



Virtual and Augmented Reality

CS-GY 9223/CUSP-GX 6004

<https://nyu-icl.github.io/courses/2022fall-vr-ar>

Prof. Qi Sun
qisun@nyu.edu | www.qisun.me

Logistics

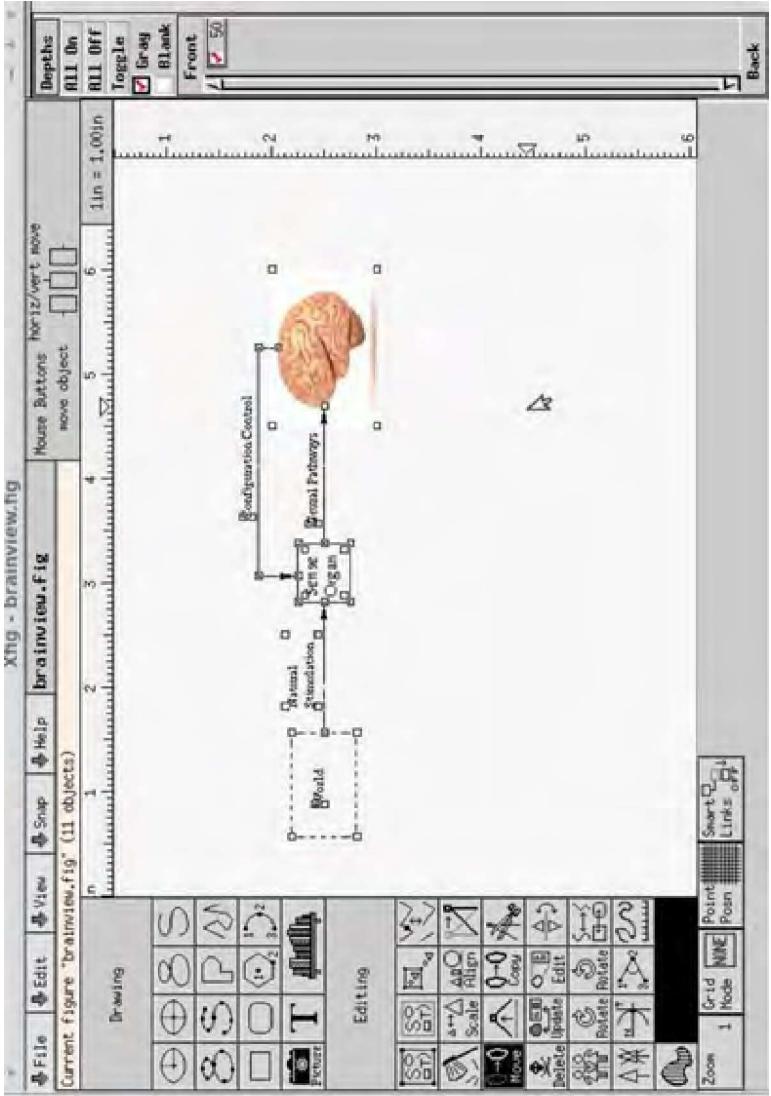
1. New AR-compatible loaner devices!
2. Assignment 3 release this week.

Universal Simulation Principle

“Any interaction mechanism from the real world can be simulated in VR.”

- Motor Programs and Remapping
- Locomotion
- Manipulation
- Social Interaction
- Alternative Interaction

Motor Programs and Remapping

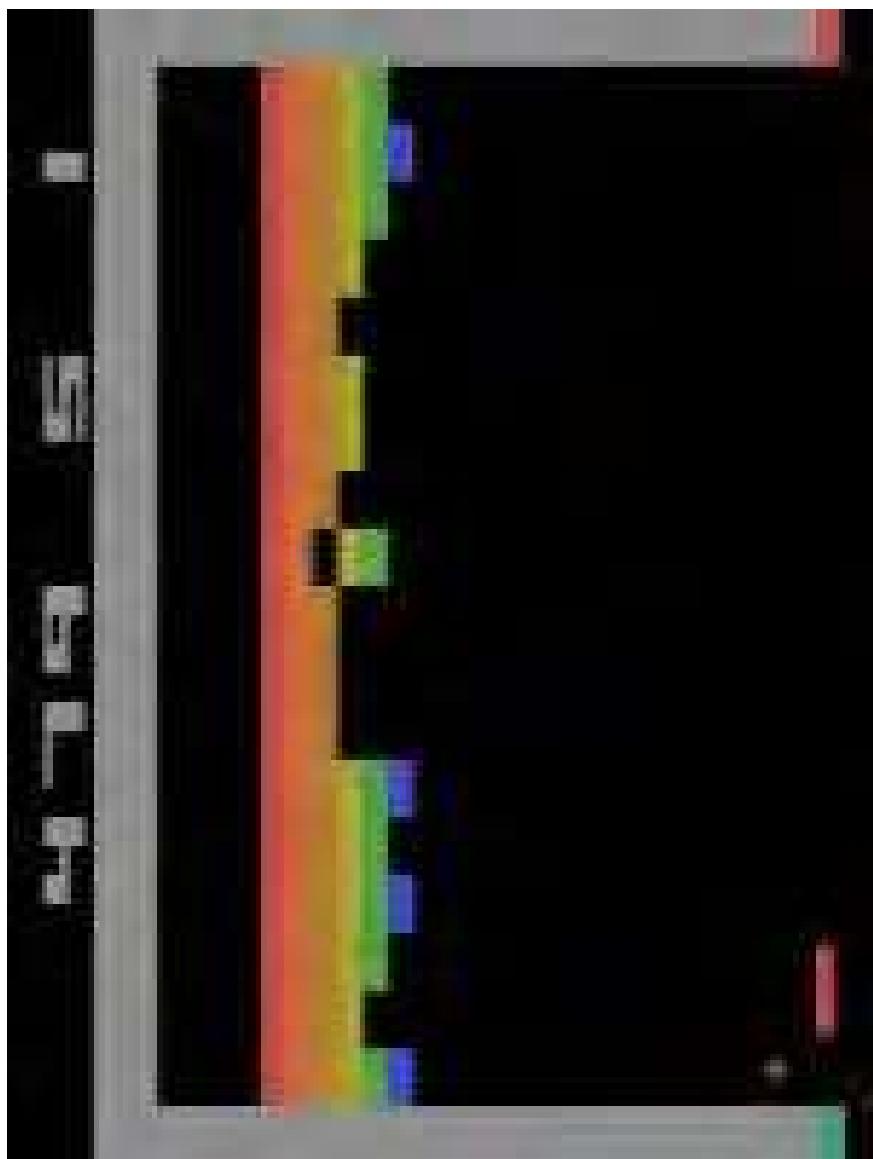


Reading: Human performance in six degree of freedom input control.
Shumin Zhai

Motor Programs and Remapping

1. Effectiveness for the task in terms of achieving the required speed, accuracy, and motion range, if applicable.
2. Difficulty of learning the new motor programs; ideally, the user should not be expected to spend many months mastering a new mechanism.
3. Ease of use in terms of cognitive load; in other words, the interaction mechanism should require little or no focused attention after some practice.
4. Overall comfort during use over extended periods; the user should not develop muscle fatigue, unless the task is to get some physical exercise.

sensor-feedback and motor control



Locomotion

All matched motions

LOCOMOTION SPECTRUM

All remapped motions



Real walking
with headset

Seated in swivel chair
with headset

Seated in fixed chair
with headset

Seated in fixed chair
and viewing a screen

Entire lookat
handled by controller

Yaw and translation
handled by controller

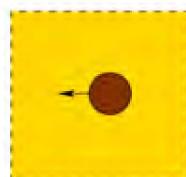
HTC Vive
CAVE systems

Gear VR
Google Daydream
Oculus Rift

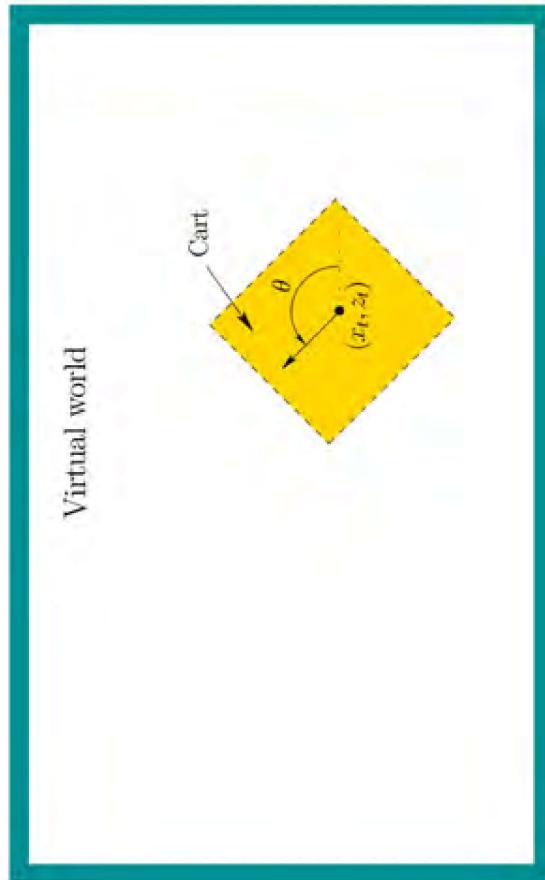
FPS game on screen
Nintendo Virtual Boy

Locomotion [Tutorial for Cardboard]

Real world



Virtual world

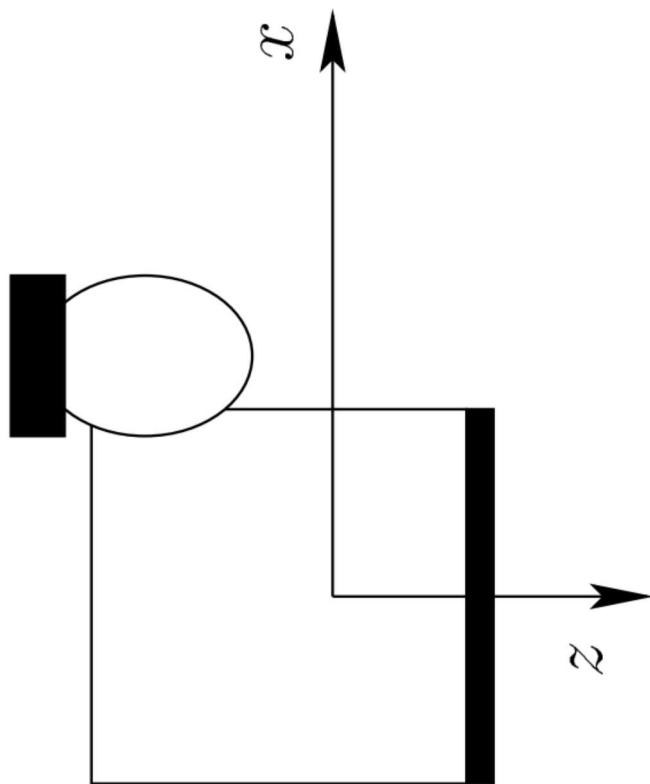


$$T_{cart} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & x_t \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

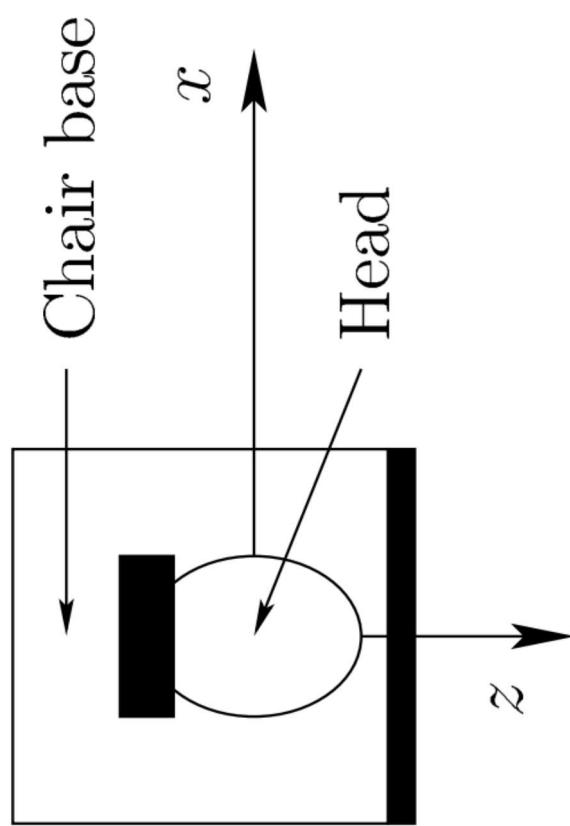
$$T_{eye} = (T_{track} T_{cart})^{-1} = T_{cart}^{-1} T_{track}^{-1}$$

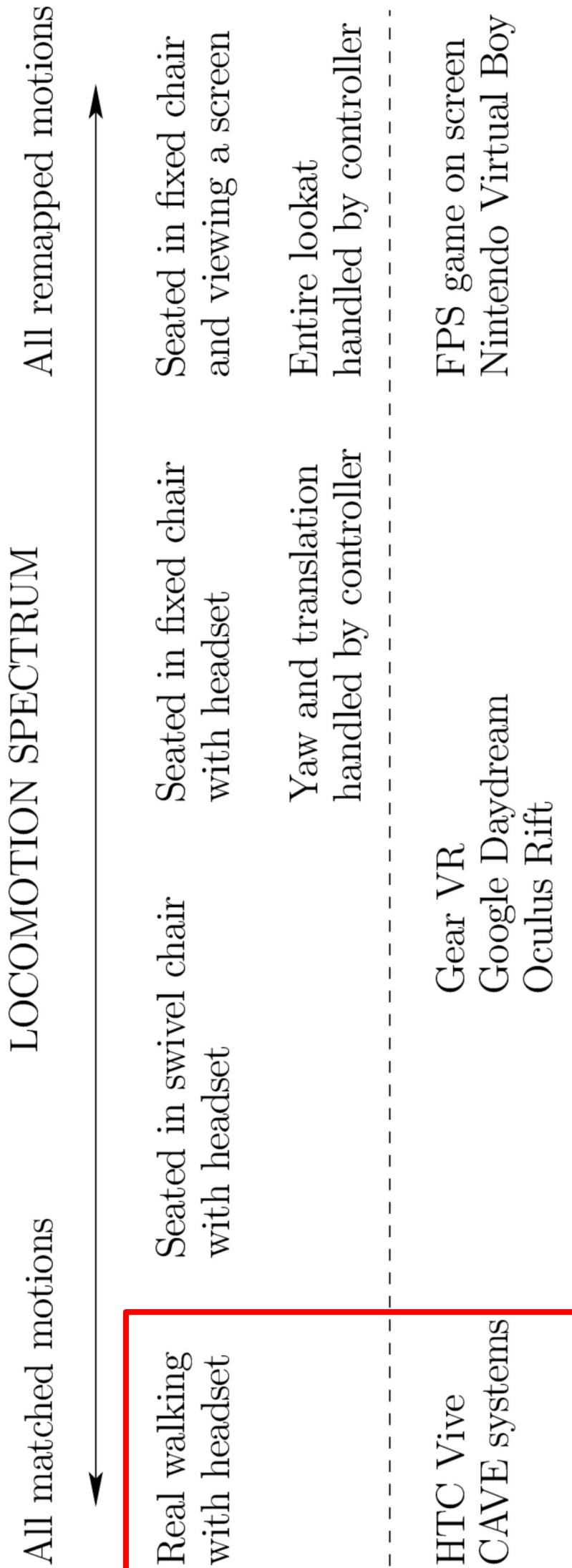
$$\begin{aligned} \dot{x}_t &= s \cos \theta & x_t[k+1] &= x_t[k] + \dot{x}_t \Delta t \\ \dot{z}_t &= s \sin \theta & z_t[k+1] &= z_t[k] + \dot{z}_t \Delta t. \end{aligned}$$

Should rotation axis be new head center or original xz origin?



Rotation axis is head center





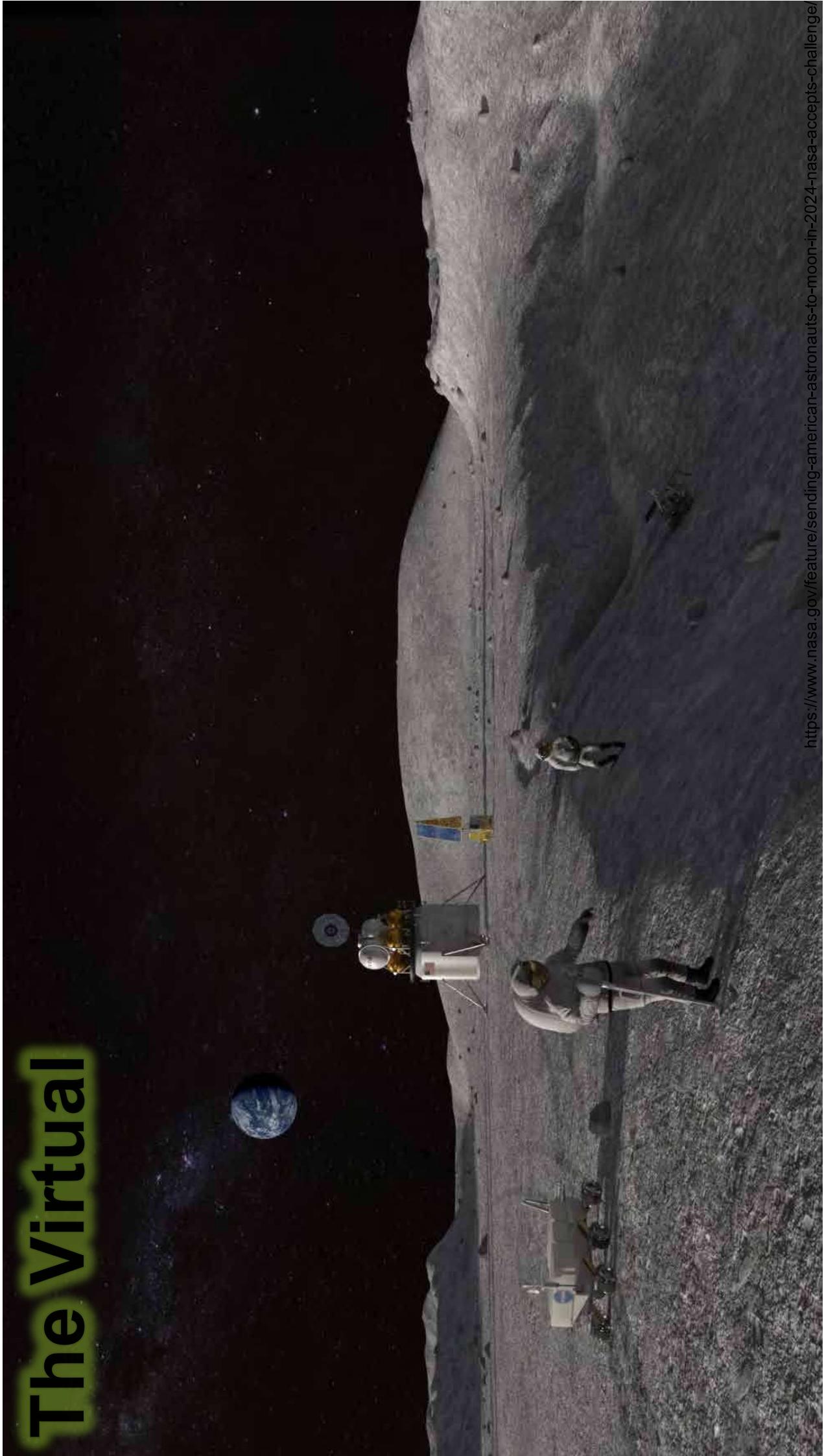
Redirected Walking

Current XR Controls are **Unsuitable** for Training



image credit: Matthew Noyes, NASA Hybrid Reality

The virtual



<https://www.nasa.gov/feature/sending-american-astronauts-to-moon-in-2024-nasa-accepts-challenge>

The Reality

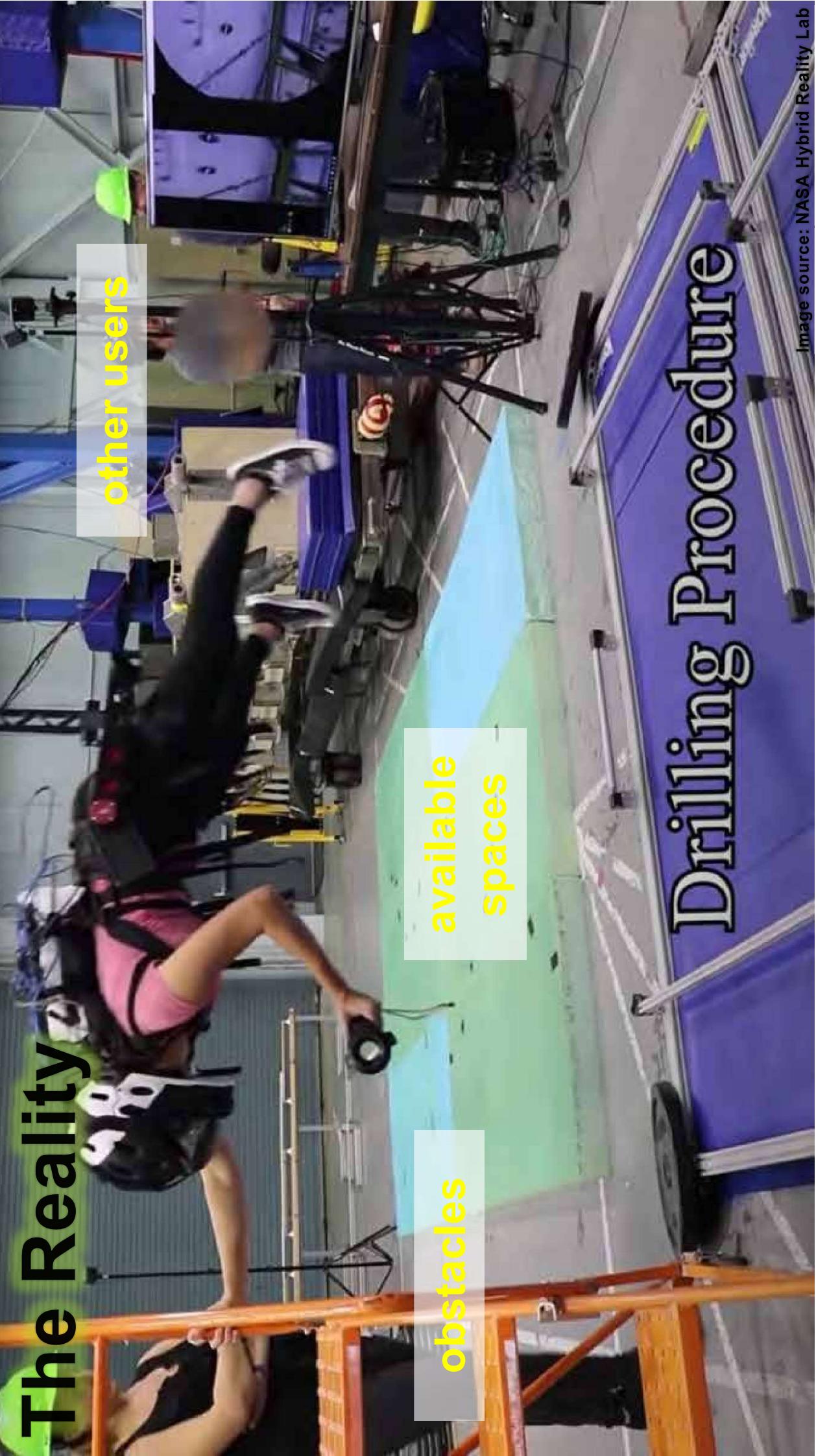
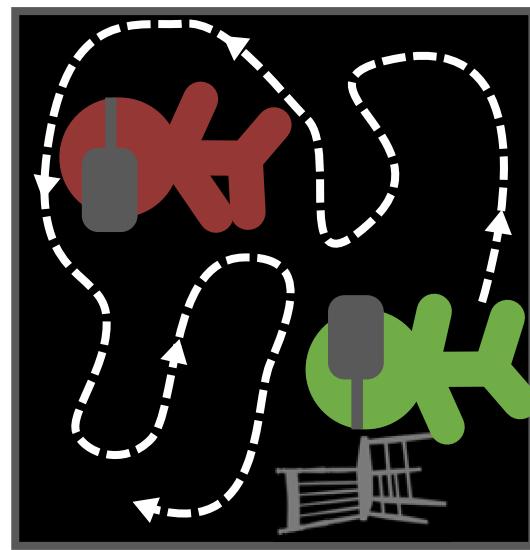
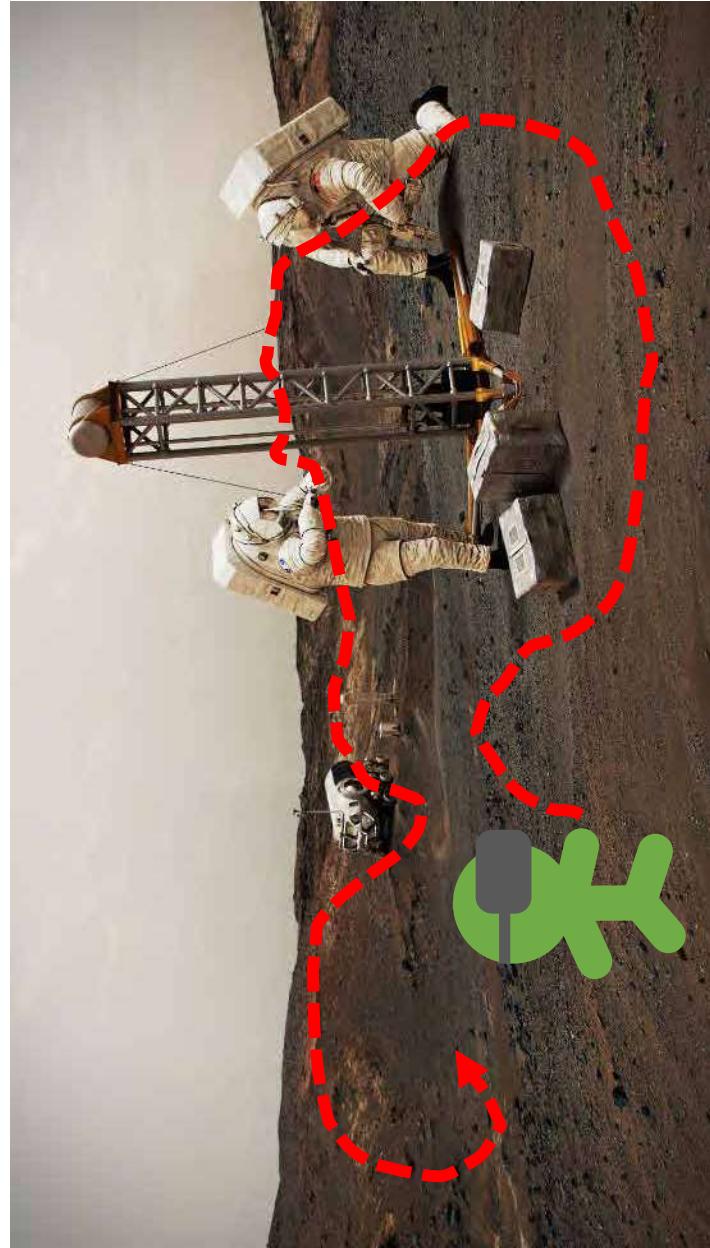
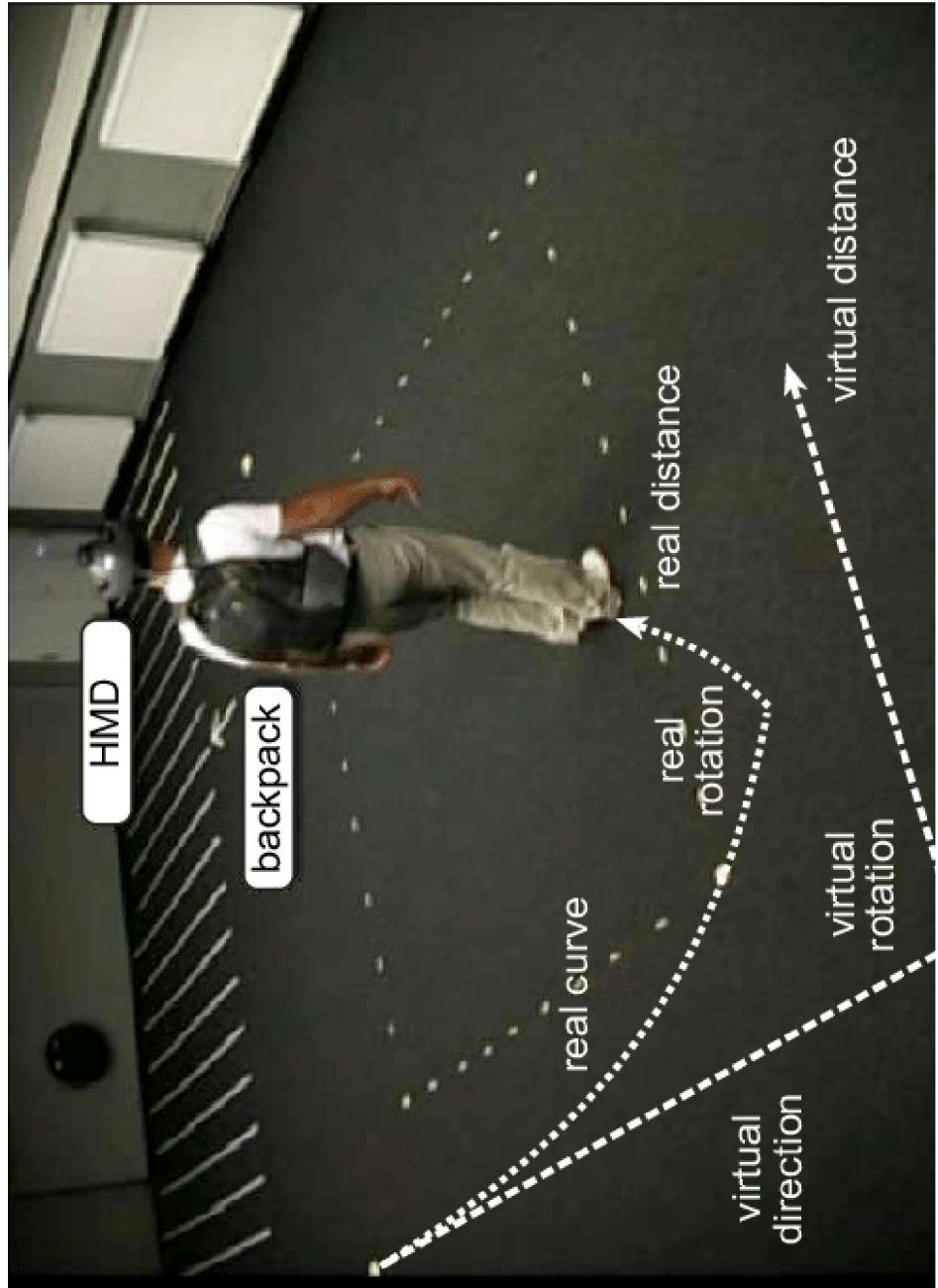


Image source: NASA Hybrid Reality Lab

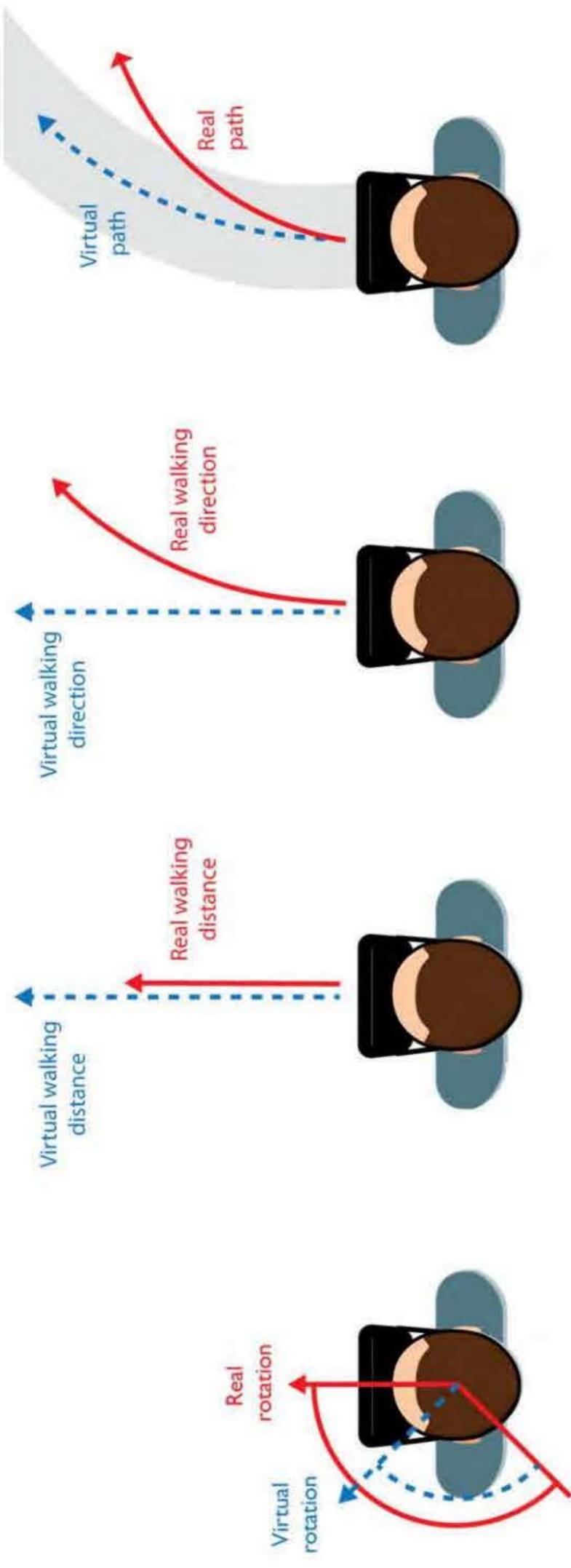
Redirected Walking – The Goal



Redirected Walking



Redirected Walking – Imperceptible Manipulation



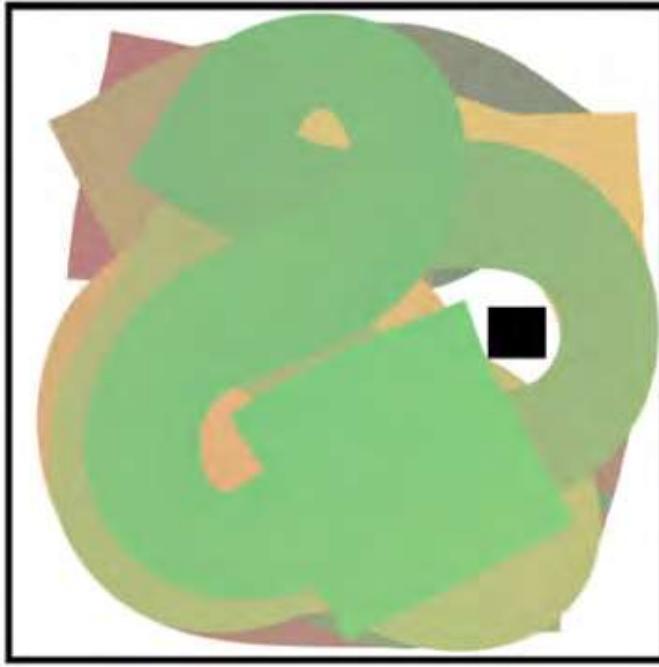
Read: “15 Years of Research on Redirected Walking in Immersive Virtual Environments”

Redirected Walking – Geometric Retargetting



Virtual scene
 $64m \times 33m$

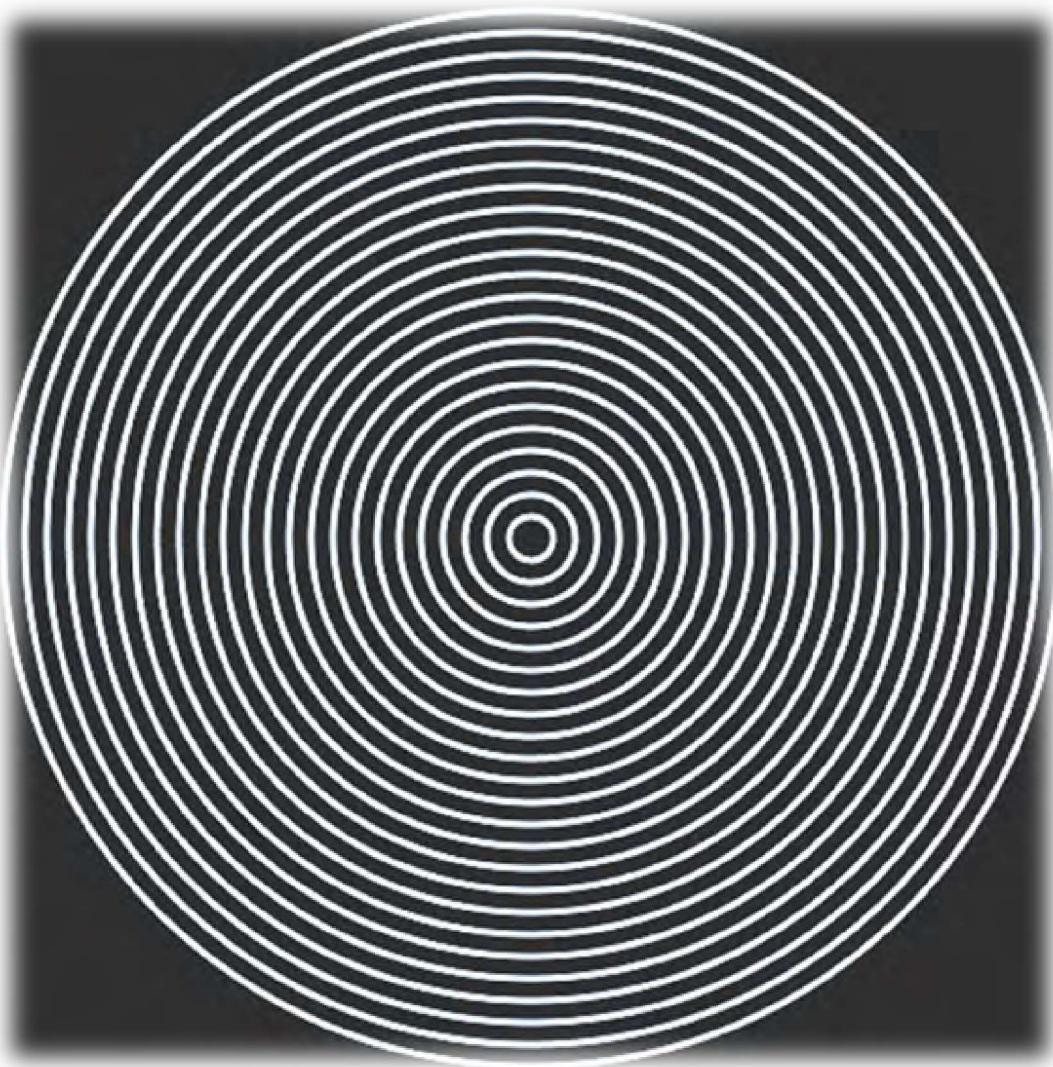
[Dong et al. 2017]



Physical space
 $10m \times 10m$

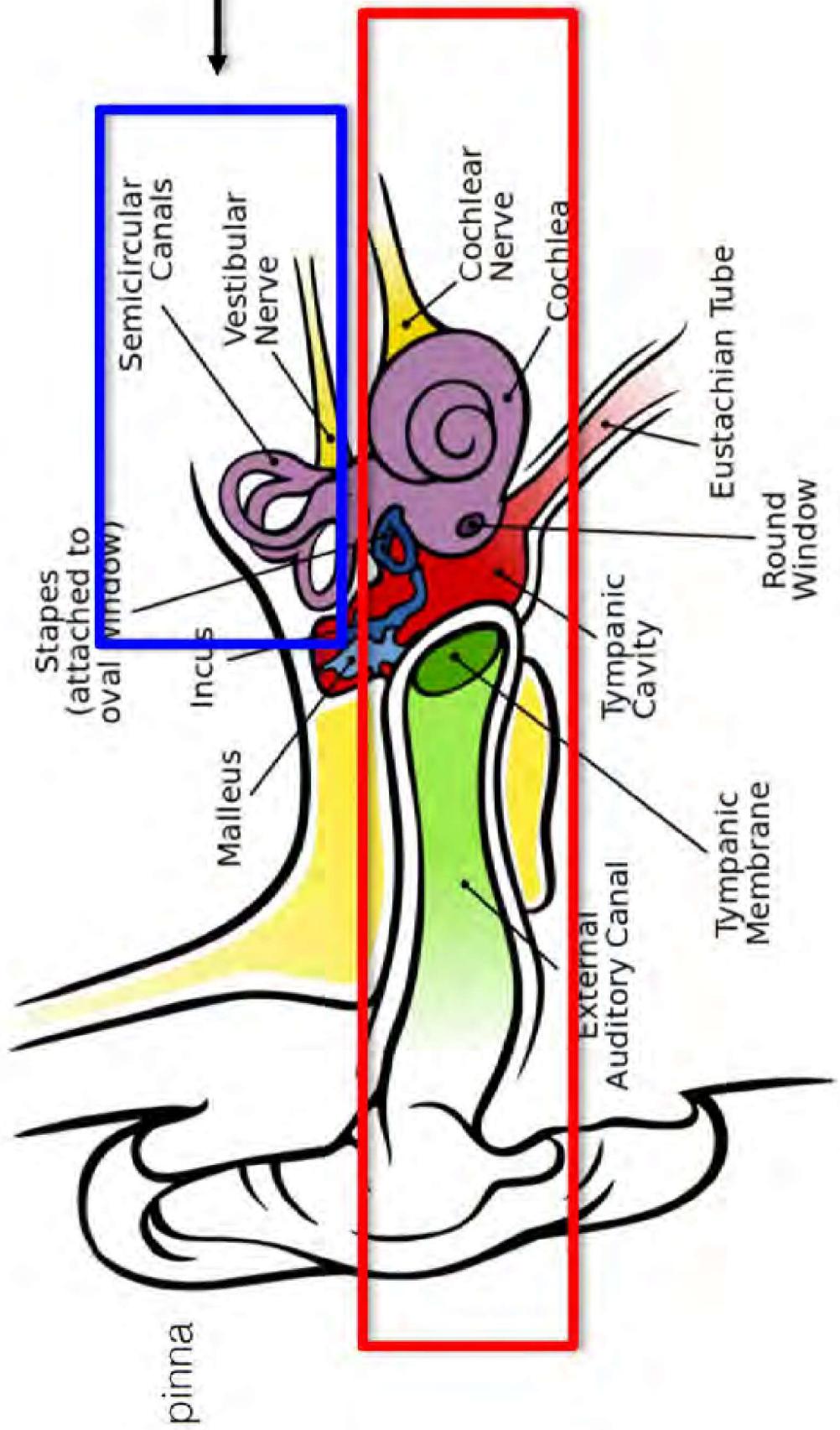
Our Solution – Distortion-Free Retargeting

Vection and Vestibular System

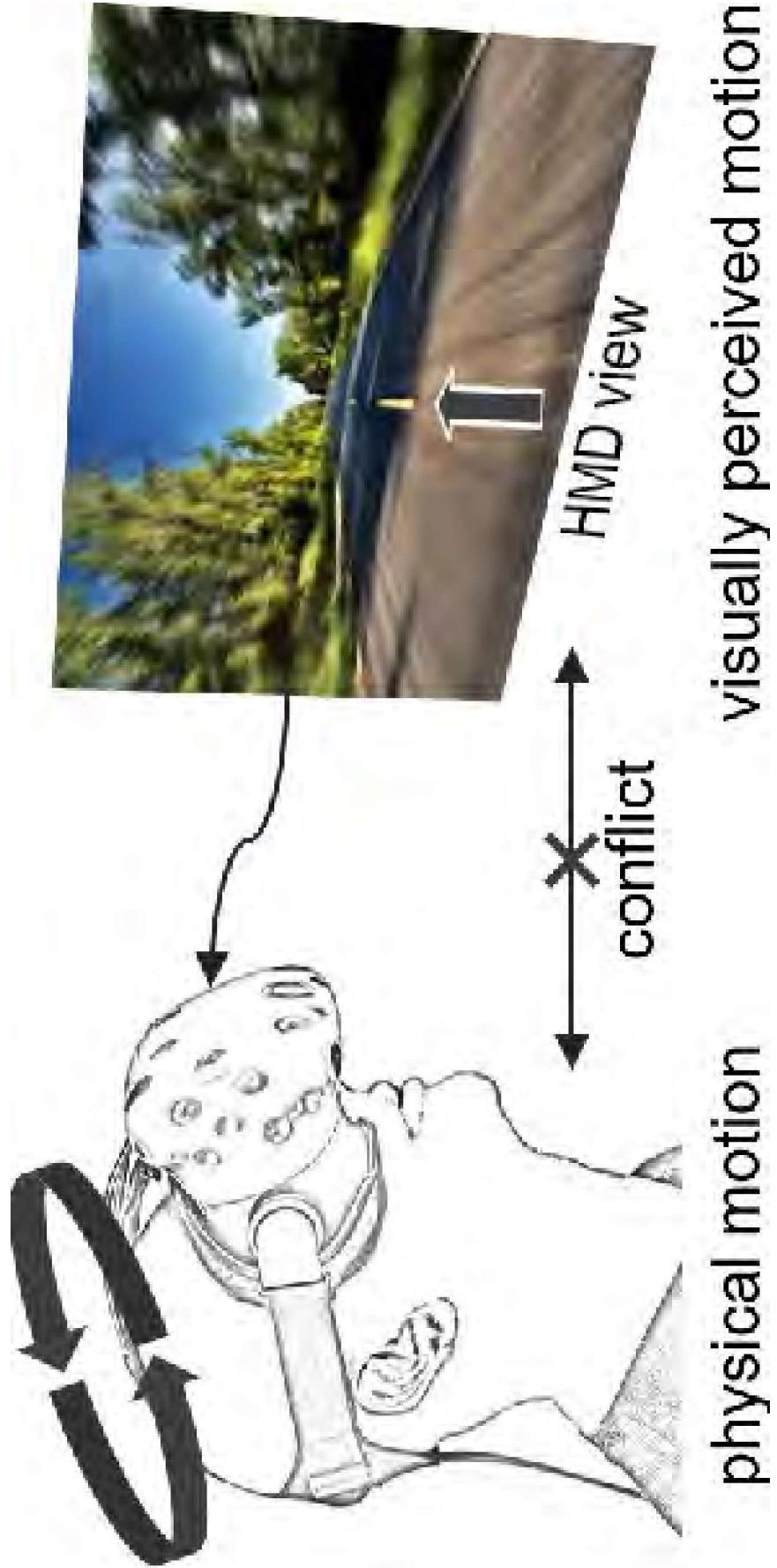


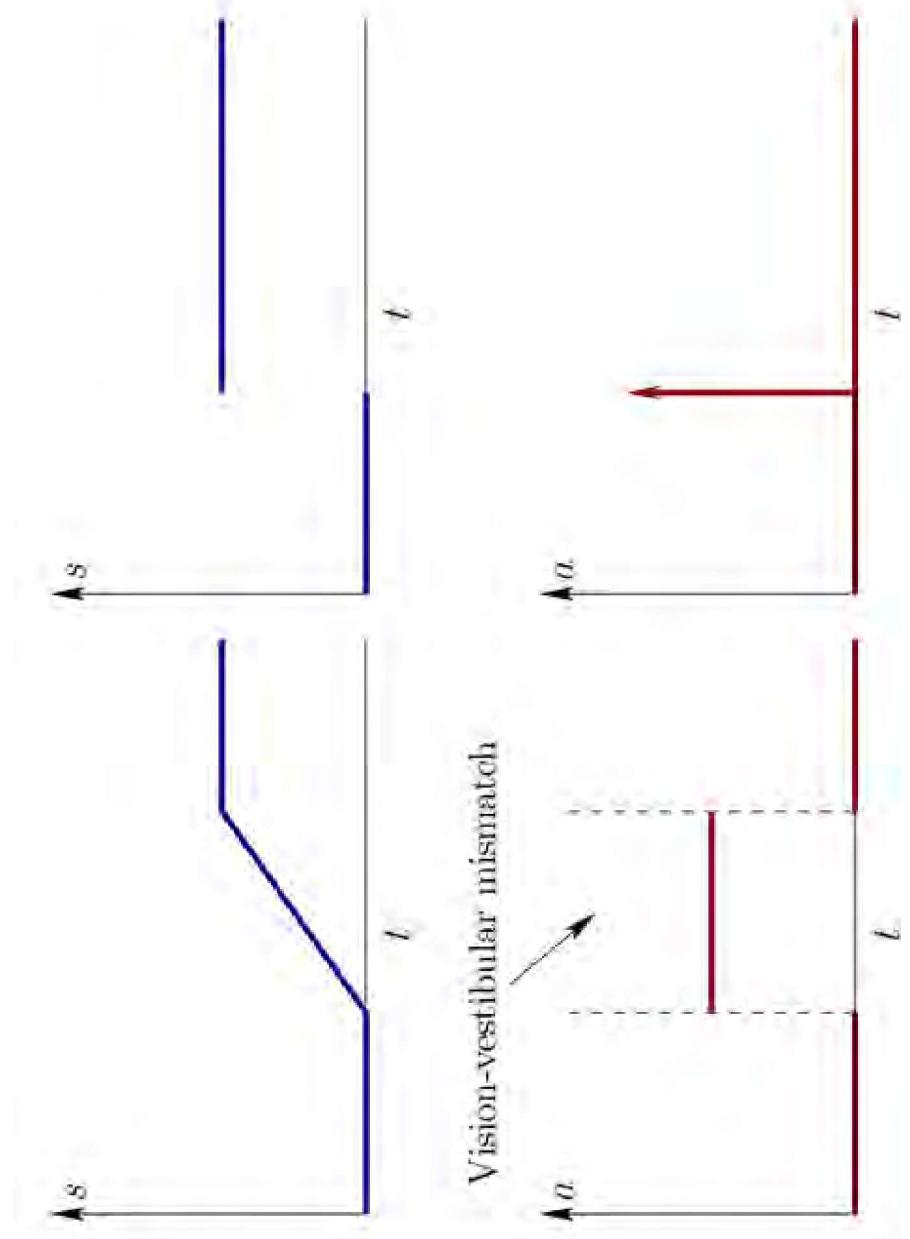
what's
this?

hearing



Vection and Vestibular System

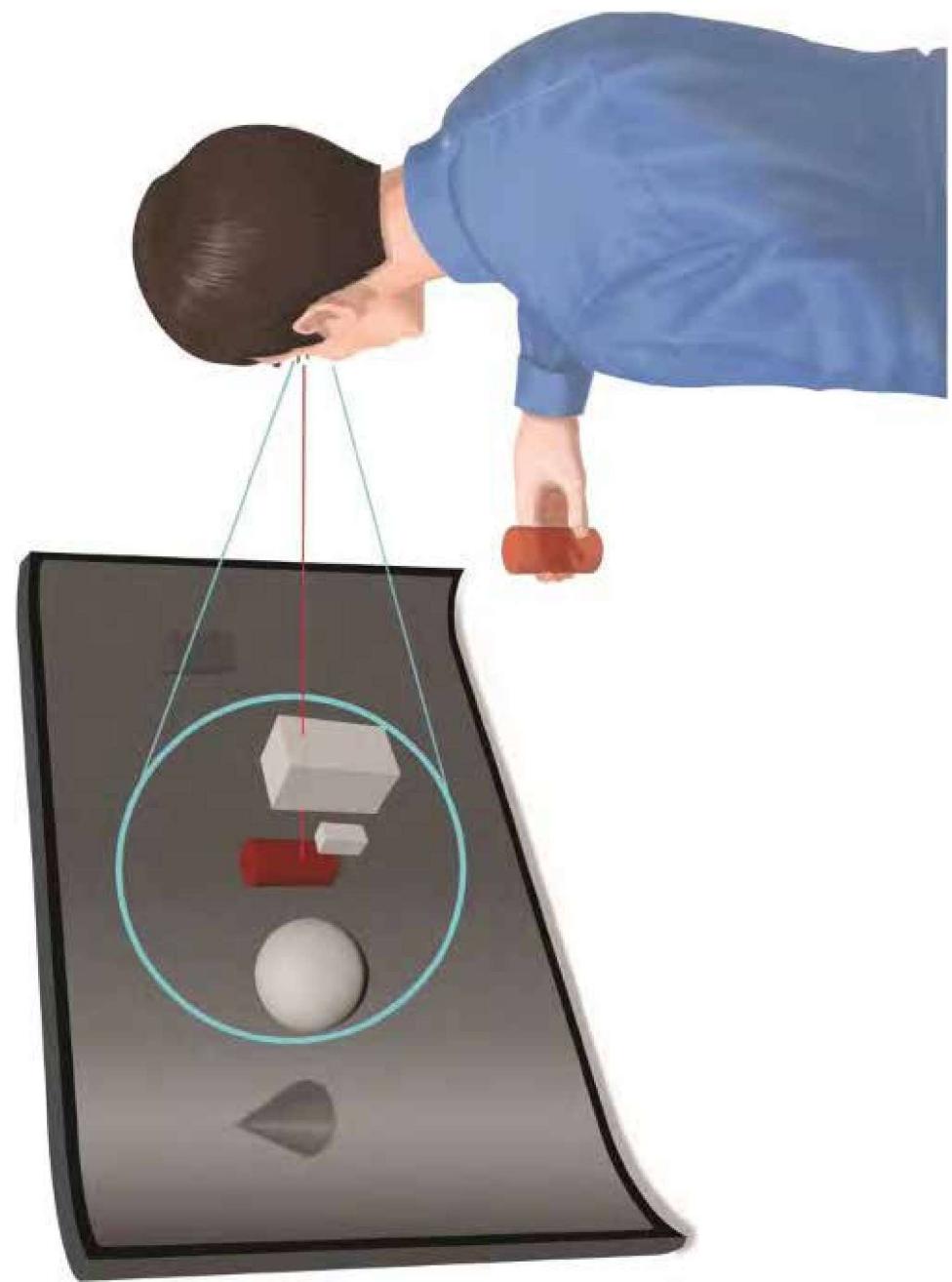




Manipulation

Avoid gorilla arms





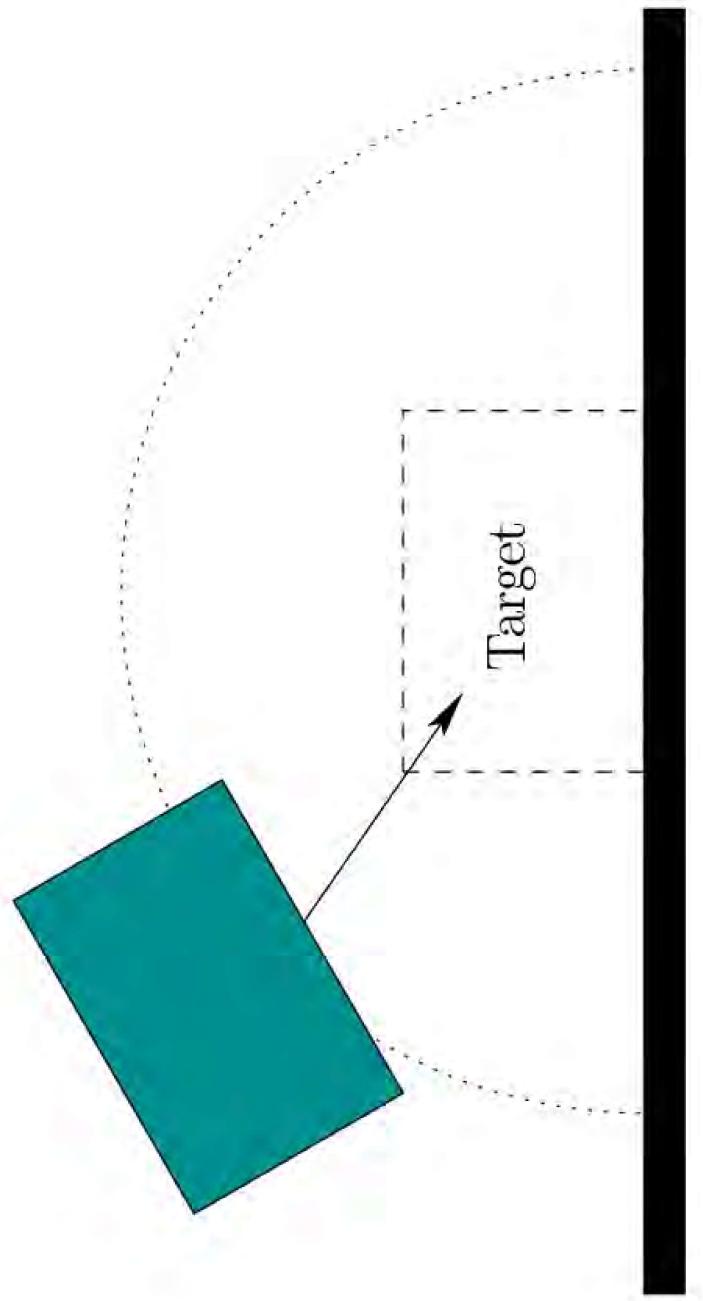
Selection



Manipulation



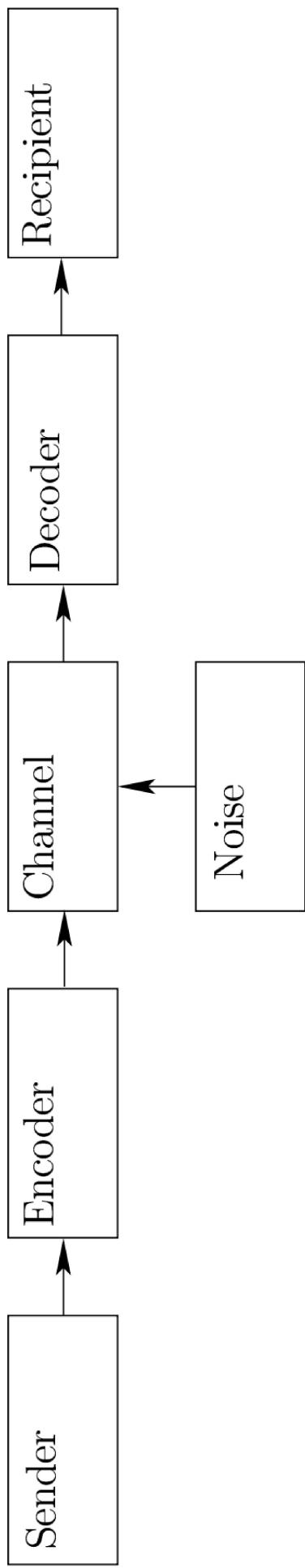
Placement



basin of attraction

Social Interaction

Shannon-Weaver communication theory



Social Interaction





Additional Interaction

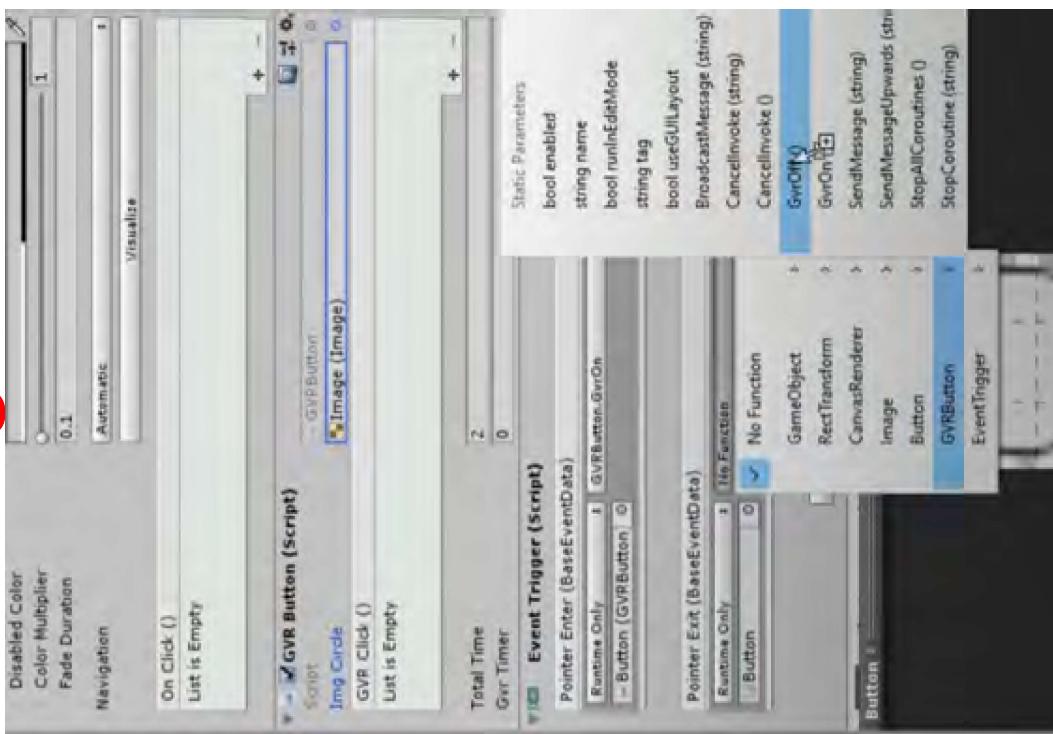


Google VR Walking by Gaze

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerWalk : MonoBehaviour {
6
7      public int playerSpeed;
8
9      // Use this for initialization
10     void Start () {
11
12     }
13
14     // Update is called once per frame
15     void Update () {
16         if (Input.GetButton("Fire1"))
17         {
18             transform.position = transform.position + Camera.main.transform.forward * playerSpeed * Time.deltaTime;
19         }
20     }
21
22 }
```

<https://www.youtube.com/watch?v=Qgpb5QzhuOg>

Google VR Interaction by Gaze



```
public float totalTime = 2;
bool gvrStatus;
public float gvrTimer;

On Click()

    // Update is called once per frame
    void Update()
    {
        if (gvrStatus)
        {
            gvrTimer += Time.deltaTime;
            imgCircle.fillAmount = gvrTimer / totalTime;
        }
    }

    if(gvrTimer > totalTime)
    {
        GVRClick.Invoke();
    }
}

public void GvrOn()
{
    gvrStatus = true;
}

public void GvrOff()
{
    gvrStatus = false;
    gvrTimer = 0;
    imgCircle.fillAmount = 0;
}
```

<https://www.youtube.com/watch?v=zdNBZsJdg9c>