3D User Interfaces Full

Theory

0_Overview

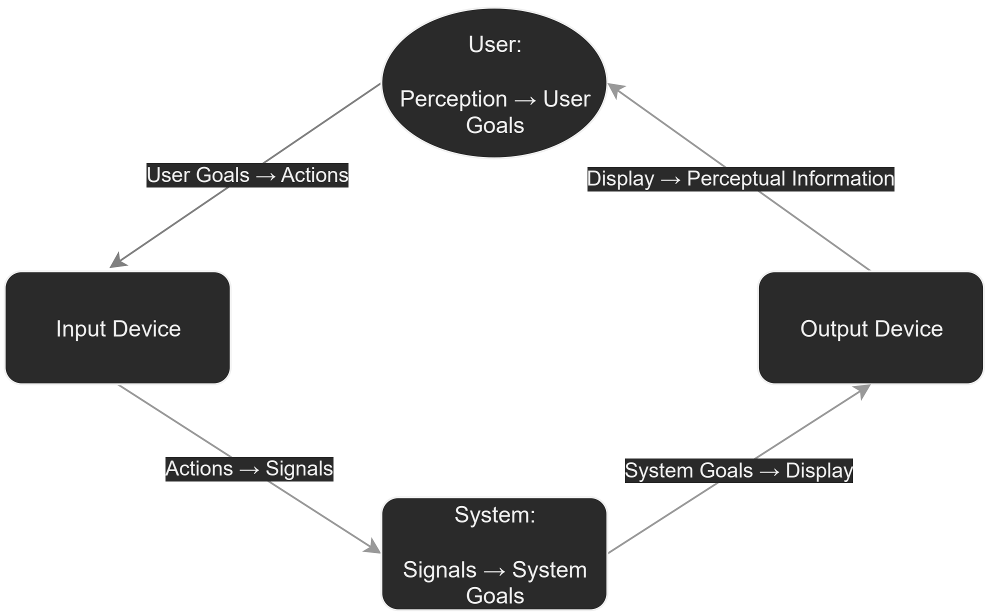
Table of Contents

Introduction to 3D User Interfaces

- Computers can help us analyze, explore and understand information (Expand our Experience)

$$stationary \xLeftrightarrow{AR \quad VR} mobile$$

Human Computer Interaction (HCI)



→ **User** perceives and interprets
→ **System** interprets input to output

→ System amplifies User in Tasks / Intelligence

Concepts

WIMP GUIs (since 1970s)

WIMP: Windows, Icons, Menus, Pointers
GUI: Graphical User Interface

Non-WIMP UIs (since 1980s)

Input: Speech, Gestures, Trackers, Sensors
Output: Head-Mounted Displays (HMDs), Heads-Up Display (HUD), 3D-Graphics, Sounds, Haptics
Computer: Networked interaction between Systems, Peer-to-peer, client-server

Augmented Reality as combination of real + virtual space in real time as a part of the Mixed Reality Conitnuum

Mixed Reality Continuum (Milgram and Kishino 94)

Mixed Reality describes the field in which we interact with a given combination of Enviroments (virtual and real):

$$\overbrace{Real\ Enviroment \leftarrow Augmented\ Reality\ (AR) \leftrightarrow Augmented\ Virtuality\ (AV) \rightarrow Virtual\ Envirome}^{Mixed\ Reality\ (MR)}$$

3D User Interface

- "UI that involves 3D interaction"
  → Interactive Systems with 3D Graphics but no 3D Interaction
  → 3D Interaction with a classical 2D Input
  → 3D Interaction with non-spatial Input (eg. Speech)
- **Focus** here is on interaction using 3D Spatial Input

Application Areas ★

- Design and Prototyping (Architecture, 3D Modelling)
- Psychiatric Treatment
- Scientific Visualization
- Tourism
- Collaboration
- Assistance
- Entertainment

#DoF #isomorphic

Selection and Manipulation

Design Guidelines

1. Use existing Manipulation Techniques (best case you already know them / tried them yourself)
2. Use Task-Analysis when choosing 3D Manipulation Technique
3. Match interaction to technique
4. Pointing for Selection and Virtual-Hand for Manipulation
5. **non-Isomorphic** techniques are useful and intuitive

Categorization

- **Selection**: Picking an Object with your hand
- **Positioning**: Moving an Object
- **Rotation**: Rotating Object from one pose to another

→ Select a representative subset of requirements for a specific task, which methods of interaction are needed

Interaction Techniques and Input Devices

DoF - Degrees of Freedom

- describes the degrees of possible movement with interactive device
  → Desktop Mouse: 2 DoFs (x-y horizontal interaction)
  → Game Controllers. $\geq$ 4 DoFs (2+2 for each hand)

Isomorphic

- Control: absolute or relative positon of hand (mouse, trackers, ...)
- View: strict geometrical 1 to 1 mapping with possible constraints

Isometric / Isotonic (Non-isomorphic)

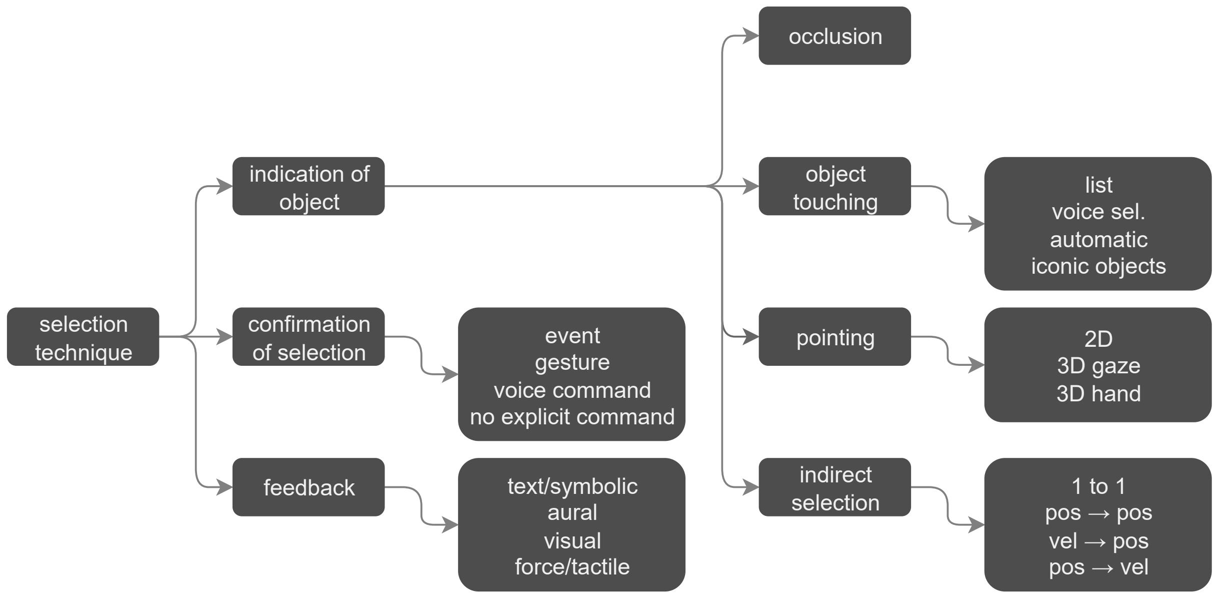
- Control: forces applied and rate of change (Joy-Stick)
- View: better reality, specifically for 3D virtual Enviroments with "magic" virtual tools (ie.: laser beams)
- **isometric:** senses foce but doesnt move
- **isotonic:** senses displacement

Task Classifications

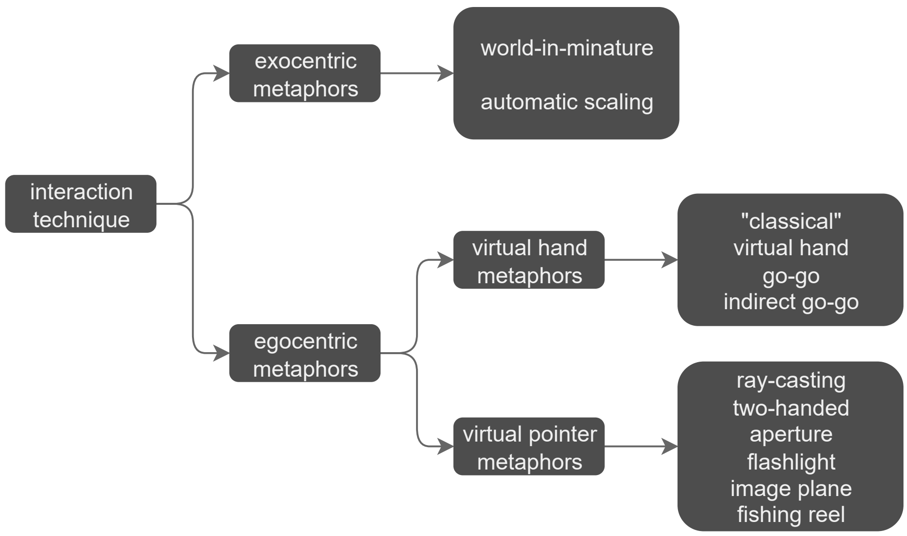
Decomposition

1. Decompose Task into Selection and Manipulation Techniques
2. Classify by "building blocks"

Classification of Selection Techniques:



Classification of interaction techniques:



Exocentric: Third-Person, Birds-eye view
Egocentric: Through the Users Eyes, First-Person

Metaphor

→ rely on known mental ideas that can be reused for the user

Virtual Pointer Methaphors - Interaction by Pointing

- generally we **raycast** from a hand-position into the hand direction and thus select or interact with an enviroment

- this can be combined for a two-handed **raycast** for more precise selection
- **flashlight** raycast where a cone is defined in the hand direction for selection, which requires less precision

Virtual Hand Metaphors - Interaction by Grasping

- Recognize Hand Movement or Pressure on 3D Controller to select and interact with objects

Exocentric Metaphors - World in miniature

- interact with an overview or downscaled model of your object
  → Metaphors can be combined for better user experience, most commonly Pointing and Grasping in VR Scenarios

Desktop 3D Manipulation

- 6 DoF Devices can be *expensive* → combination of 2DoF Devices or switch between such Devices can provide an adequate alternative

Travel

Problem: How to model Travel (Exploration, Manouvering and Search) ?

Design Guidelines

1. Match Travel Technique to Application
2. consider both natural and "artificial" techniques
3. consider combination of travel, display and input technique
4. provide multiple options for user
5. use transitional motions to reduce dizziness and improve orientation

Travel Tasks

Exploration:

- direct control of viewpoint
- little cognitive load

Search:

- naive search as "version of exploration"
- directed search (with wayfinding aids)
- different starting knowledge influences result

Manouvering:

- reposition local viewpoint
- must be intuitive
- very fast
- tight feedback loop
- otherwise frustrating

Short-Range: High-Precision
Medium-range: Combination?
Long-Range: Velocity Control, teleportation (low-precision high-speed)

Travel Techniques

	active	passive
physical	user motion	user-initiated, then system-animated
virtual	desktop UIs	animated

- active: User Controls Viewpoint
- passive: System controls Viewpoint
- physical: User body physically translates / rotates (Valve Index?)
- virtual: Body is stationary/irrelevant

Example: **VR-Roller Coaster** is passive (except Viewpoint) and virtual (except HMD that controls viewpoint)

Physical Techniques

- movement within a pre-defined VR Space
- locomotion by stationary movement (e-bike)

Steering Techniques

- User only controls direction of motion
- use gaze of HMD to travel along direction
- use pointing as direction

Route-Planning Techniques

- draw predefined path or mark locations along path

Target-based Techniques

- go directly to specific viewpoint (but not teleportation as this might decrease spatial orientation)
- use maps or specify goal and then system generates path from current location to target

Integrated Camera Controls on Desktop

- combine Mouse and keyboard mapping to achieve 6DoF Camera Control

Usability Evaluation

User-Task Analysis

- generate list of detailed task descriptions, sequences, realtionships, user work information flow
- start with protopying (ie.: with another person and on paper)

Evaluation Approaches

	Phases and Strategies	**Methods**
Before Development	Expert Reviews	Cognitive Walkthrough
	User Surveys	Heuristic Evaluation
→ What is needed?		
During Development	Usability Testing	Formative Evaluation
	User Surveys	Questionnaires
		Interviews and Demos
→ How is it working?		
After Development	Acceptance Test	Questionnaires
	Evaluation during active Use	Interviews and Demos
	User Surveys	
→ General Acceptance?		

NASA-TLX (Task Load Index)

- subjective workload assessment tool
- mental / cognitive / physical load of achieving task with your system
- score based on weighted average of demands, performance and frustration

SUS (System Usability Scale)

Subjective determination of usability:

1. Effectiveness (users can achieve their goals)
2. Efficiency (little effort is required for this achievement)
3. Satisfaction (experience was satisfactory)

Evaluation Metrics

1. System Perfomance Metrics (FPS, Network Delay, Latency, Distortion → In System)
2. Task Performance Metrics (Accuracy, Time to achieve tasks, Speed of learning → in System/Interview)
3. User Preference Metrics (Perceived eas of use, ease of learning, satisfaction → In Interview)

Usability Testing

- Usability is the measure of quality of the user experience
- Usefullness is the measure of perceived "practicality" or potential of an application

→ Test different Alternatives in Test Groups for comparison

Statistical Tools and Analysis

How to show data is different

0. Null-Hypothesis H_0 → there is only one distribution and ours is not any different ($H_1 = H_0$)
1. Prove H_1 by showing that it is unlikely that they are the same

Example: Proof that a new interaction technique is faster
0. Gather statistical Data on previous interaction technique T_0

1. Gather statistical Data on new technique T_1
2. Comparing the Data Sets suggests T_1 is faster
3. Proof by showing that it is more likely for $T_1 \neq T_0$ then $T_1 = T_0$ to be true

α and p -value

- α-Value: picked manually → probability of having a false-positive rejection of H_0
- p-Value: calculate → probability that your results are significant under H_0
- if $p > \alpha$ reject H_0

Type I and Type II Errors

- Type I ($\alpha\ Error$): false-positive → results show true but it actually is false
- Type II ($\beta\ Error$): false-negative → results show false but it is actually true

Ethical Issues

→ awarness of ethical issues for research involving humans is globally increasing

Human Categories	
Volunteers	
Person unable to give informed Consent	
Vulnerable individuals or groups	
Children/Minors	
Patients	
Healthy Individuals	

General Issues

Physical or Psychological Harm

- *virtual:* Cyber-Sickness
- *real:* Accidents, Exposure to catastrophic Situations
- Ethical Issues

Personal Data

- Private / Personal Information (*Identity or Identifying* - Retina, Face, Person)
- Compromosing Gestures / Movements

Dealing with Ethical Issues

- Inform the User as best as possible (without spoiling experiment or ruining the data)
- Always give the User options to leave or stop
- obtain proper **informed consent** (!)
- **anonymize** all data **directly**
- get approval from an ethics board

Human Cognition

Information Processing

- generally limited processing resources/capacity while perception is very fast

Attention

1. Selective Attention - choosing what to process
2. Focued attention - effort to minimize outside factors while processing
3. Divided Attention - ability to process multiple stimuli at a given poitn in time

→ Bottlenecks/Errors can occur through limits on our sensory system, **Overstimulation**
→ remove distracting factors
→ combining multiple different stimuli can be too much for multitasking

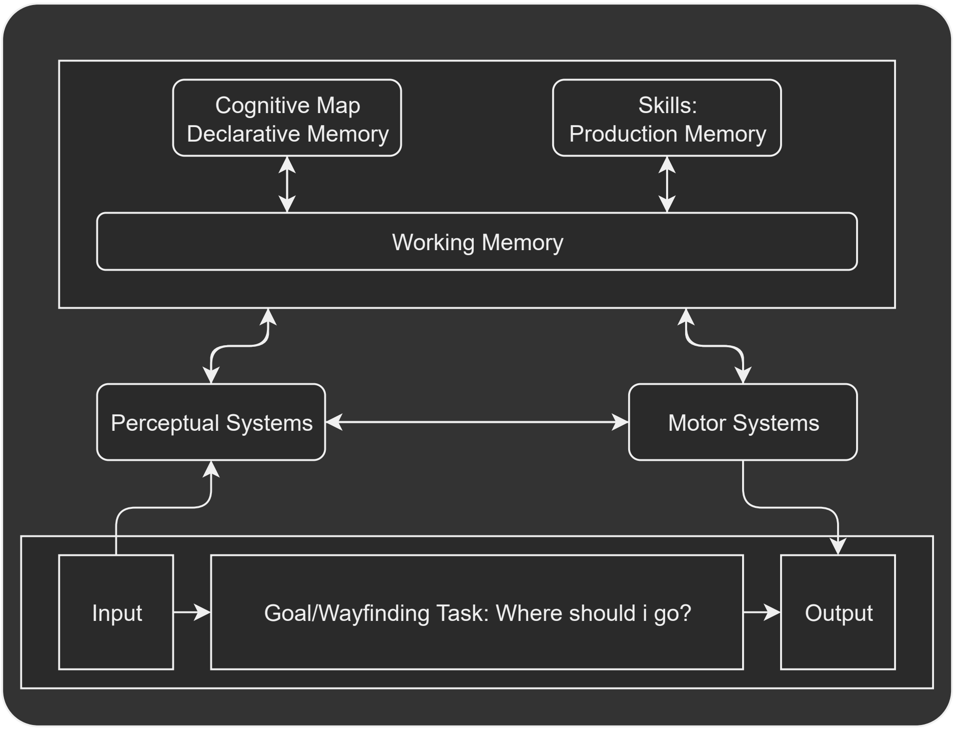
Speed and accurate Reaction Dependency

1. Stimulus-response compability
2. Stimulus-response consistency
3. Number of paralell tasks
4. Uncertainty
5. Pre-cueing

Cognition

- **Cognition** as the dirving force between generation and usage of knowledge
- Working-Memory vs. Longterm-Memory can influence results

In Wayfinding



Evaluating Cognitive Issues

1. Mental Load (task difficulty, attention / processing resources, resource allocation)
2. Human Error (lack of success, memory/sensory/motoric limitations)

→ NASA-TLX as a subjective Measure
→ task to draw/tell you what they remember
→ measure stress from body responses

Wayfinding

Design Guidelines

1. Match the cue to the task
2. Match cue to user skill
3. Cues should **not** be dominant features
4. avoid teleportation
5. integrate travel and wayfinding components

Wayfinding Tasks

Exploration:

- Acquitision of Spatial Knowledge

Search:

- Use of pre-existing spatial konwledge
- Acquisiton of further (directed or undirected) Spatial knowledge

Manouvering:

- small scale movements, e.g. for the identification within a region

→ similar to Travel Tasks

Spatial Knowledge

Landmark Knowledge:

- Visual Characteristics of the enviroment

Procedural Knowledge (route knowledge):

- Sequence of actions

Survey Knowledge:

- Topological Knowledge of Eniorment
- Highest Level of Knowledge → takes the longest to acquire

Reference Frames

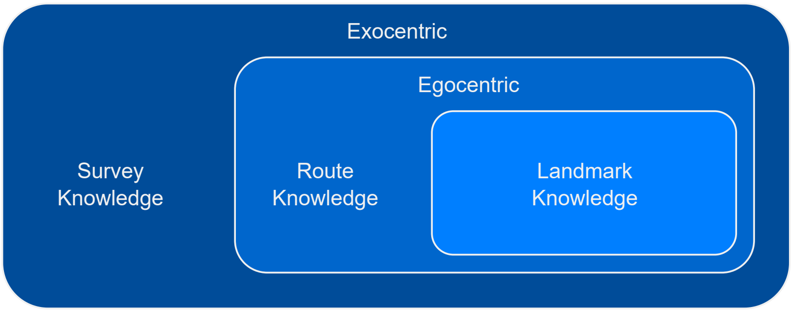
Egocentric:

- Headcentric / First Person centric reference frame for knowledge (landmarks, proc-knowledge: route)

Exocentric:

- relative to a foreign object or world (survey knowledge: map)

Transition: Egocentric → Exocentric when knowledge is acquired from Wayfinding



Wayfinding Support

User-centered

1. Motion Cues: avoid sensory conflicts
2. Multisensory Output: Audio (Enviroment/Objects), Tactility
3. Presence ("Feeling of being there"): better immersion creates better retention of information
4. Search Strategies: Teach user specific search strategies for objective (along coastline, landmarks, ...)

→ generally small FOV creates general disadvantages / discomfort for User → better starting point is big FOV

Enviroment-centered

Enviroment Design

- Examples: Zelda, Elden Ring, Residen Evil VIII
- use analogs akin to urban design principles (Paths, Edges, Districts, Nodes, Landmarks, Repetitive Structures)
- natural enviroment (horizon for directional information, atmospheric fog for distances)
- color and texture (lead with a yellow brick road)
- architectual design (special illuminaton, put a door where you want user to go)

Artificial Cues

1. Maps (**exocentric** representation → survey knowledge)
2. compasses (directional cues)
3. signs (effective when used sparingly, otherwise information overload)

4. well-known reference objects (coin, matchbox, chair, person for referncing size / distance)
5. Artifical Landmarks (Dark Souls II used many of these to give a fake senes of orientation in the world)
6. Trails (duh)

Evaluating Wayfinding Aids

1. Time-to-target measurement
2. path analysis (like real world, build a path from the way people actually take)
3. layout sketches by user

Interaction Techniques

Design Guidelines - Symbolic Input

1. Use Standardized Layouts if possible or if nothing special is needed
2. Haptic Feedback is needed, use physical buttons / feedback when possible
3. User Comfort is most important
4. Dont assume speech will always be a good solution
5. **Only** use special input when frequent input is needed

Design Guidelines - System Control

1. Avoid disturbing the flow of action in a task
2. Prevent unnecessary changes of the focus of attention
3. Avoid mode errors
4. Structure the functions in an application
5. Consider **multimodal** input

Differences 2D and 3D -UI

1. Users are often standing
2. Users may move around physically
3. no surface or vision to place keyboard
4. difficult to see keyboard in low-light enviroment

Symbolic Input

symbolic input (esp. text) is essential in desktop enviroments but hard in 3D UI

General Issues

- Frequency of Text Input
- Amount of Text
- Required Input Speed / User Reaction
- Primary vs. Secondary Task
- fatigue and blind typing difficult
- high levels of complexity for typing in 6DoF (3D Space) instead of 2DoF (Keyboard)

Keyboard-based ★

1. Miniature Keyboards
2. Low Key Count Keyboards (limited Input)

3. Chord Keyboards (key-combination defines input like chord on guitar)
4. Pinch keyboard (similar to chord keyboard but based on hand movement)
5. Soft (virtual) keyboards (hologram / virtual keys on any surface)

Pen-base

- pen-stroke gesture recognition
- arbitrary strokes are recorded and replayed (/categorized)

Gesture-Based

- sign language gestures
- numeric gestures
- instantaneous gestures (like opening optinons on oculus quest 2)

Speech-based

- no hands required (time for jazz hands)
- utilizes untapped input modality (speech)
- efficient and precise for large amounts of words

System Control

system commands are issued to request specific actions/functions from the system typically modelled WIMP-style

Metaphors for System Control in 3D UI Enivorment

System Control Method	Technique
Graphical Menu	Adapted 2D Menu
	1-DOF Menu
	3D Widget
Voice Command	Speech Recognition
	Spoke Dialog System
Gestural Command	Gesture
	Posture
Tools	Physical Tool
	Virtual Tool

Graphical Menus

1. Adapted 2D Menus (Opaque / Semi-transparent but the same as in 2D Enviroment)
   - Well known but widgets may occlude 3D Enviroment, and Enviroment can be distracting
2. 1-DOF Menus (Ring Menu, linear hand motion)
   - easy to use but only works for small lists
3. **TULIP** (Three Up, Labels in Palm)
   - Interaction via Pinch Gloves with Labels on fingers
   - easy to use, comfortable but needs special hardware
4. 3D Widgets (Menu Functionality tuned to object → diegetic interfaces)
   - non-context sensitive with interaction via pointers and buttons

Design and Implementation Issues

- Placement (accesibility vs. occlusion of 3D world)
- Selection (Mismatch 2D Menus, 3D Interaction)
- (Visual) represnetation and structure (avoid small objects and make it functional grouping)

Voice Commands

- speech recognition is user / accent dependent
- simple speech recognition can be nice but can be difficult if only input modality
- more possibility for mistakes as interfaces are not visible to user
- everything at once (instead make push to talk button for commands?)

Gestural Commands

1. Glove-based recognition
2. Camer-based recognition
3. Surface-based recognition
   → User need to discover or know the gesture
   → limited gestures should be used

Tools ★

1. Physical Tools ("Props" / Specialized Hardware/Controllers)
2. Virtual Tools (Tool Belt)
   → Examples: Virtual Smartphone in VR Space or Physical Pen selecting items

Design and Implementation Issues

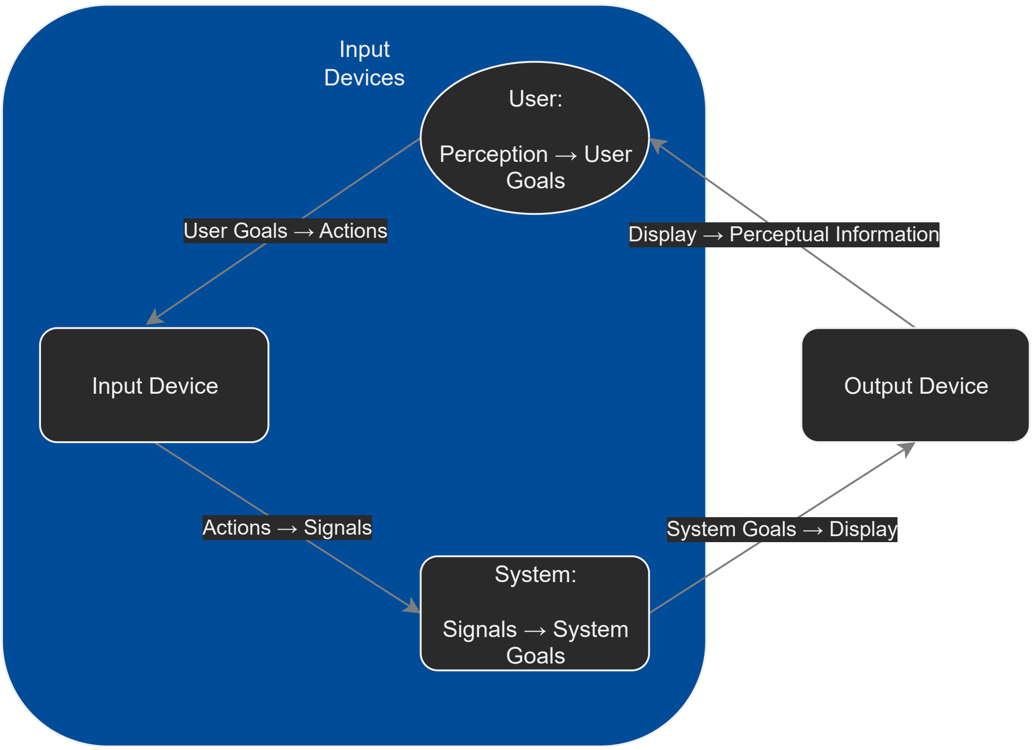
- function is communicated by form
- compliance between real and virtual world
- possible: blind operations (props must be designed for **tactile** interaction)
- Issue: Where are the props places when not needed?

→ Combine multiple techniques / input channels to control a System
→ Decoupling and error reduction

Input Devices

Design Guidelines

1. Device Ergonomics → Lightweight, Intuitive, Significant Information Transfer
2. Consider Number and type of input modes (DoFs)
3. Consider Types of Tasks
4. Many Specialized Devices > Few General Devices
5. Predict Time user needs to navigate as Empirical Evaluation
6. Speed + Short Learning Curve → free moving 6 DoF device
7. Comfort + Coordination + Precision → Desktop based 6 DoF



Property sensed by Device

	1 DoF	2 DoF	6 DoF	
Position	Bend/Linear Sensor	Tablet / Joystick	Trackers (Pos & Orientation)	Mechanical / Touch
Motion	Treadmill	Mouse / Trackball		Mechanical
Pressure	Torque sensor	Isometric Jostick	Spece Ball / Space Mouse	Touch

Desktop Input Devices

! Not 3D UI but still practical

1. Specialized Keyboards (commonly known input modality)
2. 2D Mice and Trackballs (relative motion tracking but difficult to use in immersive 3D)
3. Touch-based Tablets (Absolut Motion Tracking, useful for fully immersive displays)
4. Joysticks / Control Sticks (**isotonic** - interaction via movement, **isometric** - interaction via force)

	$Isotonic \Leftarrow$	$\Rightarrow Isometric$
Body (Muscle Tension)	Yes	Yes (but different)
Physical World (handle motion)	Yes	No
Virtual World (Object change)	Yes	Yes (but different)
Mapping	1:1	Arbitrary mathematical Mapping

Tracking Devices

Optical Sensors

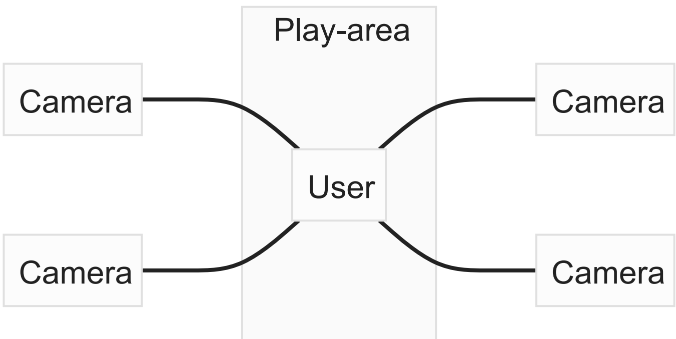
Outside-in vs. Inside-out

Placement Strategies

Sensors / Targets	**Mobile**	**Stationary**
Mobile	Inside-In	Inside-Out
Stationary	Outside-In	Outside-Out

Outside-In

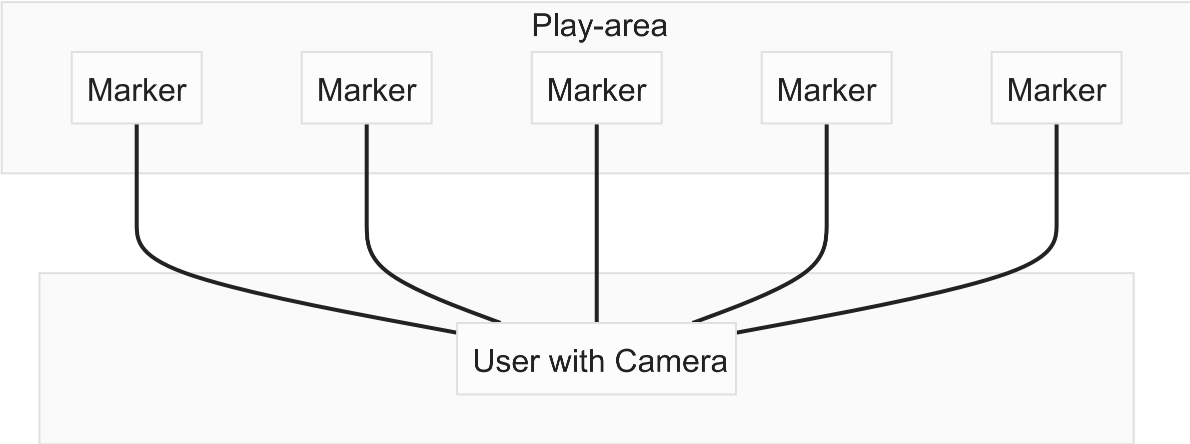
- Example: Stationary Sensor and Mobile Target (VR-HMD and fixed tracking Setup in Room)



→ Pro: Precise, Fast and Robust
→ Con: Small Range/ limited Area, Expensive, requires setup and line of sight

Inside-Out

- Example: Mobile Sensor and Stationary Target (Handheld Phone and AR-Marker)



	Marker-based	"Marker-less" - (nat. Features)	SLAM / PTAM
Pro	Cheap	Unobtrusive	Very Flexible
	Fast	Unnoticable	Unnotiable
	Precise		Relaxed Line of Sight
Con	Ugly markers	**Still** requires special markers	Untextured areas
	Line of sight !	Line of sight !	Slow
			Might drift

Inertial Sensing

Accelerometers

- measure acceleration through movement
- uses the inertia (resistance of matter to momentum) for measurement

Gyroscopes

- object-stabilized axis of rotation
- can measure rotational forces relative to rotation axis
- but is disturbed by gravity or vibration

→ mobile
→ low latency
→ minimal setup
→ low precision

Hybrid Systems

Hybrid Systems in mobile applications should automatically connect

Classification

Approaches in these Systems should use sensors with a high **difference**, specifically:

1. Complementary (different abilities → better data from more different tracking)
2. Competitive (strong approach → take the best of different options)
3. Cooperative (abilities complement each other → no system can work alone)

Example - Nintendo Wii

- Accelerometers
- IR Camera + IR Light bar (inside-out)
- Regular Controls
  → Complementary, Competitive and Cooperative !

Eye Tracking

- Pupil Tracking
- Head Mounted Cameras
- Applications include: psychophysical experiments, gaze-directed control

Data Gloves

1. Bend-Sensing Gloves (joint-angle measurements for gesture recognition)
2. Pinch Gloves (conductive material at finger tips to sense pinches, grabs, slides → selection / transformation)
3. Optically Tracked Fingertips (direct measurements of finger position)

3D Mice

- 3d (6DoF Tracker) → tethered or worn on users hand

Special Input Devices

- they exist

Direct Human Input

Speech Input

→ Where should Microphone be placed?
→ When should Computer liste?

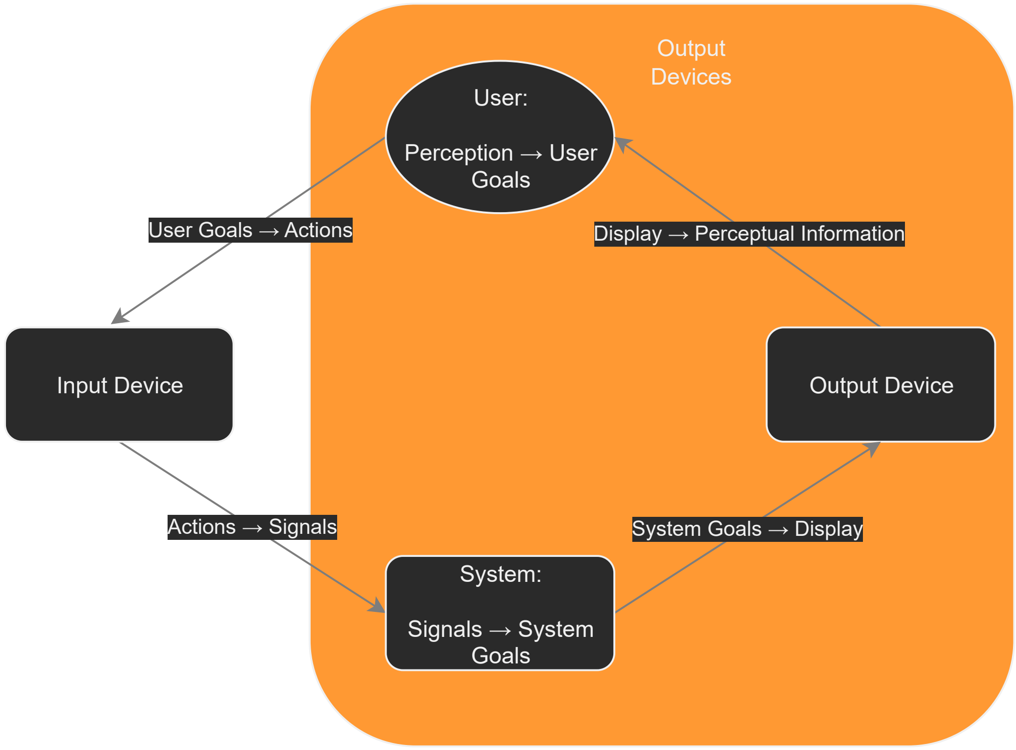
Bioelectric Input

→ Interpretation of signals between nerves / muscles directly from users body
→ ethical issues for tracking user body signals?

Brain Input

→ monitor brain waves
→ same ethical issues as with Bioelectric Input

Output Devices



Visual Displays

Characteristics

1. Field of Regard (FOR) and Field of View (FOV)
2. Spatial Resolution
3. Screen Geometry
4. Ergonomics
5. Periphal Vision
6. Cost
7. Hands Free
8. Motion Range

Depth Cues

Problem: Generally it is hard to determine Depth on Screens and Displays

→ Short distance Cues: Binoculary Disparity (Stereo Display / 3D Display)
→ Long Distance Cues: Motion Parallax and Occlusion

Visual Display Device Types

1. Monitors (self-explanatory → monocular depth cues)

2. surround-screen displays (CAVEs) → multiple projections, depth cues, motion parallax but hard and complicated
3. Workbenches ★
    - 2 Display Surfaces
    - L-Shaped (desk is screen and screen is screen)
4. Head-Mounted Displays (HMDs)
5. Heads-Up Displays (HUDs)

	Pro	Con
Monitors	inexpensive	small FOR/FOV
	high spatial resolution	no peripheral vision
	freedom of input	not very immsersive
		physical/virtual object occlusion
Surround-Screen displays (CAVEs)	Large FOV/FOR	require large amount of space
	make use of peripheral vision	expensive
	real and virtual objects can be mixed	physical/virtual object occulsion
Workbenches	higher spatial resolution	no peripheral vision
	devices can be tilted (desks)	physical/virtual object occlusion
	some permit 2D input on surface	

Auditory Displays

HRTF - Head Realted Transfer Functions: filters that modify soundwaves from a source to simulate interaction with listeners ears (listener specific)

3D Sound Localization Cues

1. Binaural Cues (Different Sounds on Left/Right for location, but directly in front/above/behind/below problematic)
2. Reverb can be difficult for localization but helps with distance measurement
3. Sound intensity as main way to determine distance

3D Sound genration

1. Binaural Mic Recording
2. Imitation, ie.: using HRTFs but very difficult

3. Auralization / Simulation → render sound field like actual room

Setup

1. Headphones (can be used with HMDs but can be uncomfortable and hard to simulate well)
2. External Speakers (user doesnt have to wear additional devices, but 3D sound must be generated specifically for user)

Audio in 3D Interfaces

1. Localization via 3D sound as a 3D Depth Cue (Wayfinding, Location)
2. Sonification (Turn Information into Sound → Enemy Special Attack?)
3. Ambient Effects (add realism / immersion)
4. **Sensory Substitution (substitute touch or haptic feedback with sound, like the fake buttons on phones)**

Haptic Displays

Cues

1. Tactile (force, vibration, pain using receptors under skin)
2. Kinesthetic (force applied to muscles / joints / tendons, maybe resistance in movement?)
   → Haptic Perception is generally combination of both (Handshake)

Displays

- Resolution typically in kHz as well as Spatial resolution (minimal proximity)
- Ergonomic (user safety and comfort are important)

	Pro	Con
Ground-referenced: physical link between user and ground point	accurate	limited movement
	high levels of force	
Body-referenced: device on body (glove)	more freedom of motion	user has to bear weight of device
	provide more control with direct manipulation	hard to put on
Tactile: Vibrators	small and lightweight	sensations might be incorrect
	useful for stimulating touch	small area
Hybrid	useful for combining tactile and force	more complex

	Pro	Con
		can be hard to wear
Passive: real cups or hammers or floor tiles	easy to design and use	limited by specificity
	useful for specific object haptics	

Practical

0_Overview

Smartphone Interaction

Inertial Measurement Units (IMUs)

Accelerometer

- measure linear acceleration
- measure gravitational acceleration

Gyroscope

- measure angular velocity
- triggers orientation changes

Magnetometer

- measure earths magnetic field
- can determine north

Isomorphic

- Control: absolute or relative positon of hand (mouse, trackers, ...)
- View: strict geometrical 1 to 1 mapping with possible constraints

Isometric / Isotonic (Non-isomorphic)

- Control: forces applied and rate of change (Joy-Stick)
- View: better reality, specifically for 3D virtual Enviroments with "magic" virtual tools (ie.: laser beams)
- **isometric:** senses foce but doesnt move
- **isotonic:** senses displacement

→ Homework 1 implemented isotonic control (Device sense movement, but only force)
→ IMUs can also be used to sense **movement patterns** (running, steps, driving a care, posture)

Unity Events

```
//Unity Events
public UnityEvent onPLayerDeath;
public void TakeDamage(float damage)
{
        health -= damage;
```

```
        if health < 0:
                onPLayerDeath.Invoke();
}


//C# Events:
public class PlayerHealth : MonoBehaviour
{
    float health=100;

    delegate void OnGameOver();
    public event OnGameOver onGameOver;
    public void TakeDamage(float damage)
    {
        health -= damage;
        if(health < 0)
        {
            onGameOver?.Invoke(); // Question (?) Mark is important, as it is a null-
pointer exception when no one subscribes
        }
    }
}

// Events need to be subscribed

public class GameController : MonoBehaviour
{
    void RestartGame()
    {
        // Restart the game!
    }

    private void OnEnable()
    {
        PlayerHealth.onGameOver += RestartGame;
    }
}
```

Selection and Manipulation

Affordance: What a User can do with device (clicking a button, scanning something)
Signifier: Hints for what you should do (visual frame, or click button symbol)

Selection Methods

Virtual Hand Method

- Collider attached to the virtual controller intersects with desired object
  → give siginifier to player (ie.: haptic feedback, color change)
  → touch event selects the object

Issues with Virtual Hand

- multiple targets might be selected when too close to each other

Raycast

- shooting laser into infinity and see if it connects to something

Issues with raycast

- Targets to close to each other makes it hard to select which
- if the target is far away the raycast might miss (flashlight method instead?)

Unity Raycast

```
public class ExampleClass : MonoBehaviour
{
    // See Order of Execution for Event Functions for information on FixedUpdate() and
Update() related to physics queries
    void FixedUpdate()
    {
        // Bit shift the index of the layer (8) to get a bit mask
        int layerMask = 1 << 8;

        // This would cast rays only against colliders in layer 8.
        // But instead we want to collide against everything except layer 8. The ~
operator does this, it inverts a bitmask.
        layerMask = ~layerMask;

        RaycastHit hit;
        // Does the ray intersect any objects excluding the player layer
        if (Physics.Raycast(transform.position,
transform.TransformDirection(Vector3.forward), out hit, Mathf.Infinity, layerMask))
```

```
        {
            Debug.DrawRay(transform.position,
transform.TransformDirection(Vector3.forward) * hit.distance, Color.yellow);
            Debug.Log("Did Hit");
        }
        else
        {
            Debug.DrawRay(transform.position,
transform.TransformDirection(Vector3.forward) * 1000, Color.white);
            Debug.Log("Did not Hit");
        }
    }
}
```

Raycast methods for Handheld VR

- Colliders can be **tracked Objects** in the real world
- Virtual Objects
- Either use center point of screen or touch position

Example: Raycasting from a Touchscreen Point into our Virtual Scene

```
if(Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
{
Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position)
if(Physics.Raycas(ray, out Hit))
        {
                ...
        }
}
```

3D Data structures

- 3D Geometrical Representations (Voxels, Meshes, Pointclouds)
- 3D Data Formats (.obj, .stl, .fbx)
- Object-oriented Formats (IFC, USD, GLTF)

Surface-based Data Structures (Meshes)

- Movies, Games, Engineering applications
- simple geometries consisting of triangles
- supported by most 3D Tools and hardware level graphic cards
- or CAD-World **Step**

Volumetric Representations (Voxels / Octrees)

- useful for non-surface oriented applications
- Computer-Tomography
- Densities and volumetric rendering
- only discrete representation and storage-intense

Human centered Development

XR and the search for the killer app

- Generally there are very few AR usages in Consumers Products that are readily used by everyone
- Social acceptance is not as high
- very niche target consumer audience for AR
- high stepping stone to learn AR interaction for what?

Use-Cases

Problem: How to understand users needs and goals
Solution: User-centered research

- The goal of a user is not to use your product !
- Goal $\neq$ Use Case

BPMN

- tool for user task analysis (Control Flow Diagram for Task Analysis)

Enviromental Analysis

- analysing the surrounding market / situation around your software idea

Personas

- not real people
- depict carefully researched and designed archetypes
  → can be used for **acceptance evaluation**
  → can be reused by marketing
  → avoid design for edge cases
  → act as point of communication for Dev Team

Examples for Car-Personas:

Allessandro	Marge	Dale
Go Fast	Be Safe	Haul big loads
Have fun!	be comfortable	be reliable

How to Design a Persona?

- start a hypothesis
- conduct in person interviews
- refine hypothesis
- sometimes its not possible to design complete personas → label provisional results as such
- it is better to write down biased assumptions (to be corrected) than to have unconcious bias

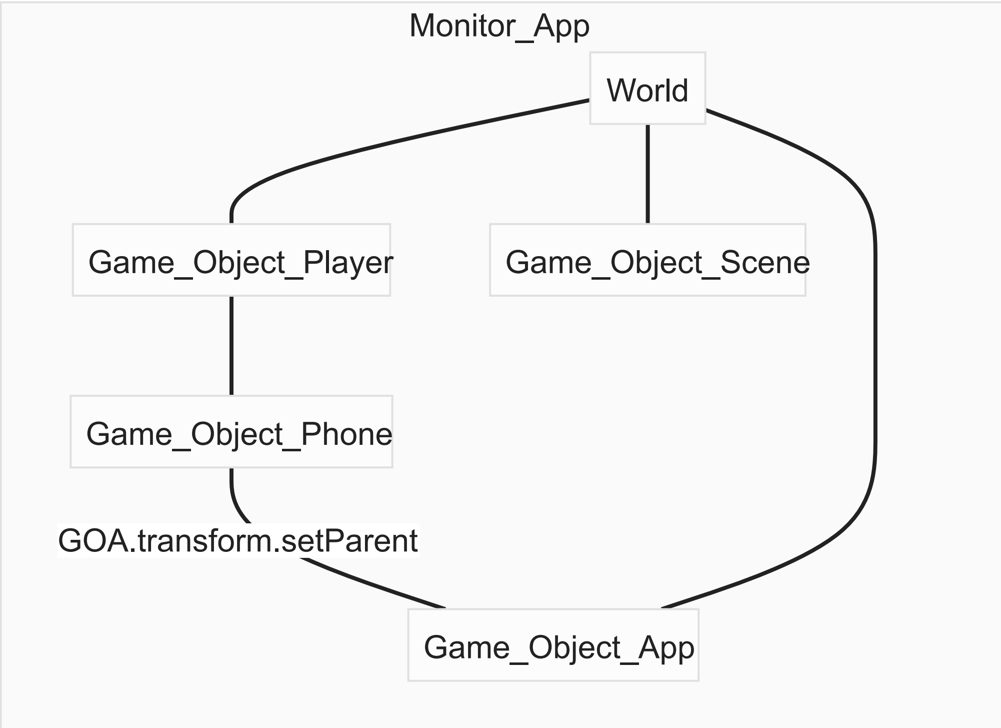
Approach

1. Write down hypothesis about goals of target group in respect to problem
2. Talk to people of the target group or research to reflect upon hypothesis
3. verbalize ad-hof personas with **behavioral patterns** and **goals**

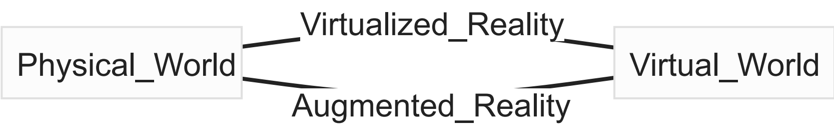
Advanced 3DUI

Scene Graphs

- also see [Augmented Reality Theory Scene Graphs]
- simple hierachy defined by relations between objects



Virtualized Reality

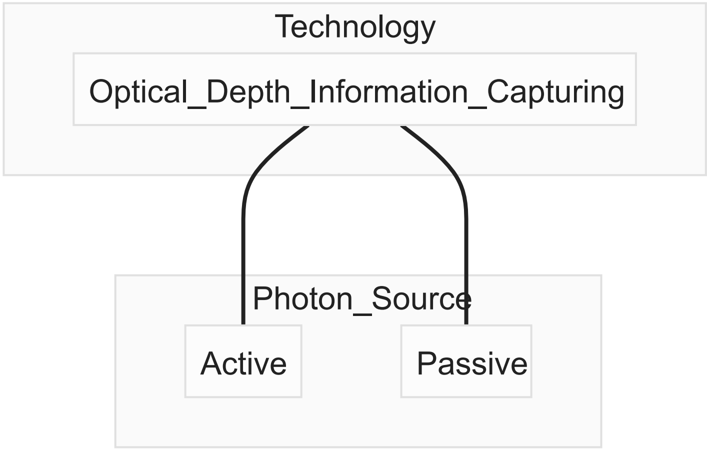


→ constructing virtual worlds from real scenes
→ asset creation for virtual enviroments
→ remote inspection
→ training

Asset Creation

1. Scanning existing Objects
2. Creating from Scratch

Optical Scanning Techniques



Active

- light source as active photon source that is then measured and used for calculation

1. SL - Structured Light calculation between projector and camera (Kinect 1st Gen)
2. iTOF - modulated time of Flight (Kinect newer Gens)
3. dTOF - direct Time of Flight (LIDAR Iphone)

Passive

- passive photon source → so light reflecting for photography

1. Photogrammetry - Multi View pictures get added together to 3D Model from many poses
2. Light-Field-based
3. Depths from Focus