridges the gap between the goal and result. Consists of planning, specifying, and performing. (Section [25.4](#))

exocentric interaction. Viewing and manipulating a virtual model of the environment from outside of it. (Section [26.2.1](#))

exocentric judgment. The sense of where a cue is relative to some other cue. Also known as object-relative judgment. (Section [9.1.1](#))

expectation violation. Resulting effects from concurrent events that are different than the expected or intended effects. (Section [32.5.1](#))

experiential fidelity. The degree to which the user's personal experience matches the intended experience of the VR creator. (Sections [4.4.2](#), [20.1](#))

experiment. A systematic investigation to answer a question through the use of a hypothesis, empirical data, and statistical analysis. (Section [33.4](#))

experimenter bias. A threat to internal validity resulting from the experimenter treating participants differently or behaving differently. (Section [33.2.3](#))

expert evaluations. Systematic approaches conducted by experts that identify usability problems from the user's perspective with the intent to improve the user experience. (Section [33.3.6](#))

expert guidelines-based evaluation. An expert evaluation that identifies potential usability problems by comparing interactions (either existing or evolving) to established guidelines. Also known as heuristic evaluation. (Section [33.3.6](#))

exploration. Browsing of space to build knowledge of the environment. (Section [10.4.3](#))

extender grab. A form of the Pointing Hand pattern where the mapping of the hand-to-object orientation is one-to-one but translations are scaled depending on the distance of the object from the user. (Section [28.5.1](#))

external validity. The degree to which the results of a study can be generalized to other settings, other users, other times, and other designs/implementations. (Section [33.2.3](#))

eye gaze selection. A form of the Pointing Pattern where the selection ray is extended from the eye(s). (Section [28.1.2](#))

eye movement theory. A motion sickness theory that states motion sickness occurs due to the unnatural eye motion required to keep the scene's image stable on the retina. (Section [12.3.5](#))

eye movements. Eye rotation controlled by six extraocular muscles around three axes. Can be classified as gaze-shifting, fixational, and gaze-stabilizing eye movements. (Section [8.1.5](#))

eye reference frames. The reference frames defined by the position and orientation of the user's eyeballs. (Section [26.3.6](#))

eye rotation gain. The ratio of the eye velocity rotation divided by head rotation velocity. (Section [8.1.5](#))

eye-tracking input devices. A class of input devices that track where the eyes are looking. (Section [27.3.2](#))

face validity. The general intuitive and subjective impression that a concept, measure, or conclusion seems legitimate. (Section [33.2.3](#))

fade out. The fading out of the virtual world to a single color or real-world provided by a video-see-through HMD. Can be used to reduce VR sickness when latency exceeds some value or tracking is lost or abated. (Sections [18.10](#), [31.15.3](#))

failure. A learning experience. (Section [33.1](#))

false positive. A finding that occurs by chance when in fact no difference actually exist. (Section [33.2.3](#))

fear-based distance cues. Fear-causing situations that increase the sense of height. (Section [9.1.3](#))

feature search. Looking for a particular feature that is distinct from surrounding distractors that don't have that feature. (Section [10.3.2](#))

feedback (interaction). Communication to a user of the results of an action or the status of a task. (Section [25.2.4](#))

field of regard. The angular measure of what can be seen by physically rotating the eyes, head, and body. (Section [8.1.1](#))

field of view. The angular measure of what can be seen at a single point in time. (Section [8.1.1](#))

figure. A group of contours that has some kind of object-like properties in our conscious. (Section [20.4.2](#))

figure-ground problem. The question of what is the foreground (figure) and what is the background (ground). (Section [20.4.2](#))

filled duration illusion. The perception that a duration filled with stimuli is perceived as longer than an identical time period empty of stimuli. (Section [9.2.3](#))

final production. The part of the Make Stage where features have been finalized so the team can focus on polishing the deliverable. (Section [32.7](#))

finger menu. A form of the Widgets and Panels Pattern where menu options are attached to the fingers. (Section [28.4.1](#))

fishing. The searching of data via many different hypotheses that can lead to wrong conclusions. Also known as data mining. (Section [33.2.3](#))

fixation. A pause to focus on an item of interest. (Section [10.3.2](#))

fixational eye movements. Small eye movements that occur when holding the head still and looking in a single direction to prevent rods and cones from becoming bleached. Can be classified as microtremors, microsaccades, or ocular drift. (Section [8.1.5](#))

flavor. The combination of smell, taste, temperature, and texture that results in a wide range of distinctive food qualities. (Section [8.6](#))

flicker. The flashing or repeating of alternating visual intensities. Can cause VR sickness. (Section [9.2.4](#))

flicker-fusion frequency threshold. The flicker frequency where flicker becomes visually perceptible. (Section [9.2.4](#))

flow. A psychological state where a person is fully focused on an activity or task. (Section [10.3.2](#))

focus group. A constructivist approach to collecting ideas and feedback similar to an interview but occurs in a group setting. (Section [33.3.5](#))

focus of expansion. The point in space around which all other stimuli seem to expand as one moves forward toward it. (Section [9.3.3](#))

fomite. An inanimate physical object that is able to harbor pathogenic organisms that can serve as an agent for transmitting diseases between different users of the same equipment. (Section [14.4](#))

formative usability evaluation. An expert evaluation that diagnoses problems by gathering critical empirical evidence from observing users interacting with an application. (Section [33.3.6](#))

forward-up map. A map that aligns the information on the map to match the direction the user is facing or traveling. (Section [22.1.1](#))

fovea. A small area in the center of the retina that contains only cones packed densely together, resulting in high visual acuity when looking directly at something. (Section [8.1.1](#))

frame. A full-resolution rendered image. (Section [15.4.3](#))

frame rate. The number of times the system renders the entire scene per second. (Section [15.4.3](#))

framing hands technique. A form of the Image-Plane Selection Pattern where the hands are positioned to form the two corners of a frame in the 2D image plane surrounding the desired object(s). (Section [28.1.3](#))

freely turnable system. A wireless reality system that allows users to easily physically turn in any direction. (Section [22.4](#))

full-body tracking. A class of input devices that consists of tracking more than just the head and hands. (Section [27.3.4](#))

fully walkable system. A reality system that enables users to stand up and physically walk around. (Section [22.4](#))

functional requirement. A requirement that specifies what some part of the system does or what the user can do. (Section [31.15.2](#))

fundamental geometry. Nearby static components in personal space and action space that add to the fundamental experience. Contains limited affordances. (Section [21.1](#))

future effect distance cues. Internal psychological beliefs of how distance may personally affect the viewer in the future. Such cues include one's intended actions and fear. (Section [9.1.3](#))

gamification. Taking what some people consider mundane tasks and making the core of that task enjoyable, challenging, and rewarding. (Section [20.2](#))

gaze-directed steering. A form of the Steering Pattern where the user moves in the direction he is looking. (Section [28.3.2](#))

gaze-shifting eye movements. Eye rotation that enables one to track moving objects or look at different objects. Can be classified as pursuit, saccades, or vergence. (Section [8.1.5](#))

gaze-stabilizing eye movements. Eye movement that enable people to see objects clearly and for objects to appear stable even as their head moves. Can be classified as vestibular-ocular reflex, optokinetic reflex, or nystagmus. (Section [8.1.5](#))

generalization. Drawing global conclusions bassed off of one or more experiences. (Section [7.9.3](#))

generative language. A form of communication that transforms the way we perceive the world through the use of concepts that cannot be described with existing/traditional descriptive languages. (Section [35.3.1](#))

gestalt groupings. Rules of how we naturally perceive objects as organized patterns and objects, enabling us to form some degree of order from the chaos of individual components. (Section [20.4.1](#))

gestalt psychology. A theory that states perception depends on a number of organizing principles, which determine how we perceive objects and the world. The human mind considers objects in their entirety before, or in parallel with, perception of their individual parts, suggesting the whole is different than the sum of its parts. (Section [20.4](#))

gesture. A movement of the body or body part. (Section [26.4.1](#))

ghosting. A second simultaneous rendering of an object in a different pose than the actual object. (Section [26.8](#))

go-go technique. A form of the Hand Selection Pattern and Direct Hand Manipulation Pattern that expands upon non-realistic hands where the virtual hands start to grow in a nonlinear manner when the arms go beyond 2/3 of their physical reach. (Section [28.1.1](#))

gorilla arm. Arm fatigue resulting from extended use of gestural interfaces without resting the arm(s). (Section [14.1](#))

gradient flow. The different speed of optic flow for different parts of the retinal image—fast near the observer and slower further away. (Section [9.3.3](#))

grating acuity. The ability to distinguish the elements of a fine grating composed of alternating dark and light stripes or squares. (Section [8.1.4](#))

ground. The background perceived to extend behind a figure. (Section [20.4.2](#))

gulf of evaluation. The lack of understanding about the results of an action. (Section [25.4](#))

gulf of execution. The gap between the current state and a known goal when it is not clear how to achieve it. (Section [25.4](#))

hand pointing. A form of the Pointing Pattern where the ray is extended from the hand or finger. (Section [28.1.2](#))

hand reference frames. The reference frames defined by the position and orientation of the user's hands. (Section [26.3.4](#))

Hand Selection Pattern. A direct object-touching interaction pattern that mimics real-world interaction—the user directly reaches out the hand to touch some object and then triggers a grab. (Section [28.1.1](#))

hand-held augmented reality. A form of video-see-through augmented reality that utilizes a hand-held display. Also known as indirect augmented reality. (Section [3.2.1](#))

hand-held display. A output device that can be held with the hand(s) and does not require precise tracking or alignment with the head/eyes. (Section [3.2.1](#))

hand-held tool. A virtual object with geometry and behavior that is attached to / held with a hand. (Section [28.2.3](#))

handrail. A linear feature of an environment, such as a side of a building or a fence, which is used to guide navigation and psychologically constrain movement. (Section [21.5](#))

hand-worn input devices. A class of input devices that are worn on the hand such as gloves, muscle-tension sensors, and rings. (Section [27.2.4](#))

haptics. The simulation of physical forces between virtual objects and the user's body. Can be classified as passive haptics or active haptics, tactile haptics or proprioceptive force, and self-grounded haptics or world-grounded haptics. (Section [3.2.3](#))

head pointing. A form of the Pointing Pattern where the selection ray is extended from the cyclopean eye. Typically presented to the user as small pointer or reticle at the center of the field of view. (Section [28.1.2](#))

head reference frame. The reference frame based on the point between the two eyes and a reference direction perpendicular to the forehead. Also see cyclopean eye. (Section [26.3.5](#))

head tracking input. A class of input that refers to interaction that modifies or provides feedback beyond just seeing the virtual environment. (Section [27.3.1](#))

head crusher technique. A form of the Image-Plane Selection Pattern where the user positions his thumb and forefinger around the desired object in the 2D image plane. (Section [28.1.3](#))

head-motion prediction. Extrapolation of future head pose. A commonly used delay compensation technique for HMD systems. (Section [18.7.1](#))

head-mounted display (HMD). A display that is more or less rigidly attached to the head. Can be categorized as a non-see-through HMD, video-see-through HMD, or optical-see-through HMD. (Section [3.2.1](#))

head-related transfer function (HRTF). A spatial filter that describes how sound waves interact with the listener's body, most notably the outer ear, from a specific location. (Section [21.3](#))

headset fit. How well an HMD fits a user's head and how comfortable it feels. (Section [14.2](#))

heads-up display (HUD). Visual information overlaid on the scene in the general forward direction. (Section [23.2.2](#))

height relative to horizon. A pictorial depth cue that causes objects to appear further away when they are closer to the horizon. (Section [9.1.3](#))

hierarchical task analysis. A task analysis that decomposes tasks into smaller subtasks until a sufficient level of detail is reached. (Section [32.1.2](#))

highlighting. Visual outlining or changing the color of an object. (Section [26.8](#))

histogram. A graphical representation of a distribution of data. (Section [33.5.2](#))

history. A threat to internal validity resulting from an extraneous environmental event that occurs when something other than the manipulated variable happens between the pre-test and the post-test. (Section [33.2.3](#))

HOMER technique (Hand-centered Object Manipulation Extending Ray-casting). A form of the Pointing Hand Pattern where the hand jumps to the object after selection by pointing, enabling the user to directly position and rotate the object as if it were held in the hand. (Section [28.5.1](#))

homunculus. A part of the brain that maps the location and proportion of sensory and motor cortex to different body parts. The "body within the brain." (Section [8.3](#))

horopter. The surface in space where there is no binocular disparity between the images on each of the eyes. (Section [9.1.3](#))

human joystick. A form of the Walking Pattern where the direction and velocity of travel is defined by where the user is standing relative to a central zone . . (Section [28.3.1](#))

human-centered design. A design philosophy that puts human needs, capabilities, and behavior first, then designs to accommodate those needs, capabilities, and ways of behaving. (Overview)

hybrid architecture. A network architecture that uses elements of peer-to-peer and client-server architectures. (Section [32.5.3](#))

hybrid tracking system A tracking system that fuses both relative and absolute measurements to provide the advantages of both. (Section [27.1.3](#))

hypothesis. A predictive statement about the relationship between two variables that is testable and falsifiable. (Section [33.4.1](#))

illusion. The perception of something that does not exist in objective reality. (Section [6.2](#))

illusory contour. The perception of boundaries when no boundary actually exists. (Section [6.2.2](#))

Image-Plane Selection Pattern. A selection pattern simulating touch at a distance, where the user holds the hand(s) between one eye and the desired object. Also known as occlusion and framing. (Section [28.1.3](#))

immersion. The objective degree to which a VR system and application projects stimuli onto the sensory receptors of users that is extensive, matching, surrounding, vivid, interactive, and plot informing. (Section [4.1](#))

inattentional blindness. The failure to perceive an object or event due to not paying attention that can occur even when looking directly at it. (Section [10.3.1](#))

independent variable. The controlled experimental input that is varied or manipulated by the experimenter. (Section [33.4.1](#))

indirect communication. The connection of two or more entities through some intermediary. Includes what we normally think of language such as spoken, written, and sign languages, as well as internal thoughts. (Section [1.2.2](#))

indirect control patterns. A set of interaction patterns that provides control through an intermediary to modify an object, the environment, or the system. Includes the Widgets and Panels Pattern and the Non-Spatial Control Pattern. (Section [28.4](#))

indirect gesture. A gesture that indicates complex semantic meaning over a period of time. The start of an indirect gesture is not sufficient for the system to start immediately responding. Conveys symbolic, pathic, and affective information. (Section [26.4.1](#))

indirect interaction. An interaction that requires cognitive conversion between input and output. (Section [25.3](#))

induced motion. Illusory motion that occurs when motion of one object induces the perception of motion in another object. (Section [9.3.5](#))

inhibition of return. The decreased likelihood of moving the eyes to look at something that has already been looked at. (Section [10.3.2](#))

input. Data from the user that affects the application. Examples include where the user's eyes are located, where the hands are located, and button presses. (Section [3.2](#))

input device. A physical tool/hardware used to convey information to the application and to interact with the virtual environment. (Chapter [27](#))

input device class. A set of input devices that share the same essential characteristics that are crucial to interaction. (Section [27.2](#))

input veracity. The degree to which an input device captures and measures a user's actions. Includes precision, accuracy, and latency. (Section [26.1](#))

instrumentation. A threat to internal validity resulting from change in the measurement tool or observers' rating behavior. (Section [33.2.3](#))

integral input device. An input device that enable control of all degrees of freedom simultaneously from a single motion (a single composition). (Section [27.1.4](#))

interaction. A subset of communication that occurs between a user and the VR application that is mediated through the use of input and output devices. (Part [V](#))

interaction fidelity. The degree to which physical actions for a virtual task correspond to physical actions for the equivalent real-world task. Consists of biomechanical symmetry, input veracity, and control symmetry. (Sections [4.4.2](#), [26.1](#))

interaction metaphor. An interaction concept that exploits specific knowledge that users already have of other domains. (Section [25.1](#))

interaction pattern. A generalized high-level interaction concept that can be used over and over again across different applications to achieve common user goals. (Chapter [28](#))

interaction technique. A more specific and technology dependent description of an interaction scheme than an interaction pattern. (Chapter [28](#))

interactive objects. Dynamic items that can be interacted with. Located in personal space when directly interacted with and most commonly in action space when indirectly interacted with. (Section [21.1](#))

interface. The system side of the interaction that exists whether a user is interacting or not. (Part [V](#))

internal representation. Thoughts that are constructed in the form of a mental model that takes the form of sensory perceptions that may or may not exist as external stimuli. (Section [7.9.4](#))

internal validity. The degree of confidence that a relationship is causal. (Section [33.2.3](#))

interpupillary distance (IPD). The distance between the eyes. (Section [17.2](#))

interquartile range. The range of the middle 50

interstimulus interval. The blanking time between flashing stimuli. (Section [9.3.6](#))

interval variable. A variable that takes on values that are ordered and equally spaced but do not have an inherent absolute zero. (Section [33.5.2](#))

interview. A series of open-ended questions asked by a real person. (Section [33.3.3](#))

INVEST. An acronym used to describe quality user stories—Independent, Negotiable, Valuable, Estimable, Small, and Testable. (Section [31.12](#))

)

isometric input device. An input device that measures pressure or force that contains no or little actual movement. (Section [27.1.5](#))

isotonic input device. An input device that measures deflection from a center point and may or may not have some resistance. (Section [27.1.5](#))

iterative design. Quickly defining and testing many implementations, continually improving upon previous ideas. Consists of the Define Stage, Make Stage, and Learn Stage. (Part [VI](#), Section refs:30.1)

iterative perceptual process. An ever-evolving sequence of processing steps that consists of stimulus to physiology to perception to stimulus as the result of action. (Section [7.5](#))

jig. A referenceable and adjustable constraints tool that can be attached to vertices, edges, and faces in order to precisely manipulate objects. (Section [28.2.3](#))

jitter. Fast shaking of an object or the world, most commonly due to imprecision. Tracker jitter (resulting from imprecise tracking) for an HMD results in the perception that the world is shaking. (Sections [17.1](#), [32.4.2](#))

judder. The appearance of jerky or unsmooth visual motion. (Section [9.3.6](#))

just-in-time pixels. Rendering from a new viewpoint every pixel instead of every frame resulting in no tearing when not waiting on vertical sync. (Section [15.4.4](#))

Kennedy Simulator Sickness Questionnaire (SSQ). The most commonly used tool for measuring simulator sickness and VR sickness. (Section [16.1](#))

key player. Someone who is essential to the success of the project. (Section [31.6](#))

kinetic depth effect. A special form of motion parallax that describes how 3D structural form can be perceived when an object moves. (Section [9.1.3](#))

landmark. A disconnected static cue in the environment that is unmistakable in form from the rest of the scene. (Section [21.5](#))

latency. The time it takes for a system to respond to a user's action. The time from the start of movement to the time a pixel resulting from that movement responds. Effective latency can be less than true latency by utilizing delay compensation techniques. (Chapter [15](#))

latency meter. A device used to measure system delay. (Section [15.5.1](#))

lateral geniculate nucleus (LGN). A relay center that sends visual signals from the eyes, the visual cortex, and the reticular activating system to different parts of the visual cortex. (Section [8.1.1](#))

leading indicator. A cue that helps users to reliably predict how the viewpoint will change in the near future. Can help alleviate motion sickness. (Section [18.4](#))

Learn Stage. The portion of the iterative design process that is all about continuous discovery of what works well and what doesn't work well. (Part [VI](#))

learned helplessness. The decision that something cannot be done, at least by the person making the decision, resulting in giving up due to a perceived absence of control. (Section [7.8](#))

lifting palm technique. A form of the Image-Plane Selection Pattern where the user flattens her outstretched hand and positions the palm so that it appears to lie below the desired object. (Section [28.1.3](#))

light adaptation. The decrease in one's sensitivity to light in light conditions. (Section [10.2.1](#))

light field. The light that flows in multiple directions through multiple points in space. (Section [21.6.2](#))

lightness. The apparent reflectance of a surface, with objects reflecting a small proportion of light appearing dark and objects reflecting a larger proportion of light appearing light/white. Also referred to as whiteness. (Section [8.1.2](#))

lightness constancy. The perception that the lightness of an object is more a function of the reflectance of an object and intensity of the surrounding stimuli rather than the amount of light reaching the eye. (Section [10.1.4](#))

likert scale. A statement about a particular topic and respondents indicate their level of agreement from strongly disagree to strongly agree. (Section [33.3.4](#))

linear perspective. The fact that parallel lines receding into the distance appear to converge together at a single point called the vanishing point. (Section [9.1.3](#))

lip sync. The synchronization between the visual movement of the speaker's lips and the spoken voice. (Section [8.7](#))

location-based VR. A VR system that requires a suitcase or more, takes time to set up, and ranges from one's living room or office to out-of-home experiences such as VR Arcades or theme parks. (Section [22.4.2](#))

locus of control. The extent to which one believes he has control. Actively being in control of one's navigation allows one to anticipate future motions and reduces motion sickness. (Section [17.3](#))

loudness constancy. The perception that a sound source maintains its loudness even when the sound level at the ear diminishes as the listener moves away from the sound source. (Section [10.1.5](#))

magical interaction. A VR hyper-natural interaction where physical movements make users more powerful by giving them new and enhanced abilities or intelligent guidance. Considered to be medium interaction fidelity. (Section [26.1](#))

magno cells. Neurons in the visual system with large bodies that have a transient response and large receptive field resulting in optimization of detecting motion, time keeping/temporal analysis, and depth perception. (Section [8.1.1](#))

Make Stage. The portion of the iterative design process where the specific design and implementation occurs. (Part VI , Chapter 32)

manipulation. The modification of attributes for one or more objects such as position, orientation, scale, shape, color, and texture. (Section [28.2](#))

manipulation patterns. A set of interaction patterns that enable the manipulation of one or more objects. Includes the Direct Hand Manipulation Pattern, Proxy Pattern, and 3D Tool Pattern. (Section [28.2](#))

map. A symbolic representation of a space, where relationships between objects, areas, and themes are conveyed. Can be a forward-up map or north-up map. (Section [22.1.1](#))

mapping. A relationship between two or more things. (Section [25.2.5](#))

marker. A user placed cue. (Section [21.5](#))

marketing prototype. A prototype built to attract positive attention to the company/project and are most often shown at meetups, conferences, trade shows, or made publicly available via download. (Section [32.6.1](#))

masking. The lack of perceiving one stimulus, or perceiving it less accurately, due to some other stimulus occurring. (Section [9.2.2](#))

maturity. A threat to internal validity resulting from personal change in participants over time. (Section [33.2.3](#))

mean. The most common form of the average that takes into account and weighs all values of a dataset equally. (Section [33.5.2](#))

mean deviation. How far values are, on average, from the overall dataset mean. Also known as the mean absolute deviation. (Section [33.5.2](#))

median. The middle value in an ordered dataset. (Section [33.5.2](#))

memories. Past experiences that are reenacted consciously in the subjective present. (Section [7.9.3](#))

mental model. A simplified explanation in the mind of how the world or some aspect of the world works. (Section [7.8](#))

meta program. The most unconscious type of psychological filter that is content-free and applies to all situations. (Section [7.9.3](#))

microphones. A class of input devices that utilize an acoustic sensor that transforms physical sound into an electric signal. (Section [27.3.3](#))

Midas touch problem. A challenge with eye tracking that refers to the fact that people expect to look at things without that look "meaning" something. (Section [27.3.2](#))

minimal prototype. A prototype built with the least amount of work necessary to receive meaningful feedback. (Section [32.6](#))

mini-retrospectives. Short focused discussions where the team discusses what is going well and what needs improvement. (Section [33.3.1](#))

mirror neuron. A neuron that responds in a similar way whether an action is performed or the same action is observed. Not proven to exist in humans. (Section [10.4.2](#))

mobile VR. A VR system that fits inside a small bag and enable instantaneous immersion at any time and any location when engagement with the real world is not required. (Section [22.4.2](#))

mode. The most frequently occurring value in a dataset. (Section [33.5.2](#))

monocular display. A display with a single image for a single eye. (Section [9.1.3](#))

morpHEME. A minimal grammatical unit of a language, with each morpheme constituting a word or meaningful part of a word that cannot be

divided into smaller independent grammatical parts. (Section [8.2.3](#))

motion aftereffect. An illusion that causes one to perceive motion after viewing stimuli moving in the opposite direction for 30 seconds or more. (Section [6.2.7](#))

motion coherence. The correlation between movements of dots in successive images. (Section [9.3.7](#))

motion depth cues. Depth cues from relative movements on the retina. (Section [9.1.3](#))

motion parallax. A motion depth cue where the images of distal stimuli projected onto the retina (proximal stimuli) move at different rates depending on their distance. (Section [9.1.3](#))

motion platform. A hardware device that moves the entire body resulting in a sense of physical motion and gravity. Such motions can help to convey a sense of orientation, vibration, acceleration, and jerking. Can be classified as active or passive. (Section [3.2.4](#))

motion sickness. Adverse symptoms and readily observable signs that are associated with exposure to real (physical or visual) and/or apparent motion. (Part III , Chapter 12)

motion smear. The trail of perceived persistence that is left by a moving object. (Section [9.3.8](#))

multicast. A one-to-many or many-to-many distribution of group communication where information is simultaneously sent to a group of addresses instead of a single address at a time. (Section [32.5.2](#))

multimodal interactions. The combining of multiple input and output sensory modalities to provide the user with a richer set of interactions. (Section [26.6](#))

Multimodal Pattern. A compound pattern that integrate different sensory/motor input modalities together. (Section [28.5.3](#))

naive search. Searching for a specific target where the location is unknown. (Section [10.4.3](#))

natural decay. The refraining of any activity such as relaxing with little movement and the eyes closed in order to reduce VR aftereffects. (Section [13.4.1](#))

navigation. Determining and maintaining a course or trajectory to an intended location. Consists of wayfinding and travel. (Section [10.4.3](#))

navigation by leaning. A form of the Steering Pattern where the user moves in the direction she is leaning. (Section [28.3.2](#))

negative aftereffect. A changes in perception of the original stimulus after the adapting stimulus has been removed. (Section [10.2.3](#))

negative afterimage. An illusion where the eyes continue seeing the inverse colors of an image even after the image is no longer physically present. (Section [6.2.6](#))

network architecture. A model that describes how users and virtual worlds are connected. Can be described classified as peer-to-peer, client-server, or hybrid architectures. (Section [32.5.3](#))

network consistency. The ideal goal for a network that, at any point in time, all users should perceive the same shared information at the same time. Can be broken down into synchronization, causality, and concurrency. (Section [32.5](#))

neuro-linguistic programming (NLP). A psychological model that explains how humans process stimuli that enter into the mind through the senses, and helps to explain how we perceive, communicate, learn, and behave. (Section [7.9](#))

nodes. An interchanges between routes or an entrances to a region. (Section [21.5](#))

noise-induced hearing loss. Decreased sensitivity to sound that can be caused by a single loud sound or by lesser audio levels over a long period of time. (Section [14.3](#))

nomadic VR. A fully walkable system that is untethered. (Section [22.4](#))

non-authoritative server. A network server that technically follows the client-server model but behave in many ways like a peer-to-peer model because the server only relays messages between clients. (Section [32.5.3](#))

non-dominant hand. The hand that best initiates manipulation, performs gross movements, and provides the reference frame for the dominant hand to work with when performing bimanual asymmetric interaction. (Section [25.5.1](#))

non-isomorphic rotation. A rotational mapping of the hand to an object that has a control/display ratio greater or less than one. (Section [28.2.1](#))

non-realistic hands. A form of the Hand Selection Pattern where virtual hands or 3D cursors corresponding to the hands do not try to mimic reality but instead focus on ease of interaction. (Section [28.1.1](#))

non-realistic interaction. A VR interaction that in no way relates to reality (i.e., low interaction fidelity) such as pushing a button on a controller to shoot a laser from the eye. (Section [26.1](#))

non-see-through HMD. An HMD that blocks out all cues from the real world. Ideal for a fully immersive VR experience. (Section [3.2.1](#))

Non-Spatial Control Pattern. An indirect control pattern that provides global action performed through description instead of spatial relationships. (Section [28.4.2](#))

non-spatial mapping. A functions that transform a spatial input into a non-spatial output or a non-spatial input into a spatial output. (Section [25.2.5](#))

non-tracked hand-held controllers. A class of input devices that are held in the hand and include buttons, joysticks/analog sticks, triggers, etc. but is not tracked in 3D space. (Section [27.2.2](#))

normal distribution. A symmetric distribution of data that peaks at the mean and varies in width and height but is well defined mathematically and has useful properties for data analysis. Also known as a bell curve or Gaussian function. (Section [33.5.2](#))

north-up map. A map that is independent of the user's orientation; no matter how the user turns or travels, the information on the map does not rotate. (Section [22.1.1](#))

nudge (selection). A step of the two-handed box selection technique where the pose of the box is incrementally adjusted no matter the distance from the user. (Section [28.1.4](#))

nulling compliance. The matching of the initial placement of a virtual object when the corresponding input device returns to its initial placement. Achievable with absolute input devices but not relative input devices. (Section [25.2.5](#))

nystagmus. A rhythmic and involuntary rotation of the eyes as a person rotates that can help stabilize the person's gaze; caused by the vestibulo-ocular reflex and optokinetic reflex. (Section [8.1.5](#))

object (software). A specific instantiation of a class in a program (Section [32.2.4](#))

object properties. Qualities of an object that tend to remain constant over time. (Section [10.1](#))

object snapping. An extension of the Pointing Pattern where the selection ray snaps or bends towards the object that is most likely to be the object of interest. (Section [28.1.2](#))

object-attached tool. A manipulable tool that is attached/colocated with an object. (Section [28.2.3](#))

objective. A high-level formalized goal and expected outcome that focuses on benefits and business outcomes over features. (Section [31.5](#))

objective reality. The world as it exists independent of any conscious entity observing it. (Chapter 6)

object-relative motion. A change of spatial relationship between stimuli. (Section [9.3.2](#))

observed score. A value obtained from a single measurement; a function of both stable characteristics and unstable characteristics. (Section [33.2.2](#))

occlusion. The strongest depth cue where close opaque objects hide other more distant objects. (Section [9.1.3](#))

ocular drift. Slow eye movement that typically occurs without the observer noticing. (Section [8.1.5](#))

oculomotor depth cues. Subtle near range depth cues that consist of vergence and accommodation. (Section [9.1.3](#))

omnidirectional treadmill. A treadmill that enables simulation of physical travel. Can be an active omnidirectional treadmill or a passive omnidirectional treadmill. (Section [3.2.5](#))

one-handed flying. A form of the Steering Pattern where the user moves in the direction the finger or hand is pointed. (Section [28.3.2](#))

onsite installation. Set up of the system and application by one or more members of the team at the customer's site. (Section [32.8.2](#))

open source. A class of licensing that allows everyone to contribute to a given project; free to use and transparent for all to see. (Section [35.4.1](#))

open standards. Standards made available to the general public that are developed (or approved) and maintained via a collaborative and consensus driven process. (Section [35.4.3](#))

open-ended question. A question with no answers to circle or check and that requires more cognitive effort to respond to in order to obtain more qualitative information. (Section [33.3.4](#))

optic flow. The pattern of visual motion on the retina—the motion pattern of objects, surfaces, and edges on the retina caused by the relative

motion between a person and the scene. (Section [9.3.3](#))

optical-see-through HMD. An HMD that enables computer-generated cues to be overlaid onto the visual field. Ideal for augmented reality experiences. (Section [3.2.1](#))

optokinetic reflex (OKR). Stabilization of gaze direction and reduction of retinal image slip as a function of visual input from the entire retina. (Section [8.1.5](#))

ordinal variable. An ordered rank of mutually exclusive categories where intervals between ranks are not equal. (Section [33.5.2](#))

otolith organs. Components of the vestibular system that acts as a three-axis accelerometers, measuring linear accelerations. (Section [8.5](#))

outlier. An observed score that is atypical from other observations. (Section [33.5.1](#))

output. The physical representation directly perceived by the user such as the pixels on the display or headphones emitting sound waves. (Section [3.2](#))

overt orienting. The physical directing of sensory receptors to optimally perceive important information about an event. (Section [10.3.2](#))

packet. A formatted piece of data that travels over a computer network. (Section [32.5.2](#))

pain. An unpleasant sensation that warns of bodily harm. Can be categorized as neuropathic pain, nociceptive pain, or inflammatory pain. (Section [8.3.4](#))

panel. A container structure that multiple widgets and other panels can be placed upon. (Section [28.4.1](#))

Panum's fusional area. The space in front of and behind the horopter where one perceives stereopsis. (Section [9.1.3](#))

partially open-ended question. A question with multiple answers that can be circled or checked and an option to fill in a different or additional information. (Section [33.3.4](#))

parvo cells. Neurons in the visual system with small bodies that have a sustained response, a small receptive field, and are color sensitive resulting in optimization of detecting local shape, spatial analysis, color vision, and texture. (Section [8.1.1](#))

passive haptics. Static physical objects that can be touched. Can be hand-held props (self-grounded haptics) or larger world-fixed objects (world-grounded haptics). (Section [3.2.3](#))

passive motion platform. A motion platform controlled by the user. (Section [3.2.4](#))

passive touch. The touching of an object on the skin where the participant has no control of the touching. (Section [8.3.3](#))

passive vehicle. A form of the Automated Pattern consisting of a virtual object users can enter or step onto that transports the user along some path not controlled by the user. (Section [28.3.4](#))

peer-to-peer architecture. A network architecture that transmits information directly between individual computers. (Section [32.5.3](#))

pendular nystagmus. Nystagmus that occurs when one rotates the head back and forth at a fixed frequency. (Section [8.1.5](#))

perceived continuity (network). The perception that all entities behave in a believable manner without visible jitter and that audio sounds smooth. (Section [32.5.1](#))

perception. High level processes that combines information from the senses and filters, organizes, and interprets those sensations to give meaning and to create subjective, conscious experiences. (Section [7.2](#))

perceptual adaptation. The alteration of a person's perceptual processes. A semipermanent change of perception or perceptual-motor coordination that serves to reduce or eliminate a registered discrepancy between or within sensory modalities or the errors in behavior induced by the discrepancy. (Section [10.2.2](#))

perceptual capacity. One's total capability for perceiving. (Section [10.3.1](#))

perceptual constancy. The impression that an object tends to remain constant in consciousness even though conditions may change. (Section [10.1](#))

perceptual continuity. The illusion of continuity and completeness, even in cases when at any given moment only parts of stimuli from the world reach our sensory receptors. (Section [9.2.2](#))

perceptual filter. Psychological processes that delete, distort, and generalize. These filters vary from deeply unconscious processes to more conscious processes and include meta programs, values, beliefs, attitudes, memories, and decisions. (Section [7.9.3](#))

perceptual load. The amount of a person's perceptual capacity that is currently being used. (Section [10.3.1](#))

perceptual moment. The smallest psychological unit of time that an observer can sense. (Section [9.2.1](#))

performance accuracy. The resulting correctness of a user's intention. (Section [31.15.1](#))

performance precision. The consistency of control that a user is able to maintain. (Section [31.15.1](#))

persistence (display). The amount of time a pixel remains on a display before going away. (Section [15.4.4](#))

persistence (perceptual). The phenomenon that a positive afterimage seemingly persists visually after the stimulus is removed. (Section [9.2.2](#))

persona. A model of a person who will be using the VR application. (Section [31.11](#))

personal space. The natural working volume within arm's reach and slightly beyond (up to about two meters from the eyes). (Section [9.1.2](#))

phoneme. The smallest perceptually distinct unit of sound in a language that helps to distinguish between similar-sounding words. (Section [8.2.3](#))

phonemic restoration effect. The perception of hearing a phoneme where it is expected to occur when that phoneme does not actually occur. (Sections [8.2.3](#), [9.2.2](#))

photic seizure. An epileptic event provoked by flicker. (Section [13.3](#))

physical panel. A real-world tracked surface that the user carries and interacts with via a tracked finger, object, or stylus. (Section [28.4.1](#))

physical trauma. An injury resulting from a real-world physical object that is an increased risk with VR due to being blind and deaf to the real world. (Section [14.3](#))

physiological measure. Objective measurement of user's physiological attributes such as heart rate, blink rate, electroencephalography (EEG), stomach upset, and skin conductance. (Section [16.3](#))

pictorial depth cues. Projections of distal stimuli onto the retina (proximal stimuli) resulting in 2D images. (Section [9.1.3](#))

pie menu. A form of the Widgets and Panels Pattern consisting of circular menus with slice shaped menu entries and selection is based on direction, not distance. Also known as marking menus. (Section [28.4.1](#))

pilot study. A smallscale preliminary experiment that acts as a test run for a more full-scale experiment. (Section [33.4.1](#))

pivot hypothesis. The phenomenon that a point stimulus at a distance will appear to move as the head moves if its perceived distance differs from its actual distance. (Section [9.3.4](#))

placebo effect. A threat to internal validity resulting from participants' expectation that the experiment causes a result instead of the experimental condition itself causing the result. (Section [33.2.3](#))

planning poker. A card game of estimating development effort. (Section [31.7](#))

Pointing Hand Pattern. A compound pattern that combines the Pointing and Direct Hand Manipulation Patterns together so that far objects are first selected via pointing and then manipulated as if held in the hand. (Section [28.5.1](#))

Pointing Pattern. A selection pattern where a ray extends into the distance and the first object intersected can then be selected via a user-controlled trigger. (Section [28.1.2](#))

position compliance. The co-location of sensory feedback with input device position. (Section [25.2.5](#))

position constancy. The perception that an object appears to be stationary in the world even as the eyes and the head move. (Section [10.1.3](#))

position-constancy adaptation. The alteration of the compensation process that keeps the environment perceptually stable during head rotation. (Section [10.2.2](#))

positive afterimage. An illusion that causes the same color to persist on the eye even after the original image is no longer present. (Section [6.2.6](#))

post-rendering latency reduction. Reduction of effective latency by first rendering to geometry larger than the final display, and then, late in the display process, selecting the appropriate subset of that geometry to be presented. (Section [18.7.2](#))

postural instability theory. A motion sickness theory that predicts sickness results when an animal lacks or has not yet learned strategies for maintaining postural stability. (Section [12.3.3](#))

postural stability test. A behavioral test for measuring motion sickness. (Section [16.2](#))

posture. A single static configuration of the body or body part. A subset of a gesture. (Section [26.4.1](#))

practical significance. The results of an experiment are important enough that they matter in practice. Also known as clinical significance. (Section [33.5.2](#))

precision. The reproducibility and repeatability of getting the same result. (Section [31.15.1](#))

precision mode pointing. A form of the Pointing Pattern that is a non-isomorphic rotation technique where the cursor seems to “slow” down due to having a control/display ratio of less than one. (Section [28.1.2](#))

presence. A sense of “being there” inside a space even when physically located in a different location. A psychological state or subjective perception in which even though part or all of an individual’s current experience is generated by and/or filtered through technology, the user fails to fully and accurately acknowledge the role of the technology in the experience. (Section [4.2](#))

primary visual pathway. A neural path that travels through the LGN in the thalamus. Also known as the geniculostriate system (Section [8.1.1](#))

primed search. Searching for a specific target where the location is previously known. (Section [10.4.3](#))

primitive visual pathway. A neural path from the end of the optic nerve to the superior colliculus. Also known as the tectopulvinar system. (Section [8.1.1](#))

principle of closure. A gestalt principle stating when a shape is not closed, the shape tends to be perceived as being a whole recognizable figure or object. (Section [20.4.1](#))

principle of common fate. A gestalt principle stating elements moving together tend to be perceived as grouped, even if they are in a larger, more complex group. (Section [20.4.1](#))

principle of continuity. A gestalt principle stating aligned elements tend to be perceived as a single group or chunk, and are interpreted as being more related than unaligned elements. (Section [20.4.1](#))

principle of proximity. A gestalt principle stating elements that are close to one another tend to be perceived as a shape or group. (Section [20.4.1](#))

principle of similarity. A gestalt principle stating elements with similar properties tend to be perceived as being grouped together. (Section [20.4.1](#))

principle of simplicity. A gestalt principle stating figures tend to be perceived in their simplest form instead of complicated shapes. (Section [20.4.1](#))

programmer prototype. A prototype created and evaluated by the programmer or team of programmers. (Section [32.6.1](#))

proprioception. The sensation of limb and whole body pose and motion derived from the receptors of muscles, tendons, and joint capsules. (Section [8.4](#))

proprioceptive forces. Haptics that provide a sense of limb movement and muscular resistance. Consists of self-grounded haptics and world-grounded haptics. (Section [3.2.3](#))

prototype. A simplistic implementation of what is trying to be accomplished without being overly concerned with aesthetics or perfection. (Section [32.6](#))

proximal stimuli. The energy from distal stimuli that reach the senses. (Section [7.1](#))

proxy. A local object (physical or virtual) that represents and maps directly to a remote object. (Section [28.2.2](#))

Proxy Pattern. A manipulation pattern where the user manipulates a local proxy object and the mapped remote object(s) is manipulated in the same way. (Section [28.2.2](#))

pursuit. The voluntary tracking with the eye of a visual target in order to maintain maximum visual acuity and to prevent motion blur. (Section [8.1.5](#))

put-that-there technique. A form of the Multimodal Pattern that uses a combination of the Pointing Pattern (to select the “that” and “there”) and the Non-Spatial Control Pattern via voice (to select the verb ‘put’). (Section [28.5.3](#))

p-value. The probability that an experiment’s result occurred by chance rather than because there is something inherently truthful about the result. (Section [33.5.2](#))

qualitative data. Subjective information that can be interpreted differently by different people. (Section [33.2.1](#))

quality requirement. A requirement that defines an overall quality or attribute of a system or application. Also known as a non-functional requirement. (Section [31.15.1](#))

quantitative data. Information about quantity that can be directly measured and expressed numerically. Also known as numerical data. (Section

[33.2.1.\)](#)

quasi-experiment. An experiment that lacks true random assignment of participants to conditions. (Section [33.4.2](#))

questionnaire. Written questions that participants are asked to respond to in writing or on a computer. (Section [33.3.4](#))

range. The minimum and maximum values of a dataset. (Section [33.5.2](#))

range of immobility. The range of displacement ratio where position constancy is perceived. (Section [10.1.3](#))

raster display. A display that scans out pixels left to right in a series of horizontal scanlines from top to bottom. (Section [15.4.4](#))

ratcheting. Discrete or instantaneous virtual turns. . (Section [18.6](#))

ratio variable. A variable that takes on values that are ordered, equally spaced, and have an inherent absolute zero. (Section [33.5.2](#))

readaptation. An adaptation back to a normal real-world situation after adapting to a non-normal situation. (Section [13.4.1](#))

re-afference. Afference that is solely due to a person's actions. (Section [7.4](#))

real environment. The real world that we live in. (Section [3.1](#))

real walking. A form of the Walking Pattern that matches physical walking with motion in the virtual environment. (Section [28.3.1](#))

realistic hands. A form of the Hand Selection Pattern where the hands appear real with full arms attached. (Section [28.1.1](#))

realistic interaction. A VR interaction that works as closely as possible to the way we interact in the real world (i.e., high interaction fidelity). (Section [26.1](#))

reality system. The hardware and operating system that full sensory experiences are built upon. The job of such a system is to effectively communicate the application content to and from the user in an intuitive way as if the user is interacting with the real world. (Section [3.2](#))

real-world prototype. A prototype that does not use any digital technology whatsoever, instead team members or users act out roles using real-world props or tools. (Section [32.6.1](#))

real-world reference frame. The reference frame defined by real-world physical space that is independent of any user motion (virtual or physical). (Section [26.3.2](#))

recognition acuity. The ability to recognize simple shapes or symbols such as letters. (Section [8.1.4](#))

redirected walking. A form of the Walking Pattern that allows users to walk in a VR space larger than the physically tracked space via rotational and translation gains. (Section [28.3.1](#))

redundant input modalities. Two or more simultaneous types of input that convey the same information to perform a single command. (Section [26.6](#))

reference frame. A coordinate system that serves as a basis to locate and orient objects. (Section [26.3](#))

reflective processing. Conscious thought ranging from basic thinking to examining one's own thoughts and feelings. (Section [7.7.3](#))

refresh rate. The number of times per second (Hz) that the display hardware scans out a full image. (Section [15.4.4](#))

refresh time. The inverse of the display refresh rate and equivalent to stimulus onset asynchrony. (Section [15.4.4](#))

regions. Areas of the environment that are implicitly or explicitly separated from each other. Also known as districts and neighborhoods. (Section [21.5](#))

registration (perceptual). The process by which changes in proximal stimuli are encoded for processing within the nervous system. (Section [10.1](#))

relative input device. An input device that senses differences between the current and last measurement. (Section [27.1.3](#))

relative/familiar size. A pictorial depth cue that causes the projection of an object on the retina to take up less visual angle when the object is further away . . (Section [9.1.3](#))

relevance filtering. A technique to reduce network traffic by only sending out a subset of information to each individual computer as a function of some criterion. (Section [32.5.5](#))

reliability (device). The extent to which an input device consistently works within the user's entire personal space. (Section [27.1.9](#))

reliability (research). The extent to which an experiment, test, or measure consistently yields the same result under similar conditions. (Section

[33.2.2](#))

rendering. The transformation of data in a computer-friendly format to a user-friendly format that gives the illusion of some form of reality. Includes visual rendering, auditory rendering (auralization), and haptics rendering. (Section [3.2](#))

rendering delay. The time from when new data enters the graphics pipeline to the time a new frame resulting from that data is completely drawn. (Section [15.4.3](#))

rendering time. The inverse of the frame rate, and in non-pipelined rendering systems is equivalent to rendering delay. (Section [15.4.3](#))

repetitive strain injuries. An injury to the musculoskeletal and/or nervous systems resulting from carrying out prolonged repeated physical activities. (Section [14.3](#))

representational fidelity. The degree to which the VR experience conveys a place that is, or could be, on Earth. (Section [4.4.2](#))

representative user prototype. A prototype designed to obtain feedback from the target audience. (Section [32.6.1](#))

requirement. A statement that conveys the expectations of the client and/or other key players such as a description of a feature, capability, and quality. A single thing the application should do. (Section [31.15](#))

research. Any systematic method of gaining and understanding new information. (Section [33.2](#))

response time (display). The time it takes for a pixel to reach some percentage of its intended intensity. (Section [15.4.4](#))

rest frame. The part of the scene that the viewer considers stationary and judges other motion relative to. (Section [12.3.4](#))

rest frame hypothesis. A motion sickness theory that states motion sickness does not arise from conflicting orientation and motion cues directly, but rather from conflicting stationary frames of reference implied by those cues. (Section [12.3.4](#))

retesting bias. A threat to internal validity resulting from collecting data from the same individuals two or more times. (Section [33.2.3](#))

reticle. A visual cue used for aiming or selecting objects. (Section [23.2.3](#))

reticular activating system. A part of the brain that enhances wakefulness and general alertness. (Section [10.3.1](#))

retina. A multilayered network of neurons covering the inside back of the eye that processes photon input. (Section [8.1](#))

retinal image slip. Movement of the retina relative to a visual stimulus being viewed. (Section [8.1.5](#))

ring menu. A form of the Widgets and Panels Pattern consisting of a rotary 1D menu with a number of options displayed concentrically about a center point. (Section [28.4.1](#))

rods. Receptors on the first layer of the retina that are responsible for vision at low levels of illumination and are located across the retina everywhere except the fovea and blind spot. (Section [8.1.1](#))

route. One or more travelable segments that connect two locations. Also known as paths. (Section [21.5](#))

route planning. The active specification of a path between the current location and the goal before being passively moved. (Section [28.3.4](#))

rumble. A simple form of haptics that vibrates an input device. (Section [26.8](#))

saccade. Fast voluntary or involuntary eye movements that allow different parts of the scene to fall on the fovea and are important for visual scanning. (Section [8.1.5](#))

saccadic suppression. The reduction of vision just before and during saccades. (Section [8.1.5](#))

saliency. The property of a stimulus that causes it to stick out from its neighbors and grab one's attention. (Section [10.3.2](#))

saliency map. A visual image that represents how parts of the scene stand out and capture attention. (Section [10.3.2](#))

scaled world grab. A form of the Pointing Hand pattern where the user is scaled to be larger or the environment is scaled to be smaller so the user can directly manipulate the object in personal space. (Section [28.5.1](#))

scene. The entire current environment that extends into space and is acted within. Can be divided into the background, contextual geometry, fundamental geometry, and interactive objects. (Section [21.1](#))

scene motion. Visual motion of the entire virtual environment that would not normally occur in the real world. (Section [12.1](#))

scene schema. The context or knowledge about what is contained in a typical environment that an observer finds himself in. (Section [10.3.2](#))

scientific method. An ongoing iterative process based on observation and experimentation that consists of predicting, testing, and refining.

(Section [33.4](#))

search. The active pursuit of relevant stimuli in the environment, scanning one's sensory world for particular features or combinations of features. (Section [10.3.2](#))

segregation. The perceptual separation of one object from another. (Section [20.4.2](#))

selection. The specification of one or more objects from a set in order to specify an object to which a command will be applied, to denote the beginning of a manipulation task, or to specify a target to travel toward. (Section [28.1](#))

selection bias. A threat to internal validity resulting from non-random assignment or self-selection of participants to groups. (Section [33.2.3](#))

selection patterns. A set of interaction patterns that enables selection. Includes the Hand Selection Pattern, Pointing Pattern, Image-Plane Selection Pattern, and Volume-Based Selection Pattern. (Section [28.1](#))

self-embodiment. The perception that the user has a body within the virtual world. (Section [4.3](#))

self-grounded haptics. Haptics that are worn/held by and move with the user. Forces are applied relative to the user. Examples include hand-held props, controllers with buzzers, and gloves with exoskeletons. (Section [3.2.3](#))

semicircular canals (SCCs). Components of the vestibular system that acts as a three-axis gyroscopes, measuring primarily angular velocities. (Section [8.5](#))

sensation. Elementary processes that enable low-level recognition of distal stimuli through proximal stimuli. (Section [7.2](#))

sensitivity. The capability of a measure to adequately discriminate between two things; the ability of an experimental method to accurately detect an effect, when one does exist. (Section [33.2.4](#))

sensory adaptation. The alteration of a person's sensitivity to detect a stimulus. (Section [10.2.1](#))

sensory conflict theory. A motion sickness theory that states sickness may result when the environment is altered in such a way that incoming information across sensory modalities (primarily visual and vestibular) are not compatible with each other and do not match our mental model of expectations. (Section [12.3.1](#))

sensory substitution. The replacement of an ideal sensory cue that is not available with one or more other sensory cues. (Section [26.8](#))

separation acuity. The smallest angular separation between neighboring stimuli that can be resolved, i.e., two stimuli are perceived as two. Also known as resolvable acuity or resolution acuity. (Section [8.1.4](#))

seperable input device. An input device that contains at least one degree of freedom that cannot be controlled simultaneously from a single motion (i.e., two or more distinct compositions). (Section [27.1.4](#))

shadowing. The act of repeating verbal input one is receiving, often among other conversations. (Section [10.3.1](#))

shadows/shading. A pictorial depth cue that provides a clue about how far away an object is and how high above the ground it is. (Section [9.1.3](#))

shape constancy. The perception that objects retain their shape, even as we look at them from different angles resulting in a changing shape of their image on the retina. (Section [10.1.2](#))

signifier. Any perceivable indicator (a signal) that communicates appropriate purpose, structure, operation, and behavior of an object to a user. (Section [25.2.2](#))

simulator sickness. Sickness that results from shortcomings of the simulation, but not from the actual situation that is being simulated. (Part [III](#))

situational properties. Qualities of an object that are more changeable than object properties. (Section [10.1](#))

size constancy. The perception that objects tend to remain the same size even as their image changes size on the retina. (Section [10.1.1](#))

sketch. A rapid freehand drawing that is not intended as a finished work but is a preliminary exploration of ideas. (Section [32.2.1](#))

skeuomorphism. The incorporation of old, familiar ideas into new technologies, even though the ideas no longer play a functional role. (Section [20.1](#))

SMART. An acronym used to describe quality objectives—Specific, Measurable, Attainable, Relevant, and Time-bound. (Section [31.5](#))

smell. The ability to perceive odors when odorant airborne molecules bind to olfactory receptors in the nose. (Section [8.6](#))

snap (selection). A step of the two-handed box selection technique where the box is brought to the hand in order to quickly access regions of interest that are within personal space. (Section [28.1.4](#))

sound amplitude. The difference in pressure between the high and lows of a sound wave. (Section [8.2.1](#))

sound frequency. The number of times per second (i.e., hertz, (Hz)) that a change in pressure is repeated. (Section [8.2.1](#))

spatial compliance. Direct spatial mapping. Consists of position compliance, directional compliance, and nulling compliance. (Section [25.2.5](#))

spatialized audio. Audio that provides a sense of where sounds are coming from in 3D space. (Sections [3.2.2](#), [21.3](#))

specialized input modality. The limitation of a single type of input for a specific task due to the nature of the task and the design of the application. (Section [26.6](#))

speech recognition. A system that translates spoken words into textual and semantic form. (Section [26.4.2](#))

speech segmentation. The perception of individual words in a conversation even when the acoustic signal is continuous. (Section [8.2.3](#))

spotter. A person closely watching a standing or walking user to help stabilize him if necessary. (Section [14.3](#))

spread. A descriptive statistic that describes how dispersed a dataset is. Measures of spread include the range, interquartile range, mean deviation, variance, and standard deviation. Also known as variability. (Section [33.5.2](#))

stable characteristics. Factors that improve reliability. (Section [33.2.2](#))

stakeholder prototype. A prototype that is semi-polished and focuses on the overall experience. (Section [32.6.1](#))

standard deviation. The most common measure of spread that is the square root of the variance. (Section [33.5.2](#))

statistical conclusion validity. The degree to which conclusions about data are statistically correct. (Section [33.2.3](#))

statistical power. The likelihood that a study will detect an effect when there truly is an effect. (Section [33.2.3](#))

statistical regression. A threat to internal validity resulting from selecting participants in advance based on high or low scores. (Section [33.2.3](#))

statistical significance. A statistical conclusion that an experiment's result did not occur by chance and thus there is likely some underlying cause for the result. (Section [33.5.2](#))

Steering Pattern. A viewpoint control pattern that continuously controls viewpoint direction without movement of the feet. (Section [28.3.2](#))

stereoblindness. The inability to extract depth signaled purely by binocular disparity. (Section [9.1.3](#))

stereopsis. The formation of a single percept that contains a vivid sense of depth due to the brain combining separate and slightly different images from each eye. Also known as binocular fusion. (Section [9.1.3](#))

stereoscopic acuity. The ability to detect small differences in depth due to the binocular disparity between the two eyes. (Section [8.1.4](#))

sticky finger technique. A form of the Image-Plane Selection Pattern where the user positions a finger on top of the desired object in the 2D image plane. (Section [28.1.3](#))

stimulus duration. The amount of time a stimulus or image is displayed. (Section [9.3.6](#))

stimulus onset asynchrony. The total time that elapses between the onset of one stimulus and the onset of the next stimulus. (Section [9.3.6](#))

storyboard. An early visual form of an experience. (Section [31.13](#))

strobing. The perception of multiple stimuli appearing simultaneously, even though they are separated in time, due to the stimuli persisting on the retina. (Section [9.3.6](#))

structural communication. The physics of the world, not the description or the mathematical representation but the thing-in-itself. For example, the bouncing of the ball off the hand or the shape of one's hand around a controller. (Section [1.2.1](#))

subconscious. Everything that occurs within our mind that we are not fully aware of but very much influences our emotions and behaviors. (Section [7.6](#))

subjective present. The precious few seconds of our ongoing conscious experience. Consists of the "now" (the current moment) and the experience of passing time. (Section [9.2.1](#))

subjective reality. The way an individual perceives and experiences the external world in his own mind. (Section [6.1](#))

subject-relative motion. A change of spatial relationship between a stimulus and the observer. (Section [9.3.2](#))

superior colliculus. A part of the brain just above the brain stem that is highly sensitive to motion and is a primary cause of VR sickness.

(Section [8.1.1](#))

super-peer. A network client that also acts as a server for regions of virtual space and can be changed between users. (Section [32.5.3](#))

symbolic communication. The use of abstract symbols (e.g., text) to represent objects, ideas, concepts, quantities, etc. (Section [35.3.2](#))

synchronization (networking). The maintenance of consistent entity state and timing of events for all users. (Section [32.5.1](#))

synchronization delay. The delay that occurs due to integration of pipelined components. (Section [15.4.5](#))

system delay. The sum of delays from tracking, application, rendering, display, and synchronization among components. Equivalent to true latency with no delay compensation. (Section [15.4](#))

system requirement. A quality requirement that describe parts of the system that are independent of the user, such as accuracy, precision, reliability, latency, and rendering time. (Section [31.15.1](#))

tactile haptics. Artificial forces that are conveyed to user's through their skin. Consists of vibrotactile stimulation and electrotactile stimulation. (Section [3.2.3](#))

target-based travel. A form of the Automated Pattern that gives a user the ability to select a goal or location he wishes to travel to before being passively moved to that location. (Section [28.3.4](#))

task analysis. Analysis of how users accomplish or will accomplish tasks, including both physical actions and cognitive processes. (Section [32.1](#))

task elicitation. Gathering information in the form of interviews, questionnaires, observation, and documentation. (Section [32.1.2](#))

task performance. The measured effectiveness of a task as performed by users. Measured with metrics such as time to completion, performance accuracy, performance precision, and training transfer. (Section [31.15.1](#))

task-irrelevant stimuli. Distracting information that is not relevant to the task with which one is involved. (Section [10.3.2](#))

taste. The chemosensory sensation of substances on the tongue. Also known as gustatory perception. (Section [8.6](#))

TCP (transport control protocol). A bidirectional reliable ordered byte stream for network communication that comes at the cost of additional latency. (Section [32.5.2](#))

team prototype. A prototype built for those on the team that are not directly implementing the application. (Section [32.6.1](#))

tearing. A visual artifact that appears as discontinuous images due to not waiting on vertical sync. (Section [15.4.4](#))

teleportation. A form of the Automated Pattern that relocates the user to a new location without any motion. (Section [28.3.4](#))

temporal adaptation. The alteration of how much time our conscious lags behind actual reality. (Section [10.2.2](#))

temporal closure. A principle of closure where the mind fills in for the time between events. (Section [20.4.1](#))

temporal compliance. The syncing in time of sensory feedback with the corresponding action or event. (Section [25.2.5](#))

testimonial. A marketing statement showing that a real person likes something. (Section [33.3.3](#))

texture gradient. A pictorial depth cue that causes texture density to increase with distance from the eye. (Section [9.1.3](#))

time perception. A person's own experience of successive change that can be manipulated and distorted under certain circumstances. (Section [9.2](#))

time to completion. How fast a task can be completed. (Section [31.15.1](#))

token. A shared data structure that can only be owned by a single user at a time. (Section [32.5.6](#))

top-down processing. Processing that is based on knowledge—that is, the influence of a user's experiences and expectations on what he perceives. Also known as knowledge-based or conceptually based processing. (Section [7.3](#))

torso reference frame. The reference frame defined by the body's spinal axis and the forward direction perpendicular to the torso. (Section [26.3.3](#))

torso-directed steering. A form of the Steering Pattern where the user moves over a terrain in the direction the torso is facing. Also known as chair-directed steering when the torso is not tracked. (Section [28.3.2](#))

tracked hand-held controllers. A class of input devices that are held in the hand, are tracked (typically with 6 degrees of freedom), and often contains functionality offered by non-tracked hand-held controllers. Also known as wands. (Section [27.2.3](#))

tracked physical prop. A tracked physical object directly manipulated by the user that maps to one or more visual objects and is often used to specify spatial relationships between virtual objects. (Section [28.2.2](#))

tracking delay. The time from when the tracked part of the body moves until movement information from the tracker's sensors resulting from that movement is input into the application or rendering component of a reality system. (Section [15.4.1](#))

trail. Evidence of a path traveled by a user that informs the person who left the path as well as other users that they have already been to that place and how they traveled to and from there. (Section [21.5](#))

training transfer. How effectively knowledge and skills obtained through the VR application transfer to the real world. (Section [31.15.1](#))

transfer (input modalities). The passing of information from one input modality to another input modality. (Section [26.6](#))

travel. The motoric component of navigation that is the act of moving from one place to another. (Section [10.4.3](#))

treadmill. A device that provides a sense that one is walking or running while actually staying in place. Can be classified as a unidirectional treadmill or omnidirectional treadmill. (Section [3.2.5](#))

trompe-l'oeil. An art technique that uses realistic 2D imagery to create the illusion of 3D from a specific static viewing location. (Part II)

true experiment. An experiment that uses random assignment of participants to groups in order to remove major threats to internal validity. (Section [33.4.2](#))

true score. The value that would be obtained if the average was taken from an infinite number of measurements. (Section [33.2.2](#))

two-handed box selection. A form of the Volume-Based Selection Pattern that uses both hands to position, orient, and shape a box via snapping and nudging. (Section [28.1.4](#))

two-handed flying. A form of the Steering Pattern where the user moves in the direction determined by the vector between the two hands and the speed is proportional to the distance between the hands. (Section [28.3.2](#))

two-handed pointing. A form of the Pointing Pattern where the selection ray originates at the near hand and extends the ray through the far hand. (Section [28.1.2](#))

UDP (user datagram protocol). A minimal connectionless transmission model for network communication that operates on a best-effort basis. (Section [32.5.2](#))

Uncanny Valley. The feeling that computer-generated characters come across as uncomfortable, creepy, or revulsive as human realism is approached but not attained. (Section [4.4.1](#))

unencumbered input device. An input device that does not require physical hardware to be held or worn (e.g., a camera system). (Section [27.1.7](#))

unified model of motion sickness. An overarching model of motion perception and motion sickness that is consistent with all five primary theories of motion sickness. (Section [12.4](#))

unstable characteristics. Factors that detract from reliability. (Section [33.2.2](#))

usability requirement. A quality requirement that describe the quality of the application in terms of convenient and practicable use. (Section [31.15.1](#))

use case. A set of steps that helps to define interactions between the user and the system in order to achieve a goal. (Section [32.2.3](#))

use case scenario. A specific example of interactions along a single path of a use case. (Section [32.2.3](#))

user story. A short concept or description of a features customers would like to see. (Section [31.12](#))

validity. The extent to which a concept, measurement, or conclusion is well founded and corresponds to real application. (Section [33.2.3](#))

values. Context-specific psychological filters that automatically judge if things are good or bad, or right or wrong. (Section [7.9.3](#))

variance. The mean squared distance from the mean. (Section [33.5.2](#))

vection. An illusion of self-motion when one is not actually physically moving in the perceived manner. (Section [9.3.10](#))

ventral pathway. A neural path from the LGN to the temporal lobe, which is responsible for recognizing and determining an object's identity. Also known as the "what" pathway. (Section [8.1.1](#))

vergence. The simultaneous rotation of both eyes in opposite directions in order to obtain or maintain binocular vision for objects at different depths. Can be classified as convergence or divergence. (Sections [8.1.5](#), [9.1.3](#))

vernier acuity. The ability to perceive the misalignment of two line segments. (Section [8.1.4](#))

vertical sync. A signal that occurs just before the refresh controller begins to scan an image out to the display. . (Section [15.4.4](#))

vestibular system Labyrinths (the otolith organs and semicircular canals) in the inner ears that act as mechanical motion detectors, which provide input for balance and sensing physical motion. (Section [8.5](#))

vestibulo-ocular reflex (VOR). Compensatory rotation of the eyes as a function of vestibular input during head rotation in order to stabilize gaze direction. (Section [8.1.5](#))

video overlap phenomenon. The phenomenon that when a person pays attention to a video that is superimposed on another video, he can easily follow the events in one video but not both. (Section [10.3.1](#))

video-see-through HMD. An HMD that enables seeing the world via real-time capture from a camera and display in a non-see-through HMD. (Section [3.2.1](#))

viewbox. A form of the World-in-Miniature Pattern that also uses the Volume-Based Selection Pattern and 3D Multi-Touch Pattern. (Section [28.5.2](#))

viewpoint control patterns. A set of interaction patterns that enables the manipulation of one's perspective and can include translation, orientation, and scale. Includes the Walking Pattern, Steering Pattern, 3D Multi-Touch Pattern, and Automated Pattern. (Section [28.3](#))

vigilance. The act of maintaining careful attention and concentration for possible danger, difficulties, or perceptual tasks, often with infrequent events and prolonged periods of time. (Section [10.3.2](#))

violated assumptions of data. Wrongly assumed properties of data that can lead to wrong conclusions. (Section [33.2.3](#))

virtual environment. An artificially created reality. In the purest cases, no content is captured from the real world. (Section [3.1](#))

virtual hand-held panel. A form of the Widgets and Panels Pattern always available in the non-dominant hand at the click of a button. (Section [28.4.1](#))

virtual reality (VR). A computer-generated digital environment that can be experienced and interacted with as if that environment were real. (Section [1.1](#))

virtual steering device. A visual representations of a real-world steering device (although it does not actually physically exist) that is used to navigate. (Section [28.3.2](#))

virtual-world reference frame. The reference frame that matches the layout of the virtual environment and includes geographic directions (e.g., north) and global distances (e.g., meters) independent of how the user is oriented, positioned, or scaled. (Section [26.3.1](#))

visceral communication. The language of automatic emotion and primal behavior, not the rational representation of the emotions and behavior. Always present for humans and the in-between of structural communication and indirect communication. (Section [1.2.1](#))

visceral processing. Reflexive and protective mechanisms tightly coupled with the motor system to help with immediate survival. (Section [7.7.1](#))

vista space. Space in the distance beyond 20 meters where one has little immediat control and perceptual cues are fairly consistent. (Section [9.1.2](#))

visual acuity. The ability to resolve visual detail. Often measured in visual angle. Types of visual acuity include detection acuity, separation acuity, grating acuity, vernier acuity, recognition acuity, and stereoscopic acuity. (Section [8.1.4](#))

visual capture. The illusion that sounds coming from one location are mislocalized to seem to come from a place of visual motion. Also known as the ventriloquism effect. (Section [8.7.1](#))

visual-physical conflict. A sensory conflict that occurs due to visual cues and corresponding proprioceptive and touch sensations not matching. Can lead to breaks-in-presence and confusion. (Section [26.8](#))

visual scanning. The looking from place to place in order to most clearly see items of interest on the fovea. (Section [10.3.2](#))

visual search. Looking for a feature or object among a field of stimuli. Can be either a feature search or a conjunction search. (Section [10.3.2](#))

voice menu hierarchy. A speech form of the Non-Spatial Control Pattern similar to traditional desktop menus where submenus are brought up after higher-level menu options are selected. (Section [28.4.2](#))

Volume-Based Selection Pattern. A selection pattern that enables selection of a volume of space and is independent of the type of data being selected. (Section [28.1.4](#))

voodoo doll technique. A form of the World-in-Miniature Pattern where image-plane selection techniques are used to temporarily create dolls.

(Section [28.5.2](#))

VR aftereffect. Any adverse health effect that occurs after VR usage but was not present during VR usage. (Section [13.4](#))

VR sickness. Any sickness caused by using VR, irrespective of the specific cause of that sickness. (Part [III](#))

walking in place. A form of the Walking Pattern where users make physical walking motions while staying in the same physical spot but moving virtually. (Section [28.3.1](#))

Walking Pattern. A viewpoint control pattern that leverages motion of the feet to control the viewpoint. (Section [28.3.1](#))

warning grid. A visual pattern presented to users when they approach the edge of tracking or a physical risk. (Section [18.10](#))

wayfinding. The mental component of navigation that does not involve actual physical movement but only the thinking that guides movement. (Section [10.4.3](#))

wayfinding aid. A cue that help people form cognitive maps and find their way in the world. Can be classified as an environmental wayfinding aid or personal wayfinding aid. (Section [21.5](#))

widget. A geometric user interface element. (Section [28.4.1](#))

Widgets and Panels Pattern. An indirect control pattern that typically follows 2D desktop widget and panel/window metaphors. (Section [28.4.1](#))

wired seated system. A reality system where the user is in a chair and it is difficult to completely turn physically due to the chair and wires. If virtual turns are not possible, the designer can assume the user is facing in a general forward direction. (Section [22.4](#))

within-subject design. An experiment that has every participant experience each condition. Also called repeated measures. (Section [33.4.1](#))

Wizard of Oz prototype. A prototype that is a basic working VR application, but a human “wizard” behind a curtain (or on the other side of the HMD) controls the response of the system in place of software. (Section [32.6.1](#))

world-grounded input devices. A class of input devices that are designed to be constrained or fixed in the real world. (Section [27.2.1](#))

world-fixed display. An output device that does not move with the head. Examples include standard monitors, CAVEs that surround the user, non-planar display surfaces, and non-moving speakers. (Section [3.2.1](#))

world-grounded haptics. Haptics that are physically attached to the real world and can provide a true sense of fully solid solid objects that don’t move. Forces are applied relative to the world. (Section [3.2.3](#))

world-in-miniature (WIM). An interactive live 3D map; an exocentric miniature graphical representation of the virtual environment one is simultaneously immersed in. (Section [28.5.2](#))

World-in-Miniature Pattern. A compound pattern that utilizes a world-in-miniature that the user can interact with and control his viewpoint from. (Section [28.5.2](#))

References

- Abrash, M. (2013). Why Virtual Reality Is Hard (and Where It Might Be Going). In *Game Developers Conference*. Retrieved from <http://media.steampowered.com/apps/valve/2013/MAbrashGDC2013.pdf> 135
- Adelstein, B. D., Burns, E. M., Ellis, S. R., and Hill, M. I. (2005). Latency Discrimination Mechanisms in Virtual Environments: Velocity and Displacement Error Factors. In *Proceedings of the 49th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 2221–2225). 164, 184
- Adelstein, B. D., Lee, T. G., and Ellis, S. R. (2003). Head Tracking Latency in Virtual Environments: Psychophysics and a Model. In *Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 2083–2987). 184
- Adelstein, B. D., Li, L., Jerald, J., and Ellis, S. R. (2006). Suppression of Head-Referenced Image Motion during Head Movement. In *Proceedings of the 50th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 2678–2682). 132, 185
- Alonso-Nanclares, L., Gonzalez-Soriano, J., Rodriguez, J. R., and DeFelipe, J. (2008). Gender Differences in Human Cortical Synaptic Density. *Proceedings of the National Academy of Sciences USA*, 105, 14615–14619. DOI: [10.1073/pnas.0803652105](https://doi.org/10.1073/pnas.0803652105).
- Anstis, S. (1986). Motion Perception in the Frontal Plane: Sensory Aspects. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. 131, 133, 185
- Anthony, S. (2013, April 2). Kinect-Based System Diagnoses Depression with 90% Accuracy. *ExtremeTech.com*. Retrieved from <http://www.extremetech.com/extreme/152309-kinect-based-system-diagnoses-depression-with-90-accuracy>
- Arditi, A. (1986). Binocular Vision. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. 185
- Attnave, F., and Olson, R. K. (1971). Pitch as a Medium: A New Approach to Psychophysical Scaling. *The American Journal of Psychology*, 84, 147–166. 100
- Azuma, R. T. (1997). A Survey of Augmented Reality. *PRESSENCE: Teleoperators and Virtual Environments*, 6 (4), 355–385. 130
- Azuma, R. T., and Bishop, G. (1995). A Frequency-Domain Analysis of Head-Motion Prediction. In *Proceedings of ACM SIGGRAPH 95* (pp. 401–408). ACM Press. 211
- Badcock, D. R., Palmisano, S., and May, J. G. (2014). Vision and Virtual Environments. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed.). Boca Raton, FL: CRC Press. 89, 92, 93, 94, 95, 121, 166
- Bailey, R. (1995, March 8). Costs Are Dropping for Virtual Reality Systems. *New York Times*. Retrieved from <http://www.nytimes.com/1995/03/08/news/08iht-ff.html> 27
- Balakrishnan, R., and Hinckley, K. (2000). Symmetric bimanual interaction. *Human Factors in Computing Systems*, 2 (1), 33–40. DOI: [10.1145/332040.332404](https://doi.org/10.1145/332040.332404). 342
- Banks, M. S., Kim, J., and Shibata, T. (2013). Insight into Vergence-Accommodation Mismatch. Proc. SPIE, 8735, 873509. DOI: [10.1117/12.2019866](https://doi.org/10.1117/12.2019866). 173
- Bhalla, M., and Proffitt, D. R. (1999). Visual-Motor Recalibration in Geographical Slant Perception. *Journal of Experimental Psychology. Human Perception and Performance*, 25, 1076–1096. DOI: [10.1037/0096-1523.25.4.1076](https://doi.org/10.1037/0096-1523.25.4.1076). 123
- Bliss, J. P., Proops, A. B., and Chancey, E. T. (2014). Human Performance Measurements in Virtual Environments. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 749–780). Boca Raton, FL: CRC Press. 397, 402
- Bolas, M., Kuruvilla, A., Chintalapud, S., Rabelo, F., Lympouridis, V., Suma, E., Barron, C., Matamoros, C., and Debevec, P. (2015). Creating Near-Field VR Using Stop Motion Characters and a Touch of Light-Field Rendering. In *ACM SIGGRAPH Posters*. 248
- Bolas, M. T. (1989). Design and Virtual Environments. Master's thesis, Stanford University. 383
- Bolas, M. T. (1992). Design and Virtual Environments. In *International Conference on Artificial Reality and Tele-Existence* (pp. 135–141). 261, 383
- Bolt, R. A. (1980). “Put-That-There”: Voice and Gesture at the Graphics Interface. In *SIGGRAPH* (pp. 262–270). DOI: [10.1145/800250.807503](https://doi.org/10.1145/800250.807503). 302, 303, 354
- Botvinick, M., and Cohen, J. (1998). Rubber Hands “Feel” Touch that Eyes See. *Nature*, 391 (6669), 756. DOI: [10.1038/35784](https://doi.org/10.1038/35784). 47
- Bowman, D. A., and Hodges, L. (1997). An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. In *ACM Symposium on Interactive 3D Graphics2* (pp. 35–38). ACM Press. 327, 332, 344, 351

Bowman, D. A., Kruifff, E., LaViola, J., Jr., and Poupyrev, I. (2004). *3D User Interfaces: Theory and Practice*. Addison-Wesley. Retrieved from <http://books.google.com/books?id=JYzmCkf7yNcC&pgis=1> 101, 153, 253, 280, 282, 307, 329, 342, 351, 475

Bowman, D. A., McMahan, R. P., and Ragan, E. D. (2012). Questioning Naturalism in 3D User Interfaces. *Communications of the ACM*, 55 (9), 78–88. DOI: [10.1145/2330667.2330687](https://doi.org/10.1145/2330667.2330687). 289, 290

Bowman, D. A., and Wingrave, C. A. (2001). Design and Evaluation of Menu Systems for Immersive Virtual Environments. In *IEEE Virtual Reality* (pp. 149–156). DOI: [10.1109/VR.2001.913781](https://doi.org/10.1109/VR.2001.913781). 348

Bowman, D. (1999). Interaction Techniques for Common Tasks in Immersive Virtual Environments. *Techniques*. 327, 328

Bowman, D., Davis, E., Badre, A., and Hodges, L. Maintaining Spatial Orientation during Travel in an Immersive Virtual Environment. *Presence: Teleoperators and Virtual Environments*, vol. 8, no. 6, 1999, pp. 618-631. 344

Bowman, D., Koller, D., and Hodges, L. A Methodology for the Evaluation of Travel Techniques for Immersive Virtual Environments. *Virtual Reality: Research, Development, and Applications*, vol. 3, no. 2, 1998, pp. 120–131. 344

Brewster, D. (1856). *The Stereoscope: Its History, Theory, and Construction, with Its Application to the Fine and Useful Arts and to Education*. London: John Murray. 15

Bridgeman, B., Van der Heijden, A. H. C., and Velichkovsky, B. M. (1994). A Theory of Visual Stability across Saccadic Eye Movements. *Behavioral and Brain Sciences*, 17 (2), 247–292. 95

Brooks, F. (1988). Grasping Reality through Illusion: Interactive Graphics Serving Science. In *CHI'88* (pp. 1–11). DOI: [10.1145/57167.57168](https://doi.org/10.1145/57167.57168). 223

Brooks, F. P. (2010). *The Design of Design: Essays From a Computer Scientist*. Addison-Wesley. 85, 229, 373, 374, 380, 388, 389, 405, 410

Brooks, F. P., Ouh-Young, M., Batter, J. J., and Jerome Kilpatrick, P. (1990). Project GROPE Haptic Displays for Scientific Visualization. *ACM SIGGRAPH Computer Graphics*, 24 (4), 177–185. DOI: [10.1145/97880.97899](https://doi.org/10.1145/97880.97899). 25

Brooks, J. O., Goodenough, R. R., Crisler, M. C., Klein, N. D., Alley, R. L., Koon, B. L., Logan, W. C., Ogle, J. H., Tyrrell, R. A., and Wills, R. F. (2010). Simulator Sickness during Driving Simulation Studies. *Accident Analysis and Prevention*, 42, 788–796. DOI: [10.1016/j.aap.2009.04.013](https://doi.org/10.1016/j.aap.2009.04.013). 202

Brooks, F. P., Jr. (1995). *The Mythical Man Month: Essays on Software Engineering* (2nd ed.). Addison-Wesley. 229, 385

Bryson, S., and Johan, S. (1996). Time Management, Simultaneity and Time-Critical Computation in Interactive Unsteady Visualization Environments. In *Proceedings of IEEE Visualization '96* (pp. 255–262). IEEE Computer Science Press. 189, 413

Burns, E., Razzaque, S., Panter, A. T., Whitton, M. C., McCallus, M. R., and Brooks, F. P., Jr. (2006). The Hand Is More Easily Fooled Than the Eye: Users Are More Sensitive to Visual Interpenetration Than to Visual-Proprioceptive Discrepancy. *Presence: Teleoperators and Virtual Environments - Special Issue: IEEE VR 2005*, 15 (1), 1–15. 109, 304, 306

Burton, T. M. W. (2012). Robotic Rehabilitation for the Restoration of Functional Grasping Following Stroke. Dissertation, University of Bristol, England. 103

Buxton, B. (2007). *Sketching User Experiences: Getting the Design Right and the Right Design. Sketching User Experiences*. Focal Press. DOI: [10.1016/B978-012374037-3/50064-X](https://doi.org/10.1016/B978-012374037-3/50064-X). 405

Buxton, W. (1986). There's More to Interaction Than Meets the Eye: Some Issues in Manual Input. In D. A. Norman and S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction* (pp. 319–337). Lawrence Erlbaum Associates. Retrieved from <http://www.billbuxton.com/eye.html> 283

Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. (1988). An Empirical Comparison of Pie vs. Linear Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 95–100). DOI: [10.1145/57167.57182](https://doi.org/10.1145/57167.57182). 346

Carmack, J. (2015). 2015 GDC - Carmack on Mobile VR (Full Livestream Recording). Retrieved April 20, 2015; from <https://www.youtube.com/watch?v=UNAmAxT7-qs> 212

Chance, S. S., Gaunet, F., Beall, A. C., and Loomis, J. M. (1998). Locomotion Mode Affects the Updating of Objects Encountered during Travel: The Contribution of Vestibular and Proprioceptive Inputs to Path Integration. *Presence: Teleoperators and Virtual Environments*. DOI: [10.1162/105474698565659](https://doi.org/10.1162/105474698565659). 336

Chen, S. E. (1995). Quicktime {VR}—An Image-based Approach to Virtual Environment Navigation. In *Proceedings of ACM SIGGRAPH 95* (pp. 29–38). ACM Press. 212

Culow, F. W. (1972). *Color: Its Principle and Their Applications*. New York: Morgan and Morgan. 92

- Cohn, M. (2005). *Agile Estimating and Planning*. Prentice Hall. [386](#)
- Comeau, C. P., and Brian, J. S. (1961, November). Headsight Television System Provides Remote Surveillance. *Electronics*, 86–90. [23](#)
- Coren, S., Ward, L. M., and Enns, J. T. (1999). *Sensation and Perception* (5th ed.). Harcourt Brace College Publishers. [86](#), [87](#), [88](#), [90](#), [91](#), [92](#), [93](#), [94](#), [98](#), [124](#), [125](#), [126](#), [128](#), [129](#), [131](#), [133](#), [134](#), [135](#), [136](#), [139](#), [140](#), [142](#), [148](#), [149](#), [296](#)
- Costello, P. (1997). *Health and Safety Issues Associated with Virtual Reality: A Review of Current Literature*. JISC Advisory Group on Computer Graphics (AGOCG) Technical Report Series No. 37, Computing Services, Loughborough University. [177](#), [179](#), [199](#)
- Craig, A. B., Sherman, W. R., and Will, J. D. (2009). *Developing Virtual Reality Applications: Foundations of Effective Design*. Elsevier. DOI: [10.1016/B978-0-12-374943-7.00005-7](https://doi.org/10.1016/B978-0-12-374943-7.00005-7). [39](#)
- Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V., and Hart, J. C. (1992). The CAVE: Audio Visual Experience Automatic Virtual Environment. *Communications of the ACM*. DOI: [10.1145/129888.129892](https://doi.org/10.1145/129888.129892). [34](#), [62](#)
- Crystal, A., and Ellington, B. (2004). Task Analysis and Human-Computer Interaction?: Approaches, Techniques, and Levels of Analysis. In *America's Conference on Information Systems* (pp. 1–9). Retrieved from <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1967&context=amcis2004> [402](#)
- Csikszentmihalyi, M. (2008). *Flow: The Psychology of Optimal Performance. Optimal Experience: Psychological Studies of Flow in Consciousness*. Harper Perennial Modern Classics. [82](#), [151](#), [228](#)
- Cunningham, D. W., Billock, V. A., and Tsou, B. H. (2001a). Sensorimotor Adaptation to Violations in Temporal Contiguity. *Psychological Science*, 12 (6), 532–535. [145](#), [184](#)
- Cunningham, D. W., Chatziastros, A., Von Der Heyde, M., and Bulthoff, H. H. (2001b). Driving in the Future: Temporal Visuomotor Adaptation and Generalization. *Journal of Vision*, 1 (2), 88–98. [184](#)
- Cutting, J. E., and Vishton, P. M. (1995). Perceiving Layout and Knowing Distances?: The Integration, Relative Potency, and Contextual Use of Different Information about Depth. In W. Epstein and S. Rogers (Eds.), *Perception of Space and Motion* (Vol. 22, pp. 69–117). San Diego, CA: Academic Press. DOI: [10.1016/B978-012240530-3/50005-5](https://doi.org/10.1016/B978-012240530-3/50005-5). [111](#), [112](#), [122](#)
- da Cruz, L., Coley, B. F., Dorn, J., Merlini, F., Filley, E., Christopher, P., et al (2013). The Argus II epiretinal prosthesis system allows letter and word reading and long-term function in patients with profound vision loss. *The British journal of ophthalmology*, 97(5), 632–636. [479](#)
- Daily, M., Howard, M., Jerald, J., Lee, C., Martin, K., McInnes, D., and Tinker, P. (2000). Distributed Design Review in Virtual Environments. In *Proceedings of the Third International Conference on Collaborative Virtual Environments* (pp. 57–63). ACM. Retrieved from DOI: [10.1145/351006.351013](https://doi.org/10.1145/351006.351013). [257](#), [419](#), [420](#)
- Daily, M., Sarfaty, R., Jerald, J., McInnes, D., and Tinker, P. (1999). The CABANA: A Reconfigurable Spatially Immersive Display. In *Projection Technology Workshop* (pp. 123–132). [34](#)
- Dale, E. (1969). *Audio-Visual Methods in Teaching* (3rd ed.). The Dryden Press. [12](#), [13](#)
- Darken, R., Cockayne, W., and Carmein, D. (1997). The Omni-directional Treadmill: A Locomotion Device for Virtual Worlds. In *ACM Symposium on User Interface Software and Technology* (pp. 213–221). DOI: [10.1145/263407.263550](https://doi.org/10.1145/263407.263550). [42](#)
- Darken, R. P. (1994). Hands-Off Interaction with Menus in Virtual Spaces. In *SPIE Stereoscopic Displays and Virtual Reality Systems* (pp. 365–371). DOI: [10.1117/12.173893](https://doi.org/10.1117/12.173893). [350](#)
- Darken, R. P., and Cevik, H. (1999). Map Usage in Virtual Environments: Orientation Issues. *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. DOI: [10.1109/VR.1999.756944](https://doi.org/10.1109/VR.1999.756944). [252](#)
- Darken, R. P., and Peterson, B. (2014). Spatial Orientation, Wayfinding, and Representation. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 467–491). Boca Raton, FL: CRC Press. [153](#), [244](#)
- Darken, R. P., and Sibert, J. L. (1993). A Toolset for Navigation in Virtual Environments. *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology UIST93*, 2740, 157–165. DOI: [10.1145/168642.168658](https://doi.org/10.1145/168642.168658). [245](#)
- Darken, R. P., and Sibert, J. L. (1996). Wayfinding Strategies and Behaviors in Large Virtual Worlds. *CHI '96*, 142–149. DOI: [10.1145/238386.238459](https://doi.org/10.1145/238386.238459). [242](#), [243](#)
- Davis, S., Nesbitt, K., and Nalivaiko, E. (2014). A Systematic Review of Cybersickness. In *Proceedings of the 2014 Conference on Interactive Entertainment* (pp. 1–9). Newcastle, NSW, Australia: ACM. [196](#)
- Debevec, P., Downing, G., M., B., Peng, H., and Urbach, J. (2015). Spherical Light Field Environment Capture for Virtual Reality Using a Motorized Pan/Tilt Head and Offset Camera. In *ACM SIGGRAPH Posters*. [248](#)
- De Haan, G., Koutek, M., and Post, F. H. (2005). IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *In Virtual*

Delaney, D., Ward, T., and McLoone, S. (2006a). On Consistency and Network Latency in Distributed Interactive Applications: A Survey—Part I. *Presence: Teleoperators and Virtual Environments*, 15 (2), 218–234. DOI: [10.1162/pres.2006.15.2.218](https://doi.org/10.1162/pres.2006.15.2.218). [415](#)

Delaney, D., Ward, T., and McLoone, S. (2006b). On Consistency and Network Latency in Distributed Interactive Applications: A Survey—Part II. *Presence: Teleoperators and Virtual Environments*, 15 (4), 465–482. [420](#)

DiZio, P., Lackner, J. R., and Champney, R. K. (2014). Proprioceptive Adaptation and Aftereffects. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 835–856). Boca Raton, FL: CRC Press. [175](#)

Dodgson, N. A. (2004). Variation and Extrema of Human Interpupillary Distance. In *SPIE Stereoscopic Displays and Virtual Reality Systems 5291* (pp. 36–46). [203](#)

Dorst, K., and Cross, N. (2001). Creativity in the Design Process: Co-evolution of Problem-Solution. *Design Studies*, 22 (5), 425–437. DOI: [10.1016/S0142-694X\(01\)00009-6](https://doi.org/10.1016/S0142-694X(01)00009-6). [380](#)

Drachman, D. A. (2005). Do We Have Brain to Spare? *Neurology*, 64 , 2004–2005. DOI: [10.1212/01.WNL.0000166914.38327.BB](https://doi.org/10.1212/01.WNL.0000166914.38327.BB).

Draper, M. H. (1998). *The Adaptive Effects of Virtual Interfaces: Vestibulo-Ocular Reflex and Simulator Sickness* . University of Washington. [97](#), [98](#), [107](#), [144](#), [175](#)

Duh, H. B., Parker, D. E., and Furness, T. A. I. (2001). An ‘Independent Visual Background’ Reduced Balance Disturbance Evoked by Visual Scene Motion: Implication for Alleviating Simulator Sickness. In *Proceedings of ACM CHI 2001* (pp. 85–89). [168](#)

Eastgate, R. M., Wilson, J. R., and D’Cruz, M. (2014). Structured Development of Virtual Environments. In Kelly S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 353–389). Boca Raton, FL: CRC Press. [237](#)

Ebenholtz, S. M., Cohen, M. M., and Linder, B. J. (1994). The Possible Role of Nystagmus in Motion Sickness: A Hypothesis. *Aviation Space and Environmental Medicine*, 65 (11), 1032–1035. [169](#)

Eg, R., and Behne, D. M. (2013). Temporal Integration for Live Conversational Speech. In *Auditory-Visual Speech Processing* (pp. 129–134). [108](#)

Ellis, S. R. (2014). Where Are All the Head Mounted Displays? Retrieved April 14, 2015, from http://humansystems.arc.nasa.gov/groups/acd/projects/hmd_dev.php [16](#)

Ellis, S. R., Mania, K., Adelstein, B. D., and Hill, M. I. (2004). Generalizeability of Latency Detection in a Variety of Virtual Environments. In *Proceedings of the 48th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 2083–2087). [98](#), [184](#)

Ellis, S. R., Young, M. J., Adelstein, B. D., and Ehrlich, S. M. (1999). Discrimination of Changes in Latency during Head Movement. In *Proceedings of Human-Computer Interaction 99* (pp. 1129–1133). L. Erlbaum Associates, Inc. [184](#)

Fann, C.-W., Jiang, J.-R., and Wu, J.-W. (2011). Peer-to-Peer Immersive Voice Communication for Massively Multiplayer Online Games. *2011 IEEE 17th International Conference on Parallel and Distributed Systems* , 759–764. DOI: [10.1109/ICPADS.2011.99](https://doi.org/10.1109/ICPADS.2011.99). [421](#)

Ferguson, E. S. (1994). *Engineering and the Mind’s Eye* . MIT Press. [380](#)

Foley, J., Dam, A., Feiner, S., and Hughes, J. (1995). *Computer Graphics: Principles and Practice* . Reading, MA: Addison-Wesley Publishing. [32](#)

Forsberg, A., Herndon, K., and Zeleznik, R. (1996). Aperture based selection for immersive virtual environments. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology* . ACM, New York, NY, USA, 95–96. DOI: [10.1145/237091.237105](https://doi.org/10.1145/237091.237105). [331](#)

Fuchs, H. (2014). Telepresence: Soon Not Just a Dream and a Promise. In *IEEE Virtual Reality* .

Gabbard, J. L. (1997). *A Taxonomy of Usability Characteristics in Virtual Environments* . Virginia Tech. [441](#)

Gabbard, J. L. (2014). Usability Engineering of Virtual Environments. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 721–747). Boca Raton, FL: CRC Press. [403](#), [440](#), [441](#), [442](#)

Ganel, T., Tanzer, M., and Goodale, M. A. (2008). A Double Dissociation between Action and Perception in the Context of Visual Illusions. *Psychological Science*, 19 , 221–225. DOI: [10.1111/j.1467-9280.2008.02071.x](https://doi.org/10.1111/j.1467-9280.2008.02071.x). [152](#)

Gautier, L., Diot, C., and Kurose, J. (1999). End-to-End Transmission Control Mechanisms for Multiparty Interactive Applications on the Internet. In *IEEE INFOCOM* (pp. 1470–1479). DOI: [10.1109/INFCOM.1999.752168](https://doi.org/10.1109/INFCOM.1999.752168). [415](#)

Gebhardt, S., Pick, S., Leithold, F., Hentschel, B., and Kuhlen, T. (2013). Extended Pie Menus for Immersive Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, 19 (4), 644–51. DOI: [10.1109/TVCG.2013.31](https://doi.org/10.1109/TVCG.2013.31). [347](#)

Gibson, J. J. (1933). Adaptation, After-Effect and Contrast in the Perception of Curved Lines. *Journal of Experimental Psychology*. DOI: 10.1037/h0074626 . [109](#)

Gliner, J. A., Morgan, G. A., and Leech, N. L. (2009). *Research Methods in Applied Settings: An Integrated Approach to Design and Analysis* (2nd ed.). Routledge. [429](#), [431](#)

Gogel, W. C. (1990). A Theory of Phenomenal Geometry and Its Applications. *Perception and Psychophysics*, 48 (2), 105–123. [132](#)

Goldstein, E. B. (2007). *Sensation and Perception* (7th ed.). Belmont, CA: Wadsworth Publishing. [73](#), [74](#), [75](#), [86](#)

Goldstein, E. B. (Ed.). (2010). *Encyclopedia of Perception* . SAGE Publications. [94](#)

Goldstein, E. B. (2014). *Sensation and Perception* (9th ed.). Cengage Learning. [72](#), [73](#), [87](#), [88](#), [91](#), [100](#), [101](#), [102](#), [152](#), [153](#)

Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. (1996). The Lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96* (pp. 43–54). DOI: 10.1145/237170.237200 . [248](#)

Gothelf, J., and Seiden, J. (2013). *Lean UX*: (E. Ries, Ed.). O'Reilly Media, Inc. [369](#), [374](#), [376](#), [388](#)

Grau, C., Ginhoux, R., Riera, A., Nguyen, T. L., Chauvat, H., Berg, M., Amengual, J. L., Pascual-Leone, A., and Ruffini, G. (2014). Conscious Brain-to-Brain Communication in Humans Using Non-invasive Technologies. *PloS One*, 9 (8), e105225. DOI: [10.1371/journal.pone.0105225](#). [477](#), [479](#)

Greene, N. (1986). Environment Mapping and Other Applications of World Projections. *IEEE Computer Graphics and Applications*, 6 (11), 21–29. [212](#)

Gregory, R. L. (1973). *Eye and Brain: The Psychology of Seeing* (2nd ed.). London: Weidenfeld and Nicolson. [73](#), [74](#), [172](#), [185](#), [186](#)

Gregory, R. L. (1997). *Eye and Brain: The Psychology of Seeing* (5th ed.). Princeton University Press. [15](#), [65](#)

Guadagno, R. E., Blascovich, J., Bailenson, J. N., and McCall, C. (2007). Virtual Humans and Persuasion: The Effects of Agency and Behavioral Realism. *Media Psychology*, 10 (1), 1–22. [49](#)

Guardini, P., and Gamberini, L. (2007). *The Illusory Contoured Tilting Pyramid* . Best illusion of the year contest 2007. Sarasota, Florida. <http://illusionoftheyear.com/cat/top-10-finalists/2007/>. [64](#)

Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior*, 19 (4), 486–517. DOI: 10.1080/00222895.1987.10735426 . [287](#)

Hallett, P. E. (1986). Eye Movements. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interactive. [95](#), [96](#), [97](#)

Hamit, F. (1993). *Virtual Reality and the Exploration of Cyberspace* . Sams Publishing. [15](#), [59](#)

Hannema, D. (2001). *Interactions in Virtual Reality* . University of Amsterdam. [299](#)

Harm, D. L. (2002). Motion Sickness Neurophysiology, Physiological Correlates, and Treatment. In K. M. Stanney (Ed.), *Handbook of Virtual Environments* (pp. 637–661). Mahwah, N. J.: Lawrence Erlbaum Associates. [165](#), [196](#), [197](#)

Harmon, L. (1973). The Recognition of Faces. *Scientific American*, 229 (5), 70–84. [59](#), [60](#)

Hartson, R., and Pyla, P. (2012). *The UX Book: Process and Guidelines for Ensuring a Quality User Experience* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. [441](#)

Heider, F., and Simmel, M. (1944). An Experimental Study of Apparent Behavior. *The American Journal of Psychology*, 57 , 243–259. DOI: 10.2307/1416950 . [225](#), [226](#)

Heilig, M. (1960). Stereoscopic-television apparatus for individual use. US Patent 2955156. [20](#), [21](#)

Heilig, M. L. (1992). El Cine Del Futuro: The Cinema of the Future. *Presence: Teleoperators and Virtual Environments*, 1 (3), 279–294. [21](#)

Hettinger, L. J., Schmidt-Daly, T. N., Jones, D. L., and Keshavarz, B. (2014). Illusory Self-Motion in Virtual Environments. In K. Hale and K. Stanney (Eds.), *Handbook of virtual environments* (2nd ed., pp. 435–465). Boca Raton, FL: CRC Press. [136](#)

Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. (1994). Passive Real-World Interface Props for Neurosurgical Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Celebrating Interdependence, CHI '94* (Vol. 30, pp. 452–458). DOI: 10.1145/191666.191821 . [334](#)

Hinckley, K., Pausch, R., Proffitt, D., and Kassell, N. F. (1998). Two-Handed Virtual Manipulation. In *ACM Transactions on Computer-Human Interaction* (Vol. 5, pp. 260–302). DOI: [10.1145/292834.292849](#) . [287](#), [315](#), [333](#)

- Ho, C. C., and MacDorman, K. F. (2010). Revisiting the Uncanny Valley Theory: Developing and Validating an Alternative to the Godspeed Indices. *Computers in Human Behavior*, 26 (6), 1508–1518. DOI: 10.1016/j.chb.2010.05.015. [51](#)
- Hoffman, H. G. (2004). Virtual-reality therapy. *Scientific American*, 291 , 58–65. DOI: 10.1038/scientificamerican0804-58. [105](#)
- Hollister, S. (2014). Oculus Wants to Build a Billion-Person MMO with Facebook. Retrieved April 8, 2015, from [http://www.theverge.com/2014/5/5/5684236/oculus-wants-to-build-a-billion-person-mmo-with-facebook 474](http://www.theverge.com/2014/5/5/5684236/oculus-wants-to-build-a-billion-person-mmo-with-facebook)
- Holloway, R. (1997). Registration Error Analysis for Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6 (4), 413–432. [164](#), [198](#), [413](#)
- Holloway, R. L. (1995). *Registration Errors in Augmented Reality Systems* . Department of Computer Science, University of North Carolina at Chapel Hill. [198](#)
- Homan, R. (1996). SmartScene: An Immersive, Realtime, Assembly, Verification and Training Application. In *Workshop on Computational Tools and Facilities for the Next-Generation Analysis and Design Environment* (pp. 119–140). NASA Conference Publication. Retrieved from [https://archive.org/stream/nasa_techdoc_19970014680/19970014680_djvu.txt 341](https://archive.org/stream/nasa_techdoc_19970014680/19970014680_djvu.txt)
- Hopkins, A. A. (2013). *Magic: Stage Illusions and Scientific Diversions, Including Trick Photography* . Dover Publications. [15](#)
- Houben, M., and Bos, J. (2010). Reduced Seasickness by an Artificial 3D Earth-Fixed Visual Reference. In *International Conference on Human Performance at Sea* (pp. 263–270). [168](#)
- Howard, I. P. (1986a). The Perception of Posture, Self Motion, and the Visual Vertical. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. [98](#), [137](#)
- Howard, I. P. (1986b). The Vestibular System. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. [107](#), [109](#), [137](#), [164](#)
- Howard, M., Tinker, P., Martin, K., Lee, C., Daily, M., and Clausner, T. (1998). The Human Integrating Virtual Environment. In *All-Raytheon Software Symposium* . [418](#)
- Hu, H., Ren, Y., Xu, X., Huang, L., and Hu, H. (2014). Reducing View Inconsistency by Predicting Avatars' Motion in Multi-server Distributed Virtual Environments. *Journal of Network and Computer Applications*, 40 (1), 21–30. DOI: 10.1016/j.jnca.2013.08.011 . [420](#)
- Hummels, C., and Stappers, P. J. (1998). Meaningful Gestures for Human Computer Interaction: Beyond Hand Postures. In *Proceedings, 3rd IEEE International Conference on Automatic Face and Gesture Recognition, FG 1998* (pp. 591–596). DOI: 10.1109/AFGR.1998.671012 . [298](#)
- Hutchins, E., Hollan, J., and Norman, D. (1986). Direct Manipulation Interfaces. In D. A. Norman and S. W. Draper (Eds.), *User-Centered System Design: New Perspectives in Human-Machine Interaction* (pp. 87–124). Boca Raton, FL: CRC Press. DOI: 10.1207/s15327051hci0104_2 . [284](#)
- Ingram, R., and Benford, S. (1995). Legibility enhancement for information visualisation. In *Proceedings Visualization '95* (pp. 209–216). DOI: 10.1109/VISUAL.1995.480814 . [245](#)
- Insko, B. E. (2001). *Passive Haptics Significantly Enhances Virtual Environments* . PhD dissertation, Department of Computer Science, UNC-Chapel Hill. [37](#)
- International Society for Presence Research. (2000). The Concept of Presence: Explication Statement. Retrieved November 6, 2014, from <http://ispr.info/>
- ITU-T. (2015). Definition of “Open Standards.” Retrieved June 20, 2015; from [http://www.itu.int/en/ITU-T/ipt/Pages/open.aspx 482](http://www.itu.int/en/ITU-T/ipt/Pages/open.aspx)
- Iwata, H. (1999). Walking about Virtual Environments on an Infinite Floor. In *IEEE Virtual Reality* (pp. 286–293). DOI: 10.1109/VR.1999.756964 . [42](#)
- Jacob, R. J. K. (1991). The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look at Is What You Get. *ACM Transactions on Information Systems* . DOI: 10.1145/123078.128728 . [318](#)
- James, T., and Woodsmall, W. (1988). *Time Line Therapy and the Basis of Personality* . Meta Publications. [81](#)
- Jerald, J. (2009). *Scene-Motion- and Latency-Perception Thresholds for Head-Mounted Displays* . Department of Computer Science, University of North Carolina at Chapel Hill. [31](#), [106](#), [132](#), [142](#), [163](#), [164](#), [183](#), [185](#), [186](#), [188](#), [189](#), [191](#), [194](#), [211](#)
- Jerald, J., Daily, M. J., Neely, H. E., and Tinker, P. (2001). Interacting with 2D Applications in Immersive Environments. In *EUROIMAGE International Conference on Augmented Virtual Environments and 3d Imaging* (pp. 267–270). [35](#)
- Jerald, J., Fuller, A. M., Lastra, A., Whitton, M., Kohli, L., and Brooks, F. (2007). Latency Compensation by Horizontal Scanline Selection for Head-Mounted Displays. *Proceedings of SPIE*, 6490 , 64901Q–64901Q–11. Retrieved from

<http://link.aip.org/link/PSISDG/v6490/i1/p64901Q/s1&Agg=doi> 33, 212

Jerald, J., Marks, R., Laviola, J., Rubin, A., Murphy, B., Steury, K., and Hirsch, E. (2012). The Battle for Motion-Controlled Gaming and Beyond (Panel). In *ACM SIGGRAPH*. [309](#)

Jerald, J., Mlyniec, P., Yoganandan, A., Rubin, A., Paullus, D., and Solotko, S. (2013). MakeVR: A 3D World-Building Interface. In *Symposium on 3D User Interfaces* (pp. 197–198). [213](#), [342](#), [438](#), [489](#)

Jerald, J., Peck, T. C., Steinicke, F., and Whitton, M. C. (2008). Sensitivity to Scene Motion for Phases of Head Yaws. In *Symposium on Applied Perception in Graphics and Visualization* (pp. 155–162). ACM Press. Retrieved from <http://portal.acm.org/citation.cfm?doid=1394281.1394310> DOI: 1394281.1394310. [132](#)

Jerald, J. (2011). iMedic. *SIGGRAPH 2011: Real-Time Live Highlights*. Retrieved Aug 24, 2015 from <https://youtu.be/n-KXs3iWyuA?t=30s>. [248](#)

Johansson, G. (1976). Spatio-temporal Differentiation and Integration in Visual Motion Perception. *Psychological Research*, 38 (4), 379–393. DOI: [10.1007/BF00309043](https://doi.org/10.1007/BF00309043). [136](#)

Johnson, D. M. (2005). *Simulator Sickness Research Summary*. Fort Rucker, AL: U.S. Army Research Institute for the Behavioral and Social Sciences. [174](#), [197](#), [201](#), [203](#), [205](#), [207](#)

Jones, J. A., Suma, E. A., Krum, D. M., and Bolas, M. (2012). Comparability of Narrow and Wide Field-of-View Head-Mounted Displays for Medium-Field Distance Judgments. In *ACM Symposium on Applied Perception* (p. 119). [27](#)

Kabbash, P., Buxton, W., and Sellen, A. (1994). Two-Handed Input in a Compound Task. In *SIGCHI Conference on Human Factors in Computing Systems* (pp. 417–423). DOI: [10.1145/191666.191808](https://doi.org/10.1145/191666.191808). [287](#)

Kant, I. (1781). *Critique of Pure Reason. Critique of Pure Reason* (Vol. 2). Retrieved from <http://www.gutenberg.org/files/4280/4280-h/4280-h.htm> 10

Kennedy, R. S., and Fowlkes, J. E. (1992). Simulator Sickness Is Polygenic and Polysymptomatic: Implications for Research. *The International Journal of Aviation Psychology*. DOI: [10.1207/s15327108ijap0201_2](https://doi.org/10.1207/s15327108ijap0201_2). [195](#)

Kennedy, R. S., Kennedy, R. C., Kennedy, K. E., Wasula, C., and Bartlett, K. M. (2014). Virtual Environments and Product Liability. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 505–518). Boca Raton, FL: CRC Press. [175](#)

Kennedy, R. S., Lane, N. E., Berbaum, K. S., and Lilienthal, M. G. (1993). A Simulator Sickness Questionnaire (SSQ): A New Method for Quantifying Simulator Sickness. *International Journal of Aviation Psychology*, 3 (3), 203–220. [163](#), [195](#), [196](#), [197](#), [207](#)

Kennedy, R. S., and Lilienthal, M. G. (1995). Implications of Balance Disturbances Following Exposure to Virtual Reality Systems. In *Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS)* (pp. 35–39). IEEE Computer Society Press. [163](#), [174](#)

Kersten, D., Mamassian, P. and Knill, D. C. (1997). Moving cast shadows induce apparent motion in depth. *Perception*, 26, 171–192. [117](#)

Keshavarz, B., Hecht, H., and Lawson, B. D. (2014). Visually Induced Motion Sickness. In K. Hale and K. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 647–698). Boca Raton, FL: CRC Press. [169](#), [175](#), [213](#)

Kim, Y. Y., Kim, H. J., Kim, E. N., Ko, H. D., and Kim, H. T. (2005). Characteristic Changes in the Physiological Components of Cybersickness. *Psychophysiology*, 42 (5), 616–625. DOI: [10.1111/j.1469-8986.2005.00349.x](https://doi.org/10.1111/j.1469-8986.2005.00349.x). [196](#)

Klumpp, R. G. (1956). Some Measurements of Interaural Time Difference Thresholds. *The Journal of the Acoustical Society of America*. DOI: [10.1121/1.1908493](https://doi.org/10.1121/1.1908493). [101](#)

Kohli, L. (2013). *Redirected Touching*. University of North Carolina at Chapel Hill. [109](#), [306](#)

Kolasinski, E. (1995). *Simulator Sickness in Virtual Environments*. U.S. Army Research Institute for the Behavioral and Social Sciences, Technical Report 1027. [163](#), [203](#), [204](#)

Kopper, R., Bowman, D. A., Silva, M. G., and McMahan, R. P. (2010). A Human Motor Behavior Model for Distal Pointing Tasks. *International Journal of Human Computer Studies*, 68 (10), 603–615. DOI: [10.1016/j.ijhcs.2010.05.001](https://doi.org/10.1016/j.ijhcs.2010.05.001). [328](#)

Korzybski, A. (1933). *Science and Sanity—An Introduction to Non-Aristotelean Systems and General Semantics*. European Society for General Semantics. [79](#)

Krum, D. M., Suma, E. A., and Bolas, M. (2012). Augmented Reality using Personal Projection and Retroreflection. *Personal and Ubiquitous Computing*, 16 (1), 17–26. DOI: [10.1007/s00779-011-0374-4](https://doi.org/10.1007/s00779-011-0374-4). [35](#)

Kumar, M. (2007). *Gaze-Enhanced User Interface Design*. PhD dissertation, Stanford University. [319](#)

Kurtenbach, G., Sellen, A., and Buxton, W. (1993). An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus.

Lackner, J. (2014). Motion sickness: More Than Nausea and Vomiting. *Experimental Brain Research*, 232 , 2493–2510. DOI: [10.1007/s00221-014-4008-8](https://doi.org/10.1007/s00221-014-4008-8). 87 , 200 , 201 , 207 , 214

Lackner, J. R., and Teixeira, R. (1977). Visual-Vestibular Interaction: Vestibular Stimulation Suppresses the Visual Induction of Illusory Self-Rotation. *Aviation, Space and Environmental Medicine*, 48 , 248–253. 164

Lang, B. (2015, January 5). Visionary VR Is Reinventing Filmmaking's Most Fundamental Concept to Tell Stories in Virtual Reality. *Road to VR*. 254

Larman, C. (2004). *Agile and Iterative Development: A Manager's Guide* . Addison-Wesley. 376

Laviola, J. (1999). *Whole-Hand and Speech Input in Virtual Environments* . Providence, RI: Brown University Press. 302

LaViola, J. (2000). A Discussion of Cybersickness in Virtual Environments. *ACM SIGCHI Bulletin* . DOI: [10.1145/333329.333344](https://doi.org/10.1145/333329.333344). 166 , 174 , 175

LaViola, J., and Zeleznik, R. (1999). Flex and Pinch: A Case Study of Whole-Hand Input Design for Virtual Environment Interaction. In *International Conference on Computer Graphics and Imaging '99* (pp. 221–225). 316

Lawson, B. D. (2014). Motion Sickness Symptomatology and Origins. In K. Hale and K. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 531–600). Boca Raton, FL: CRC Press. 163 , 197 , 200 , 207

Lehar, S. (2007). The Constructive Aspect of Visual Perception: A Gestalt Field Theory Principle of Visual Reification Suggests a Phase Conjugate Mirror Principle of Perceptual Computation. Retrieved from <http://cns-alumni.bu.edu/~slehar/ConstructiveAspect/ConstructiveAspect.pdf> 64 , 231

Leigh, J., Johnson, A., and Defanti, T. A. (1997). CAVERN: A Distributed Architecture for Supporting Scalable Persistence and Interoperability in Collaborative Virtual Environments. *Virtual Reality: Research, Development and Applications*, 2 (2), 217–237. 418

Levoy, M., and Hanrahan, P. (1996). Light Field Rendering. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '96* , 31–42. DOI: [10.1145/237170.237199](https://doi.org/10.1145/237170.237199). 248

Li, L., Adelstein, B. D., and Ellis, S. R. (2006). Perception of Image Motion during Head Movement. In *Proceedings of the ACM Symposium on Applied Perception in Graphics and Visualization* (pp. 45–50). Boston, MA: ACM Press. 132

Liang, J., and Green, M. (1994). JDCAD: A Highly Interactive 3D Modeling System. *Computers and Graphics* , 18(4), 499–506. 331 , 346

Lin, J. J. W., Abi-Rached, H., and Lahav, M. (2004). Virtual Guiding Avatar: An Effective Procedure to Reduce Simulator Sickness in Virtual Environments. In *Conference on Human Factors in Computing Systems, Proceedings* (pp. 719–726). Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-4544250250&partnerID=40&md5=e8702da495f6ad516a11210168b1ef0210>

Lindeman, R., and Beckhaus, S. (2009). Crafting Memorable VR Experiences Using Experiential Fidelity. *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology* , 187–190. DOI: [10.1145/1643928.1643970](https://doi.org/10.1145/1643928.1643970). 227

Lindeman, R. W. (1999). *Bimanual interaction, Passive-Haptic Feedback, 3D Widget Representation, and Simulated Surface Constraints for Interaction in Immersive Virtual Environments* . George Washington University. 304

Lindeman, R. W., Sibert, J. L., and Hahn, J. K. (1999). Hand-Held Windows: Towards Effective 2D Interaction in Immersive Virtual Environments. In *IEEE Virtual Reality* (pp. 205–212). DOI: [10.1109/VR.1999.756952](https://doi.org/10.1109/VR.1999.756952). 37 , 349

Link. (2015). Link Foundation Advanced Simulation and Training Fellowships. Retrieved April 15, 2015, from <http://linksim.org/> 19

Lisberger, S. G., and Movshon, J. A. A. (1999). Visual Motion Analysis for Pursuit Eye Movements in Area MT of Macaque Monkeys. *The Journal of Neuroscience*, 19 (6), 2224–2246. 129

Loomis, J. M., and Knapp, J. M. (2003). Visual Perception of Egocentric Distance in Real and Virtual Environments. In L. J. Hettinger and M. W. Haas (Eds.), *Virtual and Adaptive Environments* . Mahwah, NJ: Lawrence Erlbaum Associates. 124

Loose, R., and Probst, T. (2001). Velocity Not Acceleration of Self-Motion Mediates Vestibular-Visual Interaction. *Perception*, 30 (4), 511–518. 132

Lucas, G. M., Gratch, J., King, A., and Morency, L. P. (2014). It's Only a Computer: Virtual Humans Increase Willingness to Disclose. *Computers in Human Behavior*, 37 , 94–100. DOI: [10.1016/j.chb.2014.04.043](https://doi.org/10.1016/j.chb.2014.04.043). 477

Lynch, K. (1960). *The Image of the City* . MIT Press. 243

Mack, A. (1986). Perceptual Aspects of Motion in the Frontal Plane. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. 112 , 130

- Mack, A., and Herman, E. (1972). A New Illusion: The Underestimation of Distance during Pursuit Eye Movements. *Perception and Psychophysics*, 12 , 471–473. [141](#)
- Maltz, M. (1960). *Psycho-Cybernetics* . Prentice-Hall. [48](#)
- Mania, K., Adelstein, B. D., Ellis, S. R., and Hill, M. I. (2004). Perceptual Sensitivity to Head Tracking in Virtual Environments with Varying Degrees of Scene Complexity. In *Proceedings of the ACM Symposium on Applied Perception in Graphics and Visualization* (pp. 39–47). [184](#)
- Mapes, D., and Moshell, J. (1995). A Two-Handed Interface for Object Manipulation in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 4 (4), 403–416. [341](#)
- Mark, W. R., McMillan, L., and Bishop, G. (1997). Post-rendering 3D Warping. In *Proceedings of Interactive 3D Graphics* (pp. 7–16). Providence, RI. [212](#)
- Marois, R., and Ivanoff, J. (2005). Capacity Limits of Information Processing in the Brain. *Trends in Cognitive Sciences* . DOI: [10.1016/j.tics.2005.04.010](#).
- Martin, J. (1998). TYCOON: Theoretical Framework and Software Tools for Multimodal Interfaces. In J. Lee (Ed.), *Intelligence and Multimodality in Multimedia Interfaces* . AAAI Press. [302](#)
- Martini (1998). *Fundamentals of Anatomy and Physiology* . Upper Saddle River: Prentice Hall. [106](#)
- Mason, W. 2015, Open Source vs. Open Standard, an Important Distinction. Retrieved July 2, 2015, from <http://uploadvr.com/osvr-may-be-open-source-but-it-is-not-open-standard-and-that-is-an-important-distinction-says-neil-schneider-of-the-it/> [482](#)
- May, J. G., and Badcock, D. R. (2002). Vision and Virtual Environments. In K. M. Stanney (Ed.), *Handbook of Virtual Environments* (pp. 29–63). Mahwah, NJ: Lawrence Erlbaum Associates. [96](#)
- McCauley, M. E., and Sharkey, T. J. (1992). Cybersickness: Perception of Motion in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 1(3) , 311–318. [207](#)
- McGurk, H., and MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, 264 , 746–748. [108](#)
- McMahan, R. P., and Bowman, D. A. (2007). An Empirical Comparison of Task Sequences for Immersive Virtual Environments. In *IEEE Symposium on 3D User Interfaces* (pp. 25–32). DOI: [10.1109/3DUI.2007.340770](#). [302](#)
- McMahan, R. P., Bowman, D. A., Zielinski, D. J., and Brady, R. B. (2012). Evaluating Display Fidelity and Interaction Fidelity in a Virtual Reality Game. *IEEE Transactions on Visualization and Computer Graphics*, 18 (4), 626–633. DOI: [10.1109/TVCG.2012.43](#). [337](#)
- McMahan, R. P., Kopper, R., and Bowman, D. A. (2014). Principles for Designing Effective 3D Interaction Techniques. In K. M. Stanney and K. S. Hale (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 285–311). Boca Raton, FL: CRC Press. [299](#), [325](#), [345](#), [397](#), [398](#)
- McMahan, R.P., Ragan, E.D., Bowman, D.A., Tang, F., and Lai, C. (2015). FIFA: The Framework for Interaction Fidelity Analysis. Tech Report UTDCS-06-15. The University of Texas at Dallas. [290](#)
- Meehan, M., Brooks, F., Razzaque, S., and Whitton, M. (2003). Effects of Latency on Presence in Stressful Virtual Environments. In *Proceedings of IEEE Virtual Reality* (pp. 141–148). Los Angeles, CA. [184](#)
- Meehan, M., Insko, B., Whitton, M., and Brooks, F. P. (2002). Physiological Measures of Presence in Stressful Virtual Environments. *ACM Transactions on Graphics* . DOI: [10.1145/566654.566630](#). [37](#)
- Melzer, J. E., and Moffitt, K. (2011). *Head-Mounted Displays: Designing for the User* . CreateSpace Publishing. [178](#)
- Merriam-Webster (2015). Merriam-Webster. Retrieved May 8, 2015, from <http://www.merriam-webster.com/> [9](#)
- Milgram, P., and Kishino, F. (1994). Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information and Systems*, E77-D (12), 1321–1329. [29](#), [30](#)
- Miller, G. A., and Isard, S. (1963). Some Perceptual Consequences of Linguistic Rules. *Journal of Verbal Learning and Verbal Behavior* . DOI: [10.1016/S0022-5371\(63\)80087-0](#). [102](#)
- Miné, M. (2003). Towards Virtual Reality for the Masses: 10 Years of Research at Disney's VR Studio. In *Proceedings of the Workshop on Virtual Environments 2003* (pp. 11–17). DOI: [10.1145/769953.769955](#). [180](#)
- Miné, M., and Bishop, G. (1993). *Just-in-Time Pixels* . Technical Report TR93-005, Department of Computer Science, University of North Carolina at Chapel Hill. [191](#)
- Miné, M. R. (1993). *Characterization of End-To-End Delays in Head-Mounted Display Systems* . Technical Report TR93-001, Department of Computer Science, University of North Carolina at Chapel Hill. [187](#)

Miné, M. R., Brooks, F. P., Jr., and Séquin, C. (1997). Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction. In *SIGGRAPH* (pp. 19–26). ACM Press. [113](#), [291](#), [294](#), [328](#), [339](#), [348](#), [351](#)

Mirzaie, H. (2009, March 16). Stereoscopic Vision. Retrieved from <http://www.slideshare.net/hmirzaeee/stereoscopic-vision> [121](#)

Miyaura, M., Narumi, T., Nishimura, K., Tanikawa, T., and Hirose, M. (2011). Olfactory Feedback System to Improve the Concentration Level Based on Biological Information. *2011 IEEE Virtual Reality Conference*, 139–142. DOI: [10.1109/VR.2011.5759452](#). [108](#)

Mlyniec, P. (2013). Motion-Controlled Gaming for Neuroscience Education. Retrieved April 14, 2015, from <http://sbirsouce.com/sbir/awards/146246-motion-controlled-gaming-for-neuroscience-education> [439](#)

Mlyniec, P., Jerald, J., Yoganandan, A., Seagull, J., Toledo, F., and Schultheis, U. (2011). iMedic: A Two-Handed Immersive Medical Environment for Distributed Interactive Consultation. In *Studies in Health Technology and Informatics* (Vol. 163, pp. 372–378). [248](#), [341](#), [353](#)

Mori, M. (1970). The Uncanny Valley. *Energy*, 7 (4), 33–35. DOI: [10.1162/pres.16.4.337](#). [50](#)

Murphy, K. R., and Davidshofer, C. O. (2005). *Psychological Testing: Principles and Applications*. Pearson/Prentice Hall. [430](#)

Nakayama, K., and Tyler, C. W. (1981). Psychological Isolation of Movement Sensitivity by Removal of Familiar Position Cues. *Vision Research*, 21, 427–433. [129](#)

Neely, H. E., Belvin, R. S., Fox, J. R., and Daily, M. J. (2004). Multimodal Interaction Techniques for Situational Awareness and Command of Robotic Combat Entities. In *IEEE Aerospace Conference Proceedings* (Vol. 5, pp. 3297–3305). DOI: [10.1109/AERO.2004.1368136](#). [354](#), [407](#)

Negroponte, N. (1993, December). Virtual Reality: Oxymoron or Pleonasm. *Wired Magazine*. Retrieved from http://archive.wired.com/wired/archive/1.06/negroponte_pr.html [27](#)

Nichols, S., Ramsey, A. D., Cobb, S., Neale, H., D'Cruz, M., and Wilson, J. R. (2000). *Incidence of Virtual Reality Induced Symptoms and Effects (VRISE) in Desktop and Projection Screen Display Systems*. HSE Contract Research Report. [203](#) id=

Norman, D. A. (2013). *The Design of Everyday Things, Expanded and Revised Edition. Human Factors and Ergonomics in Manufacturing*. New York: Basic Books. DOI: [10.1002/hfm.20127](#). [11](#), [76](#), [77](#), [79](#), [227](#), [278](#), [285](#), [286](#)

Oakes, E. H. (2007). *Encyclopedia of World Scientists*. Infobase Publishing. [22](#)

Oculus Best Practices. (2015). Retrieved May 22, 2015, from <https://developer.oculus.com/documentation/> [164](#), [201](#), [441](#)

Olano, M., Cohen, J., Miné, M., and Bishop, G. (1995). Combating Rendering Latency. In *Proceedings of the ACM Symposium on Interactive 3D Graphics* (pp. 19–24). Monterey, CA. [187](#)

Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H. (2007). *Engineering Design: A Systematic Approach*. Springer. [396](#)

Patrick, G. T. W. (1890). The Psychology of Prejudice. *Popular Science Monthly*, 36, 634. [55](#)

Pausch, R., Crea, T., and Conway, M. J. (1992). A Literature Survey for Virtual Environments: Military Flight Simulator Visual Systems and Simulator Sickness. *PRESENCE: Teleoperators and Virtual Environments*, 1 (3), 344–363. [205](#)

Pausch, R., Snoddy, J., Taylor, R., Watson, S., and Haseltine, E. (1996). Disney's Aladdin: First Steps toward Storytelling in Virtual Reality. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (pp. 193–203). ACM Press. DOI: [10.1145/237170.237257](#). [180](#), [200](#), [228](#), [258](#), [313](#)

Peck, T. C., Seinfeld, S., Aglioti, S. M., and Slater, M. (2013). Putting Yourself in the Skin of a Black Avatar Reduces Implicit Racial Bias. *Consciousness and Cognition*, 22 (3), 779–787. DOI: [10.1016/j.concog.2013.04.016](#). [48](#)

Pfeiffer, T., and Memili, C. (2015). GPU-Accelerated Attention Map Generation for Dynamic 3D Scenes. In *IIEE Virtual Reality*. Retrieved from http://gpu-heatmap.multimodalinteraction.org/AttentionVisualizer/GPU_accelerated_Heatmaps.pdf [150](#)

Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., and Miné, M. R. (1997). Image Plane Interaction Techniques in 3D Immersive Environments. In *ACM Symposium on Interactive 3D Graphics* (pp. 39–44). ACM Press. [329](#), [330](#)

Pierce, J. S., Stearns, B. C., and Pausch, R. (1999). Voodoo Dolls?: Seamless Interaction at Multiple Scales in Virtual Environments. In *Symposium on Interactive 3D Graphics* (pp. 141–145). DOI: [10.1145/300523.300540](#). [352](#)

Plato. (380 BC). *The Republic*. Athens, Greece: The Academy.

Pokorny, J., and Smith, V. C. (1986). Colorimetry and Color Discrimination. In K. Boff, L. Kaufman, and J. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. [91](#)

- Polonen, M. (2010). *A Head-Mounted Display as a Personal Viewing Device: Dimensions of Subjective Experiences*. University of Helsinki, Finland. Retrieved from <http://helda.helsinki.fi/bitstream/handle/10138/19799/ahheadmou.pdf?sequence=1> 205
- Posner, M. I., Nissen, M. J., and Klein, R. M. (1976). Visual Dominance: An Information-Processing Account of Its Origins and Significance. *Psychological Review*, 83 , 157–171. DOI: [10.1037/0033-295X.83.2.157](https://doi.org/10.1037/0033-295X.83.2.157) . 109
- Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. (1996). The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. In *ACM Symposium on User Interface Software and Technology* (pp. 79–80). ACM Press. DOI: [10.1145/237091.237102](https://doi.org/10.1145/237091.237102) . 327
- Poupyrev, I., Weghorst, S., and Fels, S. (2000). Non-isomorphic 3D Rotational Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'00* (pp. 540–547). DOI: [10.1145/332040.332497](https://doi.org/10.1145/332040.332497) . 283 , 333
- Pratt, A. B. (1916). Weapon. US Patent 1183492 [18](#)
- Proffitt, D. R. (2008). An Action-Specific Approach to Spatial Perception. *Embodiment, Ego-Space, and Action* , 177–200. Retrieved from http://people.virginia.edu/~drp/reprints/Proffitt_2008_CMU_Chapter.pdf 123
- Proffitt, D. R., Bhalla, M., Gossweiler, R., and Midgett, J. (1995). Perceiving Geographical Slant. *Psychonomic Bulletin and Review* . DOI: [10.3758/BF03210980](https://doi.org/10.3758/BF03210980) . 123
- Prothero, J. D., and Parker, D. E. (2003). A Unified Approach to Presence and Motion Sickness. In L. J. Hettinger and M. Haas (Eds.), *Virtual and Adaptive Environments* (pp. 47–66). Boca Raton, FL: CRC Press. DOI: [10.1201/9781410608888.ch3](https://doi.org/10.1201/9781410608888.ch3) . 167, 196, 205
- Provancher, W. (2014). Creating Greater VR Immersion by Emulating Force Feedback with Ungrounded Tactile Feedback. *IQT Quarterly* , 18–21. [37](#)
- Ramachandran, V. S., and Hirstein, W. (1998). The Perception of Phantom Limbs. The D. O. Hebb lecture. *Brain*, 121 (9), 1603–1630. DOI: [10.1093/brain/121.9.1603](https://doi.org/10.1093/brain/121.9.1603) . 105
- Rash, C. E. (2004). Awareness of Causes and Symptoms of Flicker Vertigo Can Limit Ill Effects. *Flight Safety Foundation-Human Factors and Aviation Medicine*, 51 (2), 1–6. [174](#)
- Rasmusson, J. (2010). The Agile Samurai—How Agile Masters Deliver Great Software. Pragmatic Bookshelf. <http://pragprog.com/titles/jtrap/the-agile-samurai> . 386, 387, 392, 393, 436
- Razzaque, S. (2005). *Redirected Walking* . Department of Computer Science, University of North Carolina at Chapel Hill. [74](#), 97, 98, 107, 109, 169, 170
- Razzaque, S., Kohn, Z., and Whitton, M. (2001). Redirected Walking. In *Eurographics* (pp. 289–294). Manchester, UK. [337](#)
- Reason, J. T., and Brand, J. J. (1975). *Motion Sickness* . London: Academic Press. [165](#), 202
- Reeves, B., and Voelker, D. (1993). *Effects of Audio-Video Asynchrony on Viewer's Memory, Evaluation of Content and Detection Ability* . Research report, Stanford University. [108](#)
- Regan, D. M., Kaufman, L., and Lincoln, J. (1986). Motion in Depth and Visual Acceleration. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. [130](#)
- Regan, M., Miller, G., Rubin, S., and Kogelhik, C. (1999). A Real-Time Low-Latency Hardware Light-Field Renderer. In *Proceedings of ACM SIGGRAPH 99* (pp. 287–290). ACM Press. [212](#)
- Regan, M., and Pose, R. (1994). Priority Rendering with a Virtual Reality Address Recalculation Pipeline. In *Proceedings of ACM SIGGRAPH 94* (pp. 155–162). ACM Press. [212](#)
- Reichelt, S., Haussler, R., Fütterer, G., and Leister, N. (2010). Depth Cues in Human Visual Perception and Their Realization in 3D Displays. In *Three Dimensional Imaging, Visualization, and Display 2010* (p. 76900B–76900B–12). DOI: [10.1117/12.850094](https://doi.org/10.1117/12.850094) . 94, 95, 122
- Renner, R. S., Velichkovsky, B. M., and Helmert, J. R. (2013). The Perception of Egocentric Distances in Virtual Environments - A Review. *ACM Computing Surveys*, 46 (2), 23:1–40. [123](#), 124
- Riccio, G. E., and Stoffregen, T. A. (1991). An Ecological Theory of Motion Sickness and Postural Instability. *Ecological Psychology* . DOI: [10.1207/s15326969eco0303_2](https://doi.org/10.1207/s15326969eco0303_2) . 166
- Riecke, B. E., Schulte-Pelkum, J., Avraamides, M. N., von der Heyde, M., and Bülthoff, H. H. (2005). Scene Consistency and Spatial Presence Increase the Sensation of Self-Motion in Virtual Reality. In *Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization, APGV '05* (pp. 111–118). DOI: [10.1145/1080402.1080422](https://doi.org/10.1145/1080402.1080422) . 137
- Ries, E. (2011). *The Lean Startup* . Crown Business. [375](#)

Roberts, D. J., and Sharkey, P. M. (1997). Maximising Concurrency and Scalability in a Consistent, Causal, Distributed Virtual Reality System, Whilst Minimising the Effect of Network Delays. In *Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises* (pp. 161–166). DOI: [10.1109/ENABL.1997.630808](https://doi.org/10.1109/ENABL.1997.630808). 421

Robinson, D. A. (1981). Control of Eye Movements. In V. B. Brooks (Ed.), *The Handbook of Physiology* (Vol. II, Part 2, pp. 1275–1320). Baltimore: Williams and Wilkens. 96

Rochester, N., and Seibel, R. (1962). Communication Device. US Patent 3022878. 24

Rolnick, A., and Lubow, R. E. (1991). Why Is the Driver Rarely Motion Sick? The Role of Controllability in Motion Sickness. *Ergonomics*, 34, 867–879. DOI: [10.1080/00140139108964831](https://doi.org/10.1080/00140139108964831). 171

Salisbury, K., Conti, F., and Barbagi, F. (2004). Haptic Rendering: Introductory Concepts. *IEEE Computer Graphics and Applications*, 24 (2), 24–32. DOI: [10.1109/MCG.2004.1274058](https://doi.org/10.1109/MCG.2004.1274058). 413

Samuel, A. G. (1981). Phonemic Restoration: Insights from a New Methodology. *Journal of Experimental Psychology. General*, 110, 474–494. DOI: [10.1037/0096-3445.110.4.474](https://doi.org/10.1037/0096-3445.110.4.474). 102

Sareen, A., and Singh, V. (2014). Noise Induced Hearing Loss: A Review. *Otolaryngology Online Journal*, 4 (2), 17–25. Retrieved from http://jorl.net/index.php/jorl/article/viewFile/noise_hearingloss/pdf_55179

Schneider, N. (2015). The Secret Sauce of Standardization, Gamasutra. Retrieved July 2, 2015, from http://www.gamasutra.com/blogs/NeilSchneider/20150121/234683/The_Secret_Sauces_of_Standardization.php 481

Schowengerdt, B. T., Seibel, E. J., Kelly, J. P., Silverman, N. L., and Furness III, T. A. (2003, May). Binocular retinal scanning laser display with integrated focus cues for ocular accommodation. In *Electronic Imaging 2003* (pp. 1–9). International Society for Optics and Photonics. 479

Schultheis, U., Jerald, J., Toledo, F., Yoganandan, A., and Mlyniec, P. (2012). Comparison of a Two-Handed Interface to a Wand Interface and a Mouse Interface for Fundamental 3D Tasks. In *IEEE Symposium on 3D User Interfaces 2012, 3DUI 2012, Proceedings* (pp. 117–124). 287, 341, 343

Sedig, K., and Parsons, P. (2013). Interaction Design for Complex Cognitive Activities with Visual Representations: A Pattern-Based Approach. *AIS Transactions on Human-Computer Interaction*, 5 (2), 84–133. Retrieved from http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1057&context=thci_323

Sekuler, R., Sekuler, A. B., and Lau, R. (1997). Sound Alters Visual Motion Perception. *Nature* . DOI: [doi:10.1038/385308a0](https://doi.org/10.1038/385308a0). 108

Shakespeare, W. (1598). *Henry IV* . Part 2, Act 3, scene 1, 26–31. 22

Shaw, C., and Green, M. (1994). Two-Handed Polygonal Surface Design. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (pp. 205–212). 346

Sherman, W. R., and Craig, A. B. (2003). *Understanding Virtual Reality* . Morgan Kaufmann Publishers. 9

Sherrington, C. S. (1920). *Integrative Action of the Nervous System* . New Haven, CT: Yale University Press. 108

Siegel, A., and Sapru, H. N. (2014). *Essential Neuroscience* (3rd ed.). Lippincott Williams and Wilkins. 87, 169

Simons, D. J., and Chabris, C. F. (1999). Gorillas in Our Midst: Sustained Inattentional Blindness for Dynamic Events. *Perception*, 28 (9), 1059–1074. DOI: [10.1068/p2952](https://doi.org/10.1080/p2952). 147

Slater, M. (2009). Place Illusion and Plausibility Can Lead to Realistic Behaviour in Immersive Virtual Environments. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 364 (1535), 3549–3557. DOI: [10.1098/rstb.2009.0138](https://doi.org/10.1098/rstb.2009.0138). 47

Slater, M., Antley, A., Davison, A., Swapp, D., Guger, C., Barker, C., Pistrang, N., and Sanchez-Vives, M. V. (2006a). A Virtual Reprise of the Stanley Milgram Obedience Experiments. *PLoS ONE*, 1 (1). DOI: [10.1371/journal.pone.0000039](https://doi.org/10.1371/journal.pone.0000039). 49

Slater, M., Pertaub, D. P., Barker, C., and Clark, D. M. (2006b). An Experimental Study on Fear of Public Speaking Using a Virtual Environment. *Cyberpsychology and Behavior*, 9 (5), 627–633. Retrieved from <Go to ISI>/WOS:000241415100018 49

Slater, M., and Steed, A. J. (2000). A Virtual Presence Counter. *Presence: Teleoperators and Virtual Environments*, 9 (5), 413–434. 47

Slater, M., and Wilbur, S. (1997). A Framework for Immersive Virtual Environments (FIVE): Speculation on the Role of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 6 (6). 45

Smith, R. B. (1987). Experiences with the Alternate Reality Kit: An Example of Tension between Literalism and Magic. *IEEE Computer Graphics and Applications*, 7 (8), 42–50. DOI: [10.1109/MCG.1987.277078](https://doi.org/10.1109/MCG.1987.277078). 290

So, R. H. Y., and Griffin, M. J. (1995). Effects of Lags on Human Operator Transfer Functions with Head-Coupled Systems. *Aviation, Space and Environmental Medicine*, 66 (6), 550–556. 184

Spector, R. H. (1990). Visual Fields. In HK Walker, W. Hall, and J. Hurst (Eds.), *Clinical Methods: The History, Physical, and Laboratory Examinations* (3rd ed.). Boston, MA: Butterworths. [90](#)

Stanney, K. M., Kennedy, R. S., and Drexler, J. M. (1997). Cybersickness Is Not Simulator Sickness. In *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting* (pp. 1138–1142).

Stanney, K. M., Kennedy, R. S., and Hale, K. S. (2014). Virtual Environment Usage Protocols. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 797–809). Boca Raton, FL: CRC Press. [202](#), [207](#)

Stefanucci, J. K., Proffitt, D. R., Clore, G. L., and Parekh, N. (2008). Skating Down a Steeper Slope: Fear Influences the Perception of Geographical Slant. *Perception*, 37 (2), 321–323. DOI: [10.1068/p5796](#). [123](#)

Steinicke, F., Visell, Y., Campos, J., and Lécuyer, A. (2013). *Human Walking in Virtual Environments*. Springer. DOI: [10.1007/978-1-4419-8432-6](#). [336](#)

Stoakley, R., Conway, M. J., and Pausch, R. (1995). Virtual Reality on a WIM: Interactive Worlds in Miniature. In *ACM Conference on Human Factors in Computing Systems* (Vol. 95, pp. 265–272). DOI: [10.1145/223904.223938](#). [349](#), [352](#), [353](#)

Stoffregen, T. A., Draper, M. H., Kennedy, R. S., and Compton, D. (2002). Vestibular Adaptation and Aftereffects. In K. M. Stanney (Ed.), *Handbook of Virtual Environments* (pp. 773–790). Mahwah, NJ: Lawrence Erlbaum Associates. [96](#)

Stratton, G. M. (1897). Vision without Inversion of the Retinal Image. *Psychological Review*. DOI: [10.1037/h0075482](#). [144](#)

Stroud, J. M. (1986). The Fine Structure of Psychological Time. In H. Quastler (Ed.), *Information Theory in Psychology: Problems and Methods* (pp. 174–207). Glencoe, IL: Free Press. [124](#)

Sutherland, I. E. (1965). The ultimate display. In *The Congress of the International Federation of Information Processing (IFIP)* (pp. 506–508). DOI: [10.1109/MC.2005.274](#). [9](#), [23](#), [30](#)

Sutherland, I. E. (1968). A Head-Mounted Three Dimensional Display. In *Proceedings of the 1968 Fall Joint Computer Conference AFIPS* (Vol. 33, part 1, pp. 757–764). Washington, D.C.: Thompson Books. [25](#)

Taylor, R. M. (2010). Effective Virtual Environments for Microscopy. Retrieved April 23, 2015, from <http://cismm.cs.unc.edu/core-projects/visualization-and-analysis/advanced-graphics-and-interaction/eve-for-microscopy/> [250](#)

Taylor, R. M. (2015). OSVR: Sensics Latency-Testing Hardware. Retrieved from https://github.com/sensics/Latency-Test/blob/master/Latency_Hardware/Latency_Tester_Hardware.pdf [194](#)

Taylor, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., and Helser, A. T. (2001a). VRPN: A Device-Independent, Network-Transparent VR Peripheral System. In *Proceedings of VRST '01* (pp. 55–61). Banff, Alberta, Canada. [481](#)

Taylor, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., and Helser, A. T. (2001b). VRPN: A Device-Independent, Network-Transparent VR Peripheral System. Retrieved May 15, 2015, from http://www.cs.unc.edu/Research/vrpn/VRST_2001_conference/taylorr_VRPN_presentation_files/v3_document.htm [481](#)

Taylor, R. M., Jerald, J., VanderKnyff, C., Wendt, J., Borland, D., Marshburn, D., Sherman, W. R., and Whitton, M. C. (2010). Lessons about Virtual Environment Software Systems from 20 Years of VE Building. *Presence: Teleoperators and Virtual Environments*, 19 (2), 162–178. [413](#)

Treisman, M. (1977). Motion Sickness: An Evolutionary Hypothesis. *Science*, 197, 493–495. DOI: [10.1126/science.301659](#). [165](#)

Turnbull, C. M. (1961). Some Observations Regarding the Experiences and Behavior of the BaMbuti Pygmies. *The American Journal of Psychology*, 74, 304–308. [140](#)

Ulrich, R. (1987). Threshold Models of Temporal-Order Judgments Evaluated by a Ternary Response Task. *Perception and Psychophysics*, 42, 224–239. [124](#)

Usoh, M., Arthur, K., Whitton, M. C., Bastos, R. R., Steed, A., Slater, M., and Brooks, F. P., Jr. (1999). Walking > Walking-in-Place > Flying in Virtual Environments. In *SIGGRAPH '99 Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 359–364). DOI: [10.1145/311535.311589](#). [336](#)

Valve. (2015). Source Multiplayer Networking. Retrieved May 31, 2015, from https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking [418](#), [420](#)

Van Beers, R. J., Wolpert, D. M., and Haggard, P. (2002). When Feeling Is More Important Than Seeing in Sensorimotor Adaptation. *Current Biology*, 12 (10), 834–837. DOI: [10.1016/S0960-9822\(02\)00836-9](#). [304](#)

van der Veer, G. C., and Melguizo, M. del C. P. (2002). Mental Models. In A. Sears and J. A. Jacko (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications* (3rd ed.). Boca Raton, FL: CRC Press. [283](#)

- Virre, E., Price, B. J., and Chase, B. (2014). Direct Effects of Virtual Environments on Users. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 521–529). Boca Raton, FL: Boca Raton, FL: CRC Press. [174](#)
- Vorlander, M., and Shinn-Cunningham, B. (2014). Virtual Auditory Displays. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 87–114). Boca Raton, FL: CRC Press. [100](#)
- Wake, B. (2003). INVEST in Good Stories, and Smart Tasks. Retrieved April 18, 2015, from <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/> [392](#)
- Wallach, H. (1987). Perceiving a Stable Environment When One Moves. *Annual Review of Psychology*, 38 , 1–27. [96](#), [143](#), [144](#)
- Wallach, H., and Kravitz, J. H. (1965a). Rapid Adaptation in the Constancy of Visual Direction with Active and Passive Rotation. *Psychonomic Science*, 3 (4), 165–166. [144](#)
- Wallach, H., and Kravitz, J. H. (1965b). The Measurement of the Constancy of Visual Direction and of Its Adaptation. *Psychonomic Science*, 2 , 217–218. [142](#), [175](#)
- Wang, J., and Lindeman, R. (2014). Coordinated 3D Interaction in Tablet- and HMD-Based Hybrid Virtual Environments. In *ACM Symposium on Spatial User Interaction* . [349](#)
- Warren, R. M. (1970). Perceptual Restoration of Missing Speech Sounds. *Science*, 167 , 392–393. DOI: [10.1126/science.167.3917.392](https://doi.org/10.1126/science.167.3917.392) . [102](#)
- Warren, W. H., Kay, B. A., Zosh, W. D., Duchon, A. P., and Sahuc, S. (2001). Optic Flow Is Used to Control Human Walking. *Nature Neuroscience*, 4 (2), 213–216. DOI: [10.1038/84054](https://doi.org/10.1038/84054) . [154](#)
- Webster's New Universal Unabridged Dictionary* . (1989). New York: Barnes and Noble Books. [9](#)
- Weinbaum, S. G. (1935, June). Pygmalion's Spectacles. *Wonder Stories* . [20](#)
- Welch, R. B. (1986). Adaptation of Space Perception. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. [143](#), [144](#), [207](#)
- Welch, R. B., and Mohler, B. J. (2014). Adapting to Virtual Environments. In K. S. Hale and K. M. Stanney (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 627–646). Boca Raton, FL: CRC Press. [143](#)
- Welch, R. B., and Warren, D. H. (1986). Intersensory Interactions. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. 1). New York: Wiley-Interscience. [100](#), [108](#), [112](#)
- Wendt, J. D. (2010). *Real-Walking Models Improve Walking-in-Place Systems* . UNC-Chapel Hill. [337](#)
- Whitton, M. C. (1984). Memory Design for Raster Graphics Displays. *IEEE Computer Graphics and Applications*, 4 (3), 48–65. [190](#)
- Wilkes, C., and Bowman, D. A. (2008). Advantages of Velocity-Based Scaling for Distant 3D Manipulation. In *ACM Symposium on Virtual Reality Software and Technology* (pp. 23–29). DOI: [10.1145/1450579.1450585](https://doi.org/10.1145/1450579.1450585) . [351](#)
- Willemse, P., Colton, M. B., Creem-Regehr, S. H., and Thompson, W. B. (2009). The Effects of Head-Mounted Display Mechanical Properties and Field of View on Distance Judgments in Virtual Environments. *ACM Transactions on Applied Perception* . DOI: [10.1145/1498700.1498702](https://doi.org/10.1145/1498700.1498702) . [177](#)
- Williams, K. (2014). A Wider FOV - A Guide to Virtual Reality Demonstrations. Retrieved April 29, 2015, from <http://www.roadtovr.com/wider-fov-special-guide-virtual-reality-demonstrations/> [180](#)
- Williams, K., and Mascioni, M. (2014). *The Out-of-Home Immersive Entertainment Frontier: Expanding Interactive Boundaries in Leisure Facilities* . Gower Publishing Limited. Retrieved from http://www.amazon.com/Out-Home-Immersive-Entertainment-Frontier/dp/1472426959/ref=sr_1_1?ie=UTF8&qid=1430409256&sr=8-1&keywords=kevin+williams+immersive+256
- Wingrave, C. A., and LaViola, J. (2010). Reflecting on the Design and Implementation Issues of Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 19 (2), 179–195. [3](#), [374](#), [391](#)
- Wingrave, C. A., Tintner, R., Walker, B. N., Bowman, D. A., and Hodges, L. F. (2005). Exploring Individual Differences in Ray-Based Selection: strategies and traits. *IEEE Proceedings. VR 2005. Virtual Reality*, 2005 . DOI: [10.1109/VR.2005.1492770](https://doi.org/10.1109/VR.2005.1492770) . [324](#), [391](#)
- Wolfe, J. (2006). *Sensation & Perception*. Sunderland, Mass.: Sinauer Associates. [232](#)
- Wood, R. W. (1895). The “Haunted Swing” Illusion. *Psychological Review* . DOI: [10.1037/h0073333](https://doi.org/10.1037/h0073333) . [16](#)
- Yoganandan, A., Jerald, J., and Mlyniec, P. (2014). Bimanual Selection and Interaction with Volumetric Regions of Interest. In *IEEE Virtual Reality Workshop on Immersive Volumetric Interaction* . [233](#), [331](#)
- Yost, W. A. (2006). *Fundamentals of Hearing: An Introduction* (5th ed.). Academic Press. [100](#)

Young, S. D., Adelstein, B. D., and Ellis, S. R. (2007). Demand Characteristics in Assessing Motion Sickness in a Virtual Environment: Or Does Taking a Motion Sickness Questionnaire Make You Sick? In *IEEE Transactions on Visualization and Computer Graphics* (Vol. 13, pp. 422–428). [196](#), [202](#), [433](#)

Zaffron, S., and Logan, D. (2009). *The Three Laws of Performance: Rewriting the Future of Your Organization and Your Life*. San Francisco, CA: Jossey-Bass. [475](#)

Zhai, S., Milgram, P., and Buxton, W. (1996). The Influence of Muscle Groups on Performance of Multiple Degree-of-Freedom Input. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Common Ground - CHI '96*, 308–315. DOI: [10.1145/238386.238534](#). [333](#)

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. (1987). A Hand Gesture Interface Device. In *ACM SIGCHI* (pp. 189–192). DOI: [doi:10.1145/30851.275628](#). [26](#)

Zimmons, P., and Panter, A. (2003). The Influence of Rendering Quality on Presence and Task Performance in a Virtual Environment. In *IEEE Virtual Reality* (pp. 293–294). DOI: [10.1109/VR.2003.1191170](#). [51](#)

Zone, R. (2007). Stereoscopic Cinema and the Origins of 3-D Film, 1838-1952. Retrieved from <http://books.google.ch/books?hl=en&lr=&id=C1dgJ3-y1ZsC&oi=fnd&pg=PP1&dq=origins+cinema+psychological+laboratory&ots=IUS0TPJ1Vn&sig=SiW0qoxpCEUi0EySZDoZKNH6c> [15](#), [16](#)

Zuckerberg, M. (2014). Announcement to Acquire Oculus VR. Retrieved April 8, 2015, from <https://www.facebook.com/zuck/posts/10101319050523971474>

Index

The index that appeared in the print version of this title was intentionally removed from the eBook. Please use the search function on your eReading device for terms of interest. For your reference, the terms that appear in the print index are listed below

2D desktop integration

3D Multi-Touch Pattern

mental model

posture and approach

spindle

usage

vection

viewbox

3D Tool Pattern

hand-held tools

jigs

object-attached tools

Above-the-head widgets and panels

Accommodation-vergence conflict. *See under* sensory conflicts

Action (perceptual). *See also* locus of control, navigation; nerve impulses; neurons, mirror neuron systems

dorsal pathway

intended

mental model

performance

Action space. *See under* space perception

Adaptation. *See also* aftereffects

dual

negative training effects

optimizing

perceptual

factors

position-constancy

temporal

postural stability

rate of

readaptation

active readaptation

natural decay

sensory

dark

desensitization

motion

Adverse health effects. *See also* adaptation; comfort; fatigue; hygiene, injuries; sickness

Affordances

geometry

tools

After action reviews

Aftereffects

afterimages

negative

positive

motion

negative

negative training effects

VR

Agents. *See* avatars and characters, computer-controlled (agents)

Aliasing. *See* under rendering, visual artifacts

Analysis. *See* under data

Analysis paralysis

Apparent motion. *See also* judder; strobing

Art. *See also* content creation; sketches

abstract

aesthetics

assets

color

conceptual integrity

trompe-l'œeil

Assessment and feasibility

Assumptions (project)

changes

design specification

experiment

risk

violations of data

Attention. *See also* search; conscious perception

attentional capture

attentional gaze

auditory

cocktail party effect

characters

cycle of interaction

eye tracking

feedback

filtering

deletion

fixation

flow

inattentional blindness

change blindness

change blindness blindness

change deafness

choice blindness

continuity errors

video overlap phenomenon

inhibition of return

involuntary shift of

landmarks

maps

orienting

covert orienting

overt orienting

perceptual capacity

perceptual load

reticular activating system

salience

saliency maps

scene schemas

shadowing

sound

task-irrelevant stimuli

time

vigilance

visual scanning

Attitude

failure

feedback

team

Audio. *See* sound

Augmented reality (AR)

convergence with VR

hand-held

optical-see-through

video-see-through

Augmented virtuality

Auralization. *See also* sound

Automated Pattern

passive vehicles

route planning

target-based travel

teleportation

Automatic mode switching

Avatars and characters

animation

caricature/cartoon

computer-controlled (agents)

directing attention

motion

response

eyes

motion

tokens

Uncanny Valley

user-controlled (avatars)

examples

exocentric view

motion

scale

self-embodiment

social presence

Background. *See under* scene; customers and clients

Balance. *See* postural stability/instability; vestibular system

Behavior. *See also* avatars and characters

adaptation

affecting

body language

circadian rhythms

communication

cycle of interaction

demand characteristics

goal driven

measures

neuro-linguistic programming (NLP)

internal state

memories

programs

primal

processes

iterative

realism

sickness

subconscious

travel

Beliefs

Bimanual interaction

classification

symmetric/asymmetric

synchronous/asynchronous

handedness

dominant hand

non-dominant hand

techniques

3D multi-touch

framing hands

panels

physical props

selection box

two-handed flying

two-handed pointing

viewbox

voodoo dolls

Binaural cues

Binding (perceptual)

Binocular-occlusion conflict. *See* under sensory conflicts

Biological motion. *See also* avatars and characters; motion perception

Biomechanical symmetry

Blind spot

Block diagrams

Bottom-up processing

Boundary completion. *See* under illusions

Breadcrumbs

Break-in-presence. *See* under presence Brightness

Brooks' Law

Buttons. *See* under input device characteristics

Call of Duty Syndrome

CAVEs. *See* under displays, world-fixed

Center of action zones

Center of mass

Central vision. *See* under eye eccentricity

Change blindness. *See* under attention

Channels (navigation)

Characters. *See* avatars and characters

Classes (software)

diagrams

objects

Clients. *See* customers and clients; networked environments

Close-ended questions

Clutching

Color

afterimages

constancy

content creation

cube

dark adaptation

emotions

eye eccentricity

highlighting

salience

subconscious

todo

Color cubes

Comfort. *See also* fatigue; sensory conflicts; sickness; viewpoint, motion

configuration options

depth cues

reticle projection

hand pose

manipulating the world as an object

non-dominant hand

shooting from the hip

tracking requirements

headset fit

interviews social

stereoscopic 360° capture

Uncanny Valley

Communication. *See also* speech

brain-to-brain

direct

structural

visceral. *See also* empathy

face-to-face

indirect

interaction

language

body. *See also* gestures

emotion

generative (aka future-based)

human-computer translation

internal

perception

phonemes and morphemes

project

sign

spoken / verbal

timing

written

project

team

symbolic

team

Compasses

Compliance

spatial

directional

nulling

position

temporal

Compound patterns. *See also* Multimodal Pattern; Pointing Hand Pattern; World-in-Miniature Pattern

Cone-casting flashlight selection

Cone of Experience

Conscious perception

apprehension

changing beliefs

chunks

cycle of interaction

delayed

figure/ground

memories

subjective present

Constancies. *See* perceptual constancies

Constraints (interaction)

3D multi-touch

degrees of freedom (DOFs)

physics

real world

simulated

tools

jigs

Constraints (project)

cultural/social

feedback

indirect

intentional artificial

misperceived

obsolete

puzzle

real

resource

tools

Constructivist approaches. *See also* after action reviews; demos; expert evaluations; focus groups; interviews; questionnaires; retrospectives

Construct validity

Content creation. *See also* art; scene; real world capture; reusing content; wayfinding aids, environmental

basic

color

conceptual integrity

core experience

gamification

high-level concepts

iteration

lighting

landmarks

moving stimuli

multiple disciplines

real world lessons

skeuomorphism

transitioning to VR

wired vs wireless

Continuity errors

Continuous discovery

Contracts

assessment vs. implementation

estimating time and costs

milestones

minimizing risk

negotiating

requirements

support and updates

Control/display (C/D) ratio

Controllers. *See* input device classes; input device characteristics

Critical incidents

Crosshairs. *See* reticles

Culture

constraints

effect on perception

learning and failure

mappings

project considerations

skeuomorphism

VR community

Customers and clients. *See also* contracts; delivery

background and context

lack of standards

misunderstanding and unmet expectations

point of view

questions

requirements

user stories

wants

Cybersickness. *See* under sickness

Cycle of interaction

execution/evaluation

gulfs/bridges of execution/evaluation

task analysis

Cyclopean eye. *See also* reference frames, head

Dark adaptation. *See under* adaptation, sensory

Data

analysis. *See also* statistical conclusion validity

correlation

fishing

practical significance

p-value

statistical power

statistical significance

artifacts

average

mean

median

mode

collecting. *See also* constructivist approaches; scientific method

bias

continuous delivery

demos

programmers

prototypes

scoring system

distribution

histogram

interquartile range

mean deviation

normal

range

spread

standard deviation

variance

interpreting

false conclusion. *See also* validity

measurement types

categorical variables

interval variables

ordinal variables

ratio variables

outliers

physiological

qualitative data

quantitative data

reliability

observed score

stable characteristics

true score

unstable characteristics

review with users

sensitivity

violated assumptions

Dead reckoning (extrapolation)

Define-Make-Learn. *See also* Define Stage; Make Stage; Learn Stage

Define Stage. *See also* questions; assessment and feasibility; objectives; estimates; risk; assumptions; constraints; personas; user stories; storyboards; scope; requirements

Degrees of freedom (DOFs). *See under* constraints (interaction); input device characteristics

Delay. *See* latency

Delivery

continuous delivery

onsite installation

prioritization

Demos

attitude

vs. data collection

failure

in-house

programmers

scheduling

travel

Depth perception. *See also* illusions, depth

binocular disparity. *See also* sensory conflicts, binocular-occlusion conflict

horopter

Panum's fusional area

stereoblindness

stereopsis

compressed

consistent cues

presence

figure/ground

future effect

fear

intended action

motion

number of cues

presence

oculomotor cues

accommodation

accommodation-vergence conflict. *See* sensory conflicts, accommodation-vergence conflict

vergence

pictorial, cues height relative to horizon

pictorial cues

aerial perspective

linear perspective

occlusion. *See also* binocular-occlusion conflict

relative/familiar size

shadows/shading

texture gradient

pivot hypothesis

presence

relative importance

Design patterns (software)

Design specification. *See also* block diagrams; classes (software); design patterns; sketches; use cases

Direct Hand Manipulation Pattern

go-go technique

non-isomorphic rotations

Direct/indirect interaction continuum

direct

indirect. *See also* Indirect Control Patterns

semi-direct

Director

Discoverability (user)

Displays. *See also* heads-up displays (HUDs)

apparent motion. *See also* judder; strobing

binocular (stereoscopic)

biocular

CRT (cathode ray tube)

DLP (digital light processing)

flicker

hand-held

head-mounted displays (HMDs). *See also* comfort; field of view

accommodation-vergence conflict

calibration

depth compression

design trade-offs

fit

historical

hygiene

interpupillary distance (IPD)

latency

non-see-through

non-tracked

optical-see-through

tracking

usage

video-see-through

weight and center of mass

LCD (liquid crystal display)

monocular

OLED (organic light-emitting diode)

persistence

raster

vertical sync

refresh rate

flicker

requirements

stimulus onset asynchrony

refresh time

response time

touch screens

VRDs (virtual retinal displays)

world-fixed

CAVEs

Distal stimuli

iterative perceptual processing

motion parallax

pictorial depth cues

sensation

Distortion (perceptual filter)

Dolls (world-in-miniature)

voodoo

Dominant eye

Dominant hand. *See under* bimanual interaction, handedness

Double buffer. *See under* rendering

Dual analog sticks

Dwell selection

Early wanderings

Egocentric judgments and interaction. *See under* proprioception

Embodiment. *See* presence, self-embodiment

Emotions

body language

colors

communication

empathy

field of view

mirror neuron systems

processes

internal state

relationships

stories

subconscious

users

Empathy. *See also* communication, direct, visceral

Encumbrance

Engagement

Equipment. *See* system

Estimates (project)

planning poker

Events

anticipated

attention

blindness

priming

critical incidents

extraneous

filtering

implementation

infrequent

meetups

memory

multimodal

networked

opportunistic

perception of

binding

continuity

delayed

time. *See also* time perception

sequence and timing

unexpected

visual capture

Evolutionary theory of motion sickness

Exocentric judgments

Expectations. *See under* mental models

Experiential fidelity

Experiments. *See also* outliers; measurement types; validity; data

carryover effects

design

A/B testing

between-subjects

confounding factors

control variables

dependent variable

hypothesis

independent variable

internal validity

pilot study

quasi-experiments

true experiments

variables

within-subjects

participant selection

based on scores

bias

randomize

self-selection

replication

Expert evaluations

comparative (aka summative)

formative usability

task analysis. *See also* task analysis

guidelines-based (aka heuristic)

Experts. *See also* expert evaluations

how to read

subject-matter

users

constraints removal

marking (pie) menus

sickness

Extender grab

External validity

Eye eccentricity

central vision

eye movements

fovea

motion perception

properties

colors

neurophysiology

peripheral vision

head-mounted displays (HMDs)

interaction

light sensitivity

motion perception

motion sickness

presence

properties

sickness

vection

visual acuity

Eye gaze input

attention maps

avatars

dominant eye

input device

interface

feedback

Midas touch problem

reference frame

selection

multimodal

passive over active

pointer/reticle

redirected walking

rendering resolution

selection

specialized tasks and subtle interactions

tracking

avatars

image slip

Eye movements

affects judgments

characters

fixational eye movements

microsaccades

microtremors

ocular drift

gaze-shifting eye movements

inhibition of return

overt orienting

pursuit

saccades

saccadic suppression

vergence. *See also* accommodation-vergence conflict

visual scanning

gaze-stabilizing eye movements

eye rotation gain

nystagmus

optokinetic reflex (OKR)

pendular nystagmus

retinal image slip

vestibulo-ocular reflex (VOR)

passive vs. active

theory of motion sickness

unified model of motion sickness

Face validity

Fade outs

False conclusions

Fatigue. *See also* comfort

accommodation-vergence conflict

flicker

gorilla arm

image-plane selection

reducing

hardware weight and center of mass

physical panels

vigilance

walking

Feedback (brain). *See also* nerve signals

back projections

perceptual

Feedback (interaction). *See also* compliance; haptics; mappings

adaptation

buttons

core experience

eye tracking

force

gestures

head tracking

immediate

Multimodal Pattern

presence

rumble

sickness

sound

speech and gestures

tactile

Feedback (project). *See also* constructivist approaches

attitude

conceptualization

constraints

end users

experts. *See* expert evaluations

external

fast

market demand

naive users

programmers

prototypes. *See* prototypes

stakeholders

team

team culture

Feedforward

Fidelity

experiential

interaction

3D multi-touch

biomechanical symmetry

control symmetry

hands

input veracity

magical

non-realistic

pointing

realistic

selection

steering

walking

representational

Field of regard

Field of view

emotions

error

delay compensation

mismatched

flicker

presence

reducing

rendering

sickness

standards

vection

Film. *See also* content creation; real world capture; stories

capture

360°

light fields

stereoscopic

true 3D

continuity errors

immersive

automated pattern

camera motion. *See also* viewpoint, motion

center of action zones

content

Sensorama

L'Arrivée d'un train en gare de La Ciotat

stereoscopic glasses

strobing and judder

temporal closure

Final production

Finger menus

Fixational eye movements. *See* eye movements

Flavor

Flicker

displays

flicker-fusion frequency threshold

factors

photic seizures

Flight simulators

Link Trainers

motion platforms

sickness

Flow. *See also* motion perception, optic flow

attention

interaction

order

perception of time

Flying

one-handed

two-handed flying

Focus groups

example

Fovea

Frame. *See* under rendering

Frame rate. *See* under rendering

Framing hands selection technique

Game engines

Gamification

Gaze-irected steering

Gaze-shifting eye movements. *See* eye movements

Gaze-stabilizing eye movements. *See* eye movements

Generalizations

Geometry. *See* under affordances; real world capture; reusing content; scene

Gestalt

groupings

principle of closure

principle of common fate

principle of continuity

principle of proximity

principle of similarity

principle of simplicity

temporal closure

interfaces

segregation

feature searches

figure

figure-ground problem

ground

Gestures

accuracy

direct

error

indirect

pinch

posture

push-to-gesture

self-revealing

types of information

Getting started

reading this book

Ghosting

Gloves

examples

gestures

haptics

pinch

Go-go selection and manipulation technique

Gorilla arm. *See* under fatigue

Gravity

Grids

depth perception

jigs

networked environments

structure

warning

Gustatory system. *See* taste

Hand-held controllers. *See* input device classes; touch

Hand-held tools

Hand pointing

Handrails (visual)

Hands. *See also* bimanual interaction; comfort; Direct Hand Manipulation Pattern; Hand Selection Pattern; input device classes; mappings; Pointing Hand Pattern; Pointing Pattern; reference frames; Steering Pattern; touch

fidelity

non-realistic

realistic

semi-realistic

visual-physical conflict. *See also* sensory substitution

penetrations

Hand Selection Pattern

go-go technique

non-realistic hands

realistic hands

semi-realistic hands

Hand/weapon models

Haptics. *See also* motion platforms; touch

active

injury

gloves

passive

proprioceptive forces

self-grounded

tactile

electrotactile

vibrotactile/rumble

world-grounded

Hardware. *See* systems

Head crusher selection technique

Head-mounted displays (HMDs). *See* displays

Head pointing

Head-related transfer function (HRTF)

Heads-up displays (HUDs)

guidelines

porting from video games

Hearing. *See* sound

Highlighting

HOMER selection and manipulation technique

Homunculus

Human joystick navigation

Hygiene

fomite

Illusions. *See also* aftereffects; presence

2D

Hering illusion

Jastrow illusion

blind spot

boundary completion

illusory contours

Kanizsa

depth

Ames room

moon illusion

Ponzo railroad illusion

Pulfrich pendulum effect

trompe-l'œil

motion. *See also*vection

apparent

autokinetic effect

induced

moon-cloud illusion

Ouchi illusion

multimodal

McGurk effect

rubber hand illusion

visual capture (ventriloquism effect)

temporal, filled duration illusion

Image-based rendering

Image-Plane Selection Pattern

framing hands technique

head crusher technique

lifting palm technique

sticky finger technique

Immersion. *See also* presence

Indirect control patterns. *See also* Non-Spatial Pattern; Widgets and Panels Pattern

Indirect interaction. *See* direct/indirect interaction continuum; indirect control patterns

Induced motion. *See* illusions, motion, induced

Injuries

brain

collisions

ear damage

falling

physical trauma

reducing

spotter

warning grids

repetitive strain

Input. *See also* gestures; input device characteristics; input device classes; multimodal interactions; speech

alphanumeric

chord keyboard

neural

non-spatial. *See under* mappings

symbolic

Input device characteristics

absolute

buttons

push-to-talk/gesture

signifiers

stress

degrees of freedom (DoFs)

constraints

control symmetry

hands

integral

tracked hand-held controllers

world-grounded devices

encumbrance

haptics capable

hybrid tracking

integral

isometric

isotonic

relative

reliability

effects on performance

gestures

gloves

requirements

separable

size and shape

Input device classes

Bare-hands

eye tracking

full-body tracking

hand-worn

head tracking

microphones

non-tracked hand-held controllers

tracked hand-held controllers

world-grounded input devices

Installation (onsite)

Interaction. *See also* bimanual interaction; communication; constraints; cycle of interaction; direct/indirect interaction continuum; feedback; fidelity; interaction patterns; interaction techniques; multimodal interactions

Interaction patterns. *See also* selection patterns; manipulation patterns; viewpoint control patterns; indirect control patterns; compound patterns

Interaction techniques. *See also* interaction patterns

creating new

implementing

Interfaces. *See also* compliance; constraints; feedback; interaction design; interaction patterns; signifiers

placement

Internal validity

threats to

attrition (aka mortality)

carryover effects

confounding factors

demand characteristics

experimenter bias

history

instrumentation

maturity

placebo effects

retesting bias

selection bias

statistical regression

Interpupillary distance (IPD)

Interviews

demos

guidelines

personas

scheduling

task analysis

Intuitiveness. *See also* mental models; metaphors

Iterative design. *See also* Define-Make-Learn

philosophy

art and science

continuous discovery

human-centered

project dependence

team

Jigs

Jitter

filtering

network

physics

tracking

Judder. *See also* apparent motion; strobing

factors

display response/persistence

distance

interstimulus interval (blanking time)

stimulus duration

stimulus onset asynchrony (refresh time)

Just-in-time pixels

Kennedy Simulator Sickness Questionnaire (SSQ)

Key players

understandable requirements

Labels and icons. *See* under signifiers

Landmarks

Language. *See* under communication

Latency

compensation

2D warp (aka time warping)

cubic environment map

head-motion prediction

post-rendering

delayed perception

effective

induced scene motion

measuring delay

latency meter

parallel port

timing analysis

negative effects. *See also* sickness, motion

perception of
reducing delay
just-in-time pixels
vertical sync off
sources
application delay
display delay. *See also* displays
rendering delay
synchronization delay
tracking delay
system delay
requirements
timing analysis
thresholds
adaptation
variable latency
Learned helplessness
Learn Stage. *See also* constructivist approaches; data; scientific method; validity
Lifting palm selection technique
Lighting. *See* brightness; content creation, lighting; highlighting; lightness
Lightness
constancy
Likert scales
Lip sync
Locomotion. *See* navigation, travel
Locus of control. *See also* action (perceptual)
active motion
passive motion. *See also* Automated Pattern
leading indicators
vehicles
Make Stage. *See also* delivery; design specification; final production; networked environments; prototypes; simulation; systems; task analysis
Manipulation patterns. *See also* 3D Tool Pattern; Direct Hand Manipulation Pattern; Proxy Pattern
Mappings. *See also* compliance; World-In-Miniature Pattern
abstract maps
hands. *See also* Direct Hand Manipulation Pattern; Hand Selection Pattern; Proxy Pattern
extender grab

go-go technique

non-isomorphic rotations

non-spatial. *See also* Widgets and Panels Pattern; Non-Spatial Control pattern buttons

scaled world grab

tools

viewpoint. *See also* 3D Multi-Touch Pattern; Steering Pattern; Walking Pattern

one-to-one head tracking

world-grounded input example

Maps. *See* wayfinding aids, personal

Markers

Marketing

demos

prototype

testimonials

Marking menus. *See* menus, pie

Markup tools and measurement

Masking (perceptual)

Medication

Meetups

Memories

emotional connection

illusory conjunctions

muscle

wayfinding

Mental models. *See also* mappings; metaphors; neuro-linguistic programming (NLP)

3D Multi-Touch

spindle

audio

phonemic restoration effect

cognitive map

compliance

cycle of interaction

expectations

Midas Touch (eye gaze)

pain

placebo effects

priming
quality
scene motion
violations (network)
generalizations
interviews
intuitiveness. *See also* metaphors
leading indicators
learned helplessness
meta-structure of the world
motion
vection
non-spatial mappings
perceptual constancies
sickness
Call of Duty Syndrome
eye movements
motion
sensory conflicts
skeuomorphism
top-down processing
within world tutorials
Menus. *See also* Widgets and Panels Pattern
finger
pie (marking)
hierarchical
mark ahead
self-revealing gestures
ring
voice menu hierarchies
Metaphors. *See also* mental models
appropriate
environmental wayfinding aids
interactions. *See also* interaction patterns
2D desktop
appropriate modes

consistent

non-spatial mappings

language

power of VR

real-world

Microphones

Midas touch problem

Milestones

Mixed reality. *See also* augmented reality; augmented virtuality

Morphemes. *See* under speech, perception

Motion. *See* apparent motion; avatars and characters; motion; illusions, motion; motion perception; motion platforms; scene, motion; sickness, motion;vection; viewpoint, motion; viewpoint control patterns

Motion aftereffect

Motion blur/smear. *See* under persistence (perceptual)

Motion perception. *See also* nerve impulses;vection; vestibular system; viewpoint, motion

acceleration

sickness

apparent. *See also* judder; strobing

biological

character

coherence

figure-ground

head movement

induced motion

object-relative

optic flow

focus of expansion

gradient flow

pivot hypothesis

subject-relative

unified model

velocity

angular

Motion platforms

active motion platforms

passive motion platforms

sickness

Motion sickness. *See* sickness

Multimodal interactions

complementarity input

concurrent input

equivalent input

put-that-there

redundant input

specialized input

speech

transfer

Multimodal Pattern

automatic mode switching

put-that-there

that-moves-there

Multimodal perception. *See also* sensory substitution;vection

lip sync

McGurk effect

perceptual moments

visual capture (ventriloquism effect)

Muscle memory

Music. *See* sound

Navigation. *See also* locus of control; viewpoint control patterns

exploration

naive search

primed search

sickness

travel

constraints

pre-planned path

wayfinding. *See also* wayfinding aids

cognitive maps

Navigation by leaning technique

Negative training effects

Nerve impulses

afference

efference

efference copy

re-afference

Networked environments

architectures

authoritative servers

client-server

example

hybrid

non-authoritative servers

peer-to-peer

super-peers

causality

causality violations

concurrency

consistency

dead reckoning (extrapolation)

determinism

divergence

expectation violations

jitter

local estimation

packets

perceived continuity

physics

protocols

multicast

TCP (transport control protocol)

UDP (user datagram protocol)

reducing traffic

animations

audio

divergence filtering

dynamic grid/cells

relevance filtering

stress tests

subscribe

responsiveness
simultaneous interactions
synchronization
tokens

Neuro-linguistic programming (NLP)

attitudes
beliefs
decisions
deletion
distortion
filters
generalization
internal representations
memories
meta program
preferred modality
values

Neurons
magno cells
mirror neuron systems
parvo cells
retinal
cones
rods

Nomadic VR

Non-dominant hand. *See* under bimanual interaction, handedness

Non-isomorphic rotations

Non-realistic hands

Non-Spatial Control Pattern

gestures. *See also* gestures

voice menu hierarchies

Object-attached tools

Objective reality. *See also* distal stimuli

Objectives (project)

SMART

Object snapping (selection)

Olfactory system (smell)

Open-ended questions

Open source. *See also* standards

OSVR (open source VR)

VRPN (VR Peripheral Network)

Optic flow. *See under* motion perception

OSVR (open source VR)

Output. *See also* displays; sound; haptics; motion platforms; smell; sound; taste; wind

direct retinal

neural

non-spatial

symbolic

Pain

Panels. *See* Widgets and Panels Pattern

Perception. *See also* action; motion perception; multimodal perception; perceptual constancies; perceptual processes; pain; proprioception; smell; sound; space perception; taste; time perception; touch; vestibular system; visual system

Perceptual constancies

color constancy

lightness constancy

loudness constancy

position constancy. *See also* adaptation, perceptual, position-constancy adaptation

displacement ratio

range of immobility

shape constancy

size constancy

Perceptual continuity

blind spot

networked environments

phonemic restoration effect

Perceptual processes. *See also* adaptation; nerve impulses

apprehension

object properties

situational properties

behavioral. *See also* behavior

binding

bottom-up

emotional. *See also* emotions

iterative. *See also* cycle of interaction

reflective

registration

top-down

attention

back projections

color constancy

pathways

phonemic restoration effect

priming

sickness

visceral

Performance. *See also* requirements

effects on

beliefs

compliance

constraints

critical incidents

device reliability

interaction fidelity

interaction techniques

latency

non-isomorphic rotations

passive haptics

scene motion

task-irrelevant stimuli

wayfinding aids

measures

accuracy

precision

time to completion

training transfer

negative training effect

perception

reviewing

task

Peripheral vision. *See under eye eccentricity*

Persistence (display)

Persistence (perceptual). *See also strobining*

masking

motion blur/smear

dark adaptation

due to latency

eye pursuit

object motion

positive afterimage

Personal space. *See under space perception*

Personas

interviews

neuro-linguistic programming

questionnaires

task analysis

template

Phonemes. *See under speech, perception*

Photorealism

Physics

constraints

hands

jitter

large forces

networked

divergence

nonrealistic

structural communication

update rate

Pie (marking) menus. *See under menus*

Pilot study

Pivot hypothesis

Placebo effects

Plot

Pointing Hand Pattern

extender grab

HOMER technique

scaled world grab

Pointing Pattern

control/display (C/D) ratio

dwell selection

eye gaze selection

hand pointing

head pointing

object snapping

precision mode pointing

two-handed pointing

Postural stability/instability. *See also* vestibular system

adaptation

causes of misbalance

tests

theory of motion sickness

Posture. *See* gestures, posture; postural stability/instability

Posture and approach

Precision mode pointing

Preferred sensory modalities

Presence. *See also* immersion

break-in-presence

characters

data collection

imperfect devices

lack of physicality

latency

network challenges

real world

visual artifacts

wires

physical interaction

hands

walking

self-embodiment

bare hands

realistic hands

touch

social

stable spatial place

depth cues

place illusion

subtle motions

vection and sickness

Proprioception

biomechanical symmetry

compliance

egocentric interaction

body-relative tools

eyes-off

egocentric judgments

input device classes

torso reference frame

visual domination over

Props. *See* Proxy Pattern; haptics, passive; touch

Prototypes

core experience

marketing

minimal

programmer

real-world

representative users

stakeholder

team

Proximal stimuli

bottom-up processing

iterative perceptual processing

motion

pictorial depth cues

registration

sensation

Proxy Pattern

proxy

tracked physical props

Push-to-talk/gesture

Put-that-there

Questionnaires

close-ended questions

examples

likert scales

open-ended questions

partially open-ended questions

personas

sickness

Kennedy Simulator Sickness Questionnaire

placebo effects

simulator

task analysis

Questions (project). *See also* questionnaires

focus groups

interviewing

scientific method

Ratcheting

Readaptation. *See under* adaptation

Realistic hands

Real world capture

3D geometry

360° film

360° stereoscopic

augmented virtuality

light fields

medical and scientific

stereoscopic film

Receptors

chemo (smell and taste)

cutaneous (touch)

hair cells

cochlea (hearing)

otolith organs and semicircular canals (motion and balance)

mechano

Pacinian corpuscles (vibration)

proprioceptors

noci (pain)

photo

cones

motion

rods

Redirected walking

Reference frames

exocentric and egocentric judgments

eye

hand

non-dominant

signifiers

head. *See also* heads-up displays (HUDs)

cyclopean eye

real-world. *See also* rest frames

torso. *See also* proprioception, egocentric interaction

steering

virtual-world

Reflective processing

Refresh rate. *See under* displays

Regions (districts and neighborhoods)

Reliability. *See under* input device characteristics; data

Rendering. *See also* field of view

asynchronous

auralization

constraints

cubic environment map

delay

double buffer

frame

frame rate

minimum

image-based
just-in-time pixels
minimal points
post
requirements
resolution
sampling
time
transparency
hand
heads-up display (HUD)
voxels
world-in-miniature
visual artifacts
aliasing
delay compensation
gaps and skins
stereo
tearing
Repetitive strain injuries
Requirements
defining
document
functional requirements
line-of-sight
quality
system
accuracy
precision
reliability
task performance
performance accuracy
performance precision
time to completion
training transfer
universal VR

usability

comfort

ease of learning

ease of use

Resolution

Rest frames

background

examples

anti-seasickness display

cockpit

stabilized arrows

hypothesis

induced motion

presence

real world

reference frame

top-down processing

Reticles

eye tracking

head pointing

implementation

Retina. *See* under visual system

Retrospectives

mini

prime directive

Reusing content. *See also* content creation; real world capture

geometry

hands and weapons

heads-up displays (HUDs)

skeuomorphism

transitioning to VR

Ring menus

Risk (project)

Route planning

Routes

Saccades. *See* under eye movements, gaze-shifting eye movements

Salience. *See* under attention

Scaled world grab

Scene. *See also* film

action

center of action zones

attention

background

change blindness

rest frame

figure and ground

geometry

contextual

detail

fundamental

hacks

scaling

interactive objects

motion. *See also* sickness, motion; locus of control; rest frames;vection; viewpoint, motion

expectations

head movement

intentional

latency-induced

perception

sensitivity

unintentional/incorrect

velocity

perceptual continuity

postural instability

Scientific method. *See also* experiments

Scope (project)

Search

conjunction

feature

naive

primed

visual

Segregation. *See under gestalt*

Selection patterns. *See also Hand Selection Pattern; Pointing Pattern; Image-Plane Selection Pattern; Volume-Based Selection Pattern*

Self-embodiment. *See under presence*

Semi-realistic hands

Sensation. *See also sensory modalities*

Sensory conflicts. *See also sickness*

accommodation-vergence

binocular-occlusion

theory of motion sickness

visual-physical

visual-vestibular

latency

Sensory modalities. *See also pain; proprioception; smell; sound; taste; touch; vestibular system; visual system*

delay times

immersion

multimodal. *See also multimodal perception; multimodal interactions*

binding

consistency. *See also sensory conflicts*

integration

presence

preferred

Sensory substitution

audio

ghosting

highlighting

rumble

Sickness. *See also adaptation; comfort; fatigue; hygiene; injuries*

Call of Duty Syndrome

cybersickness

factors

application design

individual user

system

headset fit

measuring

Kennedy Simulator Sickness Questionnaire

physiological measures

postural stability tests

mitigation techniques. *See also* locus of control

active readaptation

constant visual velocity

fade outs

leading indicators

manipulate the world as an object

medication

minimize virtual rotations

minimize visual accelerations

motion platforms

natural decay

optimize adaptation

ratcheting

real-world stabilized cues

motion. *See also* sickness, theories of motion sickness

simulator

symptoms

aftereffects

discomfort. *See also* comfort

disease

disorientation

dizziness

drowsiness

eye strain

gorilla arm

headaches

nausea

noise-induced hearing loss

pallor

seizures. *See also* flicker

sweating

vertigo

vomiting

theories of motion sickness

evolutionary

eye movement

postural instability. *See also* postural stability/instability

rest frame hypothesis. *See also* rest frames

sensory conflict

unified model

Sight. *See* visual system

Signifiers

anti

constraints

false

figures

on the hand

highlighting

indirect control

finger menus

speech and gestures

widgets and panels

labels and icons

mode

object attached tools

physical

unintended

Simulation. *See also* flight simulators; physics

vs. rendering

Simulator sickness. *See* sickness

Sketches

personas

Skeuomorphism

Skewers

Smell (olfactory perception)

Social networking

Software. *See* Make Stage

Sound. *See also* attention; sensory substitution; speech

ambient

auralization

continuous contact

deafness

change

injury

interfaces

music

time

perception

binding

loudness

loudness constancy

multimodal

pitch

spatial acuity

thresholds

timbre

vection

physical

amplitude

frequency

real world break-in-presence

spatialized audio

binaural cues

head-related transfer function (HRTF)

networked

wayfinding aids

warnings

Space perception. *See also* depth perception

action space

interaction

dominant eye

egocentric judgments

exocentric judgments

illusions

personal space

comfort

device reliability

interaction

scaled world grab

social

stereopsis

vista space

Spatialized audio. *See under sound*

Speech

output

perception

lip sync

McGurk effect

morphemes

phonemes

phonemic restoration effect

segmentation

recognition

categories

context

errors

feedback

microphones

multimodal

push-to-talk

put-that-there

signifiers

strategies

voice menu hierarchies

Spindle

Spotter

Standards. *See also* open source de facto standards

open standards

organizations

Statistical conclusion validity

threats to

false positives

fishing

statistical power

violated assumptions of data

Statistics. *See* data; statistical conclusion validity

Steering Pattern

dual analog sticks

gaze-irected steering

navigation by leaning

one-handed flying

torso-irected steering

two-handed flying

usage

virtual steering device

world-grounded devices

Stereoscopic displays. *See* displays

Sticky finger selection technique

Stories. *See also* film

Disney

emotions

engagement

escape from reality

experiential fidelity

plot

reflective processing

simulation

storyboards

subjectivity

top-down processing

user created

Storyboards

Strobing. *See also* apparent motion; judder; persistence (perceptual); persistence (display)

factors

distance

interstimulus interval (blanking time)

stimulus duration

stimulus onset asynchrony

Subconscious. *See also* neuro-linguistic programming (NLP)

balance

behavioral processing

colors

cycle of interaction

filters

hands

head motion

illusions

touch

Subjective reality. *See also* illusions

Symbolic input

Systems. *See also* displays; input; output; tracking; haptics; motion platforms; treadmills; requirements; latency; networked environments

audio

block diagrams

calibration

scene motion

sickness

chair

freely turnable (aka wireless seated)

rotation

steering

tracked

wired seated

considerations

design for

fully walkable

hardware support

location-based

mobile

nomadic

sickness factors

trade-offs

VRPN (VR peripheral network)

weight. *See* weight

wired vs wireless

Target-based travel

Task analysis

cycle of interaction

diagrams

expert evaluations

hierarchical

interviews

iteration

organization and structure

personas

task elicitation

Task performance. *See* performance

Taste (gustatory perception)

flavor

Team

attitude

communication

culture

director

Tearing. *See* under rendering, visual artifacts

Teleportation

That-moves-there

Theories of motion sickness. *See* under sickness

Thresholds. *See also* vision, visual acuity

auditory

flicker-fusion frequency

latency

redirected touching

redirected walking

saccadic suppression

speech synchronization

Time perception. *See also* events

age

biological clock

supra chiasmatic nucleus (SCN)

change

filled duration illusion

circadian rhythm

cognitive clock

events. *See also* events

passage of time

flow

perceptual moment

processing effort

subjective present

temporal attention

Time warping. *See* latency; delay compensation; 2D warp

Tools. *See* 3D Tool Pattern

Top-down processing. *See* under perceptual processes

Torso-located steering

Touch. *See also* haptics; sensory substitution

active

bare hands

hand held controllers / props

homunculus

lack of

matching visuals

pain

passive

physical panels

presence

break-in

rubber hand illusion

redirected

subconscious

symbolic input

texture

vection

vibration

visual-physical conflict

Tracking. *See also* compliance; input; input device characteristics; input device classes

absolute devices

accuracy

bare hands

calibration

camera-based

line-of-sight challenges

chair

error

eyes. *See also* eye gaze input

fingers. *See also* gloves

full-body

hands. *See also* input device classes

head

avatars

calibration

requirements

walking

hybrid (sensor fusion)

inertial

input veracity

position

precision

relative vs. absolute

reliability

requirements

torso

unencumbered

Trails

Transparency. *See under* rendering

Travel. *See under* travel

Treadmills

biomechanicalvection

mental model

omnidirectional

Tutorials (within VR)

Two-handed box selection. *See under* Volume-Based Selection Pattern

Two-handed interaction. *See* bimanual interaction

Two-handed pointing

Uncanny Valley

Usage

Use cases

scenarios

User stories

Validity. *See also* construct validity; external validity; face validity; internal validity; statistical conclusion validity

task analysis

Values (perceptual filters)

Vection

acceleration

auditory

biomechanical

example

eye fixation

mental model

periphery

presence

sickness

reducing

suppression of

top-down processing

touch

unified model

velocity

Vehicles

active control

audio cues

passive

leading indicators

perception of landmarks

real-world travel sickness

rest frames

virtual steering devices

Ventriloquism effect (visual capture)

Vestibular system. *See also* motion perception; motion platforms; postural stability/instability; scene, motion; sensory conflicts, visual-vestibular; sickness, motion;vection; viewpoint, motion; viewpoint control patterns

acceleration
visual
ambiguous input
artificial input
compliance
disorientation
eye movements
nystagmus
pendular nystagmus
vestibulo-ocular reflex (VOR)
leading indicators
motion platforms
motion sickness. *See* sickness
otolith organs
position-constancy adaptation
rest frames
examples
semicircular canals (SCCs)
sensory conflict
latency
superior colliculus
velocity
Video games (traditional). *See also* heads-up displays; reticles
Call of Duty Syndrome
controllers
differences
dual analog stick steering
gaze-irected steering
geometric detail
hand/weapon models
transitioning from
zoom mode
Viewbox. *See also* 3 Multi-Touch Pattern; Volume-Based Selection Pattern, two-handed box selection; World-in-Miniature Pattern
Viewpoint. *See also* space perception
first-person
after action reviews

disembodied

egocentric judgments

self-embodiment. *See under* presence size constancy

with third person

video games

walking in someone's shoes

motion. *See also* locus of control; motion perception; motion platforms; postural stability/instability; scene, motion; sensory conflicts, visual-vestibular; sickness, motion;vection; vestibular system; viewpoint control patterns

acceleration

angular velocity

expectations

manipulate the world as an object

rendering artifacts

rotation

sensitivity

velocity

scale

3D Multi-Touch

world grab

third-person

after action review

exocentric judgments

with first person

Viewpoint control patterns. *See also* 3D Multi-Touch Pattern; Automated Pattern; Steering Pattern; Walking Pattern

Vigilance

Virtual body. *See* presence, self-embodiment

Virtual environments

Virtual steering device

Visceral processing. *See also* communication, direct, visceral

Vision. *See* visual system

Vista space. *See under* space perception

Visual acuity

color

dark adaptation

degraded

detection

eye eccentricity

grating

pursuit

recognition

separation

Snellen eye chart

stereoscopic

vernier

Visual capture (ventriloquism effect)

Visual-physical conflict. *See* under sensory conflicts

Visual system

back projections

cones

magno cells

parvo cells

retina

blind spot

fovea

image slip

virtual retinal displays (VRDs)

rods

visual pathways

dorsal (where/how/action)

lateral geniculate nucleus (LGN)

primary (geniculostriate system)

primitive (tectopulvinar system)

superior colliculus

ventral (what)

Visual-vestibular conflict. *See* under sensory conflicts

Voice menu hierarchies

Volume-Based Selection Pattern

cone-casting flashlight

two-handed box selection. *See also* viewbox

nudge

snap

VRPN (VR Peripheral Network)

Walking in place

Walking Pattern

human joystick

real walking

redirected walking

treadmills

walking in place

Wand. *See also* input device classes, tracked hand-held controllers

Warnings

grids

sound

Wayfinding aids. *See also* navigation, wayfinding

environmental

abstract data

breadcrumbs

channels

districts

edges

handrails

landmarks

markers

neighborhoods

nodes

path

regions

routes

skewers

subtle

trails

user-placed

you-are-here

personal

compasses

forward-up maps

north-up maps

you are-here maps

Weight. *See also* fatigue

center of mass

gorilla arm

reducing

physical panels

walking

Widgets

Widgets and Panels Pattern

2D desktop integration

above-the-head widgets and panels

color cubes

menus. *See* menus

panels

hand-held

physical

widgets

Wind

World-grounded devices

World-in-Miniature Pattern

dolls

moving into

viewbox

voodoo dolls

Author's Biography

Jason Jerald



Jason Jerald, PhD, is Co-Founder and Principal Consultant at NextGen Interactions. In addition to primarily focusing on NextGen Interactions and its clients, Jason is Chief Scientist at Digital ArtForms, is Adjunct Visiting Professor at the Waterford Institute of Technology, serves on multiple advisory boards of companies focusing on VR technologies, coordinates the Research Triangle Park VR Meetup, and speaks about VR at various events throughout the world.

Jason has been creating VR systems and applications for approximately 20 years. He has been involved in over 60 VR-related projects across more than 30 organizations including Valve, Oculus, Virtuix, Sixense, NASA, General Motors, Raytheon, Lockheed Martin, three U.S. national laboratories, and five universities. Jason's work has been featured on ABC's *Shark Tank*, on the Discovery Channel, on the UK's *Gadget Show*, in the *New York Times*, and on the cover of the MIT Press journal *Presence: Teleoperators and Virtual Environments*. He has held various technical and leadership positions including building and leading a team of approximately 300 individuals, and has served on the ACM SIGGRAPH, IEEE Virtual Reality, and IEEE 3D User Interface Committees.

Jason earned a Bachelor of Computer Science degree with an emphasis in Computer Graphics and Minors in Mathematics and Electrical Engineering from Washington State University. He earned a Masters and a Doctorate in Computer Science from the University of North Carolina at Chapel Hill with a focus on perception of motion and latency in VR. His graduate work consisted of building a VR system with under 8 ms of end-to-end latency; the development of a mathematical model relating latency, head motion, scene motion, and perceptual thresholds; and validation of the model through psychophysics experiments. Jason has authored over 20 publications and patents directly related to VR.