AI-DRIVEN COMPUTATIONAL DESIGN TOOLS FOR SYNTHESIZING HUMAN-CENTRIC DESIGN

A Dissertation Presented

by

Haikun Huang

Submitted to the Office of Graduate Studies, University of Massachusetts Boston, in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2020

Computer Science Program

AI-DRIVEN COMPUTATIONAL DESIGN TOOLS FOR SYNTHESIZING HUMAN-CENTRIC DESIGN

A Dissertation Presented

by

Haikun Huang

Approved as to style and content by:

Marc Pomplun, Professor
Chairperson of Committee

Dan Simovici, Professor
Member

Duc Tran, Associate Professor
Member

Lap-Fai Yu, Assistant Professor
Member

Dan Simovici, Program Director
Computer Science Program

Marc Pomplun, Chairperson
Computer Science Department

ABSTRACT

AI-DRIVEN COMPUTATIONAL DESIGN TOOLS FOR SYNTHESIZING HUMAN-CENTRIC DESIGN

August 2020

Haikun Huang,
B.S., University of Massachusetts Boston
Ph.D., University of Massachusetts Boston

Directed by Professor Marc Pomplun

In the view of the industry, the modeling design process is an overwhelming, non-trivial, and time-consuming task. As a result, creating such a design can take days, months, even several years. In this thesis, we investigate how human designers use human-centric computational-aided modeling design tools to assist the modeling design processes. We investigate:

- how we utilize the perception data obtained from the virtual environments to develop novel computational approaches of the layout designs for human movement comfort, accessibility, and convenience.
- how we utilize the semantic data obtained from real environments to develop novel computational approaches of the sound source assignment for the immersive experience in virtual reality.
- how we utilize the suggested images obtained from a set of annotated images to develop novel computational approaches of gallery design for generating style-unified gallery walls in augmented reality.

We then tackle various challenging computational design problems within our novel approaches. In the visual perception approach, we devise agent-driven optimization methods for the road signs assignment by a given layout, and we devise an adaptive algorithm to design a gallery by a suggestion system. In the auditory perception work, we devise an automatic algorithm to enhance the immersive of the given static panorama images through realistic sound sources assignment.

ACKNOWLEDGMENT

This project would not have been possible without the support and guidance of an enormous amount of people.

I want to thank my advisors, Professor Lap-Fei (Craig) Yu and Professor Marc Pomplun. I am truly fortunate that I got to know them during my Ph.D. program. I thank Professor Yu, for he has always been very supportive of my thoughts and ideas, placing his confidence in me. I thank Professor Pomplun, for he has always been providing useful comments to my thoughts and ideas. I thank Professor Duc Tran for his guidance on my calculation algorithms and principles. I would also like to thank my mentees, Yongqi Zhang, and Biao Xie. While guiding them, we are actually learning and encouraging each other.

My special thanks go to Professor Dan Simovici, who is the graduate program director of the UMass Boston Computer Science Department, financially supported me during my Ph.D. Program by the teaching and research assistantships.

I want to thank my parents, for they have dedicated so much to my education. Not only have they encouraged me to face the challenges, but they also have ignited my passion for knowledge and aroused my curiosity in the rules of nature. I would also like to thank my extended family for financial, supporting me so far away from home. Finally, a very personal thank you goes to my wife, Yi, for her patience and support, especially for the daily living during my Ph.D. program.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

Figure 1: An example of the daily use products. From left to right, they are clothes, easy transport, comfortable homes, and a variety of products.

We live in a world where we can enjoy the right clothes, secure transport, comfortable homes, and a variety of products. Therefore, we enjoy the products and creations. Nearly every aspect of our lives involves designs. We would argue that the designing process is still insufficient and inefficient. Although we can design many products, the designing process is still painstaking. The design is inefficient and suboptimal to meet our demand. To understand why we need to look at how humans design, Figure 1 shows an example of the daily use products, they are standardized, not personalized, and take a long time to design. Most designs are expensive; not many people can enjoy the products. With AI-Driven computational design tools, we will be able to create things even better and faster so that we can live in a world that is even more efficient, comfortable, and people-friendly.

Figure 2: Examples of design in various aspects.

1.1 Traditional Design

The term of *Design* is a broadly-defined. The Oxford dictionary defines it as deciding upon the *LOOK* and *FUNCTIONING* of something, such as architecture, a poster, clothes, and products. Figure 2 shows examples of design in various aspects.

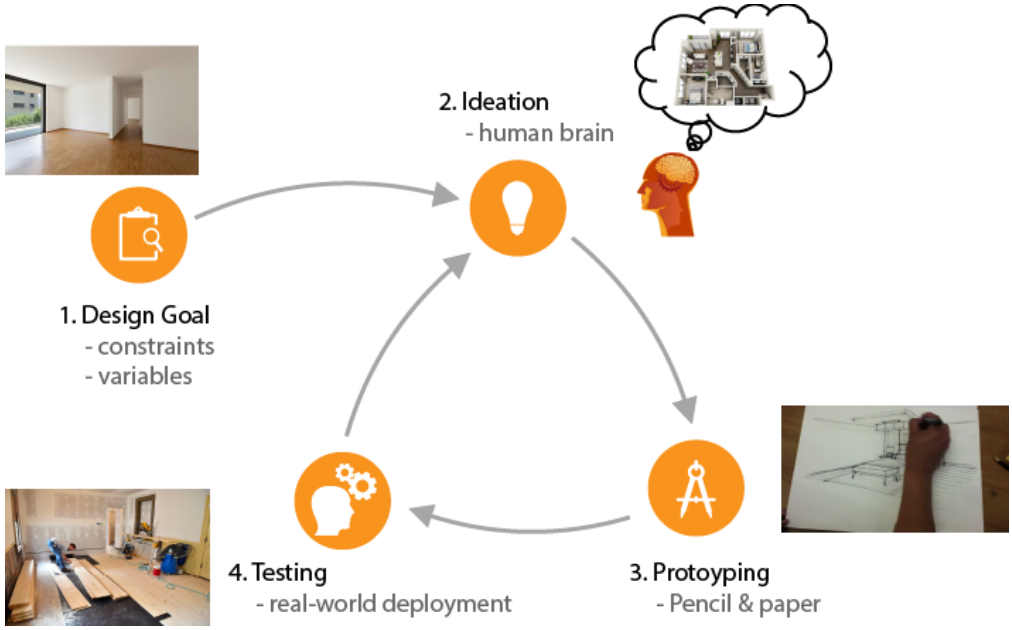


Figure 3: Examples of design in various aspects.

Figure 3 shows the traditional designing process before the computer age. Starting from a designing goal, such as creating an interior design for an apartment, the designer starts by thinking one interior design solution. He/she then sketches his/her idea on a

piece of paper. If we want to visualize the design, the designer can draw it by hand. The visualization, of course, is a prolonged and expensive process. Moreover, we only can take the visualization as a reference for the project. On the other hand, we can test the design only after we deploy it in the real world.

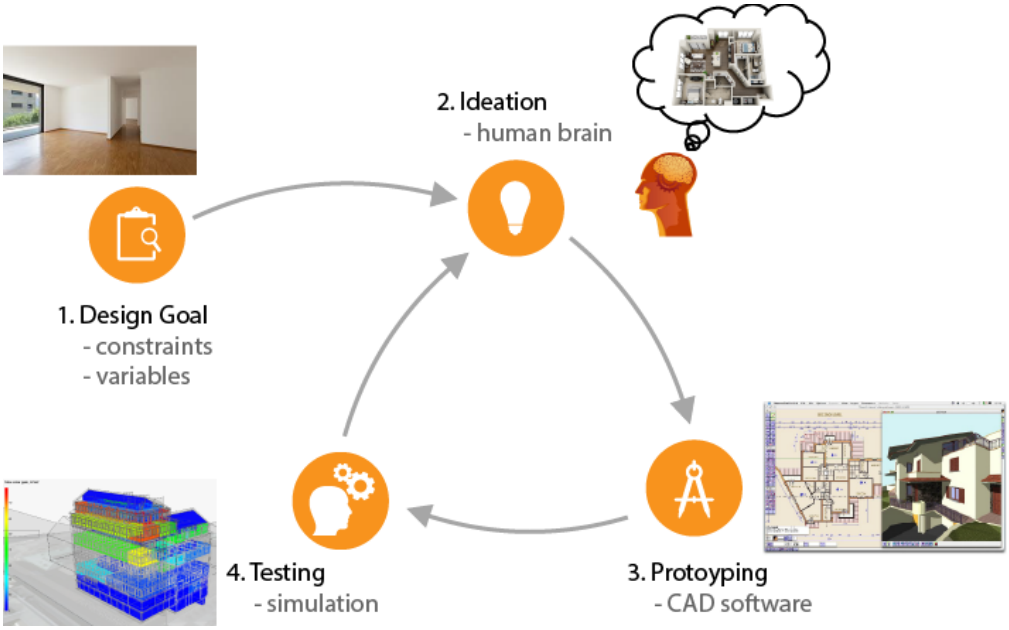


Figure 4: A workflow of the design with the CAD software.

Starting from the 1970s, computer-aided design or CAD has become an essential technology for designing a variety of things, such as architectures, cars, and consumer products. Figure 4 shows a workflow of the design with the CAD software. Compared to hand-crafting, it is much more precise and cost-effective. Designers can also quickly generate 3D previews of their designs so that it becomes easier to communicate their designs with their clients. The traditional CAD software is a passive tool used by the designer. The design already in the designer's brain, and the CAD software is just for documenting the design accordingly. Though the CAD software can provide a 3D visualization, the designing process is still complicated. To use CAD software, the

designer still needs to start with a design idea from his/her brain. However, this time he/she can use computers rather than sketch by hand to document his/her idea, which is much more precise. He/she can also generate a 3D visualization of the design and perform simulations on the virtual environment to check whether it is good or bad before the real-world deployment.

1.2 Challenges of Traditional Design

Although CAD software facilitates the designing process, there are still many challenges. Generally speaking, CAD software can be very complicated for non-specialists to use, and it requires a long time of training to become proficient. Strictly speaking, CAD software does not aid the designer in thinking of a new design. Instead, it aids the designer to digitally document a design idea that he/she already has in his/her mind. As a result, even with the use of CAD software, the designing process can still be very labor-intensive because much workforce is still needed to come up with the design idea and to document the design through manual operations. As human designers manually created the designs, and humans can make mistakes, the design flaws quite possibly arise. As a result, the design flaws might create inconvenience, and the end-users will suffer the flaws. If the design flaws happened in large-scale city planning, it would create big economic problems that would be hard to fix. Figure 5 shows an example of the designing flaws.

1.3 AI-Driven Computational Design Tool

Because of the above challenges, we would like to research on how to improve the designing processes further using the computer. The key is to let humans co-design

Figure 5: An example of the designing flaws.

with AI. This research area is called “computational design.” The branch of research is sometimes also called generative design, creative AI, parameterized design, algorithmic design, or computational creativity. In contrast, in computational design, based on our designing goal, computers help us generate a lot of different design ideas for us to start. Also, in computational design, prototyping is no longer done manually by the designers using CAD software tools. Instead, the computers automatically finish that for us. Besides, in computational design, the simulations do not just tell whether a design is good or bad; we use the simulation results to refine the design automatically through optimization. Such that the designers do not need to do many trial-and-errors to refine the designs. The features of the AI-Driven computational design are:

- fast, cheap, automatic, and scalable,
- explore creativity solution space,
- design transfer,
- design amplification,
- personalized, human-centered design,

- handle tricky tradeoffs & constraints,
- intuitive user control.

1.4 Optimization

The basic idea is to formulate the design as an optimization problem. The task of the optimization function is to obtain the solutions which represent good designs in the solution space defined by the designing goals. Depending on the optimization problems, we may have different goals [YYT12]. For example, when designing a car, the designing goals would be the size, color, horsepower, fuel consumption, and price. Figure 6 shows an example of the solution space for designing a car. It evaluates the possible solutions for designing a compact vehicle through the optimization function.

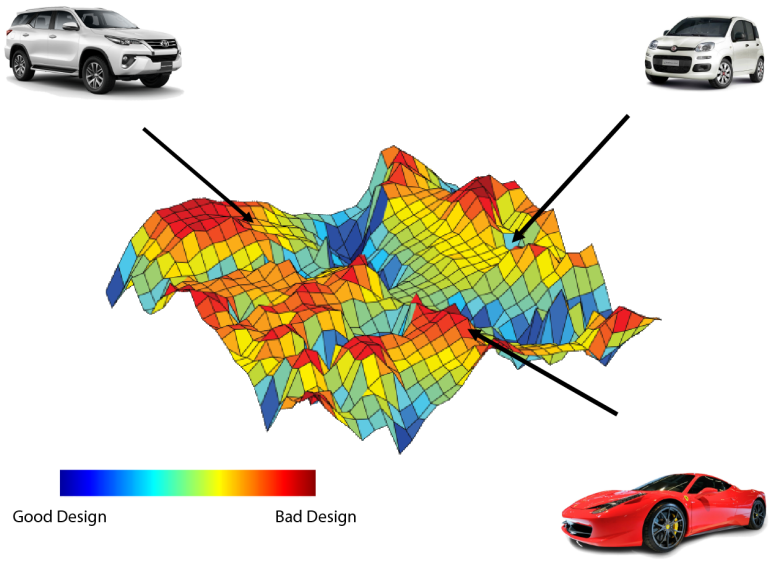


Figure 6: An example of the solution space given by an optimization function for designing a compact vehicle.

1.4.1 Large Space

Usually, the solution space is enormous for such optimization problems. The size of the solution space is defined by the number of candidates for each designing goal. For example, when designing a car, the designing goals would be the size, color, horsepower, fuel consumption, price. We assume that there are 10 candidates for each designing goal, then the solution space for designing a car would be $10,000$. In the real world, the solution space for designing a car would usually be in millions or even billions.

1.4.2 Automatic

Although computers help us with prototyping and testing, the ideation is still solely done by the human designer. Compare to manually prototyping the design by designers, and computational design will generate a lot of different designs base on our designing goals. In other words, computers will automatically generate a set of prototypes for us to start with further improvement.

1.4.3 Trade-off

The idea of trade-off is to assign weights to different objective functions and convert them into a single objective function by the weighted sum. For example, the essential features of a compact vehicle are smaller size, lower fuel consumption, and lower price. In other words, the weights of size, fuel consumption, and the price would be higher than the weights of the other features when designing the compact vehicle.

1.5 Machine Learning

The massive amount of data we have nowadays while machine learning advances have created a new wave of exciting opportunities for AI-driven computational design. Compared to 10 years ago, we now have a lot of novel devices, such as panorama cameras, depth sensors, and drones that can capture a massive amount of images, videos, and 3D data easily. We also have new sensors that can track the head movement, hand movement, body movement, and even eye gaze of users navigating in virtual reality. Figure 7 shows an example of novel devices. The data collected by such devices can help us understand how humans respond to different events and interact with different objects so that we can generate designs that are good for affordances and human perceptions.

Figure 7: An example of novel devices.

1.5.1 Perception

Perception deals with the human scenes that generate signals from the environment through sight, hearing, touch, smell, and taste. Vision and audition are the most well understood [Enn12].

In the wayfinding scenario, people look at their surroundings through the visual system to find the way and locate their destination. E.g., people navigate and locate through seeking the road signs and the landmarks [Bih17]. Figure 8 illustrates a Google street screenshot of the first-person view in New York Times Square and its viewing position corresponding to the aerial view of the street by the landmark, which is the separate building in the middle of the road.

The first-person view Aerial view of the street

Figure 8: An example of people navigate and locate through seeking the landmark. The first-person view in New York Times Square and its viewing position is corresponding to the aerial view of the street by the landmark, which is the separate building in the middle of the road.

In scene recognition, people utilize the surrounding sounds to determine if the current environment is - indoors, outdoors, busy streets, or quiet forests [LGA09] [AP15]. Notably, the distance of the sound also makes people feel that they are in a dangerous or safe situation. E.g., yelling people close to them, which is dangerous, and a quiet car engine sound far away from them, which is safe.

In the aesthetic scenario, emotion always associates with colors [Han17]. E.g., the artist conveys the emotions and messages that the artist wants to express through the color tone of the artwork. Warm colors represent warmth and hope, while cool colors represent coldness and depression. Figure 9 shows an example of the same photos with different color tones. The warm one gives people a wonderful feeling for the future, while the cool one gives people a sense of helplessness before the disaster strikes.

Figure 9: The same photos with different color tones. The warm one gives people a wonderful feeling for the future, while the cool one gives people a sense of helplessness before disaster strikes.

1.5.2 Predict

The definition of *Predict* is: say or estimate that will happen in the future or will be a consequence of something. An efficient automatic designing system should be able

to predict the human preferences according to the target parameters specified by the designers [ASH19]. For example, when designing a product for a specific group of people, designers should either pay attention to either functional diversity or lower production costs. An excellent automatic designing system should give appropriate predictions, such as designing a cutting tool for homemakers. The ideal one should be a cheap and straightforward ordinary knife rather than a functional and complicated Swiss Army knife.

1.5.3 Computational Design

There is a misunderstanding about the computational design; most people think of the computational design is to create things with a fascinating look. However, the essence of computational design is to solve problems. The process of computational design can be described in terms of needs and plans. For example, simplifying the decoration of a cup does not prevent people from recognizing it as a cup; but if this cup is simplified into a cylinder, no one will recognize it. In a broad understanding, the computational design is intuitively described by a linear model: from defining a problem to solving it. Defining a problem is an analytic process, which means to find the problem and understanding every detail, and clarifying all the needs that need to be met; solving a problem is a synthetic process, which refers to integrating these needs and balancing the various, and finally, produce a final plan that can guide actual production.

Figure 10 shows Mark Zuckerberg's Clock, designed by Mark Zuckerberg for helping his wife, Chan, knowing the time through the night. The problem with Chan is that she has a difficult time sleeping through the night. Zuckerberg said she often checks the time when she wakes up to see if the kids will be awake soon. However, knowing the time stresses her out and makes it hard for her to fall back asleep. So he constructed a wooden "sleep box" that emits a faint light between 6 a.m. and 7 a.m. The light is

noticeable enough to let her know that it might be time for the kids to wake up, but it would not wake her up if she is already asleep.

Figure 10: Mark Zuckerberg's Clock. A wooden "sleep box" that emits a faint light in the early morning, to alert his wife when it is time to wake up without looking at her phone.

1.5.4 Context

The context is a very abstract thing, for example, we can perceive the situation of the location we stayed from the context of the environment - the ambient sounds should be noisy in restaurants while the ambient sounds should be quiet in libraries.

For example, the context of designing a container depends on the usage of this container - for drinking water or for drinking wine? With the same designing goals (container) in different contexts(for drinking water or wine), the products should be differ-

ent functionalities. A container would be designed sturdy and straightforward for the drinking water propose while a container would be designed elegant and stylish for the drinking wine propose as Figure 11 shown.

Figure 11: An example of the cups for different proposes.

1.6 Advantages of Computational Designs over Traditional Design Approach

Computational design approaches are advantageous. It allows us to explore the vast designing solution space quickly, to generate many personalized and human-centered designs automatically. It also allows the designer to control the design by setting constraints intuitively. Figure 12 shows a comparison of the designing process between traditional design and computational design.

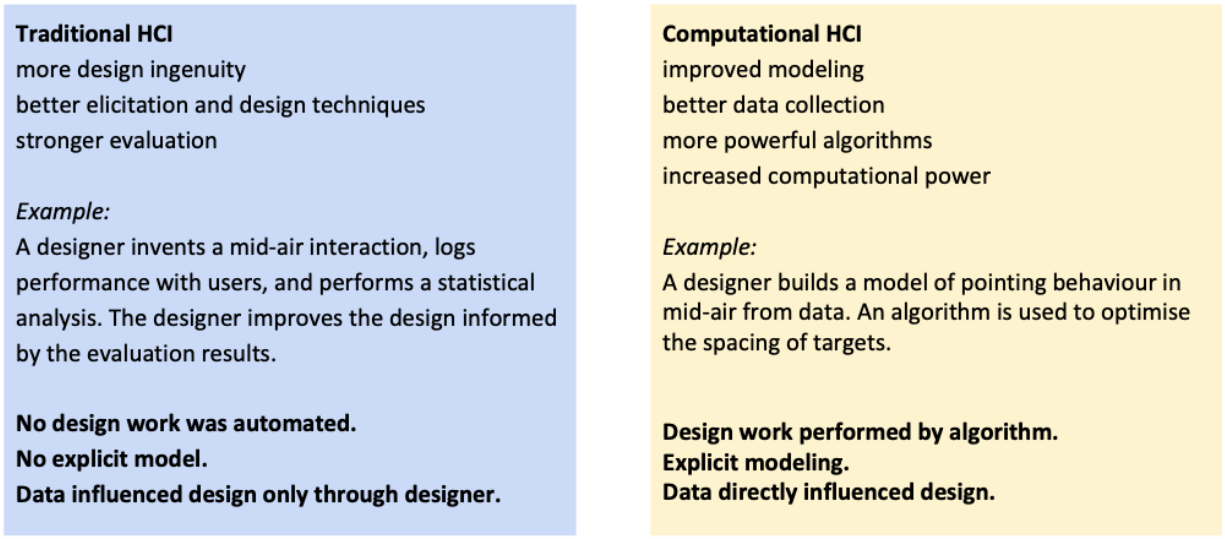


Figure 12: Comparison of the traditional designs and the computational design.

1.7 Contributions of The Thesis

This thesis develops new algorithms and design tools to help professionals and non-professionals in domains ranging from modeling and computational design to human-centric simulations. In particular,

- *Automatic Optimization of Wayfinding design*. This work devises an automatic tool to help wayfinding designers create better wayfinding designs for architectural spaces.
- *Audible Panorama: Automatic Spatial Audio Generation for Panorama Imagery*. This work automatically synthesizes realistic spatial sounds for 360° panorama images for delivering immersive VR experiences.
- *Interactive Design of Gallery Walls via Mixed Reality*. This work devises an intuitive tool to help non-professionals for gallery design via the Mixed Reality Headset.

Accordingly, the objective is to achieve the following goals:

- Devise a novel agent-based optimization approach to automatically optimize human-centric wayfinding design. Specially, we will demonstrate such an approach in outdoor and indoor virtual environments.
- Devise a novel data-based approach to synthesis realistic sound sources. Specially, we will demonstrate an application of such an approach to place the realistic sound sources into the outdoor and indoor 360° panorama images.
- Devise a novel data-based approach to automatically generate a unified style gallery. Specially, we will demonstrate an application of such an approach on a mixed-reality platform.

1.7.1 Automatic Optimization of Wayfinding design

In Chapter 2, we develop a system that automatically generates an optimized wayfinding design. Given a layout that first converted into a graph representation and a set of points, our system then performs to find the optimized paths from the points to points, considering the wayfinding criteria such as the path lengths, the turning angles, and the number of decision points. Then, an agent-based sign refinement subsequently takes place, which employs agent-based simulations to evaluate and refine the sign placement according to the human-centric navigating properties. The cost function is optimized by simulated annealing using a Metropolis-Hastings state search step to deal with the prohibitively large search space. We have major contributions,

- Introducing a novel optimization and agent-based approach for automatically generating wayfinding designs.

- Demonstrating the capability of our approach for generating wayfinding designs for different layouts.
- Showing how our approach can be further applied for visualizing and editing a generated wayfinding design.
- Evaluating the effectiveness of our automatically generated wayfinding designs in guiding the navigation of users by comparing with other wayfinding designs.

1.7.2 Audible Panorama: Automatic Spatial Audio Generation for Panorama Imagery

In Chapter 3, we develop a system that automatically places realistic sound sources into the given panorama image. We use object and scene tags assigned during scene classification and object recognition for audio assignment. These tags matched with audio tags from our audio file database and the audio files associated with the audio tags are assigned as audio sources. We estimate the depths of detected objects by comparing relative heights of objects and pre-established knowledge on the average heights of different types of objects. We have major contributions,

- We proposed a new framework for automatically assigning realistic spatial sounds to 360° panorama images based on object recognition and depth estimation.
- We constructed a large dataset of panorama images and audio files. The panorama images are made audible by running our approach on them. The dataset, the results obtained by running our approach, as well as the tool for experiencing the audible panorama will be publicly released for research purposes.
- We conducted a statistical analysis that evaluates the importance of various factors in our proposed framework.

1.7.3 Interactive Design of Gallery Walls via Mixed Reality

In Chapter 4, we develop an interactive design tool that allows users to create and visualize gallery walls via a mixed reality device. Our tool is suitable for both novice users and experienced designers. Users can directly visualize how the gallery wall design will look on the real wall while interactively creating the design. We have the following major contributions,

- Based on interviews with professional designers, we devise a computational approach for facilitating and automating the design of gallery walls, which enables a novel mixed reality interactive design tool.
- We demonstrate how mixed reality technology, which bridges the gap between real-world scene knowledge and design suggestions computed in a virtual setting, can be adopted for interior design. In our case, we particularly demonstrate how such an approach can be applied for designing gallery walls.
- We conduct experiments to evaluate the user experience and performance of using our novel tool for gallery wall design. We also conduct a perceptual study to evaluate the quality of the gallery wall designs created by users with our tool.

1.8 Thesis Organization

The remainder of the thesis is organized as follows: Aiming at demonstrating the AI-driven optimization approaches for computational design, we will present three projects in which we devise novel and AI-driven optimization for model synthesis. These will be discussed in the three chapters that follow-in Chapters 2, 3, and 4. We tackle three modeling problems from the computational design perspective. In each chapter, we will begin with a problem statement and introduction, followed by a review of related work

relevant to the problem, a presentation of the technical details of our novel AI-driven optimization method suited to the modeling task at hand, followed by experimental validations, discussion of results, and suggestions for future work. Chapter 5 concludes the thesis.

CHAPTER 2

AUTOMATIC OPTIMIZATION OF WAYFINDING DESIGN

2.1 Introduction

Imagine walking in a subway station with no wayfinding signs. How could you walk to the right platform after you buy your ticket? After some random trials, you might finally find your way to the platform, but this probably would not be a pleasant experience. You would have saved much time and energy if wayfinding signs had been placed properly in the environment to guide you through. A layout with no wayfinding signs is as confusing as a maze.

In "The VR Book" [Jer15], Jerald points out that wayfinding aids are especially important in virtual environments because it is very easy to get disoriented throughout a navigation in a virtual space. A well-constructed environment should include environmental wayfinding aids thoughtfully put by the level designers, considering the possible navigation and the navigation goals of the user. Recently, interesting experiments by Darken and Peterson [DP14] verify that most users would feel rather uncomfortable being in a largely void virtual environment, and that it is important to regularly reassure the users that they are not lost throughout a navigation.

Conventionally, level designers mainly rely on experience or a "common sense approach" [Sym15] in creating a wayfinding design. Given an environment, they think

of all likely navigation scenarios that the user will go through and then place wayfinding signs or other aids to guide the user accordingly. For example, for a train station, one common scenario is to walk from the ticket machine, through the gate, and then to the right platform. Another common scenario is to walk from the platform to the exit. Directional signs are then placed along the routes. While this design approach is straightforward, the efforts required will quickly become daunting when the number of scenarios scales up as in a real-world situation. For example, a real-world train station typically involves tens or more navigation scenarios. Moreover, when placing the signs, it is necessary to consider the user's visibility and the fact that the user may miss a sign or make mistakes throughout the navigation. Designing a wayfinding scheme that jointly considers all these factors is highly non-trivial and challenging, while a sub-optimal wayfinding scheme may easily result in a confusing and frustrating navigation experience of the users.

To tackle these problems, in this work we introduce a novel computational approach to automatically generate a wayfinding design for a given environment. To use our approach, the designer simply specifies all the navigation scenarios likely to be taken by the users. Our approach will then generate a wayfinding design to accommodate the needs of all the scenarios while considering a number of desirable factors relevant to the navigation experience and management convenience. Through agent-based simulations, our approach further refines the locations of the wayfinding signs by considering visibility and robustness with respect to the possible mistakes made by the users throughout their navigation. After generating a wayfinding design, the designer can gain further insights of the design by visualizing the accessibility of a destination from any other locations in the environment, and remove any blind zones (if necessary) by adding more signs and re-triggering the optimization.

In a real-world architectural site, typical wayfinding aids include signs, landmarks and GPS-based mobile navigation system. In a virtual environment, additional virtual wayfinding aids such as compasses [Jer15] and mini-maps [Ada14] can also be used to facilitate wayfinding. In this work, we focus on generating signs to guide the user because: 1) signs are a very common and universal mean for wayfinding; 2) signs as wayfinding aids are direct yet subtle—the user usually does not need to stop walking while reassuring his direction with a sign he sees on his way, in contrast to using other wayfinding aids such as a map which requires the user to stop his locomotion; 3) signs integrate naturally with most indoor and outdoor environments.

The major contributions of our work include:

- introducing a novel optimization and agent-based approach for automatically generating wayfinding designs.
- demonstrating the capability of our approach for generating wayfinding designs for different layouts.
- showing how our approach can be further applied for visualizing and editing a generated wayfinding design.
- evaluating the effectiveness of our automatically generated wayfinding designs in guiding the navigation of users by comparing with other wayfinding designs.

Additionally, we implement our approach as a handy plugin of the Unity game engine, which can be used by game level designers to automatically and quickly generate wayfinding schemes for their virtual worlds, hence saving their time and manual efforts spent on determining users' paths and placing wayfinding signs. We will release the plugin for public use.

2.2 Related Work

To the best of our knowledge, there is no existing work on automatically generating wayfinding designs for a given layout. We review some relevant work in wayfinding design for real-world and virtual environments. We also review some work in sign perception, navigation and path planning which bring useful insights about the human factors to consider in a wayfinding design.

2.2.1 Conventional Wayfinding Design

We give a succinct overview of the real-world wayfinding design process, which inspires our computational approach for generating wayfinding design.

In architectural design, wayfinding refers to the user experience of orientation and choosing paths within a built environment , which usually relates to the user's needs or the attractiveness of the places and streets in the environment [KW14, BMW17]. In the book *The Image of the City* [Lyn60], Lynch defined wayfinding as the "consistent use and organization of definite sensory cues from the external environment". Environmental psychologists later extended the definition of wayfinding to include also the use of signage and other graphical and visual clues that aid orientation and navigation in built environments [Pas92].

The process of wayfinding—how humans walk to their destinations in an unfamiliar environment—involves four major steps [LHB10]: *orientation*, *path decision*, *path monitoring* and *destination recognition*. *Orientation* refers to determining one's current location. *Path decision* refers to selecting paths to navigate to the destination. *Path monitoring* refers to continuously verifying that the path indeed leads to the destination. Finally, *destination recognition* refers to confirming that the destination has been

reached. Our goal in this work is to automatically generate a wayfinding design for a given environment to facilitate the above wayfinding steps.

Today almost all public spaces and private premises require a wayfinding scheme [Gib09] to ensure that they are universally accessible for all users [Uni]. To achieve this goal, after a layout is designed by architects, a wayfinding design team [Cal07] will decide about the wayfinding signs to put in the environment. In current practice, the design team manually creates a wayfinding scheme following a "Common Sense Approach" mainly based on experience [Sym15, Kat12]. Given a new premises such as a train station, designers create a wayfinding scheme following these major steps:

1. *Identifying Major Paths:* The design team first identifies the major paths likely to be taken by pedestrians, by experience or surveys with the property managers. The team examines the site's floor plan or make an on-site inspection to estimate people flows [Cal07]. The goal is to gain a comprehensive understanding of the site.

2. *Devising a Wayfinding Scheme:* Considering all the major paths, the design team determines the types and locations of the wayfinding signs, which should be placed at an appropriate height and angle clearly visible to pedestrians. Additional signs should be placed to eliminate any possible confusion caused by the architecture itself. As an example, Figure 13 shows the circulation analysis and a wayfinding scheme manually created for a concert hall.

3. *Designing, Fabricating and Placing Signs:* After devising the sign placement, the designers design the appearance of the signs to be manufactured and placed in the real environment.

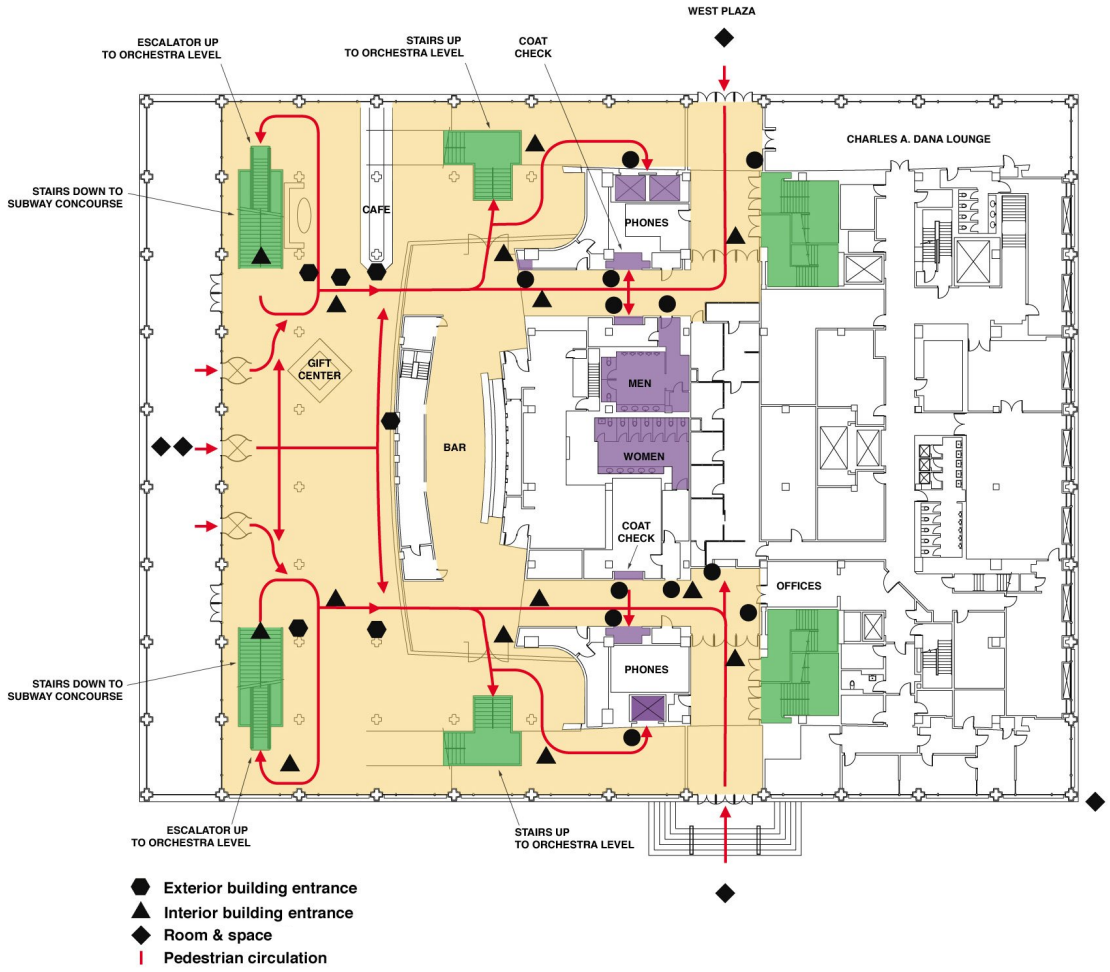


Figure 13: Circulation analysis and wayfinding scheme of a concert hall manually created by a wayfinding designers, including projected circulation paths and sign types. Courtesy of ArchitectureWeek.

4. *Evaluation, Maintenance and Update:* The team maintains the wayfinding signs in a database and reviews the sign placement periodically to replace any outdated signs.

Interested readers may refer to the literature [Cal07, Ueb10, Gib09] for more detail of the design process. Similar to the real-world wayfinding design process, our computational approach focuses on automatically identifying locations for placing signs in an environment according to the designer-specified navigation goals of the pedestrians.

2.2.2 Wayfinding Design for Virtual Environments

Wayfinding aids are crucial in virtual environments because they help users form cognitive maps, maintain a sense of position and direction of travel, and find their ways to their destinations [Jer15]. Common wayfinding aids in virtual environments include signs, maps, landmarks, light, and paths [ENK97, Jer15]. In designing a highly immersive and steerable virtual environment, it is important for level designers to make use of wayfinding aids effectively to enhance spatial understanding of the environment so that users can comprehend and operate smoothly [Mit12, Jer15]. This principle also applies to game level design. In his book, game designer Michael Salmond emphasizes the use of a road sign system in games as an important wayfinding tool to provide players with a highly immersive navigation experience [Sal16]. Figure 14 shows some example road signs used in the popular video games Fallout 4 and the Elder Scrolls IV: Oblivion.

In current practice, wayfinding aids are manually added to a virtual environment by level designers and then empirically tested for effectiveness, which depends on the quantity and quality of wayfinding aids provided to users, yet research found that it could be overwhelming to users if exposed to too many wayfinding aids [BKL04].

Darken and Sibert conducted an important study [DS96] about the wayfinding strategies and behaviors of human users in large virtual worlds. Their experiments verified that human wayfinding strategies and behaviors in large virtual worlds are strongly influenced by environmental cues. Their experiments asserted that *humans generally adopt physical world wayfinding strategies in large virtual worlds, hence common wayfinding aids in the physical world can be effectively applied to facilitate wayfinding in virtual worlds*. Based on the insights, Cliburn and Rilea [CR08] conducted a further study to compare human performance in searching for an object in a virtual environment with no aid present, with maps and with signs. The results show that subjects who navigated the virtual environment with the aid of signs achieve superior performance than under other conditions. These findings motivate us to investigate the automatic propagation of directional signs in virtual environments to enhance wayfinding.

Wayfinding Map Generation. In computer graphics, there are interesting approaches for automatically generating tourist brochures [BMW14] and destination maps [KAB10]. Though these maps are intended for real-world navigation use, they could potentially be used to assist navigation in virtual environments. Given a map and some desired destinations, these approaches select a subset of roads to reach the destinations, and visualize the important routing instructions on a generated map which is intuitive to use. Our approach is inspired by these approaches, but focuses on optimizing the placement of wayfinding signs in the layout so as to guide pedestrians to reach their destinations easily. Combining automatically generated maps with the wayfinding signs generated by our approach can potentially provide users with effective wayfinding aids to navigate smoothly in virtual environments.

(a) Fallout 4 (b) The Elder Scrolls IV: Oblivion

Figure 14: Example road signs used in video games.

2.2.3 Perception, Path Planning and Navigation

Our wayfinding design approach is also inspired by how humans perceive and navigate in everyday environments.

Perception. In everyday environments, humans continually shift their gaze to retrieve wayfinding cues for making navigation decisions [HHP09, HWP11, WWP14]. Human visual attention is known to be attracted by low-level features such as changes in color, intensity, orientation and contrast [IK01], and by high-level scene context [WWP14]. Some particular categories of objects, such as signs and texts [WP12, WLL12, WLF14], are known to strongly attract eye fixations regardless of their low-level visual saliency. Therefore, we focus on optimizing the placement of wayfinding signs in our approach.

Path Planning and Navigation. Given a layout, there are usually multiple paths a pedestrian can take to navigate from a starting point to a destination. For instance, suppose a hiker wants to walk from the bottom to the top of a hill. He may walk a path which is mostly straight, or a shortcut with sharp turns. A common strategy for path planning is to design a cost function to evaluate each path, and then search for a path that corresponds to a low cost [JCS10, Nil14, GPM10, DK03]. For a low-dimensional

Figure 15: Overview of our approach. (a–b) An input layout is first converted into a graph representation. (c) A wayfinding scheme optimization is then performed to find the optimized paths from the entrance to the points of interests, considering wayfinding criteria such as path lengths, angles and number of decision points. (d) An agent-based sign refinement subsequently takes place, which employs agent-based simulations to evaluate and refine the sign placement according to the navigation properties of agents mimicking human pedestrians.

configuration space, a grid-based search such as A* [HNR68] or D* [Ste94] can be applied to find an optimal path. For a high-dimensional configuration space, sampling-based approaches [KSL96, MT13] are commonly applied to find an optimal or near-optimal solution.

For path planning, common navigation factors to consider in the cost function involve: 1) path length (one wants to choose a short path to reduce the travel time needed to reach the destination); 2) number of turns (one wants to minimize the number of turns to reduce the complexity of the route [Win02, KAB10]); 3) number of decision points (each intersection is a decision point where the pedestrian will need to decide which road to follow next; one wants to minimize the number of decision points to reduce the chance of making mistakes). Arthur and Passini [PA92] noted that the number of decision points has an important influence on the difficulty of performing wayfinding. Casakin et al. conducted empirical studies [CBK00] which further verify these observations. We consider these criteria in the generation of our wayfinding designs, after which we will place the wayfinding signs and refine the placement based on agent's properties.

Moreover, the designer can control the importance of each criterion by adjusting its associated weight. Our approach will generate a wayfinding design accordingly.

Navigation Mistakes. Humans occasionally make mistakes in navigation. For example, it is common for pedestrians to miss a sign due to occlusion by other pedestrians; distractions such as advertisements or events happening in the environment [NA08, Bur98]; or a wrongly recognized sign or landmark [Gol99]. It is also common for pedestrians to make wrong turns in navigation [Bur98]. A well-thought-through wayfinding design should tolerate these kinds of human mistakes [DS96, PA92]—a pedestrian should still be able to reach to his destination even if he makes mistakes occasionally.

Agent-based Evaluation. Martin Raubal [Rau01] used agent-based simulation to evaluate human wayfinding in unfamiliar environments, yet the simulation used does not consider the mistakes that can be made by the agents; further, it is unsure how such evaluations can be used to enable automatic sign placement. In contrast, our agent-based simulations consider navigation mistakes and we also show how such simulations can be used to create a robust wayfinding design. Our approach is also motivated by autonomous agents [ST05] and crowd simulations [TCP06, PAB08, TMB07]. However, instead of generating realistic agent simulations, we focus on applying agent-based simulations for optimizing wayfinding designs.

To achieve a robust wayfinding design, our approach conducts agent-based simulations in placing the signs to evaluate how well the design can tolerate occasional mistakes made by agents. Using our approach, the designer can control how robust the generated wayfinding design needs to be by changing the agent parameters. For example, in creating the wayfinding design for a subway station where the pedestrians (many of whom are first-time visitors) are generally expected to be unfamiliar with the environment, the designer can adjust the agents to have a higher chance in making mis-

takes. Our approach will generate a more robust wayfinding design by placing signs in important locations so that pedestrians can still find their ways despite the mistakes.

2.2.4 Computational Layout Design

Layout design is an important problem in computer graphics. A layout typically consists of a number of sites connected by paths, with each site serving a different purpose. Computer-generated layouts can be used for creating virtual environments where virtual agents and human users can navigate for simulation and entertainment purposes. Galin et al. proposed to generate roads procedurally given a natural landscape with river and hills [GPM10]. Computationally generated layouts can also be used for architectural design [MSK10, BYM13, PYW14] and urban planning [VKW12, VAB09, VGA12, PMB16]. Refer to the survey [STB14] by Smelik et al. for a comprehensive review of the state-of-the-art procedural modeling techniques for generating layout designs for virtual environments.

An important consideration in designing a layout is the navigation experience of the pedestrians. Recently, Feng et al. proposed an approach [FYY16] which uses crowd simulation to generate mid-scale layouts optimal with respect to human navigation properties such as mobility, accessibility and coziness. However, concerning navigation, one important consideration is missing: the wayfinding experience of the pedestrians in the generated environments. We argue that their generated layouts are navigation-aware only if wayfinding signs are properly placed in the layouts.

In this regard, we consider our automatic wayfinding design approach as complementary to automatic layout design or road network generation approaches. The wayfinding signs automatically generated by our approach can enhance the navigation experience of users in virtual environments, as we show in our experiments.

2.3 Overview

Figure 15 shows an overview of our approach. We use a layout called *City* as our illustrative example to describe our approach. Our approach works on a graph representing an input layout. It consists of two major steps: *Wayfinding Scheme Optimization* and *Agent-based Sign Refinement*. In the *Wayfinding Scheme Optimization* step, our approach determines the paths for pedestrians to walk from the starting points to the destinations under different navigation scenarios specified by the user. Different human-centered navigation criteria such as turning angles and walking distances are jointly considered through an optimization to determine the paths to take. In the *Agent-based Sign Refinement* step, our approach places wayfinding signs strategically at appropriate locations along the paths. By using agent-based simulations to evaluate sign placement, our approach takes into account different human properties such as visibility and the possibility of making navigation mistakes. Depending on the requirements of the navigation scenarios, the designer can easily generate a wayfinding design that satisfies the domain-specific requirements, by changing the weights of different criteria in the wayfinding scheme optimization and the parameters of the agent-based simulations.

2.4 Problem Formulation

2.4.1 Representation

Graph Construction. To apply our approach, the user first creates a graph $G = \{V, E\}$ to represent the input layout, where V is the set of nodes representing the intersections, entrances and points-of-interest (POIs), and E is the set of edges representing the connecting paths between adjacent nodes. The creation process is simple and is similar to specifying a waypoint system in typical game level design. The user places nodes at the

intersections, entrances and POIs of the layout. For example, in the illustrative example, *City*, the POIs include the school, the post office and so forth. The user also adds an edge between two adjacent nodes if the places represented by the nodes are connected by a road.

Source-Destination Pairs. A source-destination pair encodes a navigation scenario to be considered by our approach, e.g., going from a bus stop to a restaurant, akin to an input pair a wayfinding designer creates to specify a navigation scenario in conventional wayfinding design [Cal07]. Each pair $z_i = (s_i, d_i)$ consists of a source (starting point) s_i and a destination d_i.

To facilitate the creation of source-destination pairs, by default our approach automatically generates a source-destination pair between every node representing an entrance and every node representing a POI, with the former being the source and the latter being the destination. Additionally, the user can specify any extra pair if needed. For instance, in the *City* example, he may want to create a pair connecting the hotel and the restaurant.

Importance Values. We also allow the user to assign an importance value $\kappa_i \in [0,1]$ to each source-destination pair. For instance, in the *City* example, the (*Hotel*, *Restaurant*) pair can be given a higher importance value if many pedestrians are expected to walk from the *Hotel* to the *Restaurant*, whereas the (*School*, *Restaurant*) pair can be given a lower importance value if fewer pedestrians are expected to walk from the *School* to the *Restaurant*. In the optimization, the path connecting the *Hotel* with the *Restaurant* should be given a higher priority, compared to the path connecting the *School* to the *Restaurant*. If a trade-off exists, it is important to make sure that pedestrians can walk conveniently from the *Hotel* to the *Restaurant*, while it may not matter as much for pedestrians to walk a somewhat inconvenient path from the *School* to the *Restaurant*.

2.5 Wayfinding Scheme Optimization

Given a source-destination pair $z_i = (s_i, d_i)$, there could exist multiple possible paths from s_i to d_i. Let P_{z_i} denote the set of all such paths. Our goal in this step is to generate a wayfinding scheme that takes all source-destination pairs $\{z_i\}$ into account and selects a path for each pair. In other words, we select a path $p_i \in P_{z_i}$ for each pair z_i, such that the set of all selected paths $P = \{p_i\}$ satisfies some local and global criteria defined by our cost terms. We formulate our problem as an optimization of a total cost function:

$$C^{\mathrm{P}}_{\mathrm{all}}(P) = w^{\mathrm{L}}_{\mathrm{local}} C^{\mathrm{L}}_{\mathrm{local}} + w^{\mathrm{N}}_{\mathrm{local}} C^{\mathrm{N}}_{\mathrm{local}} + w^{\mathrm{A}}_{\mathrm{local}} C^{\mathrm{A}}_{\mathrm{local}} + w^{\mathrm{L}}_{\mathrm{global}} C^{\mathrm{L}}_{\mathrm{global}} + w^{\mathrm{N}}_{\mathrm{global}} C^{\mathrm{N}}_{\mathrm{global}} \tag{2.1}$$

The total cost function $C^{\mathrm{P}}_{\mathrm{all}}(P)$ refers to a weighted sum of cost terms encoding the length, number of decision points and the amount of turns of each path, as well as the length and number of decision points of the overall wayfinding scheme. The user can adjust the importance of different design criteria by changing the weights of the corresponding cost terms, to accommodate the domain-specific needs of the layout for which the wayfinding scheme is designed. We describe each cost term in detail as follows.

2.5.1 Wayfinding Cost Terms

Local Path Length. In general, pedestrians prefer to walk a short distance [DS96, Cal07, Fol98]. Hence, for each source-destination pair, a shorter path is preferred. We define a cost to penalize the length of the selected path of each source-destination pair:

$$C^{\mathrm{L}}_{\mathrm{local}}(P) = \frac{1}{|P|L_E} \sum_{p \in P} \kappa_p L(p), \tag{2.2}$$

where $|P|L_E$ is the normalization factor with $|P|$ being the number of source-destination pairs and L_E being the total length of all edges in graph G. $L(p)$ returns the length of path p. $\kappa_p \in [0,1]$ is the importance value assigned to the source-destination pair that path p belongs to.

Local Path Node. The nodes in our formulation correspond to *decision points* in the wayfinding literature [Cal07]. Decision points are locations where pedestrians need to make a decision about which direction to go, such as an intersection between paths (e.g., a lobby in a subway station); or where pedestrians need to confirm the identity of the current location, such as a place of interest (e.g., a platform in a subway station). Directional or identification signs need to be placed at decision points to guide pedestrians to find their directions [Cal07, Ueb10] or identify their current locations. Paths with lots of decision points should be avoided [PA92] as making each navigation decision induces stresses to the pedestrians for the fear of making a wrong decision that may lead to a wrong place [PA92, Pay09]. Therefore we define a cost to penalize the number of decision points of each path:

$$C^{\mathrm{N}}_{\mathrm{local}}(P) = \frac{1}{|P||V|} \sum_{p \in P} \kappa_p N(p), \tag{2.3}$$

where $|P||V|$ is the normalization factor with $|P|$ being the number of source-destination pairs and $|V|$ being the total number of nodes in graph G. $N(p)$ returns the total number of nodes along path p.

Local Path Angle. Research in spatial orientation [Gol99] suggests that paths with varying orientation tend to confuse pedestrians in wayfinding, causing disorientation, anxiety and discomfort [DP14]. A wayfinding scheme composed of straight paths is

more intuitive for navigation [DK03]. We therefore include a cost term to penalize the selection of paths with varying orientation:

$$C^{\mathrm{A}}_{\mathrm{local}}(P) = \frac{1}{|P||V|\pi} \sum_{p \in P} \kappa_p A(p), \tag{2.4}$$

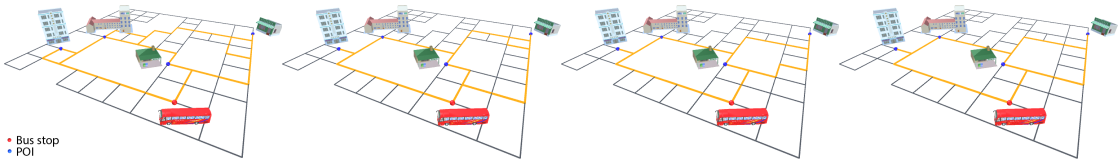
where $|P||V|\pi$ is the normalization factor with $|P|$ being the number of source-destination pairs and $|V|$ being the total number of nodes in graph G. The maximum absolute turning angle between two adjacent edges is π. $A(p)$ returns the sum of absolute turning angles between all adjacent edges along path p.

Global Path Length. Our approach encourages paths to overlap with each other so as to minimize the total length of roads (edges) that are part of a path. This property could be useful from the management's perspective [Cal07, Pas92], because by directing the flow of human movement to fewer roads, fewer roads will need to be maintained, patrolled and lightened up. We define a cost to encourage overlapping paths accordingly:

$$C^{\mathrm{L}}_{\mathrm{global}}(P) = \frac{L(P)}{L_E}, \tag{2.5}$$

where L_E is the total length of all edges in graph G as the normalization factor. $L(P)$ returns the total length of the edges that belong to any path in P.

Global Path Node. Our approach also encourages different paths to share nodes. Similar designs can be observed in the wayfinding schemes of different real-world premises, such as subway stations, shopping malls and concert halls, where people are directed to a lobby or an information desk that can lead to multiple destinations (see Figure 13 for an example). From the management's perspective, it could be easier to maintain signs centralized at certain locations in the environment [Gib09, Ueb10]. Also, centralizing signs could save space, which could be reserved for other better uses [Fol98]. We define a cost to encourage node sharing accordingly:



(a) Initialization (b) Iteration 5,000 (c) Iteration 30,000 (d) Iteration 40,000 (result)

Figure 16: Wayfinding schemes generated over the iterations of an optimization of the illustrative example, *City*. (a) Initialization. The source-destination pairs include walking from the bus stop to each POI, and walking between every pair of POIs. The path of each pair is randomly chosen from its k shortest paths. (b) Iteration 5,000. The selected paths start to overlap. (c) Iteration 30,000. The result is still sub-optimal. For example, the path connecting the bus stop to the restaurant at the upper right still shows large turning angles. (d) Optimized result.

$$C^{\mathrm{N}}_{\mathrm{global}}(P) = \frac{N(P)}{|V|}, \tag{2.6}$$

where $|V|$ is the total number of nodes in graph G as the normalization factor. $N(P)$ returns the total number of nodes that belong to any path in P.

2.5.2 Optimization

For each source-destination pair z_i, there exist a lot of possible paths going from the source to the destination. For instance, pair (*Bus Stop*, *School*) in the illustrative example (Figure 15) has more than $1,000$ possible paths. Given the many combinations of possible paths of all pairs, the solution space could be huge as it grows exponentially with the number of pairs being considered.

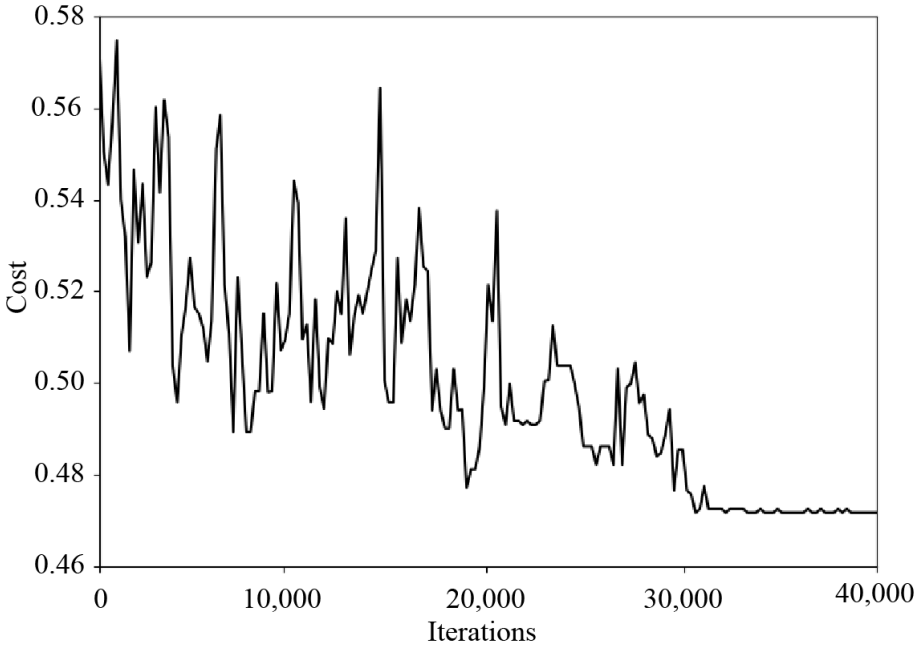


Figure 17: Cost throughout the optimization of the *City* example.

To reduce the search space for a solution, we devise a sampling-based, stochastic search algorithm to solve the optimization problem as follows. For each pair, we only consider the first loopless k shortest paths, which can be found by Yen's algorithm [Yen71] in $O(|E| + |V|\log(|V|))$ time using a Fibonacci heap, where $|E|$ is the number of edges and $|V|$ is the number of nodes. Deviation algorithms [DPS99] and alternative implementations [MP03] exist that could further enhance computational efficiency, yet we adopt the classical implementation for simplicity.

Given the k shortest paths for each source-destination pair, we find a combination of paths of all source-destination pairs which corresponds to a low cost value. Even though we reduce the size of the solution space this way, an exhaustive search for the global optimum would still require heavy computation exponential to the number of pairs being considered.

Instead, our approach finds a local optimum as an approximate solution. We apply the simulated annealing technique [KGV83] with a Metropolis Hasting [MRR53] state-searching step to explore the complex optimization landscape. The optimization proceeds iteratively. In each iteration, the current solution P is altered by a proposed move to another solution P', which may or may not be accepted depending on the acceptance probability of the proposed solution. More specifically, the acceptance probability is calculated by the Metropolis criterion:

$$\Pr(P'|P) = \min(1, e^{\frac{1}{T}(C^{\mathrm{P}}_{\mathrm{all}}(P) - C^{\mathrm{P}}_{\mathrm{all}}(P'))}), \tag{2.7}$$

where T is the temperature of the annealing process. T is high at the beginning of the optimization, allowing the optimizer to explore the solution space more aggressively; T is low towards the end of the optimization, allowing the optimizer to refine the solution. Essentially, the optimizer accepts any solution with a lower cost, while it accepts a solution with a higher cost at a probability: the higher the cost, the lower the acceptance probability. The optimization terminates if the absolute change in cost is less than 1% over $1,000$ iterations.

Figure 16 shows the wayfinding schemes generated over the iterations of the optimization process of the illustrative example. Figure 17 shows the decay in cost over the optimization process. We also experimented with changing the importance values of the source-destination pairs; the resulting wayfinding schemes are depicted in Figure 18.

Proposed Moves. Our proposed moves follow a simple design. Depending on the number of source-destination pairs $|P|$, our optimizer changes the selected paths of up to $|P|$ source-destination pairs in a single move. The probability $\Pr_x$ of drawing a move to change the selected paths of x pairs is inversely proportional to x, i.e., $\Pr_x = \frac{|P|-x+1}{X}$,

where $X = \sum_{i=1}^{|P|} i$. A selected path is randomly changed to another path from the set of k shortest paths of its corresponding source-destination pair.

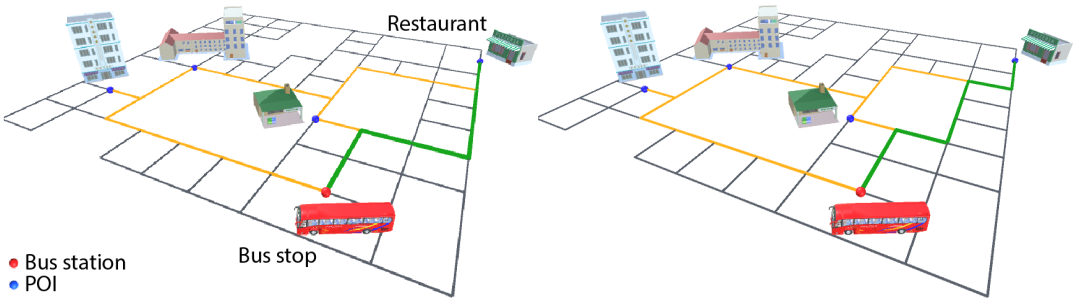
Parameter Settings. In our experiments, we initialize P by randomly selecting a path from one of the k shortest paths for each source-destination pair. By default we adaptively set k such that the length of the k-th shortest path is just within 16% of the length of the first shortest path, as research in spatial cognition finds that humans typically choose a path with a length within 16% of that of the shortest path [DK03]. Unless otherwise specified, each pair is assigned the same importance value $\kappa_i = \frac{1}{|P|}$, and we empirically set the weights $w^{\mathrm{L}}_{\mathrm{local}}$ and $w^{\mathrm{N}}_{\mathrm{local}}$ to 1, $w^{\mathrm{L}}_{\mathrm{global}}$ and $w^{\mathrm{N}}_{\mathrm{global}}$ to 5, and $w^{\mathrm{A}}_{\mathrm{local}}$ to 10. These parameters and weights can be adjusted via the interface of our tool according to domain-specific design needs—a flexibility provided by our optimization-based design framework.

2.6 Agent-based Sign Refinement

The wayfinding scheme optimization in the previous step produces a wayfinding scheme which comprises paths from the sources to the destinations. In this section, we discuss how our approach automatically places signs for each path to facilitate wayfinding.

2.6.1 Overview

Each node along a path corresponds to a decision point where a sign may be placed [Cal07]. In our experiment design, a sign shows an arrow pointing to the next node and the destination's name or symbol. Two or more signs placed at the same node are combined into a single sign showing multiple pieces of wayfinding information. Figure 19 shows an example sign placed at a street corner in the *City* scene.



(a) (b)

Figure 18: Experimenting with different importance values. (a) Wayfinding scheme generated with all source-destination pairs having the same importance value of 0.5. The path from the *Bus Stop* to the *Restaurant* is shown in green. (b) Wayfinding scheme generated with the (*Bus Stop*, *Restaurant*) pair having a lower importance value of 0.25. In this case, the global path length is improved (i.e., the global path length is shorter) while the local path angle of the path from the *Bus Stop* to the *Restaurant* becomes worse (i.e., the path has more orientation changes).

A trivial yet unrealistic solution is to place a sign showing the direction to the destination at every node along a path, so that a pedestrian walking along the path will keep reassured that he is heading to his destination. However, this solution will involve placing many redundant signs occupying a lot of space, and is generally not adopted.

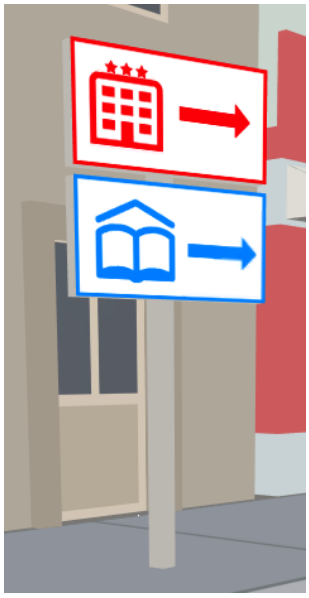


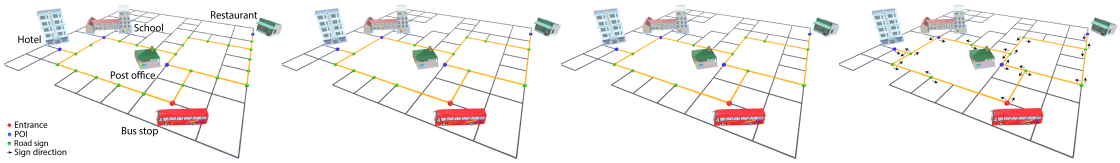
Figure 19: Example sign.

Instead, our approach places signs at strategic locations according to human vision and navigation properties which are evaluated via agent-based simulations. The sign placement process is performed as an optimization against a number of cost terms reflecting the quality of the wayfinding experience brought about by the signs. The optimization starts with the trivial solution of placing a sign at each node along a path, then iteratively alters the sign placement to optimize the costs.

One advantage of this approach compared to the alternative approach of iteratively adding signs from scratch is that the optimization process of this approach is much more tractable, because the initial solution and each intermediate solution represent a feasible wayfinding solution even though they may contain redundant signs and the sign distribution may not be ideal. As we experienced in our experiments, this approach allows the optimizer to progress stably and conservatively to achieve a refined sign placement solution effectively.

2.6.2 Representation

A sign placement solution refers to placing signs at certain nodes of the input layout. Given the path of each source-destination pair (computed from Section 2.5), a good



(a) Initialization (b) Iteration 100 (c) Iteration 500 (d) Iteration 1,000 (result)

Figure 20: Sign placement generated over the iterations of the optimization in the agent-based sign refinement step. (a) Initialization. Road signs are placed at every intersection along the paths generated in the wayfinding scheme optimization step. (b) Iteration 100. Some redundant road signs have been removed. (c) Iteration 500. Redundant road signs along the bottom path have been further removed. (d) Result. The signs on the bottom path are combined into a single sign placed near the middle of the path. The redundant sign on the upper path has been removed.

sign placement solution guides each pedestrian to walk from the source to the destination through the path effectively. Note that there could exist multiple reasonable sign placement solutions. The goal of this step is to locate one of such solutions through an optimization.

In case a road connecting two adjacent nodes is long, we may want to place signs at some intermediate locations along the road to reassure the pedestrian about his walking direction. Therefore, for roads longer than a distance threshold d_{r}, our approach adds extra nodes between the two end nodes of the road in a pre-processing step, such that the distance between any two adjacent nodes is shorter than d_{r}. These extra nodes serve as additional potential locations for placing signs. d_{r} can be empirically set by the designer depending on how frequently a pedestrian should be reassured about his direction. For example, for a subway station, the designer can use a smaller d_{r} such that more signs will be generated along a long road to reassure pedestrians that they are walking towards

a desired destination (e.g., a platform). For our illustration example, *City*, we set d_{r} to be 50 meter.

More specifically, given the graph $G = \{V, E\}$ representing the input layout, we extend V to V' to include the extra nodes added. A sign placement solution is represented by $S = \{(v_i, \phi_i)\}$, where $v_i \in V'$ is the node at which sign ϕ_i is placed. ϕ_i contains the sign's attributes such as its arrow direction and the name of the destination it is referring to. Our optimization searches for a desirable sign placement solution S^* by minimizing a total cost function $C^{\mathrm{S}}_{\mathrm{all}}(S)$:

$$C^{\mathrm{S}}_{\mathrm{all}}(S) = w^{\mathrm{N}}_{\mathrm{sign}} C^{\mathrm{N}}_{\mathrm{sign}} + w^{\mathrm{D}}_{\mathrm{sign}} C^{\mathrm{D}}_{\mathrm{sign}} + w^{\mathrm{F}}_{\mathrm{sign}} C^{\mathrm{F}}_{\mathrm{sign}}, \tag{2.8}$$

where $C^{\mathrm{N}}_{\mathrm{sign}}$ and $C^{\mathrm{D}}_{\mathrm{sign}}$ are regularization costs; $C^{\mathrm{F}}_{\mathrm{sign}}$ is the agent-based simulation cost for estimating the wayfinding failure induced by the sign placement solution S. $w^{\mathrm{N}}_{\mathrm{sign}}$, $w^{\mathrm{D}}_{\mathrm{sign}}$ and $w^{\mathrm{F}}_{\mathrm{sign}}$ are the weights of the cost terms, which are respectively set as 1, 1 and 10 by default.

2.6.3 Sign Placement Cost Terms

Number Of Signs. We include a cost term to regularize the number of signs in the sign placement solution, to penalize the existence of redundant signs:

$$C^{\mathrm{N}}_{\mathrm{sign}}(S) = \frac{N(S)}{|V'|}, \tag{2.9}$$

where $N(S)$ is the number of placed signs; $|V'|$ being the total number of nodes (i.e., potential locations for placing signs) is a normalization constant.

Distribution of Signs. In a real world design, signs are often evenly distributed along a path, which serve the purpose of regularly reassuring a pedestrian about his direction to-

wards the destination. Accordingly, we include a cost term to regularize the distribution of signs:

$$C^{\mathrm{D}}_{\mathrm{sign}}(S) = \frac{1}{|P|} \sum_{p \in P} \frac{\sigma(p)}{L(p)}, \tag{2.10}$$

where $|P|$ is the number of source-destination pairs; $\sigma(p)$ is the standard deviation of the distances between any two adjacency signs on path p, and $L(p)$ is the length of path p.

Wayfinding Failure. The placed signs should effectively guide the pedestrians from the sources to the destinations. We include a cost term to penalize wayfinding failure:

$$C^{\mathrm{F}}_{\mathrm{sign}}(S) = \begin{cases} F(S) & \text{if } F(S) \leq \mu, \\ +\infty & \text{otherwise}, \end{cases} \tag{2.11}$$

where $F(S)$ is the percentage of agents who fail to reach their destinations under the current sign placement S. $F(S)$ is obtained by performing an agent-based simulation with sign placement S. μ is a failure tolerance level specified by the designer, which is set as 20% by default.

2.6.4 Agent-based Evaluation

In each iteration of the optimization, we employ an agent-based simulation to evaluate the wayfinding experience under the current sign placement S, to obtain $F(S)$ used for computing the wayfinding failure cost.

Agent Model. Each agent mimics a pedestrian walking from a source to a destination. We model each agent with wayfinding behavior according to Montello and Sas [MS06]. The agent starts from the source. It can see any unoccluded sign within visible distance d_{v}. Whenever it sees a sign pointing to its destination, it will follow the sign to choose

a direction to walk. If it arrives at an intersection but is unsure about which road to take out of several roads connected to that intersection, it will randomly choose a road to walk with equal probability. To more realistically model mistakes that humans can make throughout a navigation, each agent has a probability $\mathrm{Pr}_{\mathrm{miss}}$ of missing a sign even within sight.

Simulation. For each source-destination pair, 100 agents are employed to walk from the source to the destination using the agent model described. At the end of the simulation, we count the number of agents that can successfully reach their destinations, and hence compute $F(S)$.

A "success" is defined as follows: let d_{b} be the "baseline" walking distance from the source to the destination if no mistake is made (i.e., $\mathrm{Pr}_{\mathrm{miss}} = 0$) under full sign placement (i.e., a sign is placed at every node along a path). If an agent, given the chances of making mistakes and under the current sign placement S, can walk from the source to the destination by a distance no longer than λd_{b}, the navigation is considered as a success. The navigation is counted as a failure otherwise. We use $\lambda = 1.5$ in our experiments.

2.6.5 Sign Refinement by Optimization

Initialization. Our optimization is initialized with the full sign placement solution, i.e., a sign is placed at every node along the path from the source to the destination of each source-destination pair. Although this sign placement can lead the pedestrians to their destinations, it consists of a lot of redundant signs that could be removed without affecting the pedestrians' ability to find their ways. We apply a stochastic, agent-based optimization to search for a reasonable sign placement solution.

Iterative Refinement. Our optimization proceeds iteratively. At each iteration, a move is randomly proposed to alter the sign placement solution whose quality is evaluated using the total cost function $C^{\mathrm{S}}_{\mathrm{all}}(S)$. The moves include:

- Adding 1 or 2 signs to 1 or 2 source-destination pairs.
- Removing 1 or 2 signs from 1 or 2 pairs.
- Moving a sign from one node to another node of a source-destination pair.

The proposed solution is accepted with an acceptance probability determined by the Metropolis criterion as described byEquation 2.7, using $C^{\mathrm{S}}_{\mathrm{all}}(S)$ as the cost function. The optimization terminates if the absolute change in cost is less than 1% over 50 iterations.

Figure 20 shows the sign placement over iterations for the illustrative example. In this example, the source-destination pairs include walking from the *Bus Stop* to each POI, and walking between every pair of POIs. Each iteration of the optimization takes about 0.01 second to finish in our experiments. It takes about $1,000$ iterations (about 10 seconds) to finish the sign placement optimization for this example.

2.7 Experiments and Results

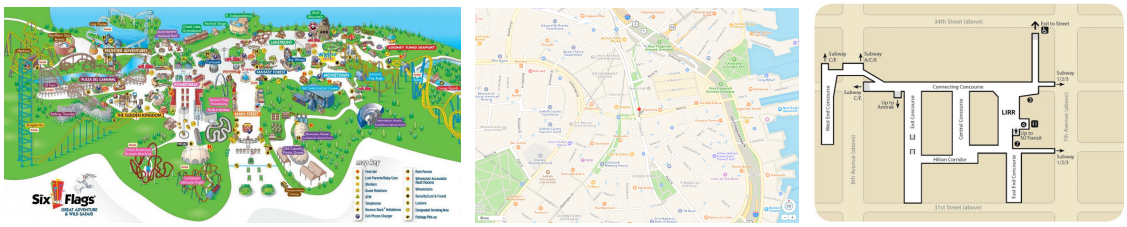
We implemented our approach as a plugin for the Unity 5 game engine using C#, which level designers can use to create a wayfinding scheme of a given layout. We run our experiments using a Macintosh machine equipped with a 2.3 GHz Intel Core i7 processor and 8GB of RAM. Generating a wayfinding scheme for a layout similar to the illustrative example, *City*, takes about 40 seconds using our current implementation.

2.7.1 Different Layouts

We used our approach to generate wayfinding designs for different layouts: *Amusement Park*, *Downtown* and *Penn Station*. Figure 21 shows the maps from which the layouts are extracted following the procedure in Section 2.4.1. Figure 22 shows the wayfinding designs generated by our approach. We describe the details of each generation in the following. Please also refer to the supplementary material for details of the generated wayfinding schemes of the layouts, and also for the results of two more layouts: *City* and *Canyon*, which demonstrate how our approach could be applied to generate wayfinding designs for 3D virtual environments and with robustness as a key consideration.

Amusement Park. We use the layout of an amusement park, Six Flags New England, as input (see Figure 22). The POIs, which in this case are the Entrance, Sky Screamer, Bonsai, Batman and Picnic Grove, represent the popular spots the visitors would like to visit. The source-destination pairs involve all pairs of entrances and popular spots, and all pairs of popular spots. In addition to the popular spots, we expect there are street performances and stalls in the park, which might distract visitors from navigating to their $\mathrm{Pr}_{\mathrm{miss}}$ in the agent simulation of the sign placement optimization step to a relatively high level of 0.2. Besides, we assume that visitors highly prefer to walk shorter and more direct paths to their destinations if possible, therefore we use larger values of 5 for the weights $w^{\mathrm{L}}_{\mathrm{local}}$ and $w^{\mathrm{N}}_{\mathrm{local}}$ of the local path length and local path node costs.

Figure 22 shows the generated design. Our approach generates a path for each source-destination pair. It places road signs densely at each intersection along the paths to ensure the robustness of the wayfinding system. The left-hand side of the layout shows a shortcut generated which is part of the paths from the popular spot at the lower left to the popular spots on the right. The shortcut allows visitors to walk shorter paths



Amusement Park Downtown Penn Station

Figure 21: Maps and 3D models from which the layouts are extracted.

to their destinations, and is also more direct for the visitors as it passes through fewer intersections (2 instead of 3) compared to the alternative path above.

Downtown. This example uses the layout of Downtown Boston as input. The goal is to place road signs that guide drivers to an available parking lot nearby. The entrances refer to the major roads through which most cars enter the Downtown area. The POIs are the parking lots, which are placed at the same locations as the real parking lots found on Google map. We suppose that all the parking lots are run by the same company, hence there are signs showing the way from one parking lot to a nearby parking lot within 0.2 mile, such that if a parking lot is full the driver can follow the signs to a nearby parking lot. Accordingly, we define the source-destination pairs to connect each entrance to its nearest parking lot, as well as to connect each parking lot to its nearby parking lot. The latter type of pairs are given a relatively larger importance value ($\kappa_p = 0.8$ instead of 0.5 given to other pairs), so that our system will prefer to find shorter paths passing through fewer intersections for the paths that connect one parking lot to a nearby parking lot, to help drivers to get to an alternative parking lot more easily in case a parking lot is full.

Figure 22 shows the generated design. A path is generated to connect each entrance to its nearest parking lot. Short and direct paths are also generated to connect parking lots to nearby parking lots. While there are many possible paths that can be chosen as

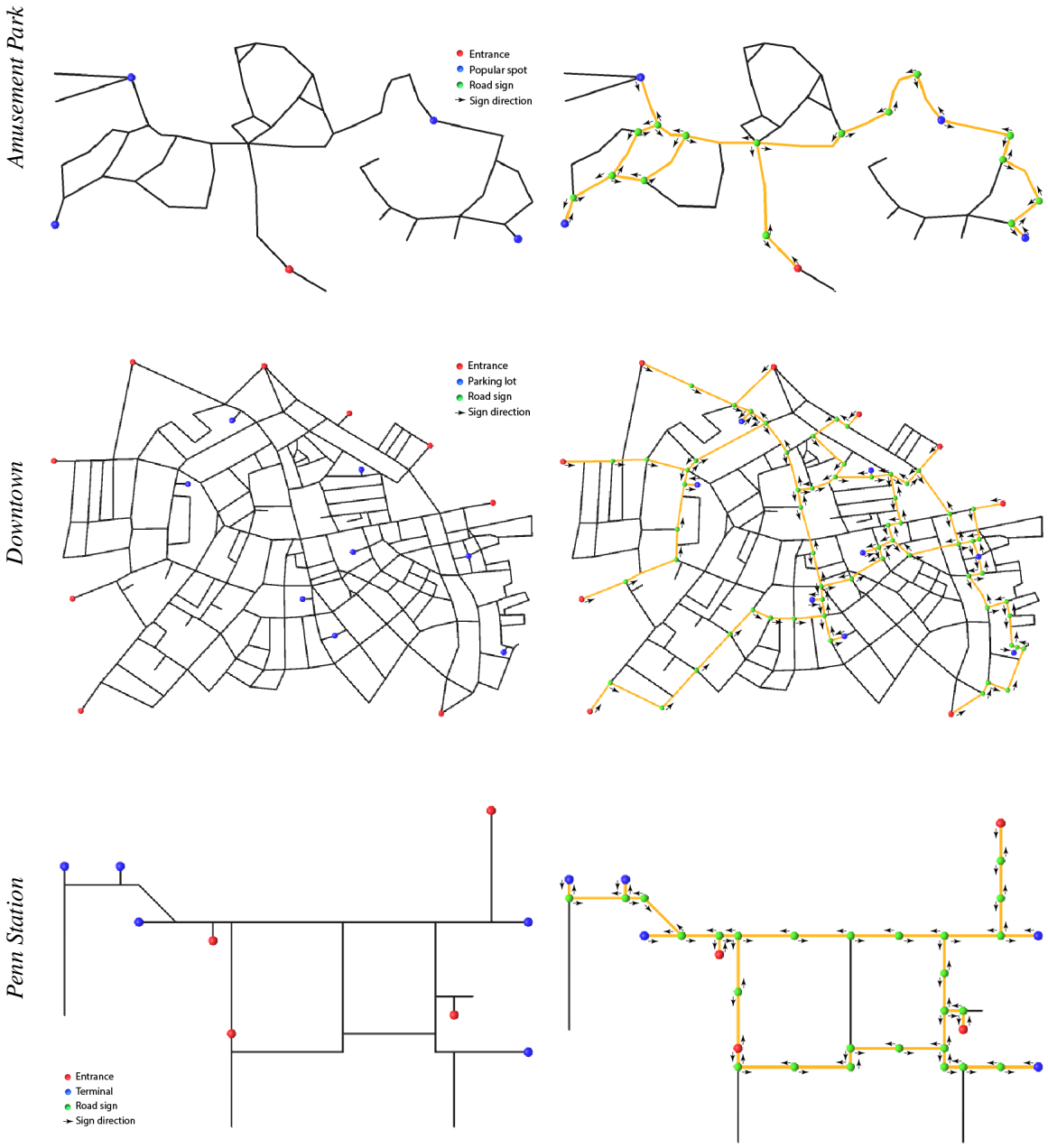
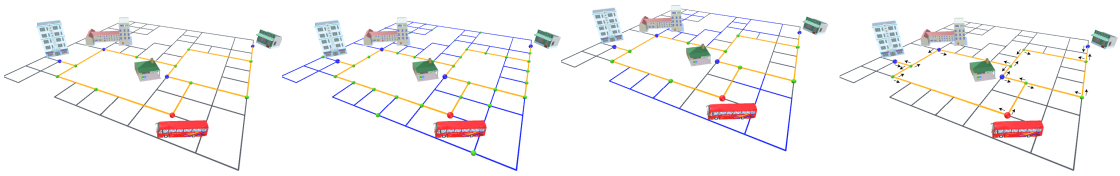


Figure 22: Wayfinding designs generated for *Amusement Park*, *Downtown* and *Penn Station*.



(a) Default Parameters (b) High Miss Chance (c) Low Wayfinding Failure Weight (d) High Visibility

Figure 23: Effects of changing the parameters of the agent-based sign placement step. (a) Default parameters. Roads that belong to a path of the wayfinding scheme are shown in yellow. (b) High miss chance. The blue lines indicate the roads that the pedestrians have walked but do not belong to any path. The generated wayfinding design is more robust against navigation mistakes. Signs are placed densely, and are also placed on some roads not belonging to any path to guide the pedestrians back to the correct paths in case they get lost. (c) Low wayfinding failure weight. Fewer signs are placed as wayfinding failure is more tolerable. (d) High visibility. Fewer signs are placed as the pedestrians can see signs far away.

the layout comprises a network of many streets, our approach chooses paths which are straight and consists of few turns, as the local path node cost penalizes the inclusion of intersections and the local path angle cost discourages orientation changes.

Penn Station. This example uses the lower level of the Penn Station as input. In this example, the entrances refer to the gates and the stairs from the upper level. The POIs refer to the terminals. The source-destination pairs include every pair of entrance and terminal, and every pair of terminals (for modeling the situations where a passenger wants to transfer from one terminal to another terminal). As the station is expected to be crowded, the visibility d_{v} of the agents is set to a relatively low value of 10 meters to account for the occlusion by human crowd, and the miss chance $\mathrm{Pr}_{\mathrm{miss}}$ is set to a relatively high level of 0.2. Figure 22 shows the generated wayfinding design. The road signs are placed densely and are also placed at non-intersection nodes, to counteract the higher miss chance by reassuring pedestrians about their directions.

2.7.2 Changing Agent Parameters

We further experimented with changing the parameters of the agent-based sign placement process using the *City* layout. In the default settings, the missing chance $\mathrm{Pr}_{\mathrm{miss}}$ is set to 0%, the weight $w^{\mathrm{F}}_{\mathrm{sign}}$ of the wayfinding failure cost term is set to 10 and the visibility distance d_{d} is set to 125 meters. Figure 23(a) shows the resulting sign placement generated with the default parameters.

We experimented with increasing the missing chance $\mathrm{Pr}_{\mathrm{miss}}$ to 10%. Figure 23(b) shows the resulting sign placement. Our system places more signs so as to increase the robustness of the wayfinding design against navigation mistakes. In addition, some signs are placed on the roads not belonging to any path for guiding the agents back to the correct paths.

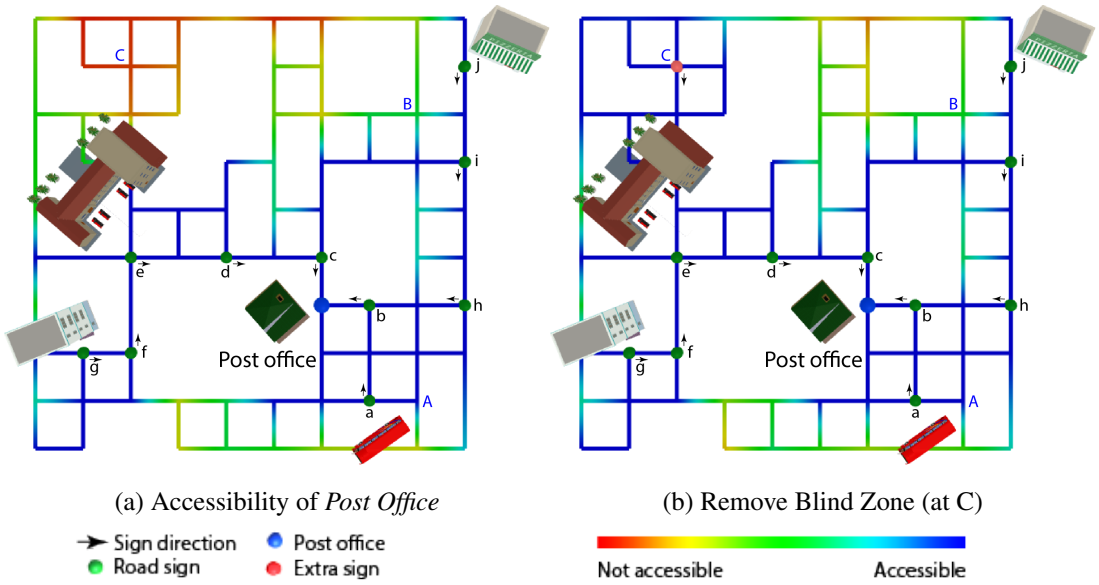
Next, we experimented with lowering the weight of the wayfinding failure cost term to 0.01. Figure 23(c) shows the resulting sign placement. Our system keeps fewer signs, because it is acceptable even if some agents make mistakes and do not walk to the destination within a desired period of walking. The pedestrians walk along some roads (shown in blue) not belonging to any path. This setting maybe useful for some situation where it is not critical for the agents to reach the destination, and when space is better preserved for other uses. For example, in a flea market, it may not be critical for the pedestrians to visit each stall as they are expected to wander around in the market.

Finally, we experimented with increasing the visibility to 250 meters. Figure 23(d) shows the result. Our system keeps only a fewer signs because the pedestrians are capable of seeing signs at a farther distance. This setting is useful for modeling situations where the signs are big (such as those shown in billboards) and can be seen far away.

2.7.3 Visualization

Destination Accessibility. Our approach also allows the designer to visualize the accessibility of a destination under the generated wayfinding design. This is a very useful functionality that can help the designer to create a wayfinding design that guides pedestrians from different locations to walk to a destination as desired. Figure 24(a) depicts this functionality. The accessibility of a destination (the *Post Office*) is visualized as a heatmap. Agents in the blue region can travel to the *Post Office* successfully by following the wayfinding signs under the current wayfinding design; while those in the red region have a low chance of success.

To compute the accessibility heatmap with respect to a destination specified by the designer, our system sample points at regular intervals along all the edges of the input layout (whether the edges are part of the paths of the generated wayfinding design or



(a) Accessibility of *Post Office*

(b) Remove Blind Zone (at C)

Figure 24: (a) Accessibility of the *Post Office* visualized as a heatmap. The accessibility from location A is good as depicted by the blue color, because a pedestrian can easily see and follow a sign on the west to walk to the *Post Office*. The accessibility from location B is fair as depicted by the green color. A pedestrian starting at B has a 50% chance to walk to the street towards the north or the east and get lost. The accessibility from location C is poor as depicted by the red color. Without any sign nearby, a pedestrian starting at C has a very low chance to reach the *Post Office*. (b) Removing the blind zone at C. The user triggers our system to place a road sign at C showing the direction to the *Post Office*. The accessibility around C is improved, as depicted by the change in color from red to blue on the updated heatmap.

not). Agents are employed to walk from each sample point to the destination, in a similar fashion as in the agent-based sign placement step (Section 2.6), to compute the rate of success of reaching the specified destination from each of the sample points. The rates of success are used to set the heatmap colors at the corresponding sample points; the heatmap color values between two sample points are interpolated.

Note that a destination typically does not need to be accessible from every region, because enforcing such full accessibility will likely involve placing a lot of signs even at some "unimportant" regions. For example, it may not be important to place signs to guide pedestrians how to walk from a post office to a restaurant. By visualizing the accessibility to a destination using a heatmap, the designer can intuitively tell what regions are covered by the current wayfinding design and if any improvement is needed.

Removing Blind Zones. If the designer wants to remove a "blind zone" (i.e., a region shown in red indicating low accessibility to the destination), he can easily do so by clicking on the red region via our user interface. Our system will automatically place signs which guide pedestrians to walk from the clicked point to the path leading to the destination. This is achieved by re-running the wayfinding scheme optimization and the agent-based sign refinement steps taking the existing solution as the initialization. Note that these additional optimization steps only change the paths and signs related to the newly-added POI; the locations of the signs and paths in the existing solution are fixed. After the optimization steps, agent-based evaluations will be re-run at each sample point to update the heatmap accordingly, which takes about 1 second for the *City* example. Figure 24(b) shows an example of removing a blind zone.

2.8 Evaluation

2.8.1 User Study

Conditions. We conducted a user study to evaluate the effectiveness of the wayfinding designs generated by our approach. Our user study was conducted in the *City* layout used as the illustrative example. Participants were asked to navigate from a starting point to a destination under 4 different wayfinding conditions:

1. No sign.

2. Mini-map. A mini-map that functions like a mini-map in a common first-person 3D video game is shown;

3. Full signs. In this case, we only run the wayfinding scheme optimization step to generate the paths for the source-destination pairs. Signs are placed at every node along each path.

4. Refined signs. In this case, we run the wayfinding scheme optimization step to generate the paths, and then the agent-based sign refinement step to refine the sign placement. Signs are placed strategically at some of the nodes along each path.

Figure 25 shows two screenshots of the user study tests under the mini-map and refined sign conditions. There are 2 different scenarios. In the first scenario, the participant was asked to walk from the *Bus Stop* to the *Restaurant*. In the second scenario, the participant was asked to walk from the *Bus Stop* to the *School*. Each scenario was tested by 80 participants under the 4 different wayfinding conditions (i.e., 20 participants for each condition).

Participants. In total, we recruited 160 participants through social networks. The participants are university students. All of them have experience with 3D video games and are familiar with the movement control of common first-person-shooting games, which our user study program similarly adopts. Before each test, a description of the task and the movement control is shown to the participant, and the participant is allowed to get familiar with the movement control in a warm-up session.

Test Sessions. The goal of the participant in each test is to walk to the destination (*School* or *Restaurant*) as fast as he can. To make sure he is clear about the destination,

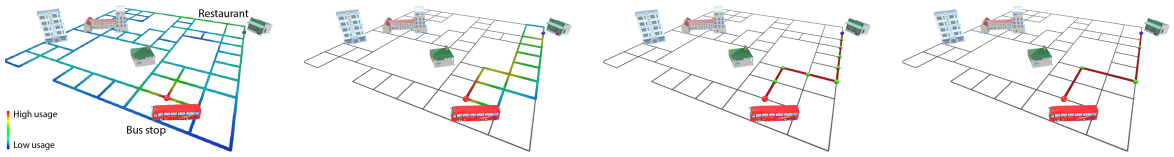
(a) Mini-map (b) Refined signs

Figure 25: Screenshots of the user study tests under the (a) mini-map and (b) refined signs conditions.

a screenshot of the destination is also shown to the participant before the user study begins. Our program records the path, the distance walked and the time taken by the participant. The test ends if the participant reaches the destination, or if the time taken exceeds the time limit, which is defined as three times the time needed to walk from the start to the destination without any stop following the path generated by the wayfinding scheme optimization step. The latter is considered as a failure case.

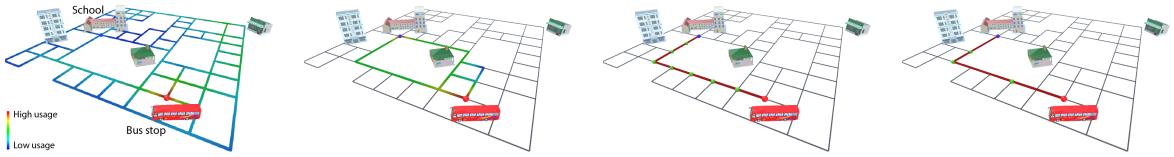
2.8.2 Results and Analysis

Path Taken. Figure 26 shows the results of the user study. The paths taken by the participants are visualized in a heatmap. The roads with high usage are shown in red, and those with low usage are shown in blue. There are some interesting observations. Under the no sign condition, the participants wandered around and could barely reach the destination. Under the mini-map condition, the participants walked towards their destinations along similar directions. However, there are considerable variations among the paths taken, as can be seen from the color dispersion on the heatmaps. For example, in scenario 2, near half of the participants took the bottom path while the other half took



(a) No sign (b) Mini-map (c) Full signs (d) Refined signs

Scenario 1 (*Bus Stop* to *Restaurant*)



(a) No sign (b) Mini-map (c) Full signs (d) Refined signs

Scenario 2 (*Bus Stop* to *School*)

Figure 26: Paths taken by the participants of the user study under different wayfinding conditions. Usage of each road is shown by its color (roads which are not walked are shown in gray). (a) With no sign, participants wandered randomly and could hardly reach the destination. (b) With a mini-map, participants walked along a similar direction towards their destinations, but the paths they took varied. (c,d) With full signs or refined signs, all participants followed the same paths to walk to their destinations. The refined sign placement is as effective as the full sign placement, but uses a significantly smaller number of signs.

the upper path. Under the full signs and refined signs conditions, all participants walked to the destinations following the same path.

Distance Walked. Table 1 shows the statistics of the distances walked by the participant under different conditions. For the no sign condition, only the data of the participants who could reach their destinations within the time limit is used to calculate the statistics. For the other conditions, all participants can reach the destinations and all data is used to calculate the statistics.

Under the no sign condition, only 55% and 25% of the participants could reach their destinations in Scenario 1 and 2 respectively. For those who could reach the destinations, they generally needed to walk a very long way as shown by the large mean values.

Under the mini-map condition, all participants could reach their destinations. In Scenario 1, the participants could reach the destination *Restaurant* by walking a distance similar to that in other conditions. However, the standard deviation (35.81m) is higher than the standard deviations (11.45m and 11.95m) of the other conditions, showing that there are larger variations in performances, due to different paths chosen as shown in Figure 26. In Scenario 2, the relative difference in standard deviation is even more pronounced (55.27m under the mini-map condition, versus 6.11m and 7.38m under the other conditions), due to the larger differences in walking distances of the paths chosen. In average, the participants walked a shorter distance to reach the destination under the full signs or refined signs conditions (471.37m and 476.32m) than under the mini-map condition (503.32m).

Under the full signs and refined signs conditions, all participants can reach their destinations. The means and standard deviations of the walked distances are similar. This shows that the refined sign placement is as effective as the full sign placement in guiding the participants to their destinations. However the refined sign placement uses

significantly fewer signs (3 signs under refined sign placement versus 8 signs under full sign placement in Scenario 1; and 3 signs under refined sign placement versus 7 signs under full sign placement in Scenario 2). Please refer to our supplemental material for the user study results and a video showing example sessions.

2.9 Summary

We verify in our experiments that our approach can be applied to automatically generate wayfinding designs for a variety of layouts, and that the designs can be used by human users to navigate to their destinations effectively in virtual worlds. Compared to the conventional approach of creating wayfinding designs manually, the novelty of our approach lies in formulating the problem as an optimization, which can be solved automatically and efficiently, hence overcoming the design challenge posed by the considerations of multiple paths and design criteria. Our optimization approach also allows the flexibility of considering additional constraints in wayfinding design and the designer can trade off between different criteria by controlling their corresponding weights. We adopt an agent-based approach to automatically place signs at strategic locations, considering human perception and navigation properties such as eyesight and the possibilities of making mistakes. The agent model makes it intuitive and flexible for designers to define agent properties and behaviors according to the specific requirements of their design projects on hand; signs will be automatically placed according to the specified agent properties.

2.10 Limitations and Future Work

Our approach only focuses on placing textual and arrow signs to facilitate wayfinding. While these are common wayfinding aids, in reality humans also make use of other wayfinding aids and cues such as maps (e.g., "You-are-here" maps [Gib09, Cal07]), landmarks and flow of people movement to determine directions. In future extension it would be useful to consider all these alternative aids and cues in generating a wayfinding design.

Our agent-based simulation model only focuses on a few properties that are relevant to wayfinding. More realistic virtual humans comprising of cognitive, perceptive, behavioral and kinematic modules, similar to the autonomous agents used for artificial life simulation [ST05], could be used to replace our agents. The perceptual data obtained from the simulations based on such agents could be used for more sophisticated wayfinding analysis to enhance the computationally-generated wayfinding design.

In our current approach, for simplicity we only consider one path for each source-destination pair. In fact, there could exist multiple paths (secondary paths) for each pair. This can be modeled by extending our framework to allow multiple paths for each pair, which will be considered jointly in the optimization.

In our approach, the source-destination pairs are manually specified rather than automatically generated. This is because our approach does not infer the layout context. An interesting future direction is to devise a data-driven approach to automatically identify the possible locations of interests given a layout based on prior statistics of human flows, and hence automatically suggest the source-destination pairs to consider. For example, given a subway station, a data-driven approach may automatically suggest that (*Entrance*, *Ticket Machine*) and (*Ticket Machine*, *Gate*) as likely source-destination pairs, based on the real-world statistics of human flows in subway stations.

2.11 Acknowledgements

We thank Ana Aravena for narrating the demonstration video. This research is supported by the UMass Boston StartUp Grant P20150000029280 and by the Joseph P. Healey Research Grant Program provided by the Office of the Vice Provost for Research and Strategic Initiatives & Dean of Graduate Studies of UMass Boston. This research is also supported by the National Science Foundation under award number 1565978. We acknowledge NVIDIA Corporation for graphics card donation.

Condition	Dist.(m)	SD(m)	Time(s)	Success	#Signs
No sign	1,128.28	361.22	54.19	55.00%	-
Mini-map	515.54	35.81	34.05	100.00%	-
Full signs	514.21	11.45	30.72	100.00%	8
Refined signs	517.83	11.95	27.59	100.00%	3

(a) Scenario 1 (*Bus Stop* to *Restaurant*)

Condition	Dist.(m)	SD(m)	Time(s)	Success	#Signs
No sign	1,205.67	187.51	53.43	25.00%	-
Mini-map	503.32	55.27	31.18	100.00%	-
Full signs	471.37	6.11	29.09	100.00%	7
Refined signs	476.32	7.38	25.25	100.00%	3

(b) Scenario 2 (*Bus Stop* to *School*)

Table 1: Average distances walked and average time taken by participants under different wayfinding conditions. For the no sign condition, only the data of the participants who successfully reached their destinations are used to compute the statistics. In general, it took the least amount of time for participants to reach their destinations under the "refined signs" condition. Compared to the "full signs" condition, users under the "refined signs" condition probably needed to spend less time looking for and reading signs. Compared to the "Mini-map" condition, users under the "refined signs" condition could probably determine the correct direction more intuitively and quickly.

CHAPTER 3

AUDIBLE PANORAMA: AUTOMATIC SPATIAL AUDIO GENERATION FOR PANORAMA IMAGERY

3.1 Introduction

Sound has been demonstrated to be an integral aspect of immersion [ZF15, KLD09], so it is no surprise that there have been numerous attempts to produce realistic sound for 3D environments [BT00, DN07].

Given a 360° panorama image, we present an approach to create a realistic and immersive audible companion.

We start by detecting the overall scene type as well as all the individual sounding objects. Leveraging scene classification and object recognition, we then assign a background audio of the scene and customized audios for each object.

We use object and scene tags assigned during scene classification and object recognition for audio assignment. These tags are matched with audio tags from our audio file database and audio associated with the audio tags are assigned as audio sources. We estimate the depths of detected objects by comparing relative heights of objects and

pre-established knowledge on the average heights of different types of objects.

We have three major contributions.

- We proposed a new framework for automatically assigning realistic spatial sounds to 360° panorama images based on object recognition and depth estimation.
- We constructed a large dataset of panorama images and audio files. The panorama images are made audible by running our approach on them. The dataset, the results obtained by running our approach, as well as the tool for experiencing the audible panorama will be publicly released for research purposes.
- We conducted statistical analysis that evaluates the importance of various factors in our proposed framework.

3.2 Related Work

As the camera hardware for 360° contents are significantly improving, there is an increasing amount of interests to facilitate better interaction with the 360° contents. However, most existing work focused on the visual aspect, for example, sharing content playback [TF17], improving educational teaching [APO17], assisting visual focus [LCH17], augmenting storytelling [PDS17, GWH16], controlling interactive cinematography [PHA17], enhancing depth perception [KJ16], and collaborative reviewing [NDH17].

On the audio side, Finnegan et al. made use of audio perception to compress the virtual space, in addition to conventional visual-only approaches [FOP16]. Rogers et al. designed and conducted a user study on sound and virtual reality (VR) in games, exploring the player experience influenced by various factors [RRW18]. Schoop et al.

proposed HindSight that can detect objects in real-time, therefore greatly enhancing the awareness and safety with notifications [SSH18]. While previous methods all rely on accompanying audio signals, our project tries to enable better interactive experience in 360° images by synthesizing realistic spatial audio from only visual content. We achieve this by constructing a comprehensive audio dataset which we will discuss later.

To enhance the sense of presence enabled by immersive virtual environments, high-quality spatial audio that can convey a realistic spatial auditory perception is important [Ste92, HB96, BT00, BM07]. Many researchers have studied realistic computer-generated sounds. One widely researched excitation mechanism is rigid body sound [OSG02]. To model the sound propagation process, efficient wave-based [RSM10] and ray-based simulation methods [OOK12] have been proposed. More closely related to our method are scene-aware audio for 360° videos [LLZ18] and automatic mono-to-ambisonic audio generation [MVL18], both of which require audio as part of the input. We draw inspirations from existing sound simulation algorithms and synthesize plausible spatial audio based on only visual information without any audio input to our system.

In existing virtual sound simulations, scene information, such as the number of objects, their positions, and motions, is assumed to be known. However, in our project, we compute this information automatically through object detection and recognition. In computer vision, robust object detection has been a grand challenge for the past decades. Early work detects objects rapidly, for example, human faces, using carefully designed features [VJ01]. Recently, more robust methods leveraging deep neural networks have been shown to achieve high accuracy [HZR16, HRS17]. We match the scene in a panorama image with an audio file, but also detect and recognize objects in the image and estimate their depth to place audio sources for those objects at convincing positions.

Since the production of sound is a dynamic process, we need to classify not only the objects, but also their movements, their actions, and their interaction with the surroundings. For example, a running pedestrian and a pedestrian talking on the phone should produce different sounds in the audible panorama. To this end, accurate action recognition can guide the generation of more realistic sounds. Most existing action analysis research requires video as input since the extra temporal information provides strong hints as to what certain human actions are [ZXW17, TKR17]. Human action recognition on still images remains a challenging task. One approach is to use word embeddings and natural language descriptions to recognize actions [SKB17].

Traditional sound texture synthesis can generate credible environmental sounds, such as wind, crowds, and traffic noise. Harnessing the temporal details of sounds using time-averaged statistics, McDermott et al. demonstrated synthesizing realistic sounds that capture perceptually important information [MSS13]. In addition to sound texture, natural reverberation also plays an important role in the perception of sound and space [TM16]. An accurate reverberation conveys the acoustic characteristics of real-world environments. However, in these ambient sound synthesis methods, the spatial information is missing since the environment is treated as a diffuse audio source. We build upon these existing ideas and augment panorama images with spatial audios.

3.3 Overview

Figure 27 depicts an overview of our approach. Our method takes a 360° image as input. After scene classification, object detection, and action recognition, the image is labeled with what we will call here on out a background tag, which matches the type of scene in the image (for example, ”Town”), and also the objects are labeled with object tags. Ob-

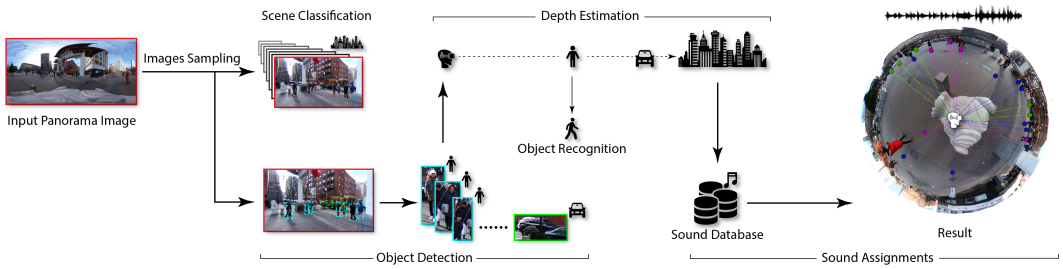


Figure 27: Overview of our approach. Given a panorama image, our approach performs scene classification and object detection on images sampled horizontally from the panorama image. Next, it performs object recognition. Based on the scene knowledge, it assigns a corresponding background sound for the scene; it also places appropriate sound sources at the estimated object locations accordingly.

ject tags either are object names or, if the object detected is a person, words for actions.

To start with, we built an audio file database. These files are organized into two types: background and object audio files. Each audio file is associated with an audio tag, which we set as the object tags for object audio files and scene tags for background audio files.

Our approach then estimates, with a single user interaction for inputting the depth of one object in the scene, the depth and hence the 3D location for each of the detected objects by using estimates of the real-life average height of objects and the relative height of objects recognized in the scene. Based on the detection and recognition results, our approach assigns appropriate audio sources at the calculated depth for each object by comparing the object tags with the audio tags in the database. If there is a match between the tags, we randomly select an audio file from our database that is labeled with the matching audio tag. These audio tags are effectively the different categories of sound that we have in our database.

For getting the audio file for the background audio of the scene, we use the same approach except that we use the scene tag instead of the object tags. The output of our approach is an audible panorama image, which can then be experienced using a virtual reality headset.

3.4 Approach

3.4.1 Sound Database

To ensure that our sound database is comprehensive, we select the sounding objects based on 1,305 different panorama images found on the internet. By running scene classification and object detection and recognition on these panorama images we were able to detect repeatedly occurring objects and scene types, which we also use as the corresponding object and scene tags. We then set the categories for all the sounds and build a database with background and single audio files. The audio files represent background sounds, human actions, sounds of motor vehicles, etc. Our sound database constitutes a total of 512 different audio sources, which are organized into two types: audio sources for objects and sounds for background ambient scenes:

- Object Sounds: There are 288 different object sounds, which include human chatting and walking, vehicle engine sounds, animal yelling and others in 23 categories. Each category, which matches previously mentioned object tags, is used for audio source assignment for objects in the scene. We normalize the volume of all sounds and generate versions of the audio files for various configurations.
- Background Audio: There are 224 different background audio files in total. We catalog these using 22 tags that match the different scene tags such as ”City”,

Tag	Type	#	Tag	Type	#
City	Bg	14	Car	Obj	6
Crowd	Bg	12	Cat	Obj	2
Library	Bg	12	Chatting	Obj	17
Room	Bg	15	Dog	Obj	13
Sea	Bg	12	Walking	Obj	2

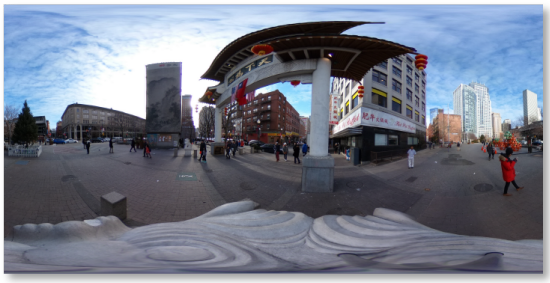
Table 2: A partial list of the audio tags used in our database. Each audio tag is a label for audio files in our database. So the ”Chatting” tag, for example, tags different files for audio of people chatting. The type refers to audio files either being for background audio (*bg*) or object audio (*obj*). Refer to our supplementary materials for a complete list of all audio tags. Our database contains a total of 512 audio files.

”Room”, ”Sea”, etc. These audio tags are used for selecting the audio source for the background based on the scene tag.

For a list of representative audio tags, refer to Table 2. A complete list can be found in our supplementary material. All audio files were obtained from the website ”freesound.org” [FRS13]. To generate 3D spatial audio from a 2D image, we estimate the relative depth of different objects in order to place the sounds reasonably in 3D space to simulate spatial audio.

3.4.2 Scene Classification

The goal of scene classification is to assign a scene tag to the scene, which matches one of the background audio tags in our audio database. Beginning with a panorama image and a camera viewing horizontally from the center of the rendered image, we rotate the



(a) A representative panorama image used for the scene before being rendered in our viewer.

(b) A sample image and the corresponding tags assigned by our automatic scene classifier. In this case, some scene tags assigned were "Crowd", "Transport", and "Town". "Town" was ultimately the highest scored and most frequently occurring tag across all tested image segments, so it was selected as the scene tag for the *Chinatown* scene.

Figure 28: An illustrative example, *Chinatown*.

viewpoint horizontally 36° to capture different segments of the image.

Our system also splits the captured samples vertically. If desired, the user may increase the number of samples adaptively to provide more input data for scene classification. Each segment is assigned a list of five scene categories, which we use as the scene tags, ranked in decreasing order of classification confidence scores.

We combine the classification results on all slices and select the most frequently-occurring, highest-scored tag as the tag for the overall scene for the panorama image. So, for example, that the two most common occurring scene tags for an image are ”Library” and ”Living Room”, and the confidence score of each is 0.8 and 0.92 respectively, then ”Living Room” will be selected as the scene tag. Refer to Figure 28(b) for an example.

3.4.3 Object Detection and Recognition

We use TensorFlow Object Detection, which is based on a Convolutional Neural Network (CNN) model pre-trained on the COCO dataset [LMB14]. Our approach slices the panorama image the same as in scene classification, and we run object detection on each slice. If there are any overlapping objects from one slice to the next, we count the detected objects as the same object. Once objects have been detected, we use Google Vision for object recognition, feeding in cropped images of all detected objects to the object recognition algorithm. We assign object tags to the objects in the same way that we tag the scene. Figure 29 depicts the object detection process.

3.4.4 Depth Estimation

Our depth estimation technique requires comparing two objects detected on a panorama image. The first object is a reference object r which we use as a baseline to estimate the depths of all other detected objects in the image. The depth of the reference object, d_r, is set by the user. This is the only user interaction required during the depth estimation process. By default, the reference object is chosen as the object with the highest score (which corresponds to the highest confidence) of running object recognition on the image.

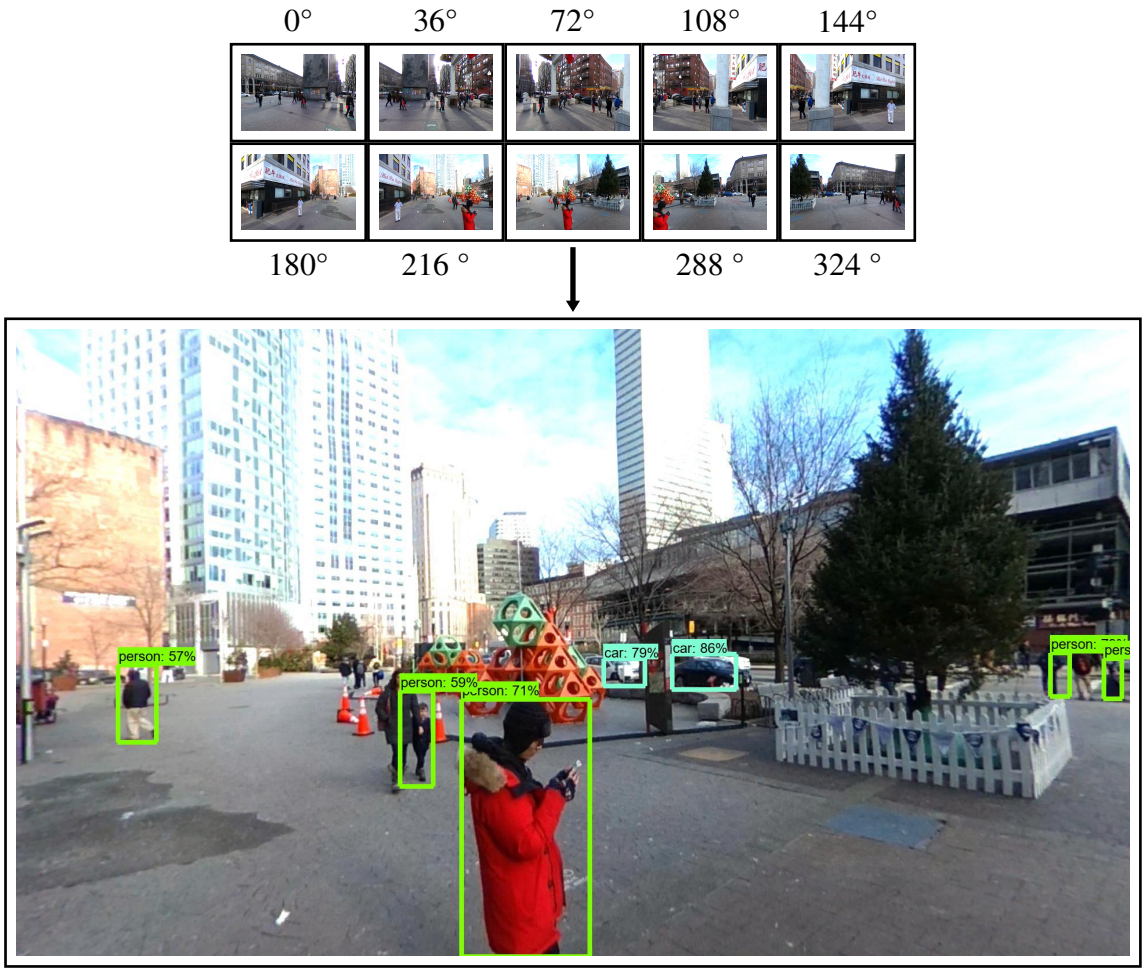


Figure 29: To perform object detection, our approach samples images of the scene by rotating the camera horizontally by 36° each time until the whole scene is covered. Here we illustrate the process for *Chinatown*, with the bounding boxes in the enlarged image showing the detected objects.

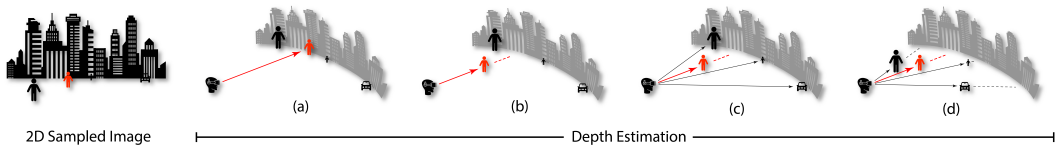


Figure 30: Depth estimation. We assume the average real-world heights of different objects categories based on estimation and previously recorded averages, which are shown in our supplementary materials. In this example, we show both people (average height: 5.3 ft) and a car (average height 5.0 ft). The person highlighted in red represents the reference object and the red line represents the baseline depth specified by the user. The depth estimation estimates the depth of the black objects by these inputs. (a) The system chooses the reference object, which corresponds to the object that received the highest confidence score during object detection. (b) The designer specifies the baseline depth to the reference object. (c) The system estimates the depths of the other objects. (d) audio sources corresponding to the objects will be placed at their estimated depths to create spatial audio.

Our goal is to estimate the depth d_i for each object i detected on the panorama image. Let R_r and R_i be the estimated real-world heights of the reference object r and the detected object i respectively (e.g., the average height of a "person" is 5.3 ft [Vis08], and that of a "car" is 5.0 ft).

The average heights for all objects in the database were either estimated by us or taken from real-world data. Please refer to our supplementary materials for a complete list of the average heights for all the object tags. Savva et al. offer a technique for automatically estimating the appropriate size of virtual objects [SCB14], which is complementary to our approach. Let N_r be the normalized height of the reference object r

(i.e., the object's height in the image divided by the image height) and N_i be the expected normalized height of the detected object i. By similar triangles, we have the following relationship:

$$\frac{N_i}{N_r} = \frac{R_i}{R_r} \tag{3.1}$$

Here, N_i is the only unknown variable, which we solve for. The final step is to calculate the estimated depth d_i. To do this, we compare the expected normalized height (N_i) with the actual normalized height (N_i') of the object in the image, whose depth we are trying to estimate. Then, by similar triangles, we obtain the estimated depth d_i of object i by:

$$d_i = \frac{N_i}{N_i'} d_r \tag{3.2}$$

This depth estimation process is applied for every detected object in the image. Figure 30 illustrates this process.

3.4.5 Audio Source Assignment

Once objects in the image have been detected and tagged, the next step is to assign an adequate audio source to each one. We accomplish this by comparing the 5 tags assigned to each object during object recognition to the tags in our database. We assign a sound if one of the tags matches a tag in the database. The tags of each object are compared to the tags in the database in order of highest to lowest confidence scores.

For objects in the image detected as persons, tags for actions are automatically assigned by the object recognition algorithm, so our approach handles action recognition

for the assignable object tags for actions. In our approach, these are ”Chatting”, ”Chatting on Phone”, ”Typing”, and ”Walking”. Some object tags for actions recognized by object recognition including ”Sitting” and ”Standing” are ignored by our approach because they are not audible.

3.4.6 Audio Equalization Processing

As a post-processing step, all the assigned audio files can be equalized using Adobe Audition. This sound equalization is done in order to match the quality of the sound to fit either an indoor or outdoor environment according to the recognized scene type. In our implementation, we first normalized the volumes of the sounds from different sources before scaling them by distance, and applied an equalization matching algorithm to create indoor and outdoor versions for each sound [GMF16].

3.5 Experiments

Our experiments were conducted with a 3.3GHz Intel Core i7 processor, an NVIDIA Quadro M5000 GPU, and 32GB of RAM. To conduct the experiments, we created a 360° panorama image viewer with the Unity engine, which supports spatial audio. We release our code, data, and viewer for research purposes. The data includes 1,305 panorama images obtained from flickr.com and four images which we captured with a Ricoh Theta V 360° camera.

3.5.1 Sound Placement Results

We discuss the results of running our approach on 4 different panorama images, namely, *Chinatown*, *Seashore*, *Living Room*, and *Cafeteria*, which are depicted in Figure **??** and

Figure 31: We run our algorithm on a dataset of 1,305 images, plus the 4 images (*Chinatown* which shown in Figure **??**, *Seashore*, *Cafeteria*, *Living Room*) we took for the user study. Here we display some of the panorama images used. Refer to our supplementary material for more examples. ”Bus” by Artem Svetlov/CC BY 2.0; ”Museum”, ”Campus” and ”Dock” by Kaleomokuokanalu Chock/CC BY-NC-SA 2.0; ”Park” by tadayoshi527/CC BY-NC-ND 2.0;

Figure 31. For audible versions of these results for the *Chinatown* scene, please refer to the supplementary video.

***Chinatown*:** Our approach automatically assigned an audio file that matched the ”Town” tag as the background audio and identified many people. These people were assigned object tags like ”Walking” and ”Chatting”. Cars were also detected and recognized, and the object tag ”Car” was assigned to these objects. The database then assigned audio files for the objects that matched these tags.

The locations of the people and vehicles are easily discernible, with the background audio making the scene experienced in VR feel like an actual city center. The background audio ”Town” matches sound that one could expect to hear in a town center including background vehicle sounds and construction sounds.

***Seashore*:** Our approach automatically assigned an audio file matching the ”Sea” tag from our database as the background audio. Our approach also detected one person far off in the distance with a cellphone so an object tag of ”Chatting on Phone” was assigned to that object. This object tag matches the audio tag ”Chatting on Phone” in the database, so an audio file associated with that tag was randomly chosen.

Since the person is far off in the distance, the audio source was placed accordingly, which can barely be heard. We used this image to test results of using background audio with few object audio sources.

The background audio assigned consists of the sounds of waves reaching the shore. This mimics what one would expect to hear at a beach shore, as the sound of waves tends to drown out other sounds, especially in a scene like this, which is not crowded.

Living Room: The background sound assigned from our database matches the background tag ”Room” and consists of quiet background noise, which mimics the background noises heard inside city apartments. Only a single audible object in the room was detected and recognized with the object tag ”TV”. By matching the object tag with the same audio tag in the database, we randomly selected an audio file with sounds from a television. In our case, music plays from the television. When viewing this scene with the Oculus Rift headset, it is immediately recognizable where and from what object the audio is coming from.

***Cafeteria*:** Our approach assigned an audio file matching the audio tag ”Restaurant” as the background sound of this scene. It also identified people with the tag ”Chatting”, so audio files for chatting were also assigned.

The restaurant audio file used as background audio, combined with the placed audio files for people chatting, produces the effect of being in an indoor crowded space.

3.6 User Study

To evaluate our approach, we conducted an IRB-approved user study with 30 participants. The users were aged 18 to 50, consisting of 17 males and 13 females, with no self-reported hearing or vision impairment. They were asked to wear an Oculus Rift headset with Oculus Touch controllers and headphones to view four panorama images. The audio was spatialized to stereo via Unity's built-in spatializer plugin (Oculus Spatializer HRTF). The 4 scenes are *Chinatown*, *Seashore*, *Cafeteria* and *Living Room*.

3.6.1 Study Design

The goal of the user study was to test how different characteristics of the audio assigned affected user ratings. To this end, we set out to test 7 sets of different audio configurations. The goal of this setup is to investigate which aspects of the synthesized audio had the largest effect on the subjectively perceived quality. Please refer to Table 3 for a description of the audio configurations included in each set.

Users were asked to view each scene while the audio was played with a certain audio configuration. For each set of audio configurations, the user experienced the same image under the different configurations belonging to that set. The 7 sets were tested in random order, with within-set audio configurations also being played in random order. The initial viewing angle of the scene was randomized before each scene and audio configuration were shown to avoid bias. Users experienced each scene and audio configuration combination once so that they could give a rating for each audio configuration under each set on each scene.

Set	Audio Configurations
1. Space	**spatial** & stereo & mono audio
2. Background	**with** & without background audio
3. Synthesized	recorded audio & **our synthesized**
4. Correctness	**correctly-** & incorrectly-detected objects
5. EQ	with & **without audio equalization**
6. Depth	**our depth** & uniform & random depth
7. No. of objects	10% & 50% & **100%** of all objects

Table 3: Audio configurations used in the user study. The bolded configuration is the standard configuration used in each set. The standard configuration is the result of running our approach on a particular scene without any modifications. We created these sets to investigate what features of the audio were important in delivering a realistic audio experience for the scenes.

3.6.2 Rating Evaluation

Users rated each configuration using a 1-5 Likert scale, with 1 meaning that the audio did not match the scene at all and 5 meaning that the audio was realistic. The audio for each configuration played for approximately 10-20 seconds. We also calculate the p-value for each audio configuration comparison using the Analysis of Variance (RM-ANOVA) test for sets with 3 configurations and using the T-Test for sets with only 2 configurations. The tests were run independently for each scene. We chose Repeated Measures ANOVA and the T-Test since each participant did all configurations under each set. Note that the audio synthesized by our approach without any modification (referred as standard configuration) was included in each set. Any p-value below 0.05

	Set						
Scene	1	2	3	4	5	6	7
Chinatown	**0.011**	0.722	0.589	$<$ **0.001**	0.596	0.123	0.288
Cafeteria	**0.004**	0.221	**0.005**	$<$ **0.001**	0.782	0.148	0.186
Seashore	0.124	$<$ **0.001**	0.126	$<$ **0.001**	0.838	N/A	N/A
Living Room	$<$ **0.001**	1.000	0.055	$<$ **0.001**	0.914	N/A	N/A

Table 4: The p-value for each scene and set of audio configurations calculated from the user study data. The p-values smaller than 0.05, which reject the null hypothesis H_0, are bolded. We performed this statistical analysis to study which aspects of our system affect the user-perceived quality of the audio assigned in each case.

indicates that we can reject the null hypothesis H_0, which refers to the situation that the average user ratings for the audio configurations in each set are about the same. So, whenever we reject H_0 for configurations in a set, it means that the difference in rating caused by the different configurations in that set is significant.

3.6.3 Results

Figure 32 shows a comparison of the different ratings given to each audio configuration by the users. The p-values calculated are shown in Table 4. Our supplementary materials contain the individual scores of each participant. We discuss the results for each set of audio configurations:

Set 1 (Space): For the space set, full spatial audio (the standard configuration)received the highest average score. If we look at the p-values for each scene, we see that they are all below the threshold of 0.05 for rejecting H_0, except for the *Seashore* scene. *Seashore* is assigned only background audio and one object audio for a person who is far away in

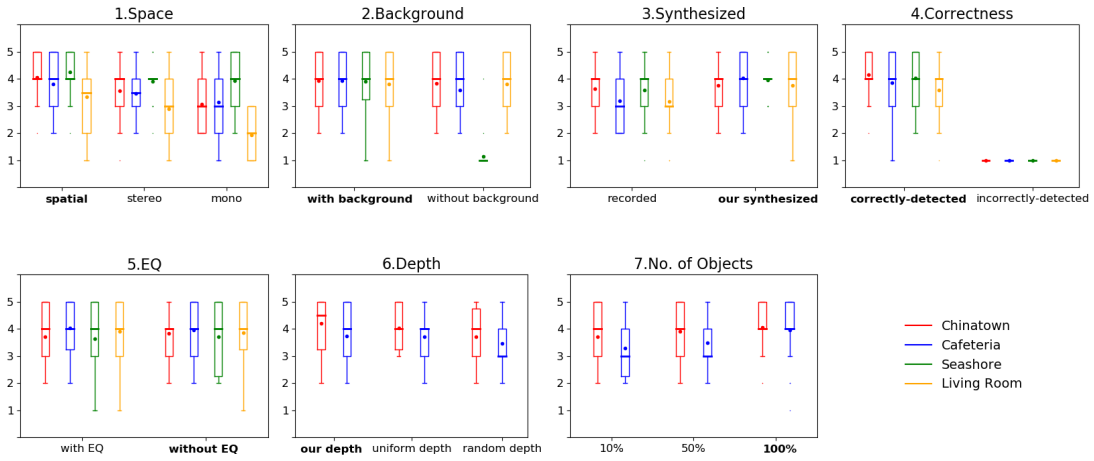


Figure 32: Results of the user study. Each plot corresponds to a set of audio configurations. The colored boxes represent the IQRs; the colored dots represent the means; the thick bars at the middle of the IQRs represent the medians; the upper and lower whiskers extend from the hinge by no further than 1.5* IQR, which show an approximately 95% confidence interval for comparing medians; and the bold texts represent the standard configurations. For each set, the average user rating under each audio configuration for each scene is shown. Note that for the *Seashore* and *Living Room* scenes, sets 6 (Depth) and 7 (Number of Objects) were not tested since each scene only has the background audio and audio for one object assigned.

the scene; the realism brought about by spatial audio may not be apparent or important for this scene. We can conclude that the spatial positioning of audio sources by our approach produces more realistic results than only using stereo or mono audio.

Set 2 (Background): For every scene except *Seashore*, configurations with background audio have scores about the same or slightly higher than the scores obtained by only including object sounds. The p-values for all scenes except *Seashore* are above 0.05. What we can conclude from this is that the effect of adding background audio is not significant when sufficient object audio sources are placed in the scene. This could be explained by the fact that for *Chinatown* and *Cafeteria* there are many distant, individual objects whose sounds may constitute a realistic background sound when mixed together, while for *Living Room* the environment is supposed to be quiet. For *Seashore*, turning off the background audio renders the scene almost silent because all the expected background beach sounds (e.g., sea waves sound) are gone while only the sound of a single detected person chatting on a phone can be heard. In conclusion, we believe that the background audio's importance depends on the scene's setting, while using both background and object audios may provide a more realistic experience in some scenes.

Set 3 (Synthesized): For all the scenes, the configuration with our synthesized audio had a higher average score than that of the recorded audio. However, only the p-value for *Cafeteria* scene was significant (below the threshold of 0.05). We conclude that our synthesized audios are perceived to be at least as realistic as those recorded with a 360° camera.

Set 4 (Correctness): Audio placed for correctly recognized objects received higher average ratings across the board. Configurations produced by our approach with correct

audio placement scored higher across all four scenes. The p-values were all below 0.05, so we can conclude that using audio files that match correctly with the objects of the scene is important.

Set 5 (EQ): Across all four scenes, the average scores for configurations with and without audio equalization were approximately the same and the p-values were all above 0.05. We can conclude that the effect brought about by sound equalization is negligible for our scenes at least.

Set 6 (Depth): On average, the configuration for audio placed at depths calculated by our approach scored higher than the configurations for uniform and random depths. While this is the case, the p-values for all scenes were above 0.05. We can conclude that while placing audio sources at proper depth may enhance realism, the effect is not significant in general. As shown in set 1, the positions of the sound sources are more important than their depths with regard to the realism perceived by the users. We conclude that while being able to recognize the depths of audible objects is important, there can be some flexibility in positioning the sound sources at their exact depths.

Set 7 (Number of Objects): On average, users preferred the configurations with all (i.e., 100%) audio sources used. However, the p-values were all above 0.05. We conclude that while using all detected audio sources produces more realistic results on average, the realism enhancement brought about by using all sources compared to using only some of the sources is not significant. In other words, having a few properly placed sound sources is enough to create the sense that the sound is realistic, even if other objects in the scene are not assigned audio files.

Post-hoc Tests: We also run pairwise post-hoc tests on sets 1, 6, and 7 for the four scenes, with a total of 24 post-hoc tests. For set 1 (space), there is significant difference (p-value smaller than 0.05) in ratings between the standard configuration (spatial) and the mono configuration in the *Living Room* scene. For set 6 (depth), there is significant difference in ratings between the standard configuration (using our estimated depth) and the random depth configuration in the *Chinatown* scene. For set 7 (number of objects), there is significant difference in ratings between the standard configuration (using all detected objects) and configuration using 10% of the detected objects. Please refer to our supplementary documents for full results of the post-hoc tests.

3.6.4 User Feedback

Most users commented that they found the synthesized sounds to be realistic and immersive. However, some users commented that they found some audio sources unrealistic because they could not see moving objects. This was especially true in the *Chinatown* scene, where some users complained that they found the sound of cars unrealistic since no vehicles were moving. While this is a limitation posed by static panorama images, it does relate to the most common suggestion that users had on extending our approach to videos.

For the *Living Room* scene, some users stated that while the TV sound source was realistic, the audio for the sound source was too centralized. In other words, when turning their head away from the TV, they expected to hear more sound waves bouncing back across the room. We could explore incorporating sound waves bouncing off surfaces in our approach, which would require semantic understanding of what surfaces are in the scene.

Many users claimed that they could clearly tell the difference between the configurations that had audio sources positioned with respect to objects in the image and those that did not. Overall, most users were able to identify where audio sources were placed in the images. Most said that such 3D placement of sounds enhanced the scenes. Refer to our supplementary material for information on the frequency of certain types of comments made by users.

3.6.5 Discussion

Out of the 7 sets of audio configurations tested, audio placed by our approach with no modifications tested best in most experiments. From our user study, we see that our configuration received the same or highest average score among all audio configurations in each set. From the parameters we tested, the most relevant ones are spatial audio and correct objects. In comparison, audio equalization, accurate depth, and using all detected objects are not as important as the spatialness and correct detection. As for the background audio, our results show that its importance depends on the scene complexity and it can enhance the realism in some cases (e.g., *Seashore*).

One application of these findings is for prioritizing in power-sensitive or real-time computing, for example, mobile AR applications where certain computations can be conducted with a lower priority without significantly deteriorating overall user experience. We advocate an emphasis on a high-quality ambisonic audio engine and robust object detection algorithm. On the other hand, estimating accurate, high-resolution depth and performing audio equalization could be given a lower priority.

3.7 Conclusion

We presented Audible Panorama, an approach for automatically generating spatial audio for panorama images. We leveraged scene understanding, object detection, and action recognition to identify the scene and objects present. We also estimate the depths of different objects, allowing for realistic audio source placement at desired locations. User evaluations show that the spatial audio synthesized by our approach can bring realistic and immersive experience for viewing a panorama image in virtual reality. We are open-sourcing the audiovisual results we ran on Flickr panorama images, the viewer program, and the manually curated audible sound database.

3.8 Limitation and Future Work

Our current approach only applies to 360° panorama images. As an early attempt, we focus on panorama images which are abundant but usually lack an accompanying audio. A useful and natural extension would be to make our approach compatible with 360° videos with temporal consistency.

As with other data-driven synthesis approaches, one inherent issue with our approach is generalization. In our current study, only 4 scenes are comprehensively evaluated.

While we synthesized and release the audios for all the panorama images we collected on our project website, we did not conduct a full-scale evaluation on all the 1,305 results. One future work is to determine whether those results are as good as the 4 evaluated ones. Our approach may not perform well on panorama images with a) moving

objects with dynamic sound; b) tiny objects too small to be detected (e.g., a bird); and 3) partially occluded objects that result in object detection failure. For example, while a user may expect a partially occluded car on a road to give car engine sounds, an object detection algorithm may fail to detect the car due to partial occlusion and hence no car engine sound is assigned by our approach. We are interested in developing a more systematic way of measuring audio quality and perceptual similarity, especially for immersive audiovisual contents.

The user feedback we received also hints that exploring how to synthesize sounds for moving objects could help improve our approach. Inferring semantic behavior from still images is inherently challenging due to the lack of temporal information which carries important object movement and action cues. However, as humans can infer the object motions on a single image in many cases, with the advancements of computer vision techniques, we believe it would be possible for computers to infer similarly, perhaps by leveraging a more sophisticated understanding of the scene context (e.g., a car near the curb rather than in the middle of the road is more likely to be parked and static) or by analyzing the subtle details (e.g., motion blur) on different regions of the image.

Another interesting extension is to apply our approach for panoramic cinemagraphs: still panorama images in which a minor and repeated movement occurs on a few objects, giving the illusion that the viewer is watching an animation. Our approach could be applied to assign sounds of this repeated motion to the moving objects on a cinemagraph.

We have created a sound database containing audio sources for many types of scenes and objects observed on common panorama images. One further augmentation is to devise sound synthesis algorithms that can synthesize novel audios adaptively based

on observations from new images. Such synthesized audios may match with the scene even more closely and realistically, as well as introducing more natural variations. By releasing our results and working toward a more comprehensive research toolkit for spatial audio, we look forward to user experience enhancement in virtual reality.

3.9 Acknowledgments

We are grateful to the anonymous reviewers for their useful comments and suggestions. We would also like to thank the user study participants, and we are also thankful for the free audio files from freesound.org. The authors would also like to thank all the Flickr users for sharing their panorama images. This research project is supported by the National Science Foundation under award number 1565978.

CHAPTER 4

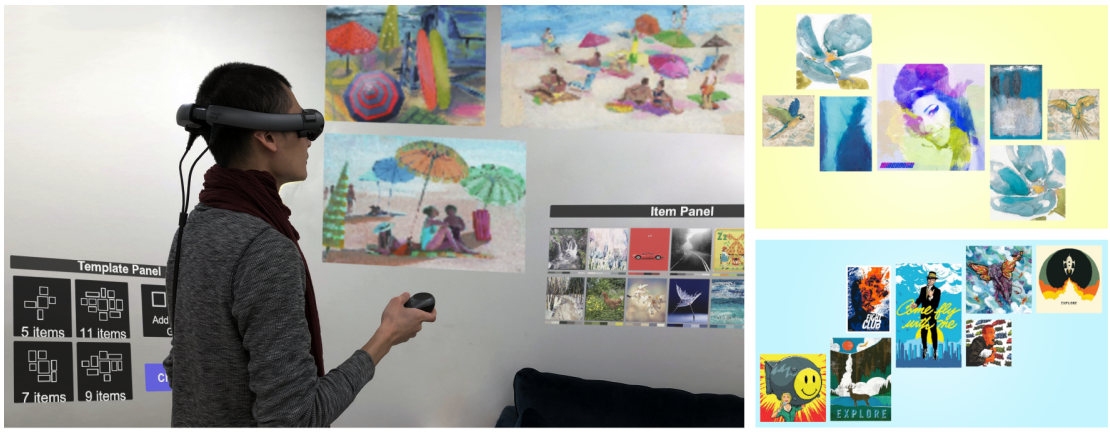
INTERACTIVE DESIGN OF GALLERY WALLS VIA MIXED REALITY

4.1 Introduction

The advent of mixed reality devices (e.g., Microsoft Hololens, Magic Leap One) gives rise to new and exciting opportunities for spatial computing. The superior immersive visualization and interaction experience provided by these devices promises to change the way interior design is performed as they allow users to instantly preview and modify designs in real spaces.

Interior design has historically been a costly and time-intensive process. The conventional design process involves contemplating f abric swatches and inspirational photos, as well as talking to a designer. A professional designer may make use of 3D modeling software to preview a design on screen through sophisticated manual operations. The designer's client must then mentally translate what they see on the 2D screen to how the design may look in a real living space. Without convenient means for visualizing and modifying designs, the design process can be tedious and nonintuitive. Such limitations restrict the ability of general users to engage in interior design even though they may have creative ideas.

In our work, we attempt to address these challenges by devising an interior design tool leveraging the visualization and interaction capabilities of the latest consumer-grade



(a) Using our mixed reality tool for designing a gallery wall

(b) Gallery wall designs

Figure 33: (a) A user wearing a Magic Leap One headset designs a gallery wall using our tool. The figure shows what the user sees in mixed reality while designing: the control panels and his gallery wall design overlaid on the real wall. (b) Some gallery walls designed by users with our tool.

mixed reality devices. Since interior design is a very broad area, as an early attempt to investigate such design applications based on mixed reality, we particularly focus on the design of gallery walls, which are common for decorating interior spaces such as living rooms, hotel lobbies, galleries, etc. Figure 33 shows a user designing a gallery wall using our tool.

A **gallery wall** refers to a cluster of wall **art items** artistically arranged on a wall. A gallery wall commonly contains a **focal item** near the center of the arrangement that sets the tone for the overall design. The other items are called **auxiliary items** and are placed around and with reference to the focal item. The auxiliary items are generally compatible with the focal item in terms of color and style. These definitions follow

the conventions used by designers in creating a gallery wall design using a traditional workflow.

Our tool is suitable for novice users. Users are able to directly visualize how the gallery wall design will look on the real wall while interactively creating the design. The visualization and user-interaction components of our tool keeps the user in the loop of the design process, allowing quick exploration of desirable designs through trial-and-error.

Furthermore, by comparing the color and style compatibility between different art items, our suggestion engine can let the user quickly browse through many desirable design suggestions automatically generated by our tool, hence saving the manual and mental efforts involved in browsing through a large database of wall art items.

We make the following major contributions in this work:

- Based on interviews with professional designers, we devise a computational approach for facilitating and automating the design of gallery walls, which enables a novel mixed reality interactive design tool.
- We demonstrate how mixed reality technology, which bridges the gap between real-world scene knowledge and design suggestions computed in a virtual setting, can be adopted for interior design. In our case, we particularly demonstrate how such an approach can be applied for designing gallery walls.
- We conduct experiments to evaluate the user experience and performance of using our novel tool for gallery wall design. We also conduct a perceptual study to evaluate the quality of the gallery wall designs created by users with our tool.

We believe these contributions will inspire future research in creating mixed reality interfaces for interior design.

4.2 Related Work

To the best of our knowledge, there is no existing work on using mixed or augmented reality for gallery wall design. Regardless, we briefly review the existing research work and commercial tools relevant to our problem domain.

4.2.1 Extended Reality for Interior Design

Companies have been exploring the use of virtual, augmented, or mixed reality technologies for creating and visualizing interior design. Matterport uses a 3D camera to capture the color and depth of real-world living spaces, which can be visualized in 3D by users wearing a virtual reality headset. Such an approach finds promising applications for virtual real estate tours. On the other hand, roOomy provides virtual staging services, enabling users to see previews of interior designs via augmented reality devices showing virtual furniture objects overlaid on a real scene. Furniture retailers such as Wayfair also develop virtual and augmented reality experiences with capabilities such as customizing the design of outdoor spaces with furnishings and décor.

It is noteworthy that several companies provide web interfaces or mobile applications for designing gallery walls. For example, Shutterfly allows users to upload their photos and arrange them using preset layouts provided by the company. Art.com provides a mobile application that allows users to select individual art items or preconfigured gallery wall layouts from their large wall art collection and visualize them via augmented reality on a mobile device.

On the research side, Zhang et al. [ZCC16] proposed an approach to add furniture items and relight scenes on a RGBD-enabled tablet. Yue et al. [YYR17] developed a mixed reality system which allows users to efficiently edit a scene for applications such as room redecoration. Virtual content needs to be adapted to fit the current scene.

Nuernberger et al. [NOB16] devised a technique to align virtual content with the real world. Chae et al. [CHS18] proposed a space manipulation technique for placing distant objects by dynamically squeezing surrounding spaces in augmented reality. Lindlbauer and Wilson [LW18] showed how to manipulate space and time in augmented reality.

Compared to the existing approaches, our tool not only uses extended reality technologies for visualizing gallery wall designs, but also reasons about the spatial and color compatibility. Our tool facilitates the design process by suggesting desirable combinations and placements of art items, taking the color of the wall into account.

4.2.2 Automated Layout Design

Recently, layout design automation has received much research attention. There are previous efforts on automatic 2D graphics layout design [OAH15, OAH14], poster design [QFG16], website design [PCL16], magazine covers [**?**], photo collages [**?**, **?**], etc. We focus on reviewing automatic scene layout design works.

Merrell et al. [MSL11] proposed an interactive tool for furniture layout design based on interior design guidelines, while Yu et al. [YYT11] devised an optimization framework for automatic furniture layout design. Fisher et al. [FSH11] characterized structural relationships in scenes based on graph kernels and later proposed an example-based approach [FRS12] for synthesizing 3D object arrangements for modeling partial scenes. More recently, Wang et al. [WSC18] applied deep convolutional priors for indoor scene synthesis, while Weiss et al. [WLD18] proposed a physics-based approach for fast and scalable furniture layout synthesis. The aforementioned works are mostly focused on the geometrical aspects of populating spaces with furniture. Complementary to this line of work, Chen et al. [CXY15] created a tool called *Magic Decorator* for automatically assigning materials for objects in indoor scenes. Xu [XFI14, XFT15] proposed a tool

for layout beautification and arrangement. However, none of these approaches considers the stylistic arrangement of wall art items in a scene. We propose a novel approach for modeling common factors such as colors, spatial relationships, and semantic compatibility among a number of art items to generate desirable gallery walls, which could complement the existing automated interior design approaches.

We also note that recently CAD software companies such as Autodesk are applying generative design for automating layout synthesis [NLL17]. Along with this line of work, we believe our generative design tool for semi-automating gallery wall design will also find good practical uses.

4.2.3 Interactive Scene Modeling

Typically, users want the ability to control, modify, and visualize the design during the design process so that they can infuse their personal stylistic preferences in their designs. As such, interactive modeling tools play an important role in the design process. There is a large body of work on interactive modeling tools. We review recently proposed scene modeling interfaces.

Along the direction of suggestive interfaces for scene modeling, Yu et al. proposed a suggestive interface called *ClutterPalette* [YYT16] that uses object distribution statistics learned from real-world scenes for suggesting appropriate furniture items to add to a scene. Matthew et al. [FH10] devised a context-based search engine for 3D furniture models to add to a scene.

Another line of work focuses on providing users with easy controls for creating objects while modeling a scene. As humans are accustomed to drawing sketches, a promising approach is to devise sketch-based interfaces for modeling scenes. For example, Xu et al. [XCF13] proposed a sketch-based interface for retrieving and placing 3D

Figure 34: A gallery wall created by a designer using a conventional workflow. The focal item (in yellow), as well as a diagonal pair (in orange) and a triangular group (in cyan) of auxiliary items are highlighted.

models in scenes. Recently, Li et al. [LLZ17] devised an interface called *SweepCanvas* that allows users to perform 3D prototyping on an RGB-D image of a partial scene by sketching. Users can conveniently create virtual objects overlaid on top of the point cloud of a real scene.

Compared to the existing interfaces, we propose a novel mixed reality-based interface for designing gallery wall layouts *in situ*, with the user seeing the design overlaid on top of the target wall. Seeing how the generated design fits into the real world, the user can easily modify the design by a few intuitive operations. We demonstrate in our

evaluation experiments that our tool can allow not only designers but also novice users to quickly generate desirable gallery wall designs.

4.3 Interview with Designers on Workflow

To devise a computational approach and a practical tool for designing gallery walls, we interviewed 4 professional designers from a large furnishings and décor company to better understand the way professional designers create gallery walls under current practice. Each of the designers has at least 5 years of interior design or staging experience and has designed dozens of gallery walls in their professional capacity.

Figure 34 shows a gallery wall created by a designer. The general process of designing a gallery wall is as follows:

1. The designer first observes the style of the room, particularly paying attention to the colors of the wall and the furniture objects that the designer would like to decorate around.

2. The designer explores a database containing many art items and chooses a focal art item, which is to be placed near the center of the gallery wall and serves as the reference for placing other auxiliary art items. The color of the focal art item should be compatible with the wall color.

3. The designer selects and adds the auxiliary art items to the gallery wall. The colors and styles of these art items should contain some variety, yet they should all be compatible with those of the focal art item.

4. The designer lays out the auxiliary art items around the focal art item nicely. One common consideration is balance: pairs of similar art items are placed at opposite sides of the focal art item.

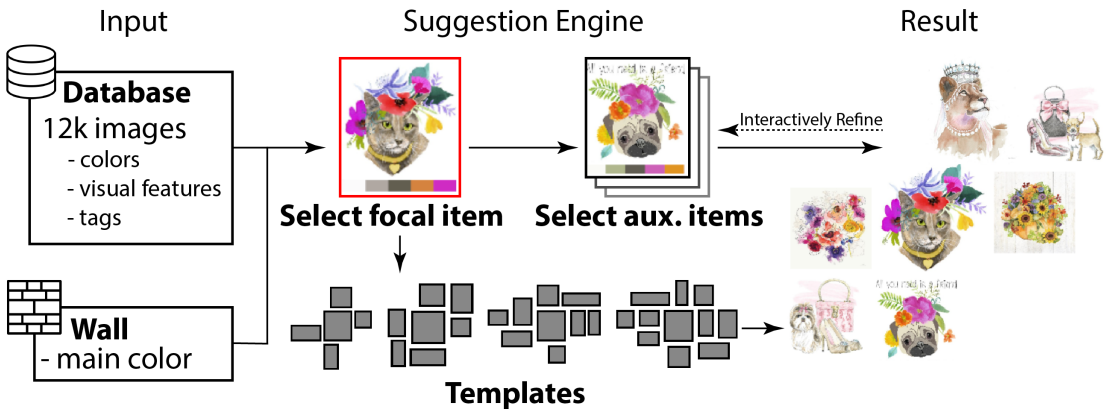


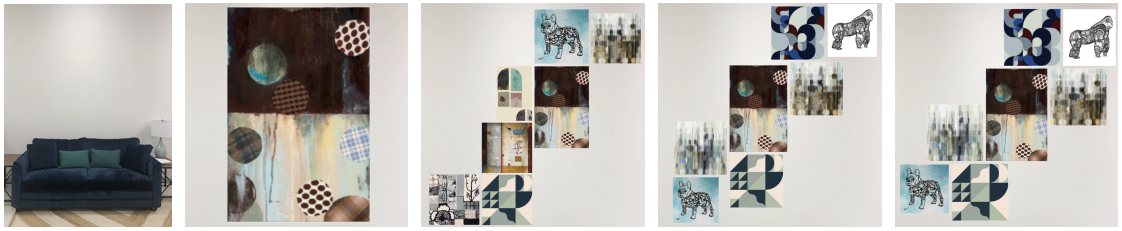
Figure 35: System overview. Taking a database of art items and a wall as input, our tool suggests a focal art item as the starting point for designing a gallery wall. The user may apply a template to generate an initial design, and then interactively refines the design with the help of our suggestion engine.

5. The designer refines the locations of the art items to achieve proper spacing and alignment between items.

We devise our computational approach based on the above observations to automate the conventional design process. Typically, in creating a gallery wall for a common scene like a living room, it takes about 20 minutes for the designer to select compatible wall art items and another 20 minutes to arrange the items into a good layout.

4.4 System Overview

Our tool is realized as an application that runs on the Magic Leap One mixed reality headset. Figure 33 shows a user designing a gallery wall using our tool. The user wears the headset when using our application to create a gallery wall.



(a) Input Wall (b) Focal Item Selection (c) Initialization (d) Interactive Refinement (e) Result

Figure 36: Designing a gallery wall with our tool. (a) The input empty wall. (b) The user selects a focal item. (c) The user applies a template for initializing the design. (d) The user modifies the design interactively. (e) The finished gallery wall design.

The display on the mixed reality headset shows a user interface as well as the virtual gallery wall design overlaid on the real wall, such that the user can instantly preview how the gallery wall design will look on the real wall while designing. The user interacts with the user interface via a handheld controller (e.g., a Magic Leap One's Control) to perform operations such as selecting and dragging.

Figure 35 shows an overview of our tool. It consists of two major components: a suggestion engine for generating design suggestions and a user interface for modifying the gallery wall design. The tool is connected to a large database of wall art items (pictures) from which suitable art items are automatically retrieved and suggested to the user.

4.4.1 Workflow

Figure 36 illustrates the typical user workflow while using our tool for designing a gallery wall. The input is a wall color. The output, which can be reached with as few as three user decisions, is a gallery wall design that goes well with the wall color. Our

tool achieves color compatibility by using the wall color to retrieve from the database a number of candidate focal art items whose colors are compatible with the wall color. A user selects a focal art item from these suggestions, picks a focal item size from among offered art sizes, and either chooses a gallery wall template to launch the synthesis of a gallery wall design or browses recommended auxiliary items.

The selection of auxiliary items, whether human-selected or template-selected, begins with the generation of a color palette based on the colors of the wall and the selected focal item. Based on this color palette and the style of the focal item, the system then retrieves from the database a selection of auxiliary art items that are either presented to the user as options or automatically incorporated into a layout.

After the initialization, the user can interactively modify the gallery wall design via operations such as dragging-and-dropping and selecting in the 3D space using a handheld controller. For example, the user can move, resize, replace, add, or remove any art items. Our tool also provides semi-automatic operations to help the user refine the design, for example, by performing automatic snapping of the art items. The design session ends when the user is satisfied with the gallery wall.

4.4.2 Art Items Data

Database. Our tool is built upon a suggestion engine which helps users browse and make selections from a large database of wall art items while designing their gallery walls. The database, which contains 12,000 wall art items created by professional artists, belongs to a company specialized in furniture and interior design business. Figure 37 depicts some example art items. The art items are mostly paintings and stylized photographs. Each art item comes with 1 to 5 realistic dimensions (from small to large) whose real replicate can be ordered and put on a real wall.

Figure 37: Samples of wall art items in our database, which contains more than 12,000 art items.

Annotations. To compute the compatibility between different art items for making suggestions, each item is annotated with:

1. **Colors.** The 5 most dominant colors of the art item are extracted by the k-means clustering algorithm and stored.

2. **Visual Features.** 256 visual feature values are computed by a convolutional neural network, which encode the visual style of the art item. We trained a Siamese Network to perform such feature extraction using the aforementioned database containing many art items. The network takes image pairs as input, where a positive pair consists of images of items from the same category and a negative pair consists of images of items from different categories. Each input image is processed by a modified Inception-ResNet [SIV17], whose last layer is changed to

Tags: Style

American Traditional	Asian Inspired	Beachy	Bohemian & Bold	Eclectic Modern
Cabin / Lodge	Coastal	Cottage / Country	Cottage Americana	Eclectic
French Country	Glam	Global Inspired	Industrial	Mid-Century Modern
Modern & Contemporary	Modern Farmhouse	Modern Rustic	Nautical	Ornate Glam
Ornate Traditional	Posh & Luxe	Rustic	Scandinavian	Sleek & Chic Modern
Traditional	Tropical			

Tags: Subject

Abstract	Abstract	Bath & Laundry	Buildings & Cityscapes	Cities & Countries
Entertainment	Fantasy & Sci-Fi	Fashion	Floral & Botanical	Food & Beverage
Geometric	Humor	Inspirational Quotes & Sayings	Landscape & Nature	Maps
Nautical & Beach	People	Spiritual & Religious	Sports & Sports Teams	Transportation

Tags: Color

Beige	Black	Blue	Brown	Chrome
Clear	Gold	Gray	Green	Orange
Pink	Purple	Red	Silver	Tan
White	Yellow			

Table 5: The tags in our database, which are manually assigned to the art items by professional designers. The database consists of 3 categories, namely, style (27 tags), subject (20 tags), and color (17 tags).

a fully connected layer, resulting in a 256-dimensional vector as the output. The network was trained to minimize the contrastive loss [HCL06] of the vectors (embeddings) of the input image pairs.

3. **Tags.** Each item also carries tags manually specified by designers of the company. (See Table 5)

Similarity between two art items is computed based on the L2 distance between their annotation vectors: the smaller the distance, the more similar the two items are. We find these annotations very useful in devising our suggestion engine as they characterize the art items and are also common criteria used by designers for comparing art items. By

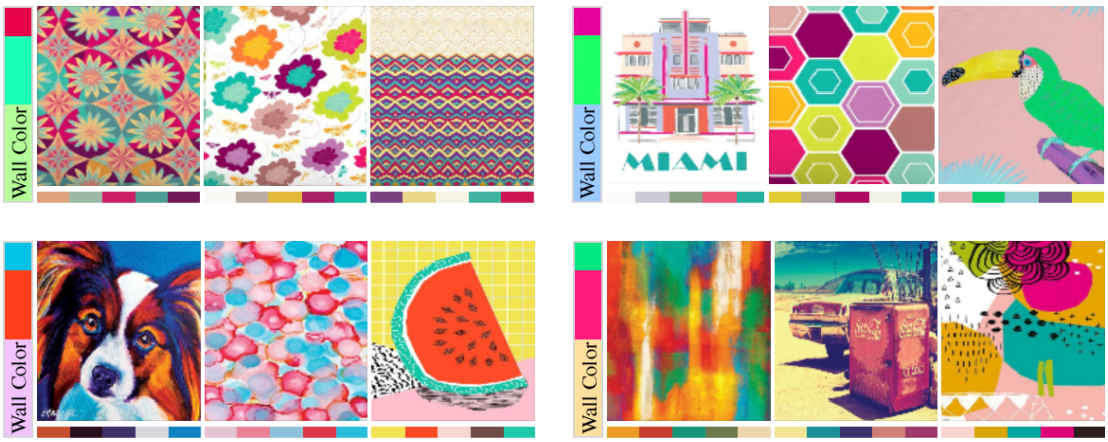


Figure 38: Examples of focal items retrieved by the suggestion engine based on different wall color schemes.

formulating our scoring functions using these three types of annotations, we are able to devise an interface that allows the user to flexibly apply filters using a subset or all of the three types of annotation to retrieve relevant art item suggestions, making it easy for the user to browse through the large database of art items.

4.5 Technical Approach

We provide details for our design suggestion engine. There are two major components: art items suggestion and templates. By using these components, the user can quickly browse through the database of art items and select items that fit with the wall, as well as obtaining a decent spatial arrangement of the items as an initialization of their design.

4.5.1 Wall Plane and Color

Akin to the conventional workflow for designing a gallery wall, our approach starts with considering the wall color. The user wears the Magic Leap One headset and faces

the target wall to be decorated. The wall plane is detected and extracted based on the headset's built-in functionality. The user can manually specify the wall color via a color picker in the user interface, or by using the headset's camera to take a picture of the wall whose average color is taken as the wall color.

Based on the wall color, a neighbor color within $\pm(60^\circ$ to $90^\circ)$ of the wall color is randomly selected from the HSV circular color space. The complementary color of the neighbor color (180° from the neighbor color in the HSV circular color space) is also selected. Figure 39 shows an example. The wall color is used as the basis for retrieving other colors for suggesting relevant art items.

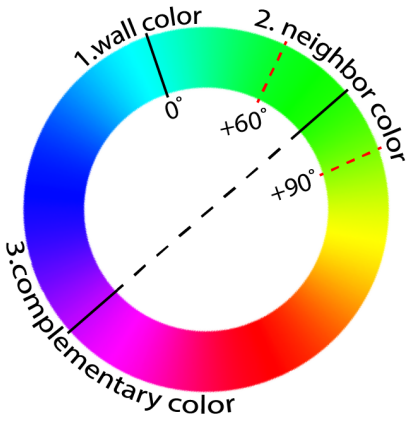


Figure 39: Wall's color palette example.

4.5.2 Suggested Focal Items

Our goal in this step is to retrieve and suggest a list of art items from the database as candidate focal art items for the user. Figure 38 shows some suggested focal art items based on different wall colors.

To achieve this, the wall color, neighbor color and complementary color then form a 3-color palette based on which compatible focal art items are selected. The wall compatibility score S_{foc} of a candidate focal art item ϕ is defined as:

$$S_{\text{foc}}(\phi) = 1 - \frac{1}{9} \sum_{\mathbf{c}_\text{w} \in C_\text{w}} \min\{d(\mathbf{c}_\text{w}, \mathbf{c}_\phi) \mid \mathbf{c}_\phi \in C_\phi\}, \tag{4.1}$$

where C_w is a set containing the wall color, neighbor color and the complementary color in the HSV space; C_ϕ is a set containing the 5 dominant colors of the candidate focal art item ϕ in the HSV space. This scoring function evaluates how close the candidate focal

art item ϕ's color palette is with respect to the wall's color palette. The closer they are, the higher the wall compatibility score.

Note that $\mathbf{c}_{\mathrm{w}}, \mathbf{c}_{\phi} \in \mathbb{R}^3$ are colors in the HSV space. $d(.)$ is a distance metric function to project the two colors $\mathbf{c}_{\mathrm{w}}$ and $\mathbf{c}_{\phi}$ into the HSV cone and to compute the squared distance between them in that cone [**?**]:

$$\begin{aligned} d(\mathbf{c}_{\mathrm{w}}, \mathbf{c}_{\phi}) = & (\sin(H_{\mathrm{w}})S_{\mathrm{w}}V_{\mathrm{w}} - \sin(H_{\phi})S_{\phi}V_{\phi})^2 \\ & + (\cos(H_{\mathrm{w}})S_{\mathrm{w}}V_{\mathrm{w}} - \cos(H_{\phi})S_{\phi}V_{\phi})^2 \\ & + (V_{\mathrm{w}} - V_{\phi})^2, \end{aligned}$$

where $H \in [0, 2\pi)$, $S \in [0,1]$, and $V \in [0,1]$ are the HSV channel values. The range of $d(\mathbf{c}_{\mathrm{w}}, \mathbf{c}_{\phi})$ is $[0,3]$. Equation (4.1) sums up the differences of 3 pairs of colors, hence a normalization of 9 is used.

Our approach computes the compatibility scores for all the art items in the database. The top-20 art items are retrieved and displayed in order of the compatibility scores with the highest-scoring item shown first. The user is supposed to select a focal art item from the list of suggested art items. However, if needed, the user can also explore the database to select any other art item as the focal art item using the Item Panel which we describe in a later section.

4.5.3 Suggested Auxiliary Art Items

Akin to the conventional gallery wall design approach, the selected focal art item serves as a reference for the suggestion engine to suggest other compatible, auxiliary art items to add to the gallery wall design. To retrieve auxiliary art items from the database as suggestions, an overall compatibility score S_{aux} is computed for each candidate auxiliary

art item, which evaluates the style and color compatibility between the auxiliary art item and the selected focal art item:

$$S_{\text{aux}}(\phi) = w_{\text{c}} S^{\text{c}}_{\text{aux}}(\phi) + w_{\text{s}} S^{\text{s}}_{\text{aux}}(\phi), \tag{4.2}$$

where w_{c} is the weight of the color compatibility score $S^{\text{c}}_{\text{aux}}$ and w_{s} is the weight of the style compatibility score $S^{\text{s}}_{\text{aux}}$.

Color Compatibility Score: A candidate auxiliary art item ϕ has a high color compatibility score $S^{\text{c}}_{\text{aux}}$ if its colors are close to the dominant colors of the selected focal art item. Specifically, the color compatibility score of auxiliary art item ϕ is defined as follows:

$$S^{\text{c}}_{\text{aux}}(\phi) = 1 - \frac{1}{15} \sum_{\mathbf{c}_{\text{f}} \in C_{\text{f}}} \min\{d(\mathbf{c}_{\text{f}}, \mathbf{c}_{\phi}) \mid \mathbf{c}_{\phi} \in C_{\phi}\}, \tag{4.3}$$

where C_{f} and C_{ϕ} are respectively sets containing the 5 dominant colors of the selected focal art item and of the auxiliary art item ϕ. $\mathbf{c}_{\text{f}}, \mathbf{c}_{\phi} \in \mathbb{R}^3$ are colors in the HSV space. As Equation (4.3) sums up the differences of 5 pairs of colors, and each difference has a range of $[0,3]$, a normalization of 15 is used. This scoring function evaluates how close the auxiliary art item ϕ's color palette is with respect to the selected focal art item's color palette. The closer they are, the higher the score is.

Style Compatibility Score: A candidate auxiliary art item ϕ has a high style compatibility score $S^{\text{s}}_{\text{aux}}$ if its visual feature vector is close to the selected focal art item's visual feature vector. The style compatibility score of auxiliary art item ϕ is defined as follows:

$$S^{\text{s}}_{\text{aux}}(\phi) = 1 - \frac{1}{\sqrt{n}} \|\mathbf{v}_{\text{f}} - \mathbf{v}_{\phi}\|, \tag{4.4}$$

where $\mathbf{v}_{\text{f}}, \mathbf{v}_{\phi} \in \mathbb{R}^n$ are the n-dimensional visual feature vectors of the focal item and auxiliary art item ϕ computed by the convolutional neural networks. The size of the dimension in our database is 256.

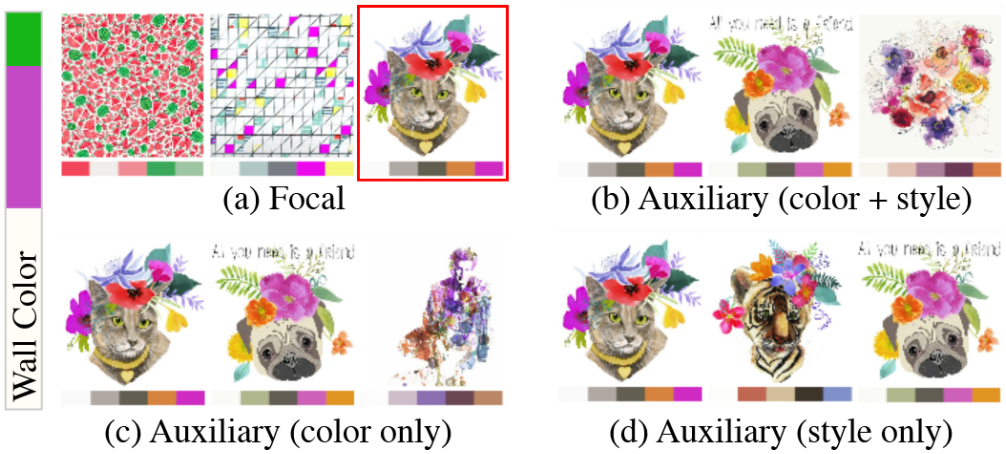


Figure 40: Art item suggestions. Based on the wall's color palette shown on the left, (a) a number of focal art items compatible in color are suggested. Based on a selected focal art item (highlighted in red), (b) a number of auxiliary art items compatible in color and style are suggested. Auxiliary art items suggested by considering (c) only color or (d) only style.

Overall, the compatibility score is computed for each art item in the database. The user interface displays the top-20 art items sorted in descending order of their compatibility scores as auxiliary art item suggestions. To provide flexibility in retrieving suggestions, our user interface allows the user to turn on and off the consideration of the color or the style compatibility score, which correspond to setting w_c or w_s as 1 or 0. Figure 40 shows an illustration. The user can also select which of the 5 dominant colors C_f of the selected focal art item to consider in computing the color compatibility score.

4.5.4 Templates

To allow the user to quickly generate an initial gallery wall design based on a focal item, our tool provides a number of preset templates that the user can choose from

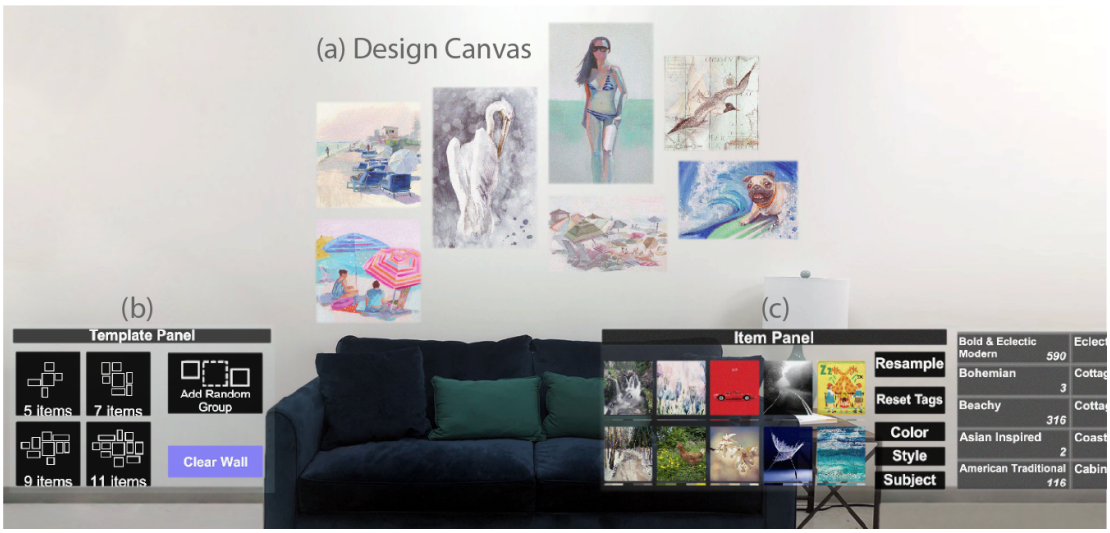


Figure 41: User interface of our tool via which a user wearing a mixed reality headset visualizes and designs a gallery wall. It consists of three components: (a) Design Canvas; (b) Template Panel; and (c) Item Panel.

and apply. Figure 44 shows some example templates that our tool provides. These templates encode spatial relationships of art items that are commonly applied by gallery wall designers. A template arranges groups of auxiliary art items symmetric about and around the focal art item, akin to the gallery wall designs created by the conventional design workflow. Figure 41(b) shows the 4 templates (with 5, 7, 9, and 11 items) that we provide with our tool in our experiments. These templates resemble the common patterns used by designers as we learned from the interview. Auxiliary items within a group have compatible color and style by default (Equation (4.2)).

Initializing a Gallery Wall Design. Figure 42 illustrates how to apply a template to generate a gallery wall design. According to the layout of the chosen template, starting from the root (the focal art item), our approach inserts auxiliary art items which are compatible with the focal art item, group by group. More specifically, the items are

added on the circumference of a circle centered at the focal item with a random radius $r \in [0.5d, 3d]$, where d is the diagonal length of the focal item. A pair of two auxiliary items would be added at the opposite sides of the circle. A group of three items would be added at the vertexes of a randomly-oriented equilateral triangle circumscribed by the circle. The gallery wall design generation finishes as all groups of auxiliary art items have been placed. The generated design is taken as an initial design based on which the user can modify interactively.

Spatial Refinement. Our approach refines the spatial relationships between the items after the initialization and after every user interaction with the gallery wall design such as adding an item, removing an item, and moving an item.

Snapping: To keep the gallery wall design compact, by default, all auxiliary items steer towards the focal item at the center while they maintain a certain minimum space between each other to avoid overlapping.

Alignment: To keep the gallery wall design neat and uncluttered, by default, our approach aligns neighboring art items either horizontally or vertically by their edges so long as the alignment does not cause overlapping. (see Figure 43(b))

We include more examples of interactively modifying actions in the supplementary video.

4.6 User Interaction

Figure 41 shows the user interface of our tool which is displayed in mixed reality. It consists of three components: a) the *Design Canvas* where the user can interactively modify the current gallery wall design visualized on the real wall; b) the *Template Panel* where the user can select and apply a preset template for synthesizing an initial gallery

(a) Template (b) Result

Figure 42: Applying a template to generate a gallery wall design. (a) A template and its tree structure. The auxiliary items are placed symmetrically about the focal item at the center. (b) A gallery wall created by applying this template.

(a) Replacing an Art Item (b) Alignment

Figure 43: The user can interactively modify a design in mixed reality using the functionalities of our user interface.

wall design; and c) the *Item Panel* where the user can retrieve art items from the database by specifying different criteria. Each of the components provides a number of functionalities that allow the user to refine a gallery wall design conveniently and desirably. We describe them in the following.

4.6.1 Design Canvas

The design canvas visualizes and overlays the current gallery wall design on the real wall via the mixed reality headset's display. It also provides a number of functionalities for interactively adjusting the art items and the gallery wall layout:

- **Add.** The user selects an art item in the current design and retrieves a number of art items from the database that are compatible in terms of dominant colors, visual features and tags, which he can add to the current design.
- **Replace.** The user replaces an art item with another compatible art item from the database. (see Figure 43(a))
- **Move.** The user moves an art item by dragging it.
- **Resize.** The user chooses another size for an art item.
- **Remove.** The user removes an art item.

4.6.2 Template Panel

The Template Panel allows the user to quickly generate an initial gallery wall design with items decently placed. It provides a number of preset templates that the user can apply to synthesize a gallery wall design based on a selected focal art item. It also provides other functionalities to enable automatic refinement of the spatial layout of the current design. A list of functionalities supported:

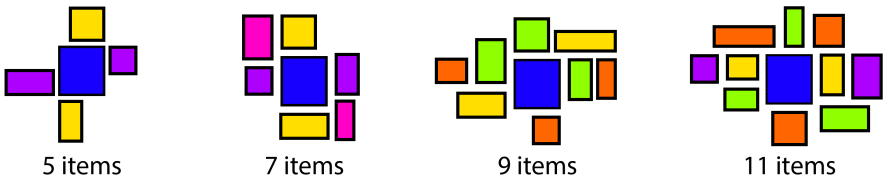


Figure 44: The templates we use in our system. The blue item represents the focal item. The colored items represent the groups of 2 or 3 auxiliary items.

- **Apply a Template.** Based on a placed focal art item, the user applies a template to synthesize a gallery wall design.

- **Add Random Group.** Our tool automatically adds a group of 2 or 3 auxiliary art items, which are compatible with the focal art item, to the current design. The group of auxiliary items are symmetric about the focal item (see Figure 44).

- **Align All.** The user triggers our tool to align all the art items with respect to each other. The alignment is done along the horizontal (left, right, or center) or vertical (top, bottom, or center) direction. This functionality comes in handy because it could be tiring and difficult for users to make a lot of precise adjustment in the 3D space [Mac92] using a handheld controller.

- **Snap.** If enabled, an art item is snapped to its neighbor item as the user drags the art item around, i.e., it will steer towards the center of its neighbor item until a minimum spacing between the two items is reached. (see Figure 45)

- **Clear Wall.** All the art items are removed from the design.

4.6.3 Item Panel

The Item Panel is connected to the suggestion engine and the database of art items. Its primary function is to display a relevant list of art items that the user can add to the gallery wall design as a focal art item or auxiliary art items. The panel shows a number of buttons that the user can click to set the criteria for retrieving relevant art items from the database. For example, the user can select whether to use color, or style, or both as criteria for determining compatibility between items. The user can also select which color(s) out of the 5 dominant colors of the focal art item to use for determining color compatibility. A list of functionalities supported:

- **Update Wall's Color Palette.** Our tool re-generates the neighbor and complementary colors of the wall's color palette.

Figure 45: Snapping example. (a) The user drags an art item, which is (b) snapped towards the center of its neighbor item until a minimum spacing between the two is reached.

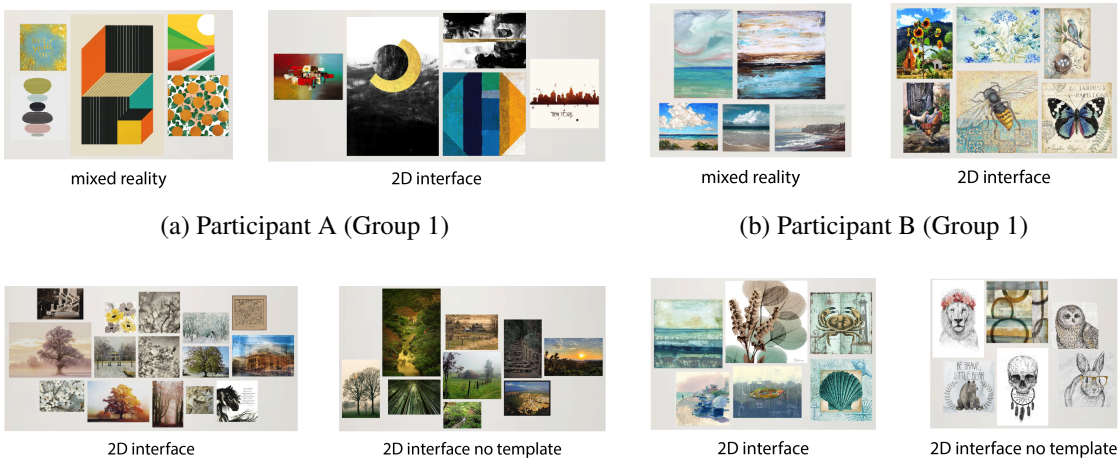
- **Find Focal.** Based on the wall's color palette, our tool retrieves a number of compatible art items as focal art item suggestions (according to equation (4.1)).
- **Find Auxiliary.** The user selects criteria (e.g., colors, visual features, tags) based on which our tool retrieves 20 compatible art items as auxiliary art item suggestions. By default, the art items are sorted by their compatibility scores.

4.7 User Evaluation

We developed our tool using C# on the Unity Game Engine installed with the Magic Leap Lumin SDK. We deployed our tool onto a Magic Leap One headset which we used for our user evaluation experiments.

User Groups. We recruited two different groups of users to evaluate our tool.

Group 1: The first group was recruited to evaluate the user experience of designing a gallery wall using our mixed reality interface based on Magic Leap One versus using a 2D interface which mimics a traditional design tool on a laptop. We recruited 17 participants, who are the employees of a company, consisting of 12 males and 5 females, aged from 20 to 45, the average age was 32. All participants did not have experience



(a) Participant A (Group 1)

(b) Participant B (Group 1)

(c) Participant C (Group 2)

(d) Participant D (Group 2)

Figure 46: Example gallery wall designs created by the user evaluation participants under different conditions.

using the Magic Leap One headset. Each participant designed 2 gallery walls, under ***Condition MR*** and ***Condition 2D*** in a random order.

Group 2: The second group was recruited to evaluate the user experience of designing a gallery wall without and without the template functionality. We recruited 24 participants, who are the college students, consisting of 16 males and 8 females, aged from 19 to 24 the average age was 22. Each participant designed 2 gallery walls under ***Condition 2D*** and ***Condition 2DNT*** in a random order. t

Conditions. We asked the participants to create gallery wall designs under different conditions. The goal of each task was to design a gallery wall that fits with a living room with a pale gray wall and a blue sofa as shown in Figure 41.

- ***Condition MR:*** The participant used our tool delivered through a mixed reality interface to create a gallery wall.

- ***Condition 2D:*** The participant used our tool delivered through a 2D interface to create a gallery wall.

- ***Condition 2DNT:*** The participant used our tool delivered through a 2D interface to create a gallery wall with *no template functionality*. That is, the "Apply a Template" and "Add Random Group" buttons under the Template Panel were disabled.

Note that, both ***Condition MR*** and ***Condition 2DNT*** used a background image showing a pale gray wall and a blue sofa. We include the 2D interface we used for the user evaluation in the supplementary material.

Procedure. Before each task, we briefed and trained the participant in creating a gallery wall design using our tool. We asked the participant to follow a 5-minute tutorial which guided him how to use our tool to create a gallery wall design step by step. Note that we showed the participant the tutorial whether he was tasked with creating a gallery wall using the mixed reality interface (Condition MR) or the 2D interface (Condition 2D). The participant could ask any question about the user interface to make sure he was familiar with using it.

The participant was then asked to create a gallery wall design that fits with the living room under a given condition. For ***Condition 2D*** and ***Condition 2DNT***, the participant was presented with a photo of the living room when designing a gallery wall using the 2D interface (please refer to the supplementary material for a screenshot of the interface). Our tool tracked the interaction metrics for later analysis.

Figure 46 shows some gallery walls designed by the participants. Our supplementary material contains the results created by all participants.

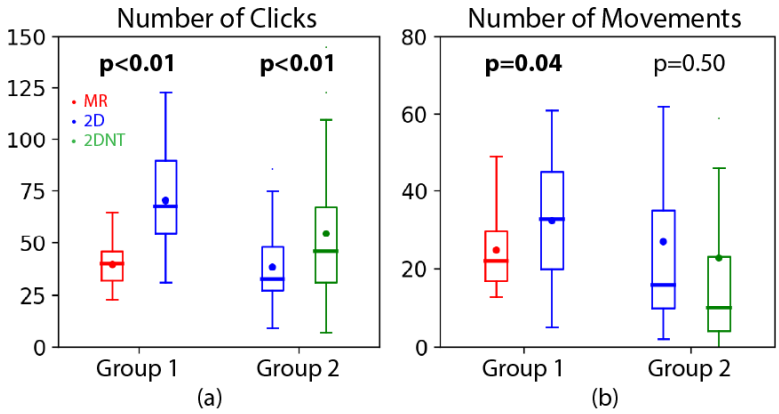


Figure 47: Performance results under different conditions. Color dots and bars show the means and medians. The p-value of t-test computed between the results of the two conditions in each group is shown. The p-values smaller than 0.05 which reject the null hypothesis are bolded.

4.8 Experiment Results

We discuss the user evaluation results with regard to performance, usage and user feedback. We use t-tests to evaluate if there is any significant difference between the results obtained under different pairs of conditions by reporting the p-values. We show our results in box plots for easy interpretation. Our supplementary material shows the numeric results and all designs created by participants under different conditions.

4.8.1 Performance

We tracked the performance of retrieving items from the large database we used in our experiments on a desktop computer equipped with a i7-7700K 4.2GHz CPU, 16GB RAM, and an NVIDIA GeForce GTX 1080 8GB graphics card. Retrieving focal items took 3.12 seconds and retrieving auxiliary items took 1.21 seconds. All the other user interface operations ran at an interactive rate.

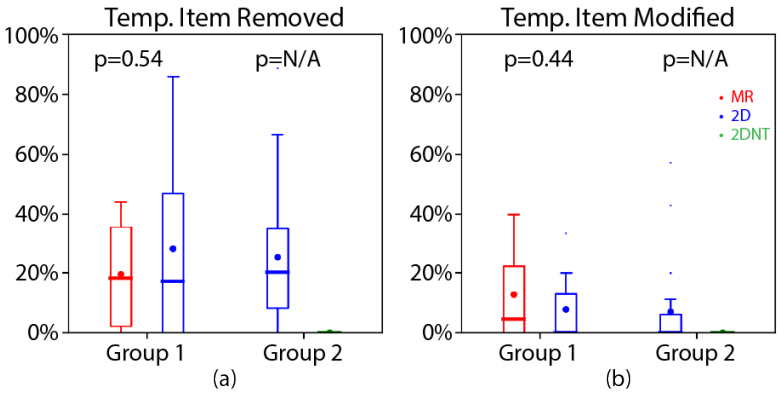


Figure 48: Usage statistics of the items placed by a template under different conditions.

Our tool tracks the participant's performance under each given condition. Here are the performance metrics tracked:

- ***Number of Clicks:*** The total number of times the participant clicked on a user interface component.
- ***Number of Movements:*** The total number of times the participant adjusted the position of an art item.

Mixed Reality vs. 2D Interface. As the Group 1 results in Figure 47 show, under the MR condition where the participants designed via a Magic Leap One headset, they made fewer clicks (p¡0.01 in t-test) and movements (p=0.04), compared to the 2D condition where they designed via a 2D interface.

Under the MR condition, the participants could see their design directly visualized on the real wall, which might result in fewer adjustments. In our perceptual study, we find that there is no significant difference between the visual quality of the designs created under the MR and the 2D conditions. In all, the direct visualization brought about

by mixed reality allows the participants to create gallery wall designs of similar visual quality (compared to designs created on a 2D screen) with fewer manual adjustments.

Templates vs. No Templates. As the Group 2 results in Figure 47(a) show, under the 2D condition where the participants created designs with the aid of templates they made fewer clicks (p¡0.01) compared to the 2DNT condition where they created designs without the aid of templates. The template functionality helped them create designs more efficiently.

4.8.2 Usage

Our tool also tracks the usage of the design suggestions generated by the Template Panel or Item Panel. Here are the metrics used:

- ***Template Item Removed:*** If a participant applied a template for initializing a gallery wall design, the percentage of items generated by the template that he/she removed.
- ***Template Item Modified:*** If a participant applied a template for initializing a gallery wall design, the percentage of items generated by the template that he/she modified.
- ***Suggested Item Usage:*** What percentage of items in the final gallery wall design was chosen among the top-20 suggestions from the Item Panel retrieved according to the user's specified criteria.
- ***Selected Item's Rank:*** The average rank of the items the participants selected from the suggestions of the Item Panel.

Template Items. Figure 48 shows the usage statistics of the items placed by a template under different conditions. As Figure 48(a) shows, the participants removed about 20% and 28% of the items placed by the templates under the MR and 2D conditions respectively. As a template places highly similar items in a design, it seems that the participants tended to replace a few items with other less similar items to introduce some variation or contrast into the overall design. On the other hand, as Figure 48(b) shows, in average only about 13% and 8% of the items were modified under the MR and 2D conditions respectively. Overall, the participants kept a majority of the items placed by a template in their final gallery wall design.

Suggested Items. Figure 49 shows the usage statistics of the items suggested by the Item Panel under different conditions. As Figure 49(a) shows, under the MR and 2D conditions, about 82% and 76% of the items used in the final design were chosen from the top-20 suggestions in the Item Panel retrieved according to their specified criteria. The Group 2 results show that there was a significant difference (p¡0.01) between suggested item usages under the 2D and 2DNT conditions. Under the 2DNT condition when the participants could not use a template to initialize a gallery wall design, they tended to use items from the database more randomly.

On the other hand, as Figure 49(b) shows, the average rank of the selected items ranges from about 8 to 10 under different conditions. It seems that, in using the Item Panel, the participants tended to choose items that match with their specified criteria but not in a very strict sense.

4.8.3 User Feedback

We talked to the participants after the experiments. They generally merited the visualization brought about by our mixed reality tool for showing the design on the real wall,

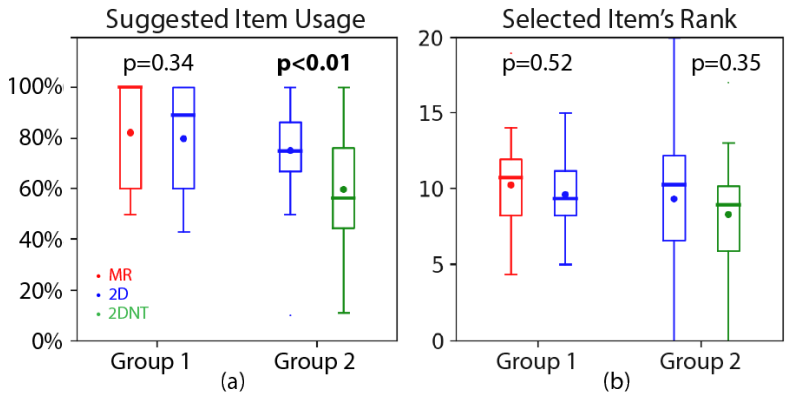


Figure 49: Usage statistics of the items suggested by the Item Panel under different conditions.

which made the interactive design process more intuitive and interesting, compared to designing on a 2D screen, as they could directly see how their design could fit with the real space. On the downside, some participants reported initial user experience challenges while acclimating themselves to the device's field-of-view and headset fit. We include the participants' comments in our supplementary material.

We also conducted a two-alternative forced-choice approach to evaluate the quality of the gallery wall designs created by the participants under two different conditions (***MR*** & ***2D*** or ***2D*** & ***2DNT***). We include the details of the perceptual study in the supplementary material.

4.9 Summary

We proposed a novel mixed reality-based interactive design tool for gallery walls. By overlaying a virtual gallery wall on a real wall, our tool allows users to directly visualize their design in the real world as an integrated part of the creative process. Our suggestive

design interfaces facilitate users in retrieving compatible art items for creating their desired gallery walls.

4.9.1 Limitations

We used a Magic Leap One headset for experimenting with our tool. The hardware still has limitations for consumer use. For example, the field-of-view is still a bit narrow for the user to see the overall design and the user interface at once; the user has to look around to see different things, which could be inconvenient.

It could be tiring to use the handheld controller for 3D interaction and manipulation in the 3D space for a prolonged period of time. Because of this, it would be challenging to use a sophisticated design tool that requires tedious and precise user input in mixed reality.

We believe that the visualization of the design in the real space by mixed reality may benefit the users in envisioning and communicating their designs, allowing them to design in the real space directly and intuitively. In the future, it would be interesting to extend our mixed reality design tool to consider more sophisticated context of the 3D scene to facilitate other interior design tasks, such as suggesting furniture placement. Performing advanced scene analysis in real time for enabling interior design applications in mixed reality presents a practical challenge, which could be resolved as the computaional power of mixed reality devices continue to increase.

Due to the scope of this project, in this work we only focus on a subset of interior design, namely, gallery wall designs. We show that a mixed reality approach for gallery wall design is feasible. Our approach could be extended to consider 3D decorations too (see Figure 50), though they are less common and we did not consider them in our current approach.

It would be helpful to have a large database of gallery wall designs from which our tool could learn the spatial relationships and compatibility between different art items in a gallery wall design, and use such knowledge for synthesizing new designs.

4.9.2 Future Work

With the advances of artificial intelligence and natural language processing techniques, we hope that such techniques can be adopted in mixed reality for enabling natural and convenient user interaction experiences in the interior design process. For example, it would be very helpful if a mixed reality interior design software application could understand voice commands from the user to decorate a space, minimizing the need for manual user input. With such advances, using a mixed reality software application for interior design will be as natural as talking to an interior designer who can instantly visualize the design for the user through mixed reality. We believe this is an exciting goal for enabling human-AI collaboration in design.

For commercial purposes, art items could be further annotated with non-aesthetic tags, such as their prices, frequencies of being viewed or selected, and the semantic meanings they carry. Incorporating such additional tags could provide more desirable item recommendations while a user is creating their gallery wall designs.

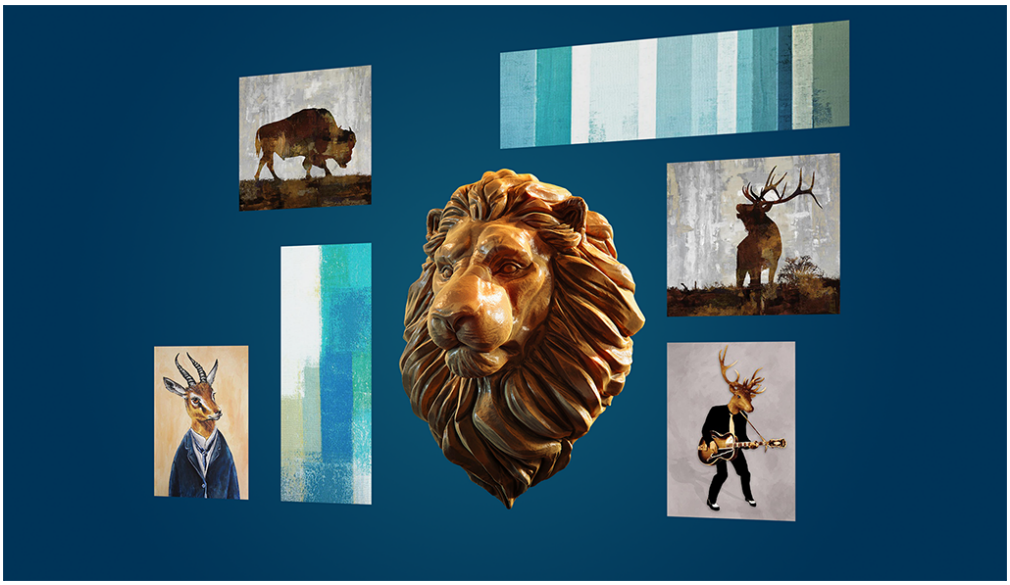


Figure 50: Example of a gallery wall design containing a 3D decoration object (a lion head) that can be visualized in mixed reality using our tool.

CHAPTER 5

CONCLUSION

5.1 General Summary

This dissertation has demonstrated how AI-driven optimization approaches can be applied to tackle the human-centric modeling problem. From the visual perspective, a novel approach has been proposed for wayfinding design. The proposed approach shares personal perception properties, which can be used to guide the optimization in order to accomplish the modeling task. It results in the generation of different realistic wayfinding designs that can be used for indoor and outdoor scenarios. From the auditory perspective, a novel approach has been proposed to enable accurate sound sources placement for a 360° panorama image. The approach enables the system to estimate appropriate distance and orientation of the detected objects and place the sound sources in a 3D space, which centered from the observer while it classifies the background of the given 360° image and places the appropriate ambient sound source as the background sound. From the aesthetic perspective, a novel approach has been proposed for gallery designs with a smart suggestive engine. The approach can automatically generate gallery designs by the smart suggestive engine for providing insights into the overall design of the stylizing and colorizing.

5.2 Future Trends in AI-Driven Human Centric Modeling

Recent advances in machine learning, virtual reality, and argument reality have made these technologies increasingly accessible to the general public. Through AI-Driven modeling, such human-centric perception properties can be readily utilized to create high-quality and robustness models for various applications. With XR devices, one can experience the procedure generated world in mixed reality with real physical perception just like the real world. Advances in haptic devices may even allow us to touch and feel those virtual objects. These possibilities are inspiring as it will open up many opportunities for researches and application of AI-driven human-centric modeling.

Low-cost eye-tracking device (e.g., Tobii) and body motion capture (e.g., Kinect), free and open-source databases(e.g., Coco Dataset, Youtube labeled Video Dataset), and the easy-to-use machine learning languages and libraries (e.g., Python & Scikit Learn), make it easy for researchers to obtain the human-centric data. These data will probably be transformed into human-centric perception properties.

These trends foretell an even stronger future role for AI-driven modeling. With the advancement of XR device research, there are more and more methods for obtaining data, and the methods for obtaining data are getting more straightforward and more accessible. They provide reasonable grounds for AI-driven modeling approaches. On the other hand, the modeling process will no longer reside solely in empirical. For example, a product must satisfy important physical, material, and mechanical properties to sustain physical and usability requirements in the real-world. This is exciting as it opens up many opportunities for research. Our research on AI-driven computational design tools intended for general users, which ease the designing process, also closely aligns with the trend of generic and personalized designs. Such promising trends will

also give impetus to more sophisticated creative e activities at the household and small business levels, e.g., to create video games, movies, products, and other applications.

REFERENCE LIST

[Ada14] Ernest Adams. *Fundamentals of game design*. Pearson Education, 2014.

[AP15] Pavel Zahorik Silvia Cirstea Andrew J. Kolarik, author Brian C. J. Moore and Shahina Pardhan. "Auditory distance perception in humans: a review of cues, development, neuronal bases, and effects of sensory loss." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4744263/`, 2015.

[APO17] Fathima Assilmia, Yun Suen Pai, Keiko Okawa, and Kai Kunze. "IN360: A 360-Degree-Video Platform to Change Students Preconceived Notions on Their Career." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017, Extended Abstracts.*, pp. 2359–2365, 2017.

[ASH19] Rawan Alghofaili, Yasuhito Sawahata, Haikun Huang, Hsueh-Cheng Wang, Takaaki Shiratori, and Lap-Fai Yu. "Lost in Style: Gaze-driven Adaptive Aid for VR Navigation." In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19. ACM, 2019.

[Bih17] Peter Robinson Eduardo Dias Lee Skrypchuk Bihao Wang, Quentin Stafford-Fraser. "Landmarks based human-like guidance for driving navigation in an urban environment." In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[BKL04] Doug Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice, CourseSmart eTextbook*. Addison-Wesley, 2004.

[BM07] Doug A Bowman and Ryan P McMahan. "Virtual reality: how much immersion is enough?" *Computer*, **40**(7), 2007.

[BMW14] Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer. "Automatic generation of tourist brochures." In *Computer Graphics Forum*, volume 33, pp. 449–458. Wiley Online Library, 2014.

[BMW17] M. Birsak, P. Musialski, P. Wonka, and M. Wimmer. "Dynamic Path Exploration on Mobile Devices." *IEEE Transactions on Visualization and Computer Graphics*, **PP**(99):1–1, 2017.

[BT00] Durand R Begault and Leonard J Trejo. "3-D sound for virtual reality and multimedia." 2000.

[Bur98] Peter C Burns. "Wayfinding errors while driving." *Journal of Environmental Psychology*, **18**(2):209–217, 1998.

[BYM13] Fan Bao, Dong-Ming Yan, Niloy J Mitra, and Peter Wonka. "Generating and exploring good building layouts." *ACM Transactions on Graphics (TOG)*, **32**(4):122, 2013.

[Cal07] C. Calori. *Signage and Wayfinding Design: A Complete Guide to Creating Environmental Graphic Design Systems*. Wiley, 2007.

[CBK00] Hernan Casakin, Thomas Barkowsky, Alexander Klippel, and Christian Freksa. "Schematic maps as wayfinding aids." In *Spatial cognition II*, pp. 54–71. Springer, 2000.

[CHS18] Han Joo Chae, Jeong-in Hwang, and Jinwook Seo. "Wall-based Space Manipulation Technique for Efficient Placement of Distant Objects in Augmented Reality." In *The 31st Annual ACM Symposium on User Interface Software and Technology*, pp. 45–52. ACM, 2018.

[CR08] Daniel C Cliburn and Stacy L Rilea. "Showing users the way: Signs in virtual worlds." In *2008 IEEE Virtual Reality Conference*, pp. 129–132. IEEE, 2008.

[CXY15] Kang Chen, Kun Xu, Yizhou Yu, Tian-Yi Wang, and Shi-Min Hu. "Magic decorator: automatic material suggestion for indoor digital scenes." *ACM Transactions on Graphics (TOG)*, **34**(6):232, 2015.

[DK03] Matt Duckham and Lars Kulik. ""Simplest" Paths: Automated Route Selection for Navigation." In *International Conference on Spatial Information Theory*, pp. 169–185. Springer, 2003.

[DN07] Janki Dodiya and Vassil N. Alexandrov. "Perspectives on Potential of Sound in Virtual Environments." *HAVE 2007. IEEE International Workshop on Haptic, Audio and Visual Environments and Games*, pp. 15–20, 2007.

[DP14] Rudolph P Darken and Barry Peterson. "Spatial orientation, wayfinding, and representation." In K.S. Hale and K.M. Stanney, editors, *Handbook of Virtual Environments*, pp. 467–491. CRC Press, 2014.

[DPS99] ERNESTO DE QUEIRÓS VIEIRA MARTINS, Marta Margarida Braz Pascoal, and JOSÉ LUIS ESTEVES DOS SANTOS. "Deviation algorithms for ranking shortest paths." *International Journal of Foundations of Computer Science*, **10**(03):247–261, 1999.

[DS96] Rudolph P Darken and John L Sibert. "Wayfinding strategies and behaviors in large virtual worlds." In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 142–149. ACM, 1996.

[ENK97] T Todd Elvins, David R Nadeau, and David Kirsh. "Worldlets—3D thumbnails for wayfinding in virtual environments." In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pp. 21–30. ACM, 1997.

[Enn12] Christopher G. Healeyand James T. Enns. "Attention and Visual Memory in Visualization and Computer Graphics." In *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, 2012.

[FH10] Matthew Fisher and Pat Hanrahan. "Context-based search for 3D models." In *ACM Transactions on Graphics (TOG)*, volume 29, p. 182. ACM, 2010.

[Fol98] Mark A Foltz. *Designing navigable information spaces*. PhD thesis, Massachusetts Institute of Technology, 1998.

[FOP16] Daniel J. Finnegan, Eamonn O'Neill, and Michael J. Proulx. "Compensating for Distance Compression in Audiovisual Virtual Environments Using Incongruence." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, pp. 200–212, 2016.

[FRS12] Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. "Example-based Synthesis of 3D Object Arrangements." In *ACM SIGGRAPH Asia 2012 papers*, SIGGRAPH Asia '12, 2012.

[FRS13] Frederic Font, Gerard Roma, and Xavier Serra. "Freesound Technical Demo." In *ACM International Conference on Multimedia (MM'13)*, pp. 411–412, Barcelona, Spain, 21/10/2013 2013. ACM, ACM.

[FSH11] Matthew Fisher, Manolis Savva, and Pat Hanrahan. "Characterizing structural relationships in scenes using graph kernels." In *ACM Transactions on Graphics (TOG)*, volume 30, p. 34. ACM, 2011.

[FYY16] Tian Feng, Lap-Fai Yu, Sai-Kit Yeung, KangKang Yin, and Kun Zhou. "Crowd-driven Mid-scale Layout Design." *ACM Transactions on Graphics*, **35**(4), 2016.

[Gib09] David Gibson. *The Wayfinding Handbook: Information Design for Public Places*. Princeton Architectural Press, 2009.

[GMF16] François G. Germain, Gautham J. Mysore, and Takako Fujioka. "Equalization matching of speech recordings in real-world environments." In *IEEE ICASSP 2016*, 2016.

[Gol99] Reginald G Golledge. "Human wayfinding and cognitive maps." *Wayfinding behavior: Cognitive mapping and other spatial processes*, pp. 5–45, 1999.

[GPM10] Eric Galin, Adrien Peytavie, Nicolas Maréchal, and Eric Guérin. "Procedural generation of roads." In *Computer Graphics Forum*, volume 29, pp. 429–438. Wiley Online Library, 2010.

[GWH16] Jan Gugenheimer, Dennis Wolf, Gabriel Haas, Sebastian Krebs, and Enrico Rukzio. "SwiVRChair: A Motorized Swivel Chair to Nudge Users' Orientation for 360 Degree Storytelling in Virtual Reality." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, pp. 1996–2000, 2016.

[Han17] Mitsuhiko Hanada. "Correspondence analysis of color–emotion associations." In *Color Research and Application*, 2017.

[HB96] Claudia Hendrix and Woodrow Barfield. "The sense of presence within auditory virtual environments." *Presence: Teleoperators & Virtual Environments*, **5**(3):290–301, 1996.

[HCL06] Raia Hadsell, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping." In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

[HHP09] Alex D Hwang, Emily C Higgins, and Marc Pomplun. "A model of top-down attentional control during visual search in complex scenes." *Journal of Vision*, **9**(5):25, 2009.

[HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics*, **4**(2):100–107, 1968.

[HRS17] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors." In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 3296–3297, 2017.

[HWP11] Alex D Hwang, Hsueh-Cheng Wang, and Marc Pomplun. "Semantic guidance of eye movements in real-world scenes." *Vision research*, **51**(10):1192–1205, 2011.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[IK01] Laurent Itti and Christof Koch. "Computational modelling of visual attention." *Nature reviews neuroscience*, **2**(3):194–203, 2001.

[JCS10] Léonard Jaillet, Juan Cortés, and Thierry Siméon. "Sampling-based path planning on configuration-space costmaps." *IEEE Transactions on Robotics*, **26**(4):635–646, 2010.

[Jer15] Jason Jerald. *The VR Book: Human-Cerntered Design for Virtual Reality*. Morgan & Claypool, 2015.

[KAB10] Johannes Kopf, Maneesh Agrawala, David Bargeron, David Salesin, and Michael Cohen. "Automatic Generation of Destination Maps." *ACM Trans. Graph.*, **29**(6):158:1–158:12, 2010.

[Kat12] Joel Katz. *Designing Information: Human Factors and Common Sense in Information Design*. Wiley, 2012.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by simulated annealing." *SCIENCE*, **220**(4598):671–680, 1983.

[KJ16] Arun Kulshreshth and Joseph J. LaViola Jr. "Dynamic Stereoscopic 3D Parameter Adjustment for Enhanced Depth Discrimination." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, pp. 177–187, 2016.

[KLD09] FZ. Kaghat, C. Le Prado, A. Damala, and P. Cubaud. "Experimenting with Sound Immersion in an Arts and Crafts Museum." *In: Natkin S., Dupire J. (eds) Entertainment Computing - ICEC 2009.*, **5709**, 2009.

[KSL96] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." *IEEE transactions on Robotics and Automation*, **12**(4):566–580, 1996.

[KW14] A. Kachkaev and J. Wood. "Automated planning of leisure walks based on crowd-sourced photographic content." In *46th Annual Universities' Transport Study Group Conference*, 2014.

[LCH17] Yen-Chen Lin, Yung-Ju Chang, Hou-Ning Hu, Hsien-Tzu Cheng, Chi-Wen Huang, and Min Sun. "Tell Me Where to Look: Investigating Ways for Assisting Focus in 360° Video." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017.*, pp. 2535–2545, 2017.

[LGA09] Robert Leech, Brian Gygi, Jennifer Aydelott, and Frederic Dick. "Informational factors in identifying environmental sounds in natural auditory scenes." *The Journal of the Acoustical Society of America*, **126**:3147–55, 12 2009.

[LHB10] William Lidwell, Kritina Holden, and Jill Butler. *Universal Principles of Design, Revised and Updated: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach Through Design*. Rockport Publishers, 2010.

[LLZ17] Yuwei Li, Xi Luo, Youyi Zheng, Pengfei Xu, and Hongbo Fu. "SweepCanvas: Sketch-based 3D Prototyping on an RGB-D Image." In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pp. 387–399. ACM, 2017.

[LLZ18] Dingzeyu Li, Timothy R. Langlois, and Changxi Zheng. "Scene-Aware Audio for 360°Videos." *ACM Trans. Graph. (SIGGRAPH)*, **37**(4), 2018.

[LMB14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. "Microsoft COCO: Common Objects in Context." *eprint arXiv*, (1405.0312), 2014.

[LW18] David Lindlbauer and Andy D Wilson. "Remixed reality: Manipulating space and time in augmented reality." In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 129. ACM, 2018.

[Lyn60] Kevin Lynch. *The Image of the City*. The MIT Press, 1960.

[Mac92] Ian Scott Mackenzie. *Fitts' law as a performance model in human-computer interaction*. PhD thesis, University of Toronto, 1992.

[Mit12] Briar Lee Mitchell. *Game design essentials*. John Wiley & Sons, 2012.

[MP03] Ernesto QV Martins and Marta MB Pascoal. "A new implementation of Yen's ranking loopless paths algorithm." *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, **1**(2):121–133, 2003.

[MRR53] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics*, **21**(6):1087–1092, 1953.

[MS06] D.R. Montello and Corina Sas. *Human Factors of Wayfinding in Navigation*, pp. 2003–2008. CRC Press/Taylor & Francis, Ltd., 2006.

[MSK10] Paul Merrell, Eric Schkufza, and Vladlen Koltun. "Computer-generated residential building layouts." In *ACM Transactions on Graphics (TOG)*, volume 29, p. 181. ACM, 2010.

[MSL11] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. "Interactive furniture layout using interior design guidelines." In *ACM Transactions on Graphics (TOG)*, volume 30, p. 87. ACM, 2011.

[MSS13] Josh H McDermott, Michael Schemitsch, and Eero P Simoncelli. "Summary statistics in auditory perception." *Nature neuroscience*, **16**(4):493–498, 2013.

[MT13] Hui Miao and Yu-Chu Tian. "Dynamic robot path planning using an enhanced simulated annealing approach." *Applied Mathematics and Computation*, **222**:420–437, 2013.

[MVL18] Pedro Morgado, Nuno Vasconcelos, Timothy Langlois, and Oliver Wang. "Self-Supervised Generation of Spatial Audio for 360°Video." In *Neural Information Processing Systems (NeurIPS)*, 2018.

[NA08] Khaled Nassar and Ahmed Al-Kaisy. "Assessing sign occlusion in buildings using discrete event simulation." *Automation in Construction*, **17**(7):799–808, 2008.

[NDH17] Cuong Nguyen, Stephen DiVerdi, Aaron Hertzmann, and Feng Liu. "CollaVR: Collaborative In-Headset Review for VR Video." In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST 2017, Quebec City, QC, Canada, October 22 - 25, 2017*, pp. 267–277, 2017.

[Nil14] Nils J Nilsson. *Principles of artificial intelligence*. Morgan Kaufmann, 2014.

[NLL17] Danil Nagy, Damon Lau, John Locke, Jim Stoddart, Lorenzo Villaggi, Ray Wang, Dale Zhao, and David Benjamin. "Project Discover: An application of generative design for architectural space planning." In *SimAUD 2017 Conference proceedings: Symposium on Simulation for Architecture and Urban Design*, 2017.

[NOB16] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D Wilson. "SnapToReality: Aligning augmented reality to the real world." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 1233–1244. ACM, 2016.

[OAH14] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. "Learning layouts for single-pagegraphic designs." *IEEE Transactions on Visualization and Computer Graphics*, **20**(8):1200–1213, 2014.

[OAH15] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. "DesignScape: Design with interactive layout suggestions." In *Proceedings of the 33rd annual ACM Conference on Human Factors in Computing Systems*, pp. 1221–1224. ACM, 2015.

[OOK12] Masashi Okada, Takao Onoye, and Wataru Kobayashi. "A ray tracing simulation of sound diffraction based on the analytic secondary source model." *IEEE Transactions on Audio, Speech, and Language Processing*, **20**(9):2448–2460, 2012.

[OSG02] James F O'Brien, Chen Shen, and Christine M Gatchalian. "Synthesizing sounds from rigid-body simulations." In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 175–181. ACM, 2002.

[PA92] Romedi Passini and Paul Arthur. "Wayfinding: People, signs, and architecture.", 1992.

[PAB08] Nuria Pelechano, Jan M Allbeck, and Norman I Badler. "Virtual crowds: Methods, simulation, and control." *Synthesis Lectures on Computer Graphics and Animation*, **3**(1):1–176, 2008.

[Pas92] Romedi Passini. *Wayfinding in Architecture*. John Wiley and Sons Inc, 1992.

[Pay09] Andrew Phillip Payne. *Understanding Change in Place: Spatial Knowledge Acquired by Visually Impaired Users Through Change in Footpath Materials*. ProQuest, 2009.

[PCL16] Xufang Pang, Ying Cao, Rynson WH Lau, and Antoni B Chan. "Directing user attention via visual flow on web designs." *ACM Transactions on Graphics (TOG)*, **35**(6):240, 2016.

[PDS17] Vanessa C. Pope, Robert Dawes, Florian Schweiger, and Alia Sheikh. "The Geometry of Storytelling: Theatrical Use of Space for 360-degree Videos and Virtual Reality." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017.*, pp. 4468–4478, 2017.

[PHA17] Amy Pavel, Björn Hartmann, and Maneesh Agrawala. "Shot Orientation Controls for Interactive Cinematography with 360 Video." In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST 2017, Quebec City, QC, Canada, October 22 - 25, 2017*, pp. 289–297, 2017.

[PMB16] Chi-Han Peng, Niloy J Mitra, Fan Bao, Dong-Ming Yan, and Peter Wonka. "Computational Network Design from Functional Specifications." *ACM Transactions on Graphics*, **35**(4), 2016.

[PYW14] Chi-Han Peng, Yong-Liang Yang, and Peter Wonka. "Computing layouts with deformable templates." *ACM Transactions on Graphics (TOG)*, **33**(4):99, 2014.

[QFG16] Yuting Qiang, Yanwei Fu, Yanwen Guo, Zhi-Hua Zhou, and Leonid Sigal. "Learning to Generate Posters of Scientific Papers." In *AAAI*, pp. 51–57, 2016.

[Rau01] M Raubal. "Human wayfinding in unfamiliar buildings: a simulation with a cognizing agent." *Cognitive Processing*, (2-3):363–388, 2001.

[RRW18] Katja Rogers, Giovanni Ribeiro, Rina R. Wehbe, Michael Weber, and Lennart E. Nacke. "Vanishing Importance: Studying Immersive Effects of Game Audio Perception on Player Experiences in Virtual Reality." In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*, p. 328, 2018.

[RSM10] Nikunj Raghuvanshi, John Snyder, Ravish Mehra, Ming Lin, and Naga Govindaraju. "Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes." *ACM Transactions on Graphics (TOG)*, **29**(4):68, 2010.

[Sal16] Michael Salmond. *Video Game Design: Principles and Practices from the Ground Up*. Bloomsbury, 2016.

[SCB14] Manolis Savva, Angel X. Chang, Gilbert Bernstein, Christopher D. Manning, and Pat Hanrahan. "On Being the Right Scale: Sizing Large Collections of 3D Models." In *SIGGRAPH Asia 2014 Workshop on Indoor Scene Understanding: Where Graphics meets Vision*, 2014.

[SIV17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning." In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[SKB17] K. Sharma, A. C. Kumar, and S. M. Bhandarkar. "Action Recognition in Still Images Using Word Embeddings from Natural Language Descriptions." In *2017 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 58–66, March 2017.

[SSH18] Eldon Schoop, James Smith, and Bjoern Hartmann. "HindSight: Enhancing Spatial Awareness by Sonifying Detected Objects in Real-Time 360-Degree Video." In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018*, p. 143, 2018.

[ST05] Wei Shao and Demetri Terzopoulos. "Autonomous pedestrians." In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 19–28. ACM, 2005.

[STB14] Ruben M Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. "A survey on procedural modelling for virtual worlds." In *Computer Graphics Forum*, volume 33, pp. 31–50. Wiley Online Library, 2014.

[Ste92] Jonathan Steuer. "Defining virtual reality: Dimensions determining telepresence." *Journal of communication*, **42**(4):73–93, 1992.

[Ste94] Anthony Stentz. "Optimal and efficient path planning for partially-known environments." In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 3310–3317. IEEE, 1994.

[Sym15] Paul Symonds. "Designing a Wayfinding Strategy: Common Sense Approach." `http://www.travelwayfinding.com/designing-a-strategy/`, 2015.

[TCP06] Adrien Treuille, Seth Cooper, and Zoran Popović. "Continuum crowds." In *ACM Transactions on Graphics (TOG)*, volume 25, pp. 1160–1168. ACM, 2006.

[TF17] Anthony Tang and Omid Fakourfar. "Watching 360° Videos Together." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017.*, pp. 4501–4506, 2017.

[TKR17] Y. Tian, Y. Kong, Q. Ruan, G. An, and Y. Fu. "Hierarchical and Spatio-Temporal Sparse Representation for Human Action Recognition." *IEEE Transactions on Image Processing*, **PP**(99):1–1, 2017.

[TM16] James Traer and Josh H McDermott. "Statistics of natural reverberation enable perceptual separation of sound and space." *Proceedings of the National Academy of Sciences*, **113**(48):E7856–E7865, 2016.

[TMB07] Daniel Thalmann, Soraia Raupp Musse, and Adriana Braun. *Crowd simulation*, volume 1. Springer, 2007.

[Ueb10] Andreas Uebele. *Signage Systems and Information Graphics: A Professional Sourcebook*. Thames and Hudson, 2010.

[Uni] United States Equal Employment Opportunity Commission. "Americans with Disabilities Act of 1990." `https://www.eeoc.gov/eeoc/history/35th/1990s/ada.html`. [Online; accessed 2-July-2016].

[VAB09] Carlos A Vanegas, Daniel G Aliaga, Bedrich Benes, and Paul A Waddell. "Interactive design of urban spaces using geometrical and behavioral modeling." In *ACM Transactions on Graphics (TOG)*, volume 28, p. 111. ACM, 2009.

[VGA12] Carlos A Vanegas, Ignacio Garcia-Dorado, Daniel G Aliaga, Bedrich Benes, and Paul Waddell. "Inverse design of urban procedural models." *ACM Transactions on Graphics (TOG)*, **31**(6):168, 2012.

[Vis08] Peter M Visscher. "Sizing up human height variation." *Nature genetics*, **40**(5):489, 2008.

[VJ01] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features." In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pp. I–I. IEEE, 2001.

[VKW12] Carlos A Vanegas, Tom Kelly, Basil Weber, Jan Halatsch, Daniel G Aliaga, and Pascal Müller. "Procedural generation of parcels in urban modeling." In *Computer graphics forum*, volume 31, pp. 681–690. Wiley Online Library, 2012.

[Win02] Stephan Winter. "Modeling costs of turns in route planning." *GeoInformatica*, **6**(4):345–361, 2002.

[WLD18] Tomer Weiss, Alan Litteneker, Noah Duncan, Masaki Nakada, Chenfanfu Jiang, Lap-Fai Yu, and Demetri Terzopoulos. "Fast and Scalable Position-Based Layout Synthesis." *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[WLF14] Hsueh-Cheng Wang, Yafim Landa, Maurice Fallon, and Seth Teller. "Spatially prioritized and persistent text detection and decoding." In *Camera-Based Document Analysis and Recognition*, pp. 3–17. Springer, 2014.

[WLL12] Hsueh-Cheng Wang, Shijian Lu, Joo-Hwee Lim, and Marc Pomplun. "Visual Attention is Attracted by Text Features Even in Scenes without Text." In *Proceedings of Annual Meeting of the Cognitive Science Society (CogSci)*, 2012.

[WP12] Hsueh-Cheng Wang and Marc Pomplun. "The attraction of visual attention to texts in real-world scenes." *Journal of vision*, **12**(6):26, 2012.

[WSC18] Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. "Deep convolutional priors for indoor scene synthesis." *ACM Transactions on Graphics (TOG)*, **37**(4):70, 2018.

[WWP14] Chia-Chien Wu, Hsueh-Cheng Wang, and Marc Pomplun. "The roles of scene gist and spatial dependency among objects in the semantic guidance of attention in real-world scenes." *Vision research*, **105**:10–20, 2014.

[XCF13] Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. "Sketch2Scene: sketch-based co-retrieval and co-placement of 3D models." *ACM Transactions on Graphics (TOG)*, **32**(4):123, 2013.

[XFI14] Pengfei Xu, Hongbo Fu, Takeo Igarashi, and Chiew-Lan Tai. "Global beautification of layouts with interactive ambiguity resolution." In *Proceedings of the 27th annual ACM Symposium on User interface software and technology*, pp. 243–252. ACM, 2014.

[XFT15] Pengfei Xu, Hongbo Fu, Chiew-Lan Tai, and Takeo Igarashi. "GACA: Group-aware command-based arrangement of graphic elements." In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2787–2795. ACM, 2015.

[Yen71] Jin Y Yen. "Finding the k shortest loopless paths in a network." *management Science*, **17**(11):712–716, 1971.

[YYR17] Ya-Ting Yue, Yong-Liang Yang, Gang Ren, and Wenping Wang. "SceneCtrl: Mixed Reality Enhancement via Efficient Scene Editing." In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pp. 427–436. ACM, 2017.

[YYT11] Lap-Fai Yu, Sai Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F. Chan, and Stanley Osher. "Make it home: automatic optimization of furniture arrangement." *ACM Trans. Graph.*, **30**(4):86, 2011.

[YYT12] Lap-Fai Yu, Sai Kit Yeung, Demetri Terzopoulos, and Tony F. Chan. "DressUp!: Outfit Synthesis Through Automatic Optimization." *ACM Transactions on Graphics*, **31**(6):134:1–134:14, 2012.

[YYT16] Lap-Fai Yu, Sai-Kit Yeung, and Demetri Terzopoulos. "The clutterpalette: An interactive tool for detailing indoor scenes." *IEEE Transactions on Visualization and Computer Graphics*, **22**(2):1138–1148, 2016.

[ZCC16] Edward Zhang, Michael F Cohen, and Brian Curless. "Emptying, refurnishing, and relighting indoor spaces." *ACM Transactions on Graphics (TOG)*, **35**(6):174, 2016.

[ZF15] Jiulin Zhang and Xiaoqing Fu. "The Influence of Background Music of Video Games on Immersion." *Journal of Psychology and Psychotherapy*, **5**(191), 2015.

[ZXW17] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Dahua Lin, and Xiaoou Tang. "Temporal Action Detection with Structured Segment Networks." *CoRR*, **abs/1704.06228**, 2017.