Virtual Reality & Physically-Based Simulation
Interaction Metaphors

GOSH, WHAT DO YOU SUPPOSE THE FUTURE WILL BE LIKE?
WHO KNOWS? FLYING CARS AND CITIES BUILT ON CLOUDS, MAYBE!
© Universal Press Syndicate

G. Zachmann
University of Bremen, Germany
cgvr.cs.uni-bremen.de

The First(?) Interactive Computer Program

- First really interactive, real-time computer game (arguably):
  - Spacewar, 1961, MIT
  - Two players, two spaceships ("wedge" and "needle"), each can fire torpedos
  - With it came the first real interaction devices and metaphors

A Classification of Interaction Tasks

- *Basic interaction tasks (BITs) in 2D GUIs* [Foley / vanDam]:
  - Selection (objects, menus, ..)
  - Positioning (incl. orientation) or manipulation
  - Entering quantities (e.g., numbers)
  - Text input (via keyboard or speech input)
- *Universal Interaction Tasks (UITs) in VEs* [Bowman]:
  1. Navigation = change of viewpoint
  2. Selection = define object or place for next task
  3. Manipulation = grasp, move, manipulate object
  4. System control = menus, widgets sliders, number entry, etc.

Non-UIT Interactions

- Search
  - E.g., searching a scene for a specific object, excluding the navigation
- Ambient, implicit, playful, or non-purposeful interaction
  - E.g., playing around with a virtual spraying can
- Sculpting / modeling surfaces / painting (Tiltbrush, Quilt)
- Making an avatar dance by whole body interaction

Examples of Non-UIT Interaction

Sandscape (sand as terrain)

http://tangible.media.mit.edu

GranulatSynthese (interactive art installation)

http://imve.informatik.uni-hamburg.de/projects/GranulatSynthese

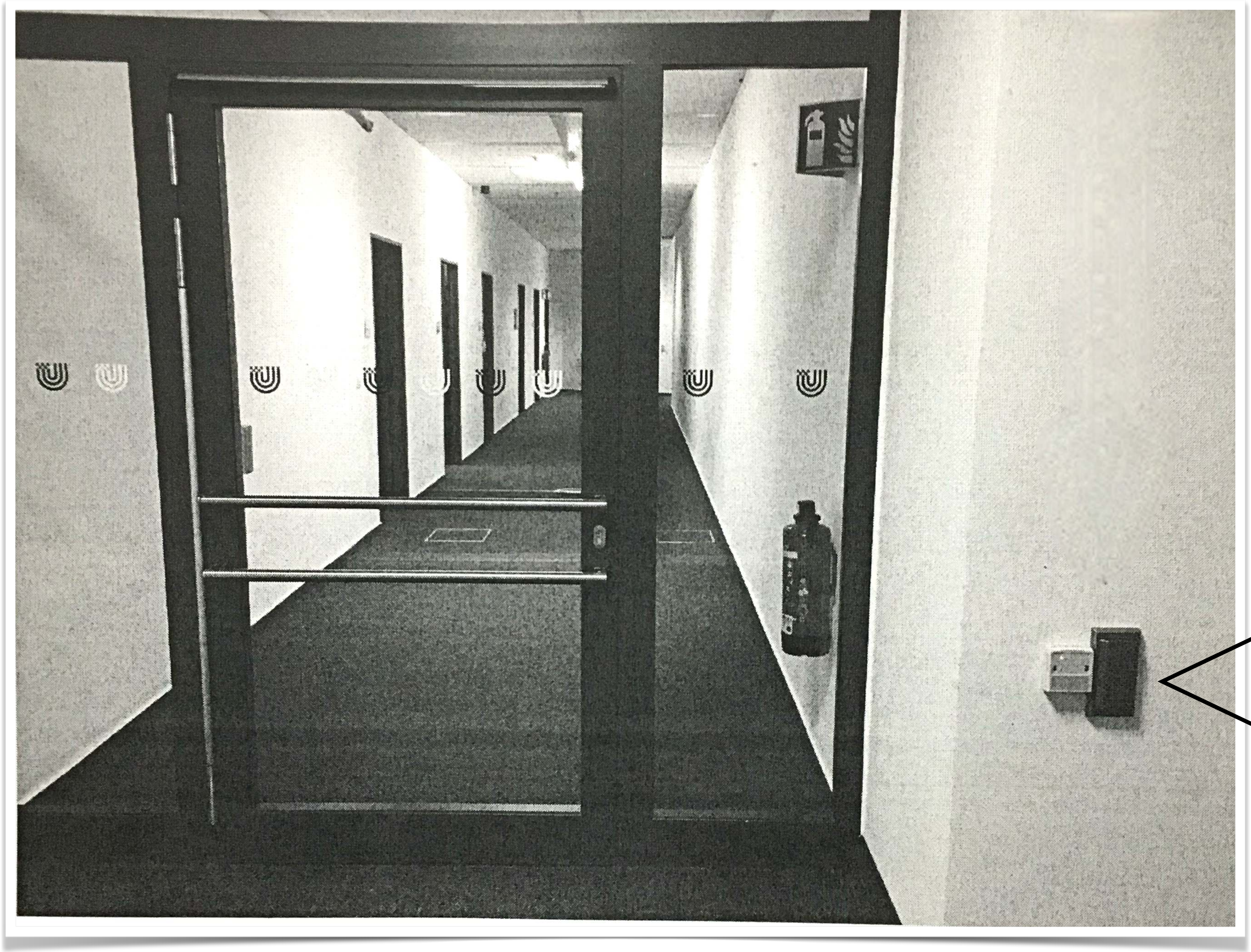
Approaches to the Design of User Interfaces

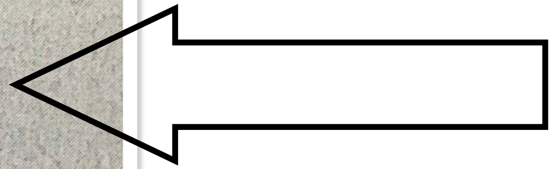
- There are two main approaches: natural and magic interaction
  - Natural interaction:
    - Try to resemble reality and the interaction with it as closely as possible
  - "Magic" interaction
    - Give the user new possibilities beyond reality
    - Challenge: keep the cognitive overhead as low as possible, so that users don't get distracted from their task!
- Technologies:
  - Direct *user action* (e.g., tracking of the body, gesture, head turning, ...)
    - Pro: well suited if intuitive; con: possibilities are somewhat limited
  - Physical devices (e.g., steering wheel, button, ...)
    - Pro: haptic feedback affords precise control
    - Con: not easy to find/devise novel & useful devices
  - Virtual devices (e.g., menus, virtual sliders, etc., embedded in the VE)
    - Pro: very flexible, reconfigurable, "anything goes"
    - Con: can be difficult to use because of lack of force feedback

- Goals (in particular in VR):
  1. Intuitive / natural interaction (usability)
     - By definition: easy to learn
     - Adjust to the user's expertise (expert vs. novice)
  2. Efficient interaction (user performance)
     - Precision, speed, productivity of the users
- Problems (especially in VR):
  - No physical constraints (interaction in mid-air) ⟶ fatigue
  - In particular: no haptic feedback
  - Efficient interaction with objects outside of the user's reach
  - Noise / jitter / imprecision in tracking data
  - No standards (yet)

> *There has never been a high performance task done in the history of this planet, to the best of my knowledge, that has ever been done well with an intuitive interface.*
> [Brian Ferran]

Example of Unintuitive User Interaction





Button for opening the door!

When a device as simple as a door has to come with an instruction manual — even a one-word manual — then it is a failure, poorly designed.

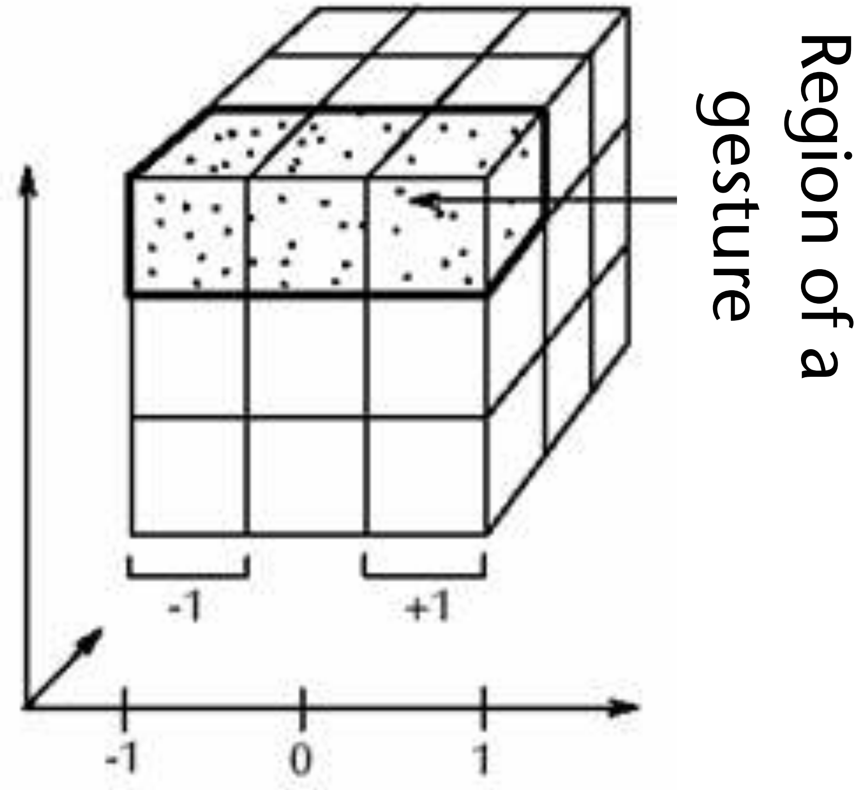
(Donald Norman, The Design of Everyday Things, 2002, p. 87)

Hand Pose Recognition

- Is basically a simple classification problem:
  - Given: a flex vector $x \in \mathbb{R}^d,\ d \approx 20$, i.e. the joint angles
  - Desired classification: pose $G(x) \in \{$ “Fist“, “Hitch-hike“ , $\ldots \}$
- Wanted: an algorithm that is ...
  - .. user independent
  - .. robust (> 99%)
  - .. Fast
- General solution: machine learning algorithms (e.g, neural network)

An Extremely Simple (yet Sufficient) Pose Recognition Algorithm

- Neural network is fine, in case of lots of gestures, or some of them are inside the parameter space
    - However, experience shows: users can remember only a small set (e.g. 5)
- Consider only a few poses near the border of the parameter space
    - Discretize the flex vector $f \in [0,1]^d \rightarrow f' \in \{-1, 0, +1\}^d$
      0 = the corresp. flex value is "somewhere in the middle"
    - Pose = a region of the *d*-dimensional parameter cube
    - Represent each region/pose by a discrete vector:
      $g \in \{-1, 0, +1\}^d$, 0 = *don't care*
    - Classify *f* as pose *i* $\Longleftrightarrow$ *f*' "matches" g^i
      $\Longleftrightarrow \forall j \,:\, f'[j] = g[j]$
      and ignore those *f*'[*j*] where *g*[*j*] = 0



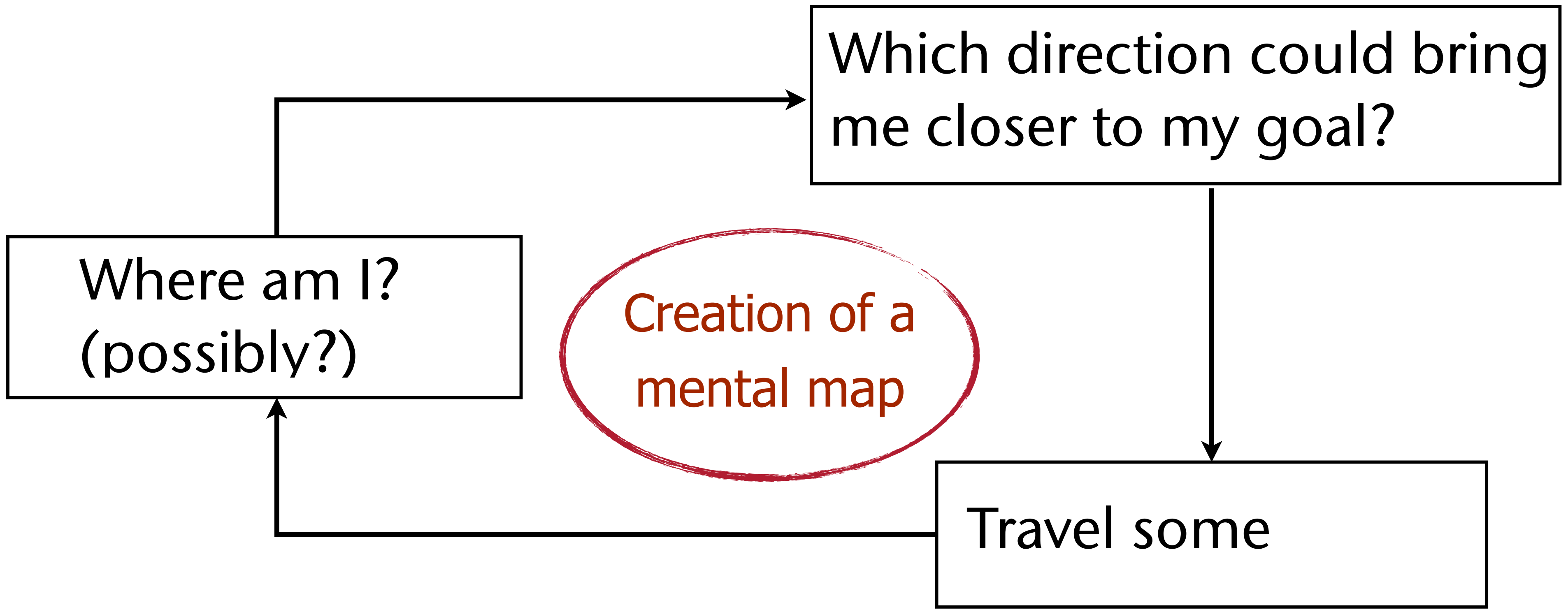
- Implementation details:
  - Do automatic calibration at runtime to fill the range [0,1]:
    - Maintain a running min/max and map it to [0,1]
    - Over time, shrink min/max gradually (for robustness against outliers)
  - Ignore transitory gestures
- Dynamic gestures =
  1. Sequence of static poses/postures (e.g., sign language)
  2. Path of a finger / hand
  - Utility for VR?

Navigation

- Comprises: *Wayfinding* & *Locomotion*
- Locomotion / Travel =
  - Cover a distance (in RL or in VR)
  - Maneuvering (= place viewpoint and/or viewing direction exactly)
- Wayfinding =
  - Strategy to find a specific place (in an unknown building / terrain)
  - Comprises: experience, cognitive skills, …

How do People Solve a Wayfinding Task

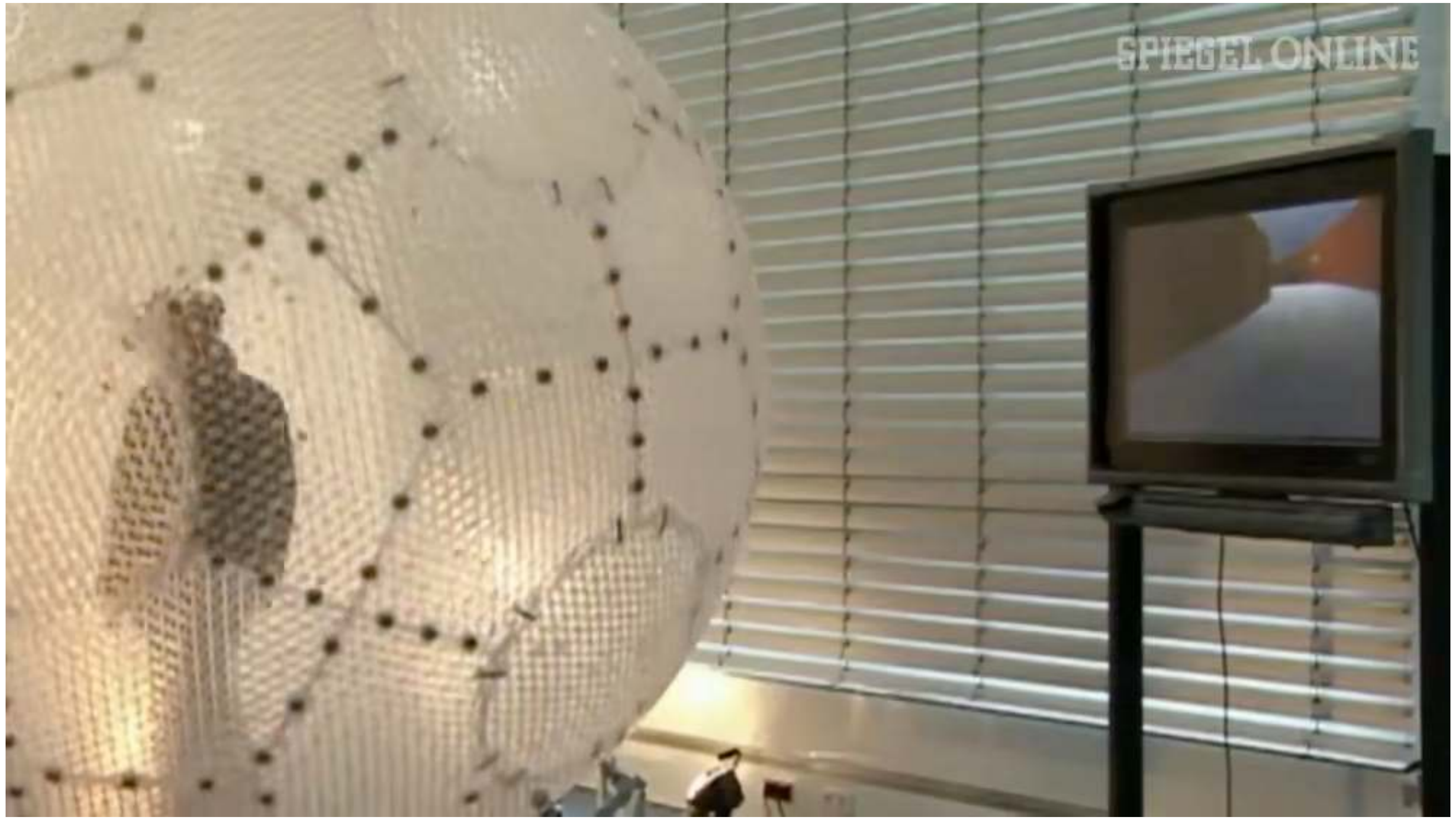
- How do people find their way:
  - Natural hints/clues
  - Signs (man-made)
- A simple user model for way finding:



- In VEs, there can be two kinds of wayfinding aids:
  - Aids for improving the user's performance in the virtual environment
  - Aids that help increase the user's performance later in the real world (i.e., that increase the training effect)

- Question: do humans create a mental map of their environment in order to solve wayfinding tasks?
- Answer: probably yes, but not like a printed street map; rather like a non-planar graph that stores edge lengths

Kerstin Schill, Neuro-Informatics, Uni Bremen



http://www.spiegel.de/wissenschaft/technik/0,1518,739416,00.html

Definition Interaction *Task* vs. Interaction *Metaphor*

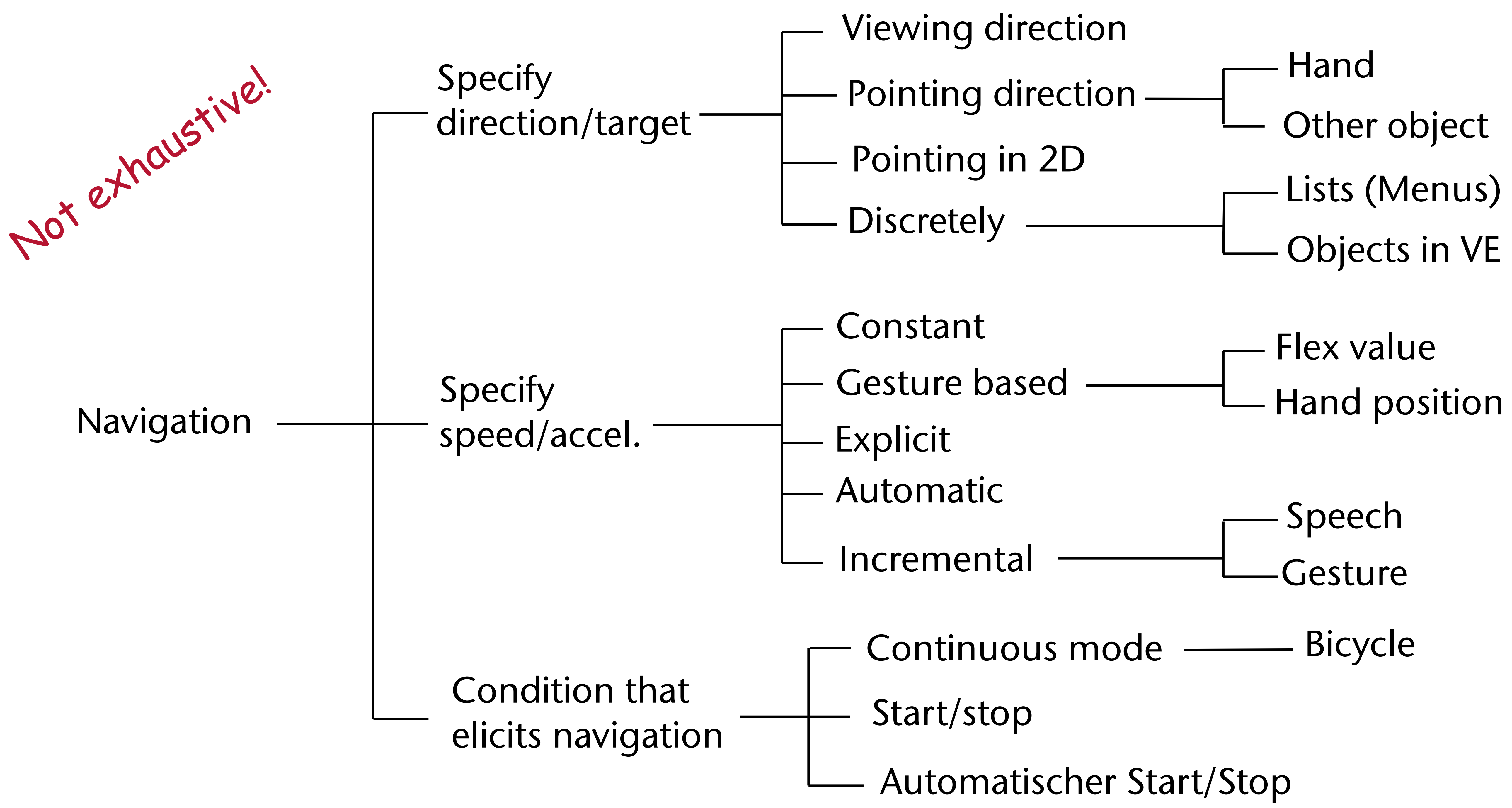
- Interaction task := what to achieve
- Interaction metaphor / technique := how to achieve it
- Examples:

Interaction Task	Interaction Devices	Interaction Metaphor/Technique
Define scaling on tablet (e.g., for photo)	Touch screen	Pinch gesture
Define position on screen of desktop PC	Mouse	Move mouse on table, watch mouse pointer on screen, click
Start a timer	Hour glass	Turn hour glass
Change gears in car	Pedals and gear stick	First, push left pedal, move gear stick in desired position, release left pedal slowly
Switch room lights	Microphone	Clap hands three times

Techniques for Navigation in VEs

- A.k.a. "viewpoint control"
- Real user navigation, e.g., walking, turning head, ...
- *Point-and-fly* (especially in Caves and HMDs)
- *Walking in place*
- *Scene-in-hand*
- *World-in-Miniature*
- Orbital mode
- And some more ...

A Taxonomy for this Interaction Task



Not exhaustive!

- Navigation
  - Specify direction/target
    - Viewing direction
    - Pointing direction
      - Hand
      - Other object
    - Pointing in 2D
    - Discretely
      - Lists (Menus)
      - Objects in VE
  - Specify speed/accel.
    - Constant
    - Gesture based
      - Flex value
      - Hand position
    - Explicit
    - Automatic
    - Incremental
      - Speech
      - Gesture
  - Condition that elicits navigation
    - Continuous mode
      - Bicycle
    - Start/stop
    - Automatischer Start/Stop

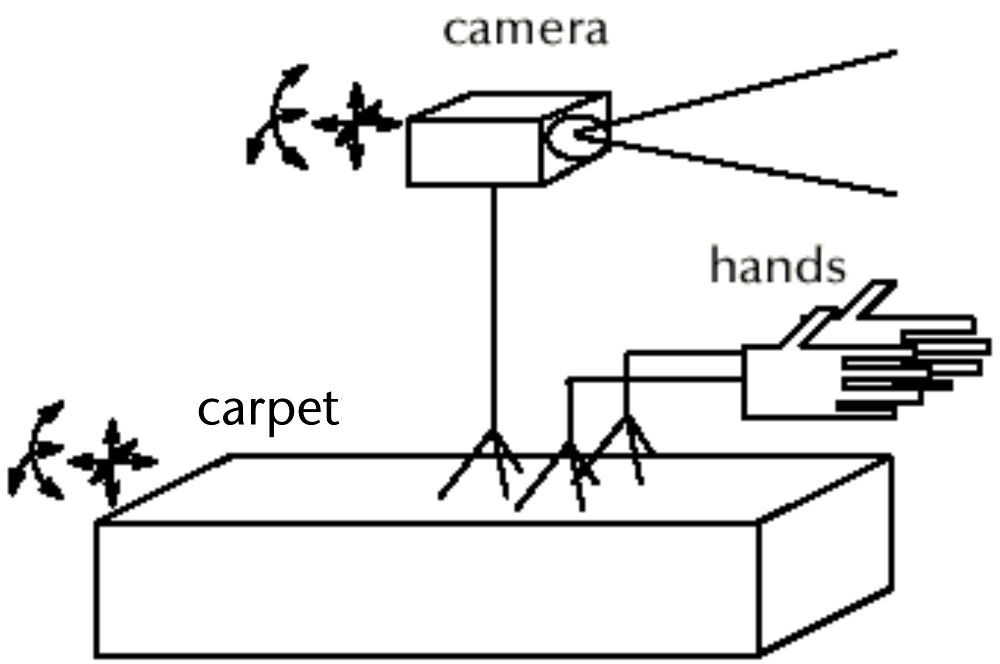
Taxonomies are a technique to explore the design space of an interaction task!

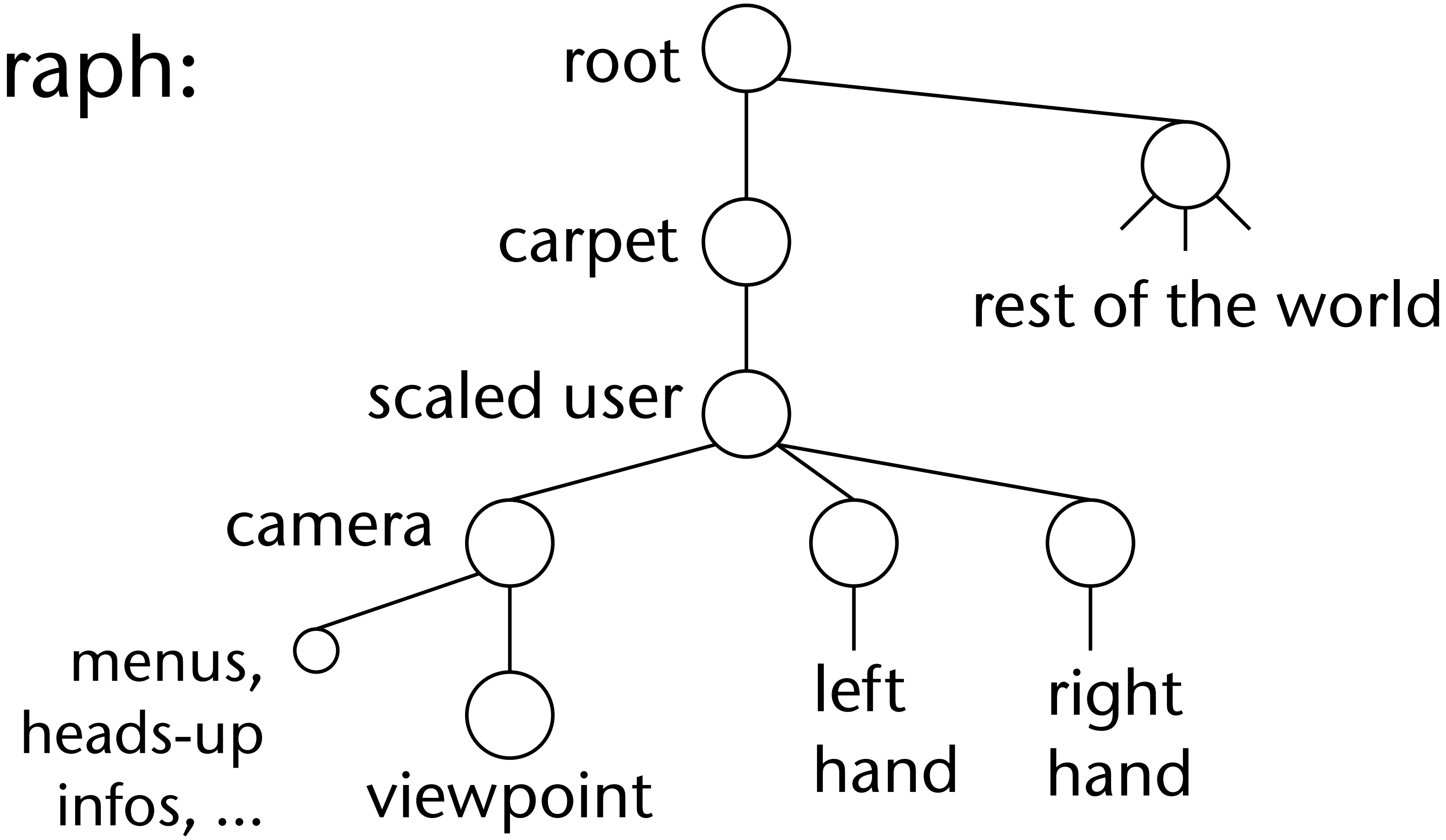
Task: Drive a Car, Develop a Taxonomy

https://www.menti.com/p9rdrjti47

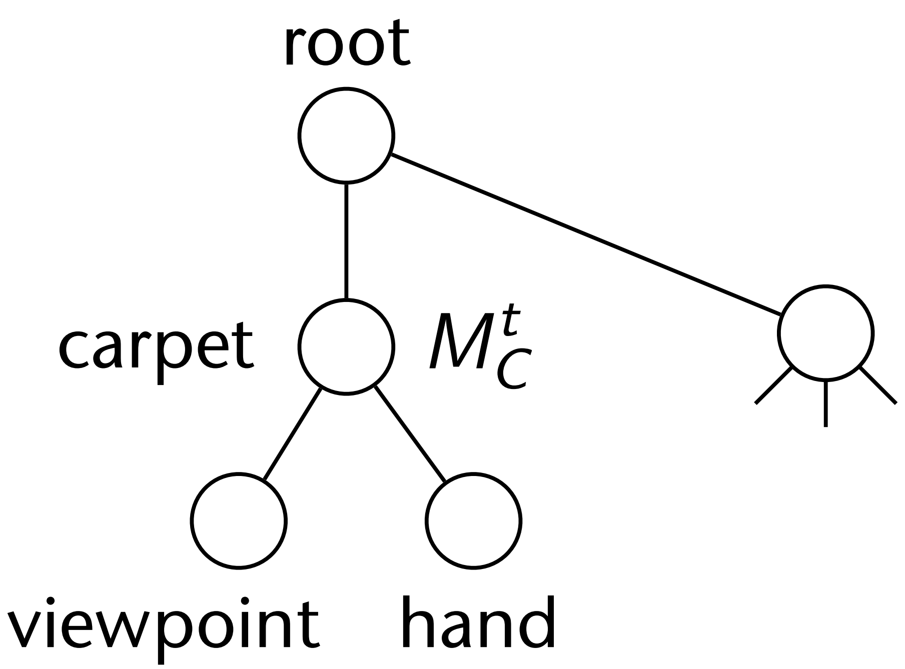
An Abstract Representation of the User

- User = head, hand, perhaps whole body (avatar)
- The "flying carpet" metaphor :
  - User = camera
  - Camera is placed on a carpet
  - User controls camera and carpet
- Representation as (part of) a scenengraph:

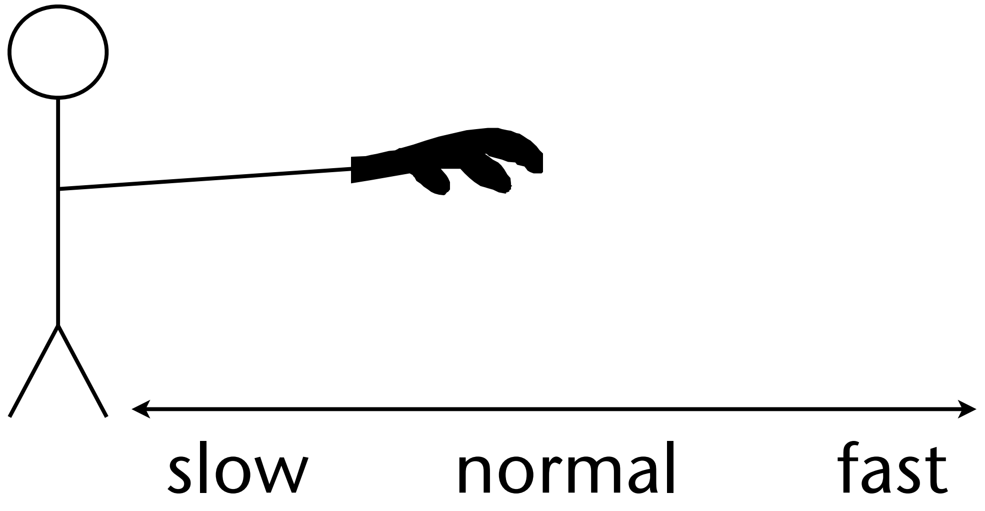




The Point-and-Fly Metaphor



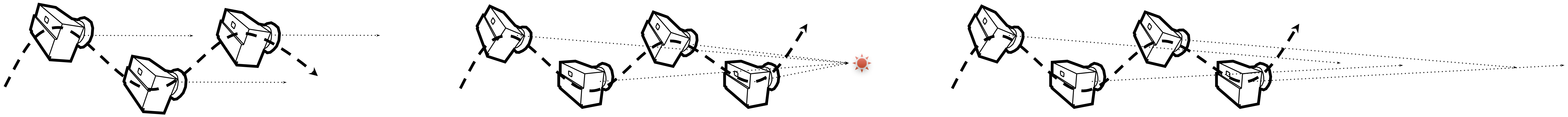
- Controlling sensors:
  - Head sensor controls viewpoint
  - Hand sensor controls carpet: $M_C^t = M_C^{t-1} \cdot \text{Transl}(s \cdot \mathbf{t})$
    s = speed,
    $\mathbf{t}$ = direction, e.g., pointing direction of hand tracking sensor in camera frame(!)
- Generalization: use graphical objects instead of sensor to derive translation direction (e.g., controller)
- Some options for the specification of the speed:
  - Fixed
  - Flexion of the thumb
  - Distance between hand and body
  - Make it independent of framerate



- This navigation technique implements the point-and-fly metaphor, too, even though there is no pointing with a hand:

Perception of the Distance Travelled in VR

- Question: when using point-and-fly, how can the sense of presence be increased while navigating in a VE?
- Idea:
  - Make the viewpoint oscillate, like in reality (First-person-shooter games invented this earlier ;-))
  - Variants:



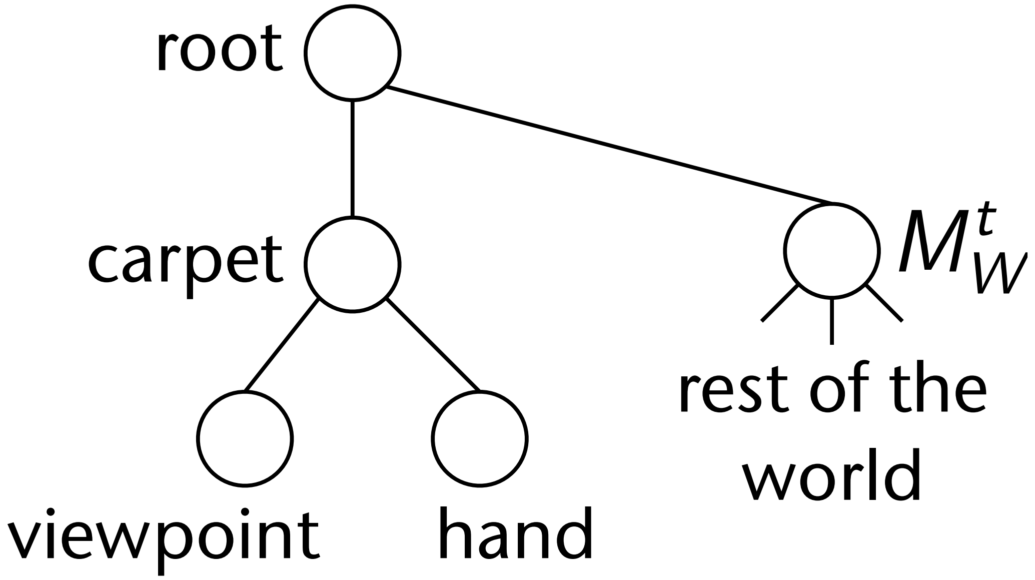
- Results:
  - Only vertical oscillation helps increase presence
  - Users prefer slight oscillation over no oscillation
  - Short "travel distances" can be estimated more precisely (~ factor 2)

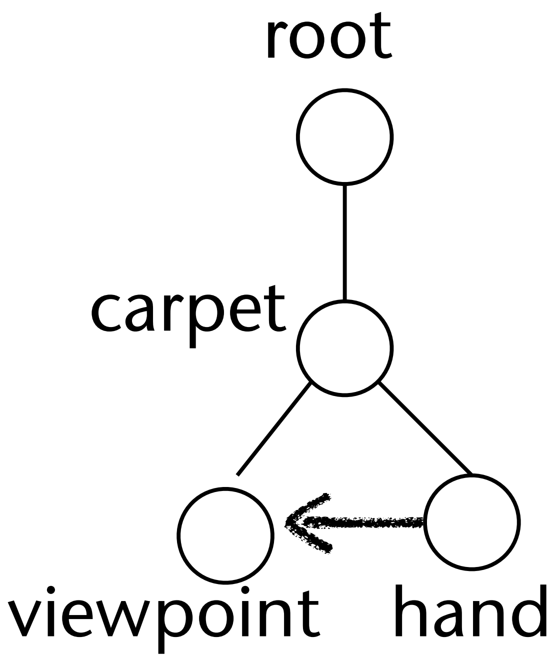
The Scene-in-Hand / Eyeball-in-Hand Metaphor

- *Scene-in-hand*:
  - "*Grabbing the air*" technique
  - *Carpet* remains stationary, scene gets rotated about a specific point in space
  - The transformation:

$$M_W^t = M_H^t \cdot (M_H^{t_0})^{-1} \cdot M_W^{t_0}$$

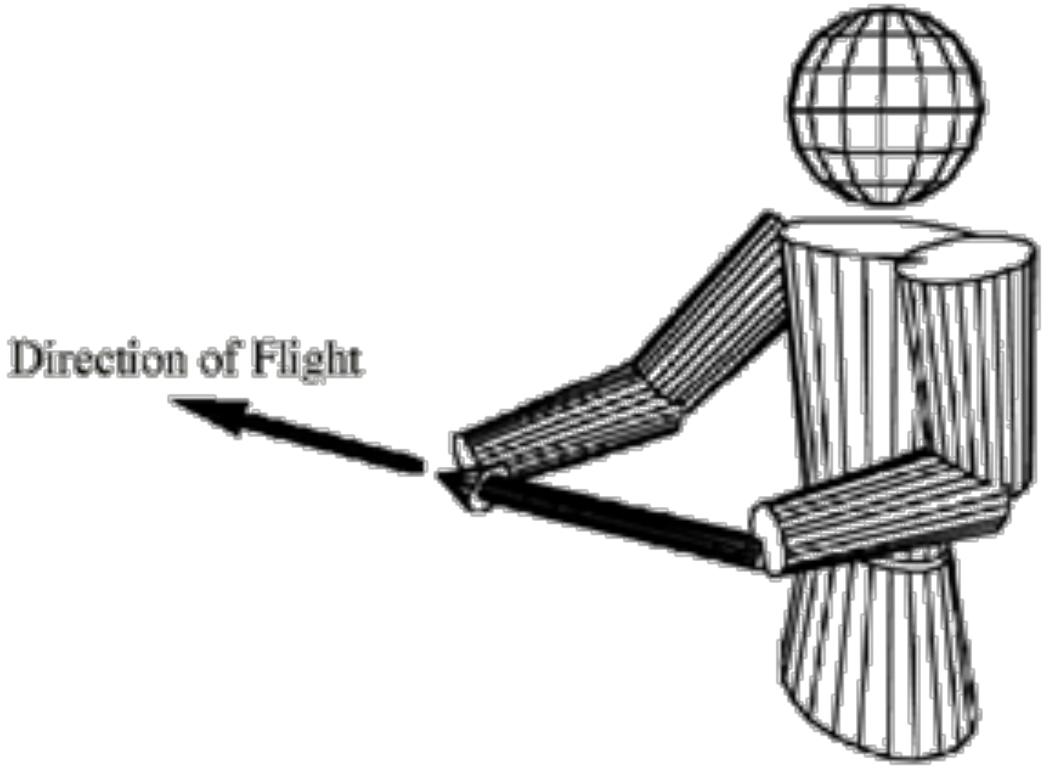
  - Frequent way of implementation: instead of user's hand, use specific device, e.g. the CAT
- *Eyeball-in-hand*:
  - Viewpoint is controlled directly by user's hand
  - Can be absolute or relative (accumulating) mode



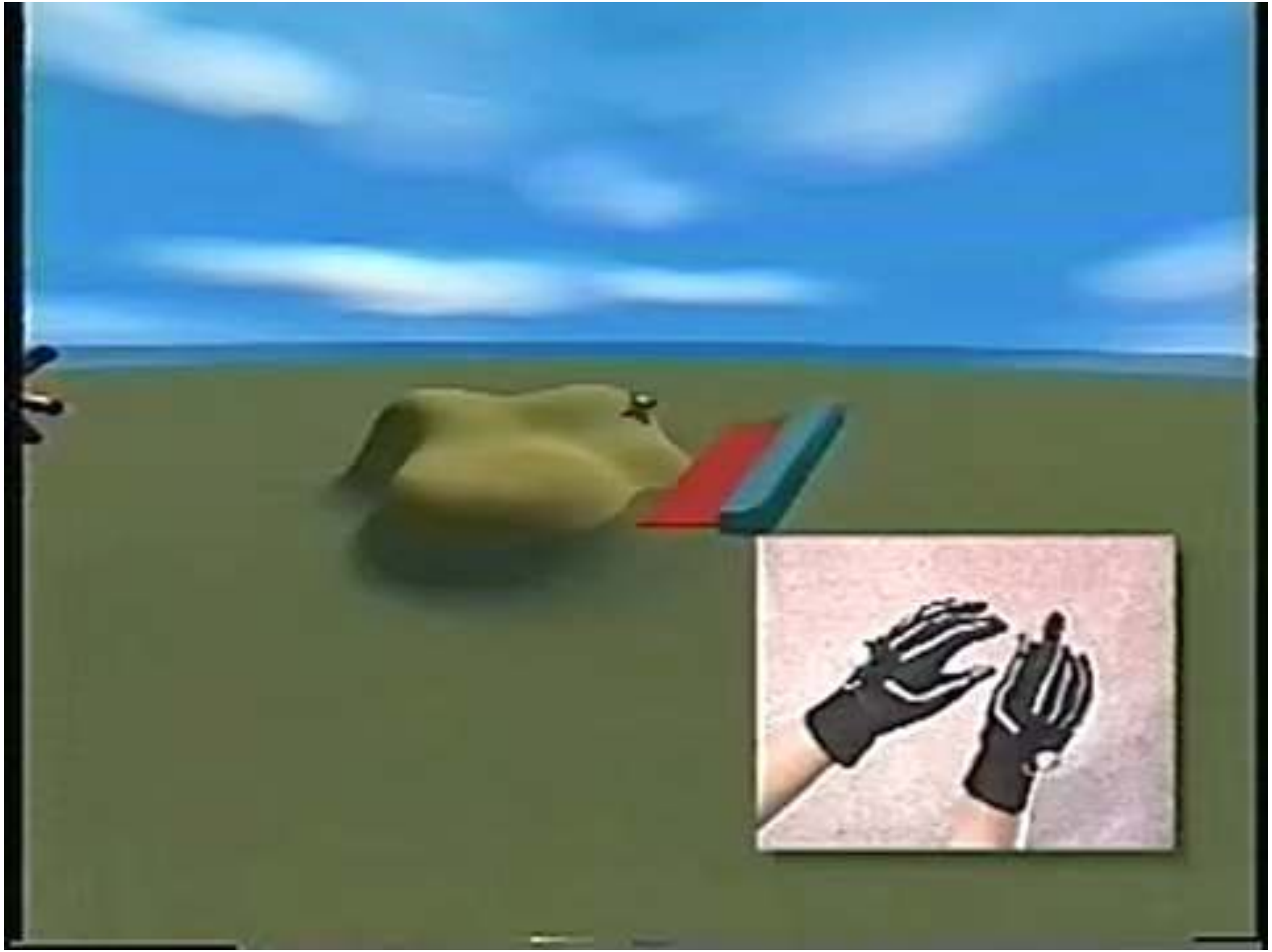


Two-Handed Navigation (a.k.a. 3D Multitouch)

- Approach: we only need to fix/grasp 2 points in space and 1-2 triggers (→ pinch gloves) to implement a "*scene-in-hand*" technique
- The metaphor:
  - 1 trigger, 1 moving point → translate the scene
  - 2 trigger, 1 fixed point , 1 moving point → rotate the scene
  - 2 trigger, 2 moving points → scale the scene
- Variant:
  - Direction = vector between both hands
  - Speed = length of vector
- Multi-touch was invented in 3D,
  only later used in 2D (smartphones, workbench)



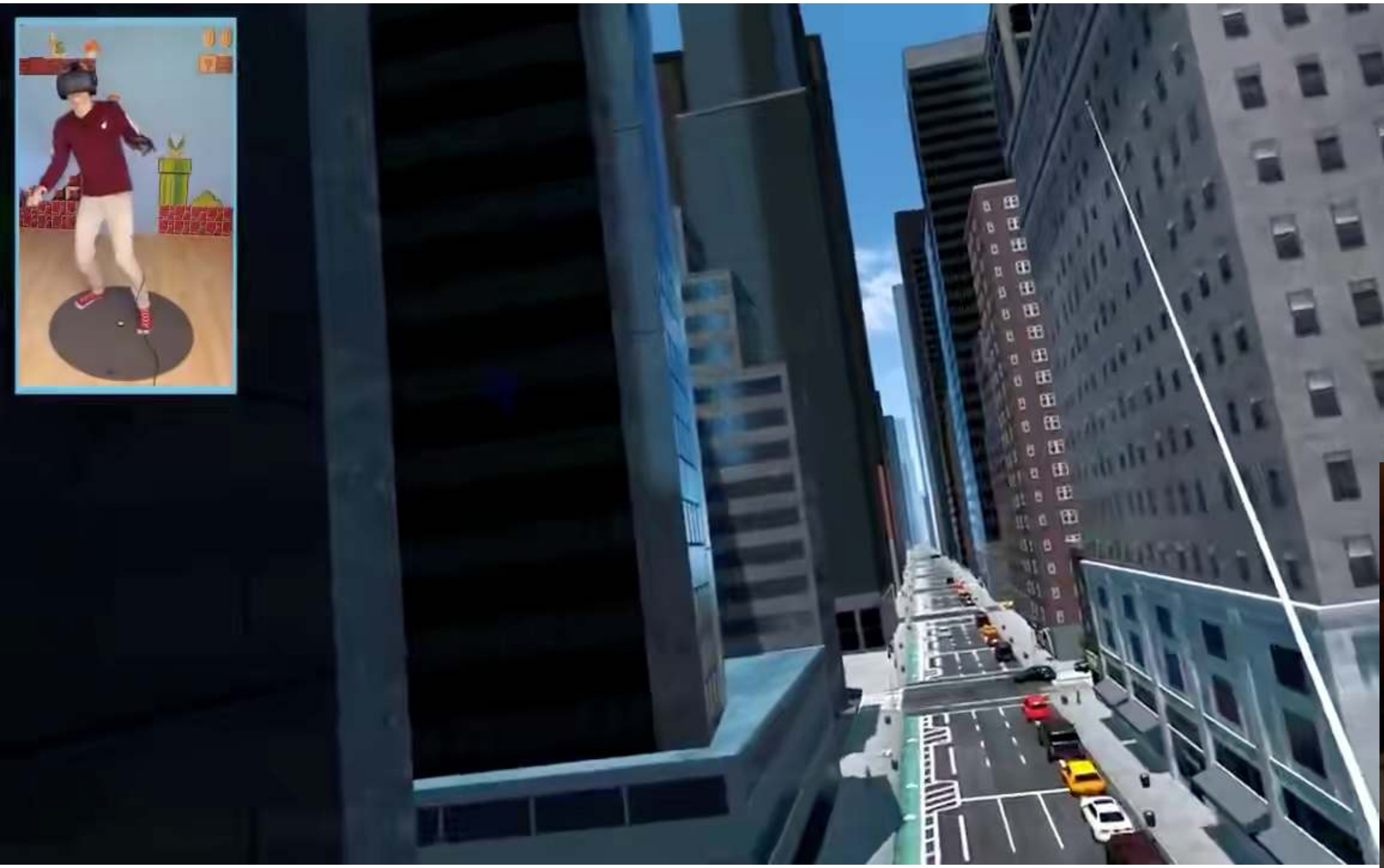
Examples of Scene-in-Hand Navigation



SmartScene, MultiGen, Inc.

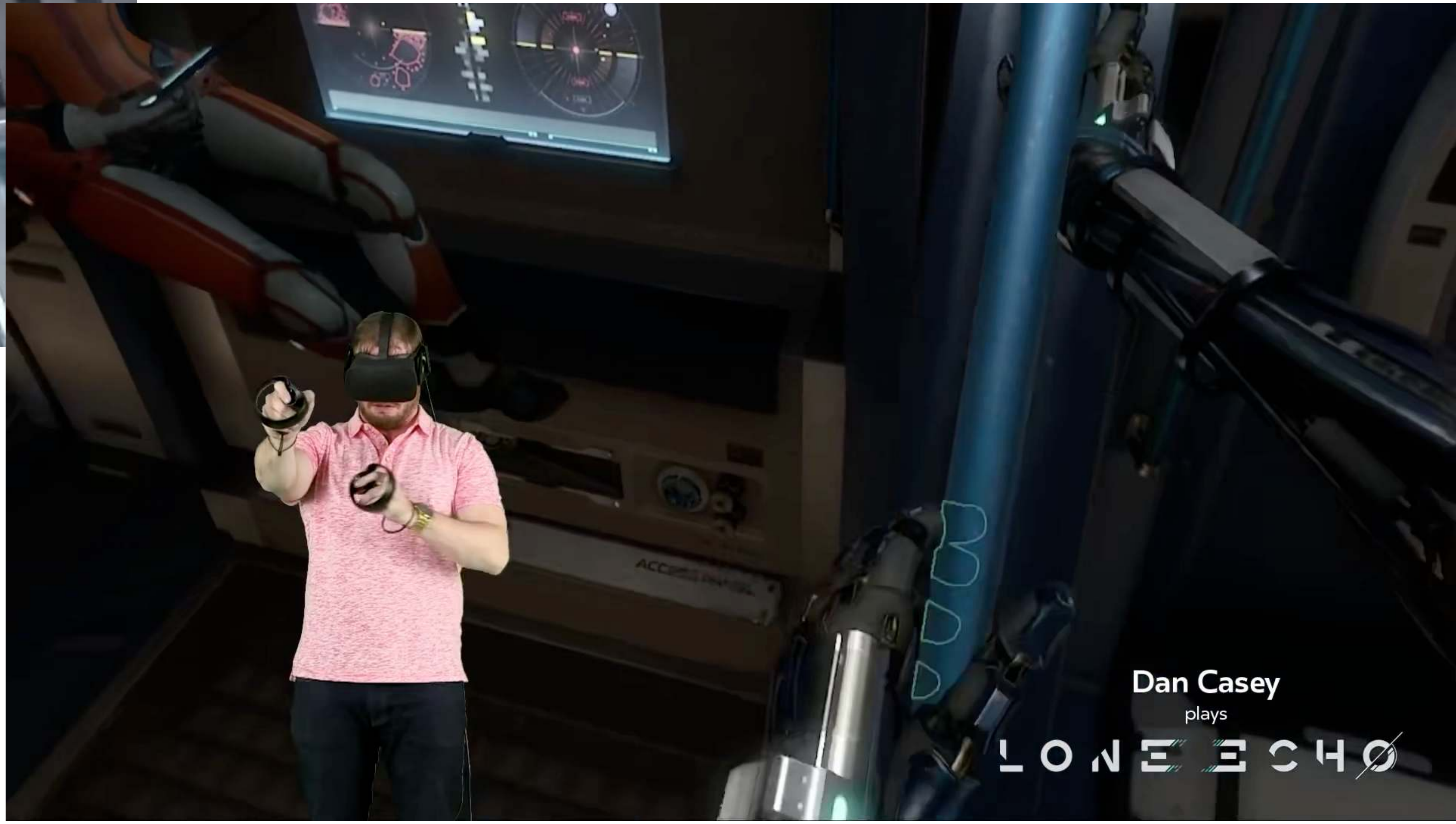
Example of Scene-in-Hand: Exploration of a Medical Volume Visualization

Scene-in-Hand Navigation in Games



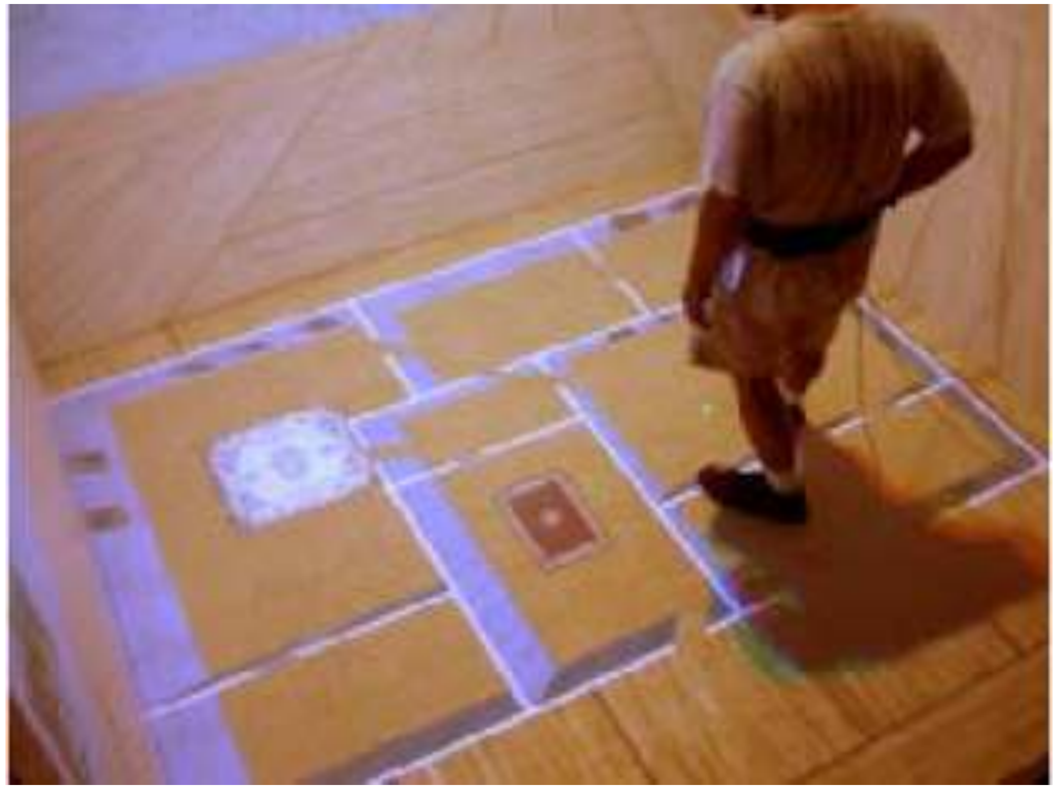
Spider-Man: Scene-in-Hand in disguise

Lone Echo: Scene-in-Hand with inertia



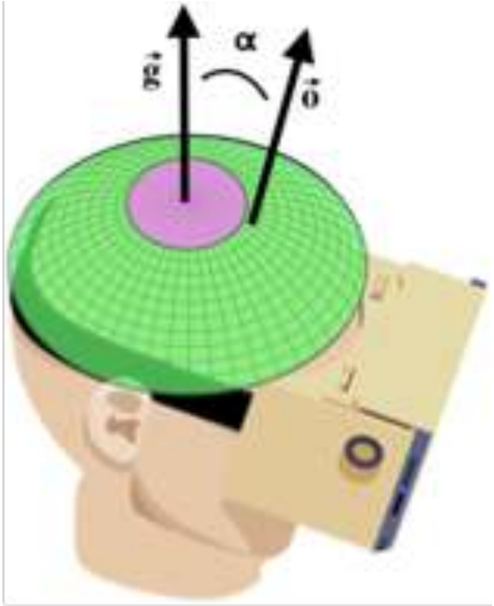
Real Walking on a (Mini-)Map

- Idea: project a scaled down version of the VE on the floor (map) and use feet
- Coarse navigation by teleportation: user walks to the new place/viewpoint on the map and triggers teleportation
- System commands involved:
  1. Bring up map = look at floor + trigger
  2. Teleportation = look at floor + trigger
  3. Dismiss map = look up + trigger
  - Trigger = speech command or "foot gesture"
- Accurate navigation: "lean" towards desired direction; speed = leaning angle

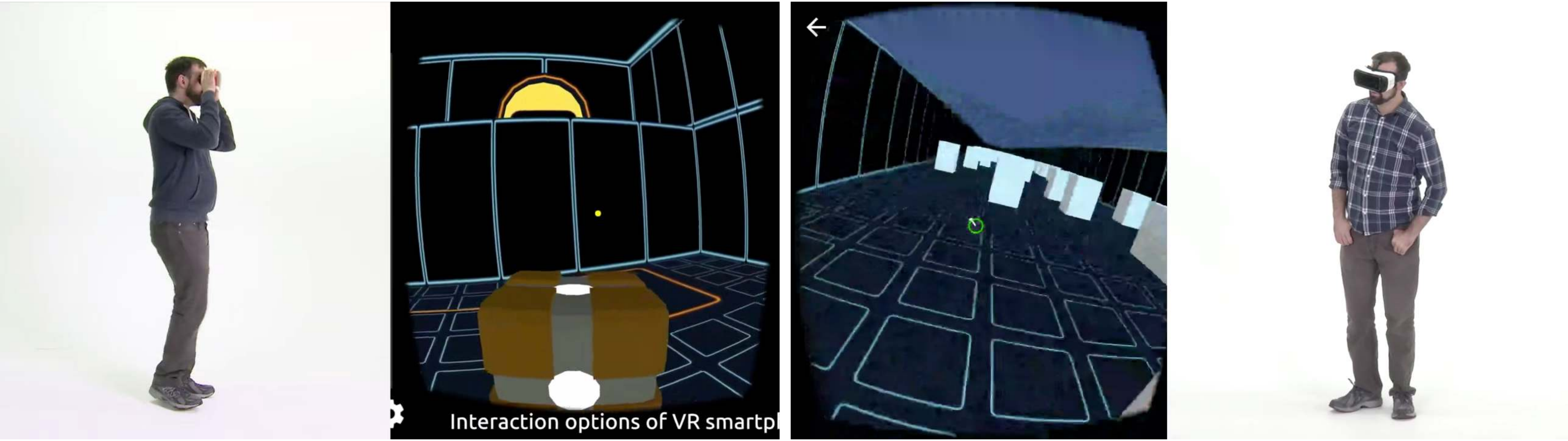


Walking in Place (WIP)

- The technique:
  - Head of user is tracked (optical tracker, or accelerometer) ⟶ time series
  - Classifier recognizes walking pattern in time series (and speed)
  - Cart is moved in gaze direction
  - Only forward direction is possible
- Extension: TILT-WIP
  - WIP pattern just provides triggers
  - Tilting angle of head provides navigation direction (omnidirectional)
- Can increase the level of subjective presence in the VE
  - Reason is probably that proprioceptive information from human body movements better matches sensory feedback from the computer-generated displays
- Probably makes sense only with a virtual ground (terrain, etc.)



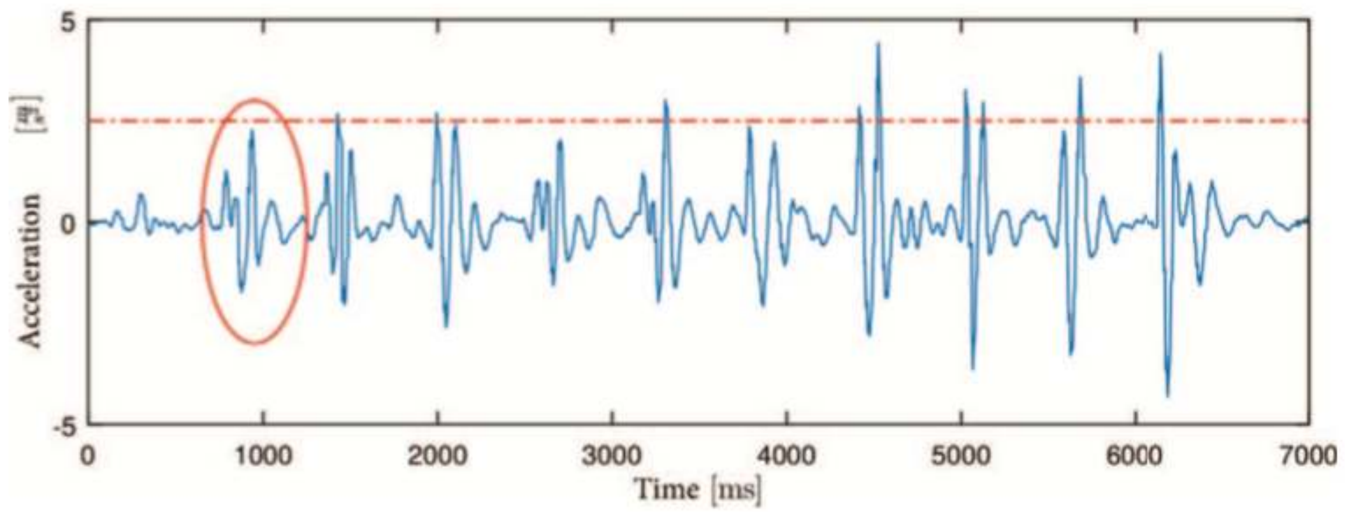
Example in Mobile VR

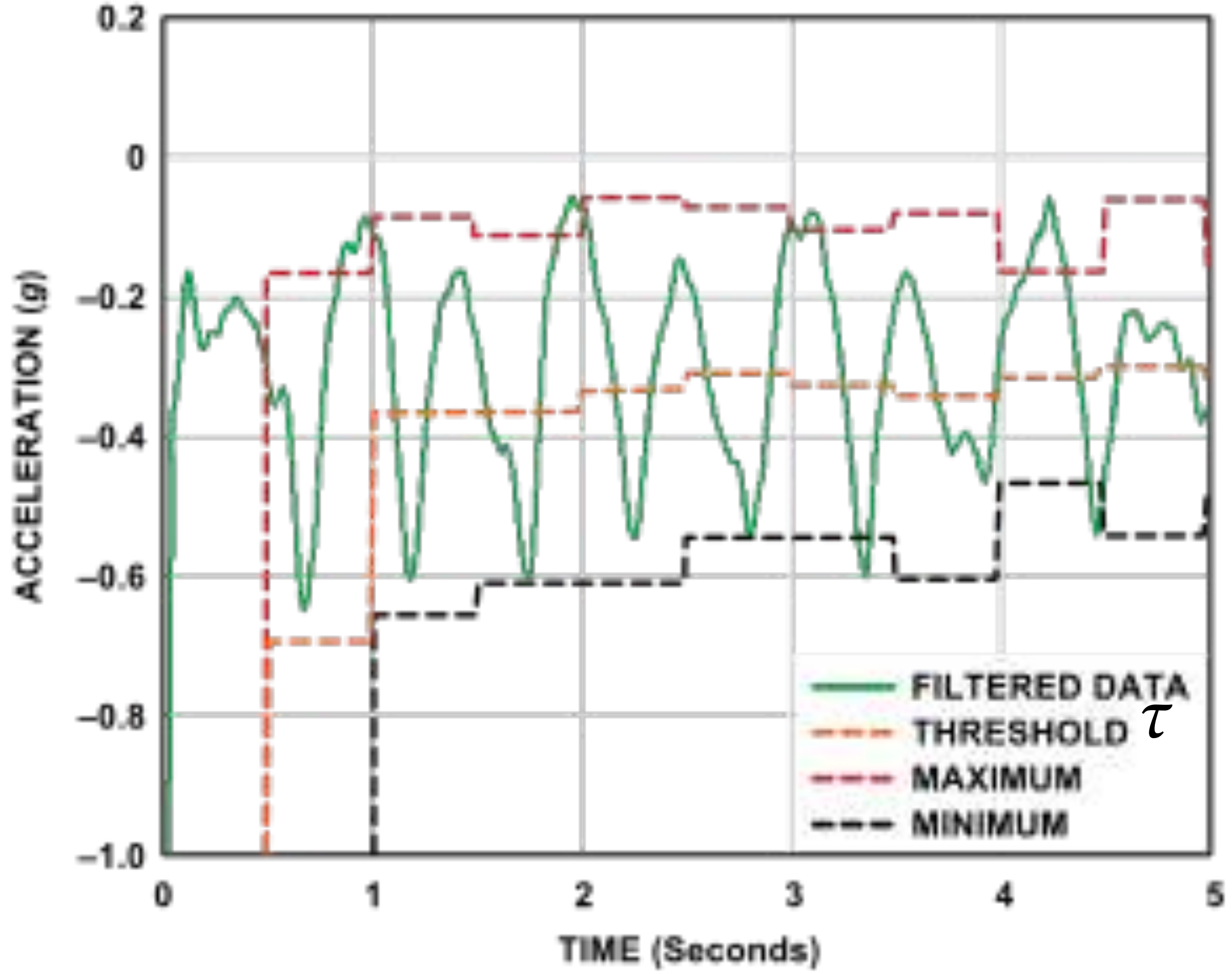


Eelke Folmer: VR-STEP

Simple Method for Detecting the Walking Motion

- Naïve thresholding of accelerometer data from head
  - Project acceleration onto vertical axis
  - Step := acceleration > threshold
- More sophisticated thresholding:
  - Every ½ second, calculate min/max over last ½ sec of data
  - At the same time, calculate threshold
    $$\tau = (\max + \min) / 2$$
  - With every new sample a_i, check
    $$a_i < \tau \;\wedge\; a_{i-1} > \tau$$
    ⇒ step detected

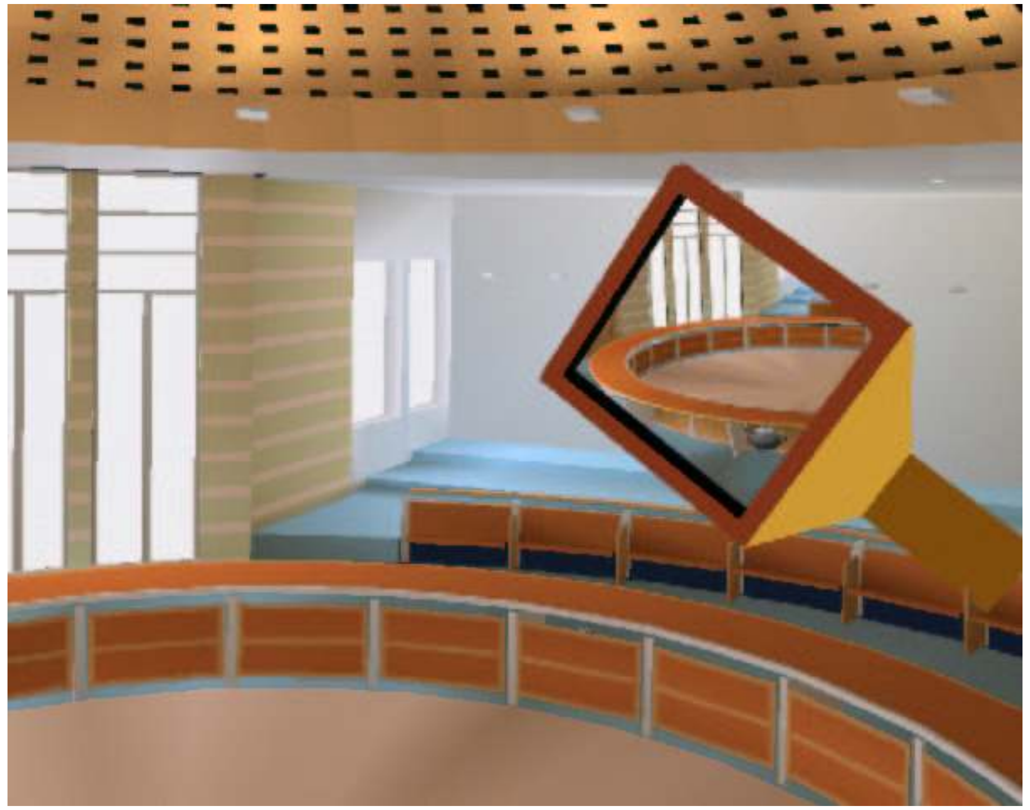


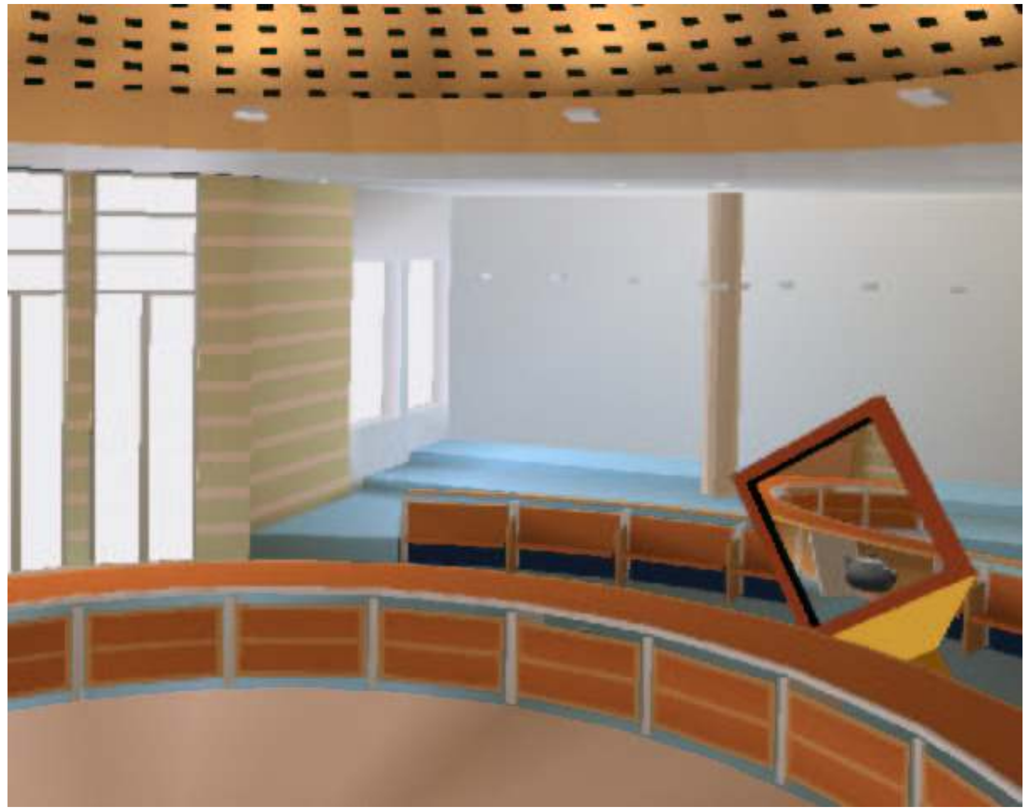


Exploration of VEs using a Magic Mirror

- Task/goal: present a second viewpoint (like inset in an image) intuitively in a VE, and allow for its manipulation
- Idea: use the mirror as a metaphor → "magic mirror"
  - One object serves as hand mirror (could even look like it)
  - Keeps a fixed position relative to camera (follows head motions)
  - Can be manipulated like any other object in the VE
- Additional features (not possible with real mirrors):
  - Zooming
  - Magnification / scaling down of image in mirror
  - Clipping of objects in front of mirror (which occlude mirror)

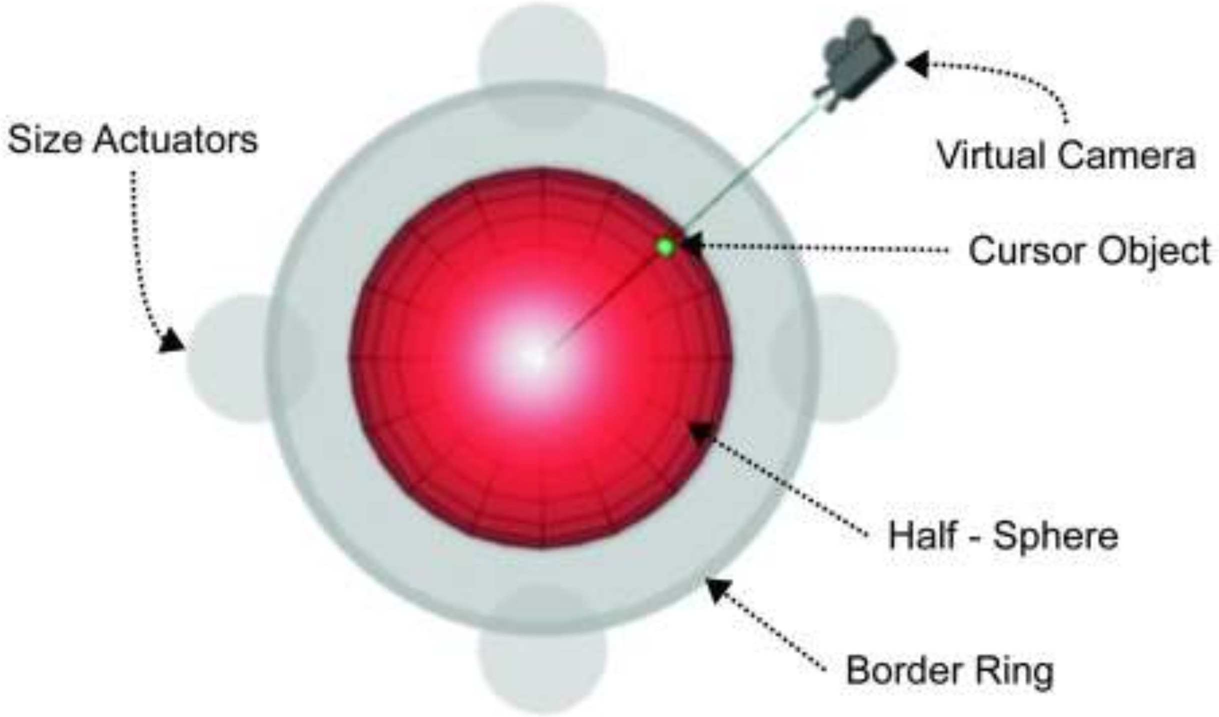
- Implementation:
  - Render scene 2x
  - First, render only into a small viewport (in the shape of the mirror) with mirrored viewpoint
  - Save as texture
  - Second, render into complete viewport from main viewpoint
  - Third, render texture on top of mirror object (no z test)
- Or, use method presented in Computer Graphics class

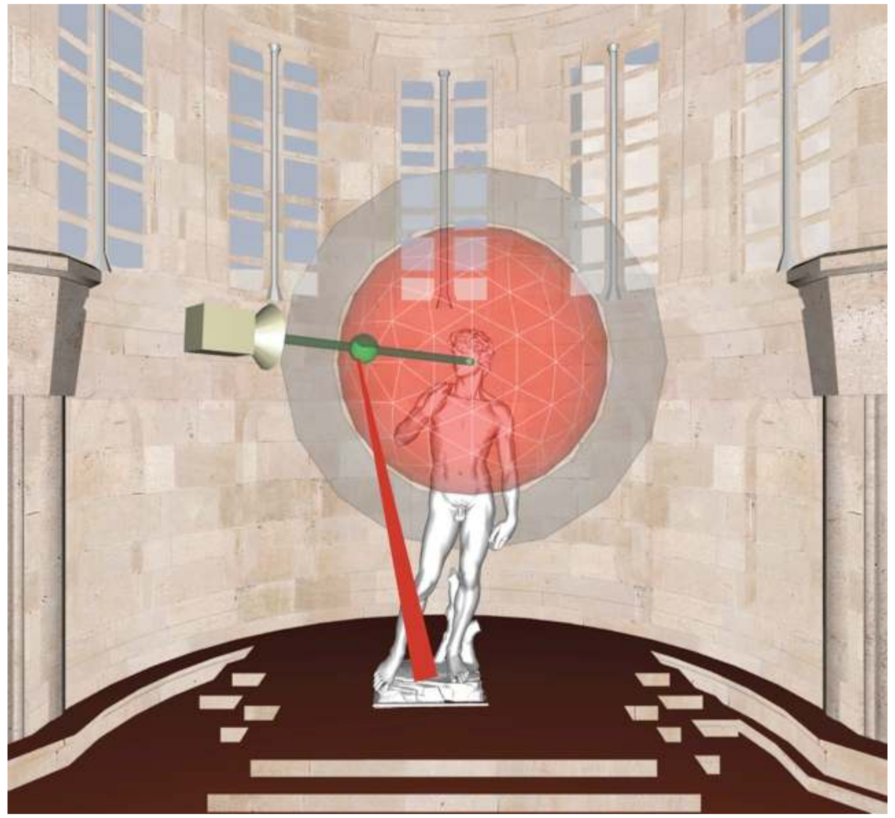


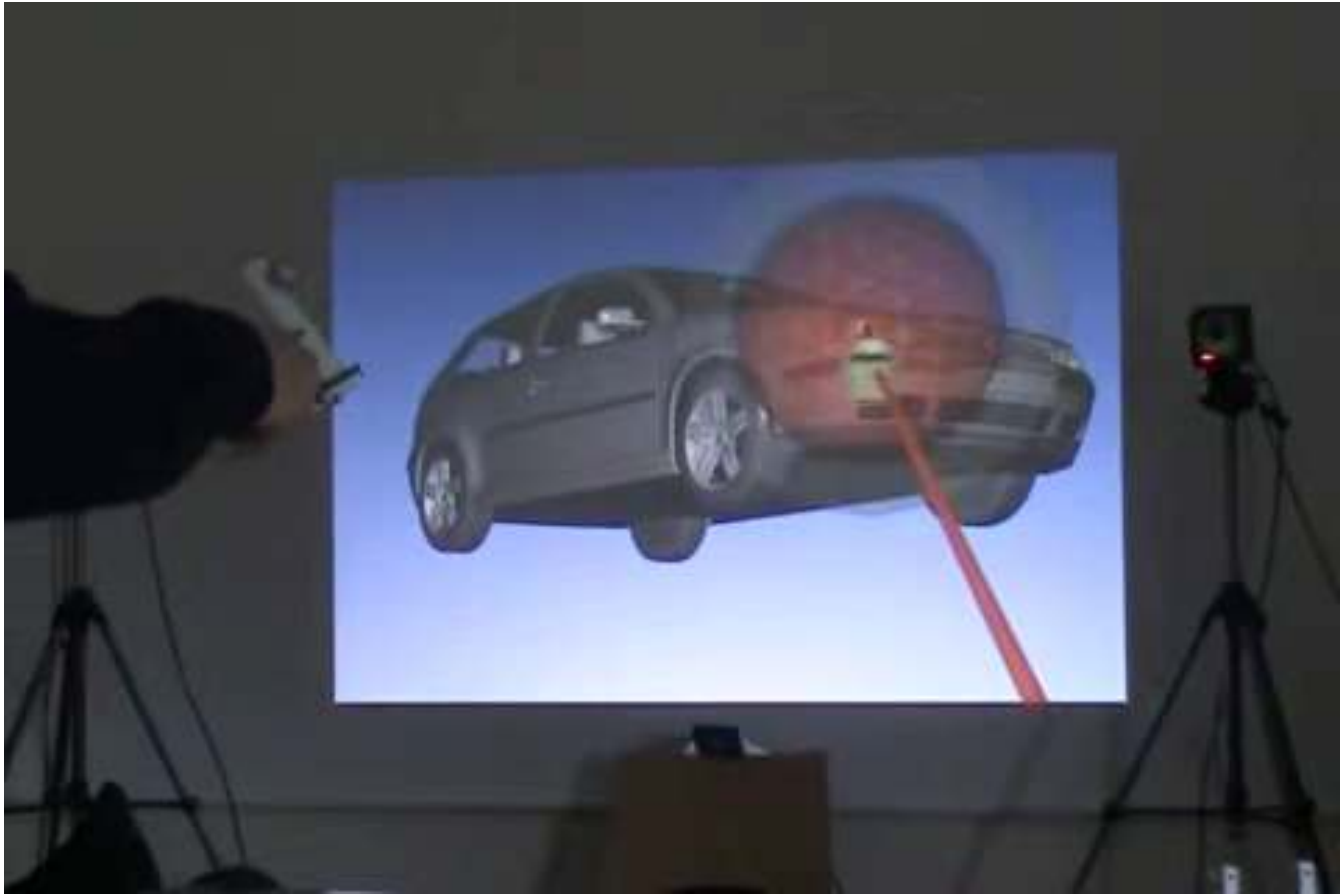


The Immersive Navidget – Example for Task Decomposition

- Metaphor for defining the viewpoint directly
- Input device: *controller* with wheels and buttons
- Decomposition of the task:
  1. Define the center of interest (COI)
     - E.g. by ray casting: intersection point = new COI
     - Make it the center of a sphere
  2. Define radius of sphere = distance of new viewpoint from COI
     - Here: specified using wheel on wand
  3. Define viewpoint on sphere (using ray)
  4. Animate viewpoint on path towards new viewpoint (= smooth teleportation)
- Switch phases using a button on wand







Fidelity Score of Navigation Techniques

- One way for ranking different techniques: classify components w.r.t. several levels, then accumulate all individual scores

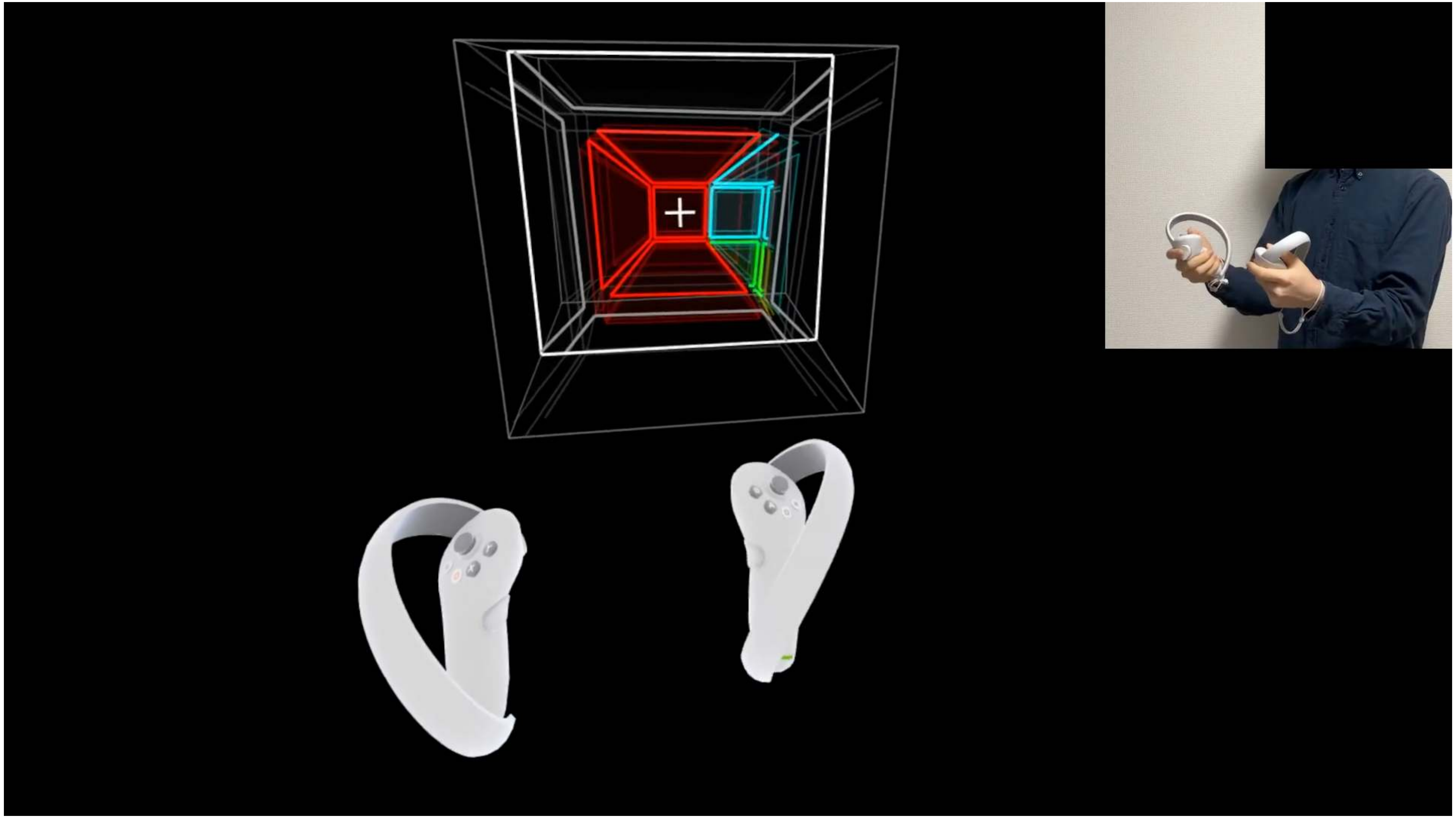
Translation		Rotation	
Body motion	Physicality	Body motion	Physicality
Full body motion (3)	Physical (3)	Full body motion (3)	Physical (3)
Partial body motion (2)	Mixed (2)	Partial body motion (2)	Mixed (2)
Anatomical substitution (1)	Virtual (1)	Anatomical substitution (1)	Virtual (1)
Any other case (0)	Any other case (0)	Any other case (0)	Any other case (0)

[Molina, EuroXR 2023]

Do You Know, or Can You Think of Other Techniques?

https://www.menti.com/p9rdrjti47

Challenge: How to Navigate Intuitively in 4D Space?



[ICAT-EGVE 2023]

Interlude: User Models

- Idea: if we had a model of how users "work", then we could predict how they will interact with a specific UI and what their user performance will be
- Advantage (theoretically): no user studies and no UI mock-ups necessary any more
- Related fields: psychophysics, user interface design, usability

The Power Law of Practice

- Describes, what time is needed to perform an activity after the *n*-th repetition

$$T_n = \frac{T_1}{n^a}$$

T_1 = time needed for first performance of the activity,

T_n = time for *n*-th repetition,

$a \approx 0.2 \dots 0.6$

- Warning:
  - Applies only to mechanical activities, e.g. using the mouse, typing on the keyboard
  - Does not apply to cognitive activities, e.g., learning for exams! ;-)
- This effect, called learning effect, must be kept in mind when designing experiments and analyzing the data!

Hick's Law

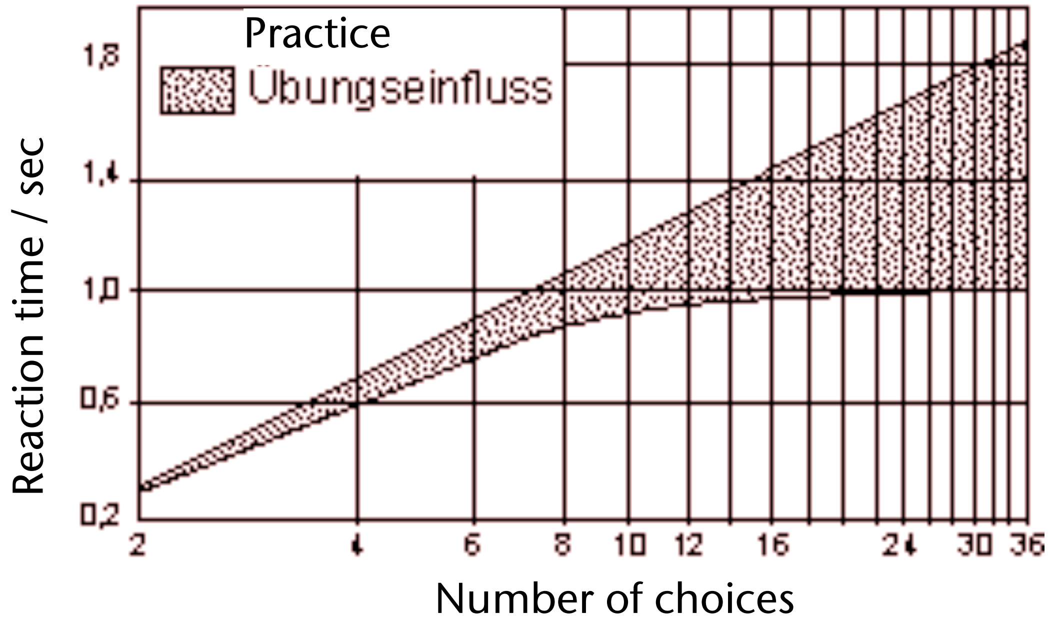
- Describes the time needed to make a 1-out-of-*n* selection, but there cannot be any cognitive workload involved:

$$T = I_c \log_2(n+1) \quad , \quad I_c \approx 150 \text{ msec}$$

  where *n* = number of choices (assumption: choices are uniformly distributed!)

  - Example: *n* buttons + *n* lights, one is lighted up randomly, user has to press corresponding button

- Warning: don't apply this law too blindly!
  - E.g., practice has a big influence on reaction time



- Note: sometimes, Hick's law is taken as proof that one large menu is more time-efficient than several small submenus ("rule of large menus") ...
  I argue this is — mathematically — correct only because of the "+1", for which there is no clear experimental evidence! Besides, there are many other factors involved in large menus (clarity, Fitts' law, ...)

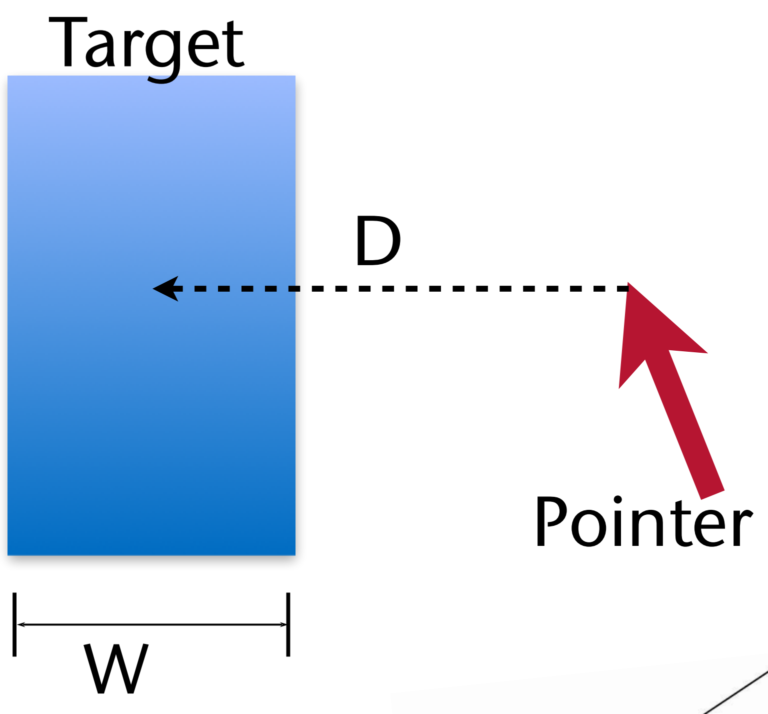
Fitts's Law

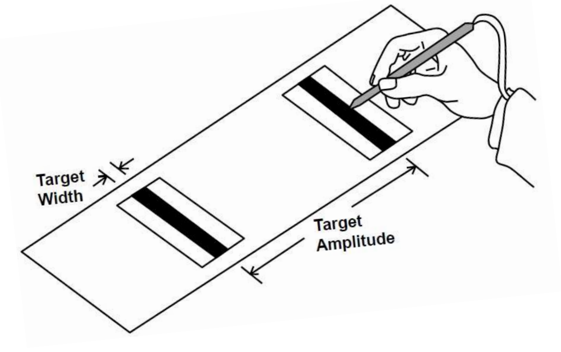
- Describes the time needed to reach a target
- Task: reach and hit a specific target as quickly and as precisely as possible with your hand/ pencil/ mouse/ etc., from a resting position → "target acquisition"
- The law:

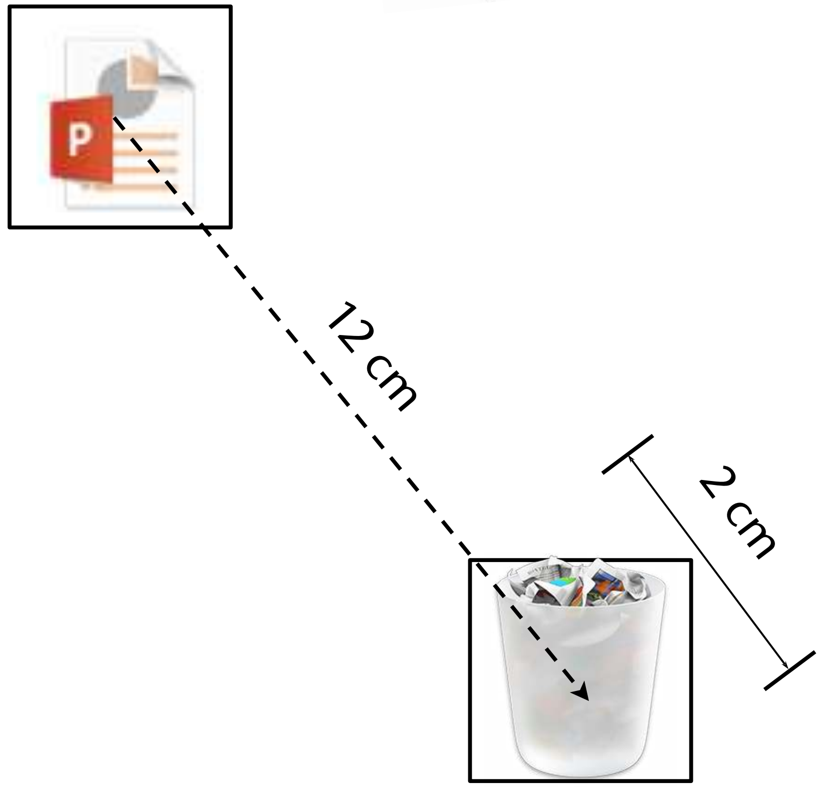
$$T = b \log_2(\frac{D}{W} + 1) + a$$

where D = distance between resting position and target, W = diameter of the target

- The "index of difficulty" (ID) = $\log_2(\frac{D}{W} + 1)$



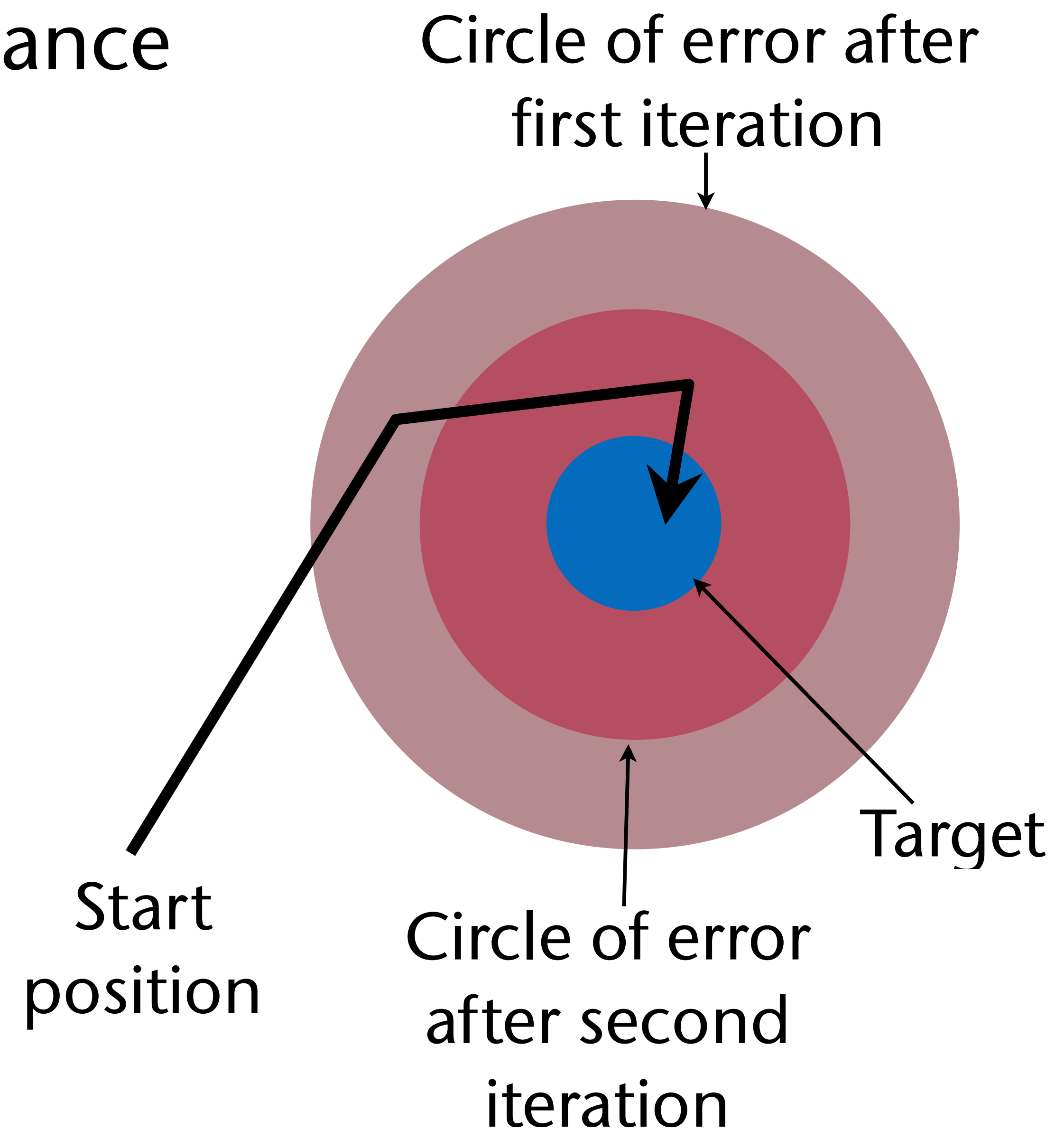




Derivation of Fitts' Law (at least, a plausible explanation)

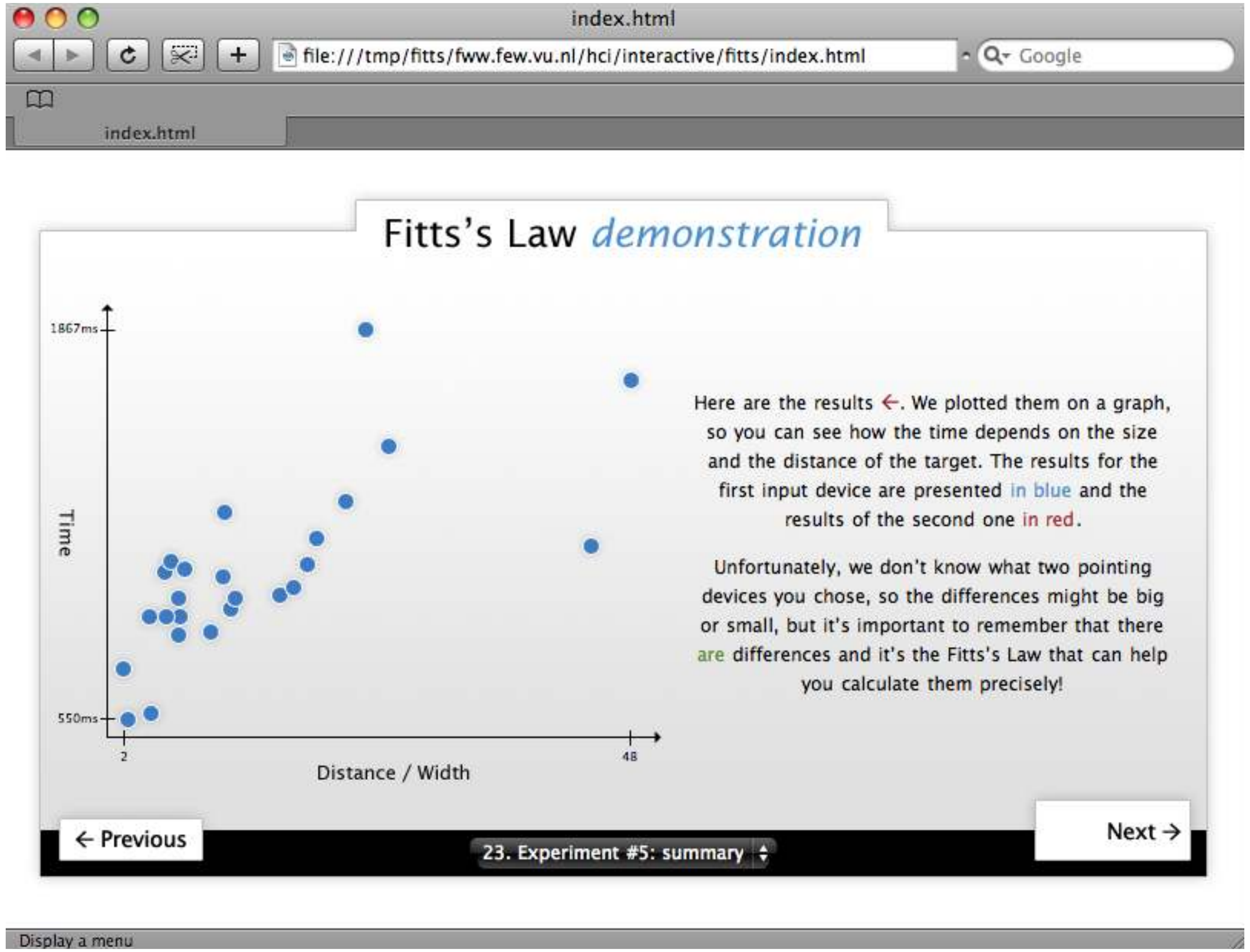
- Assume a control loop "muscle-eye-brain" with
  - Constant processing power of the brain
  - Inaccuracies of movement are proportional to target distance
- Simplification: discretize control loop over time
- Distance of pointer from target
  - $D_0 = D$, $D_i = \lambda D_{i-1} = \lambda^i D$, with $\lambda < 1$
  - After n steps, the pointer is inside the target:

    $$D_n = \lambda^n D < W$$
  - Solving for n yields $n = \log_\lambda(\frac{W}{D})$
  - Each steps takes time τ, brain's "setup" time = a, overall: $T = a + n\tau = a + \tau \log_\lambda(\frac{W}{D}) = a + b \log(\frac{D}{W})$


Circle of error after first iteration
Start position
Circle of error after second iteration
Target

Demo / Experiment

- Fitt's Law applies to many other kinds of target acquisition tasks


index.html
file:///tmp/fitts/fww.few.vu.nl/hci/interactive/fitts/index.html
Google
index.html
Fitts's Law *demonstration*
1867ms
550ms
Time
2
48
Distance / Width
← Previous
23. Experiment #5: summary
Next →
Display a menu

Here are the results ←. We plotted them on a graph, so you can see how the time depends on the size and the distance of the target. The results for the first input device are presented in blue and the results of the second one in red.

Unfortunately, we don't know what two pointing devices you chose, so the differences might be big or small, but it's important to remember that there are differences and it's the Fitts's Law that can help you calculate them precisely!

Marcin Wichary , Vrije Universiteit: http://fww.few.vu.nl/hci/interactive/fitts/

Taking Users' Traits Regarding Error/Speed Into Account

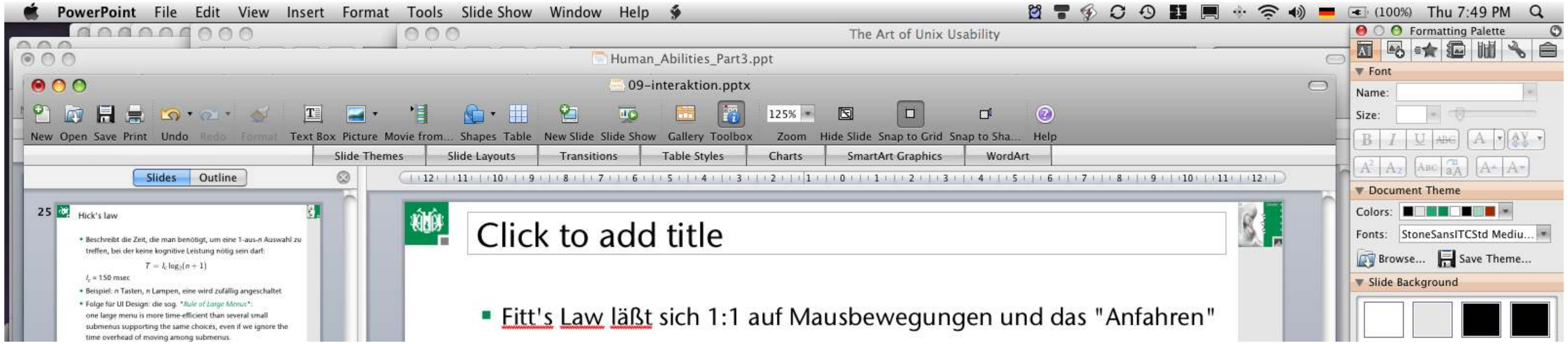
- Some user's prefer speed over "hit accuracy", some the opposite
- One idea: adjust effective target width by "error tolerance"
- Method (w/o statistical derivation):
  - Calculate e = error percentage of the user, $e \in [0,1]$
  - Determine z such that $\pm z$ around the average contains 1-e percent of the area under the *unit* Gauss curve (look up in a table)
  - Then, set

$$W_e = \frac{2.066}{z} W$$

and use this instead of W in Fitts' equation

Consequences from Fitts' Law

- "Rule of Target Size": The size of a button should be proportional to its expected frequency of use
- Other consequences: "*Macintosh fans like to point out that Fitts's Law implies a very large advantage for Mac-style edge-of-screen menus with no borders, because they effectively extend the depth of the target area off-screen. This prediction is verified by experiment.*"
  [Raymond & Landley: "The Art of Unix Usability", 2004]



25 Hick's law

- Beschreibt die Zeit, die man benötigt, um eine 1-aus-n Auswahl zu treffen, bei der keine kognitive Leistung nötig sein darf:

$$T = I_c \log_2(n+1)$$

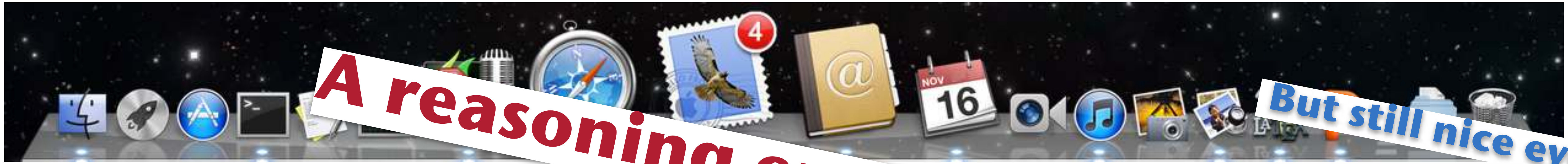
I_c = 150 msec

- Beispiel: n Tasten, n Lampen, eine wird zufällig angeschaltet
- Folge für UI Design: die sog. "*Rule of Large Menus*": one large menu is more time-efficient than several small submenus supporting the same choices, even if we ignore the time overhead of moving among submenus.

Click to add title

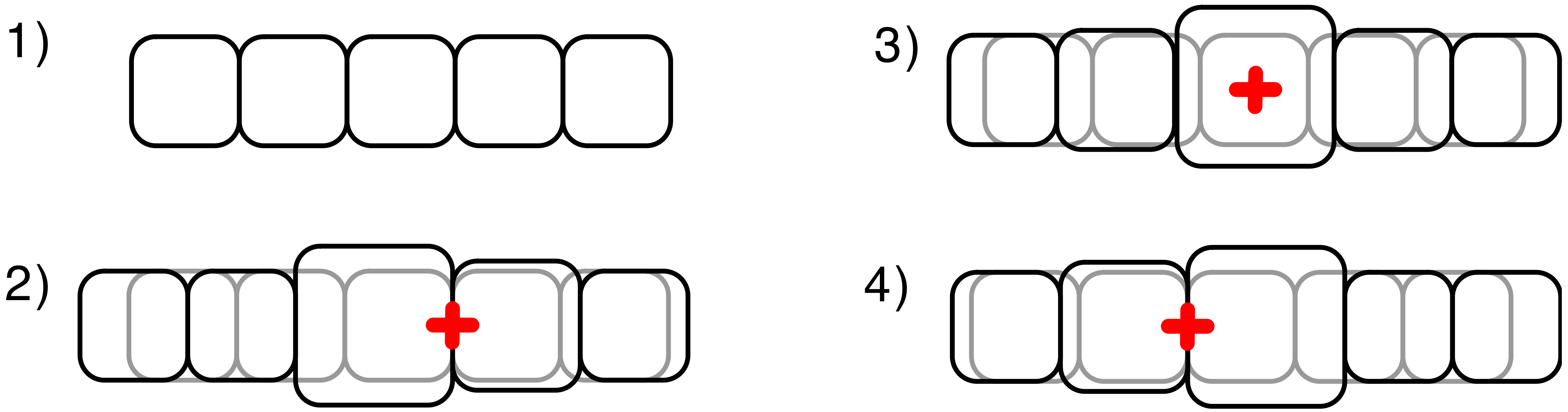
- Fitt's Law läßt sich 1:1 auf Mausbewegungen und das "Anfahren"

- *Tear-off menus* and *context menus*: they decrease the average travel distance *D*
- Apple's "Dock": the size of the icons gets adjusted dynamically



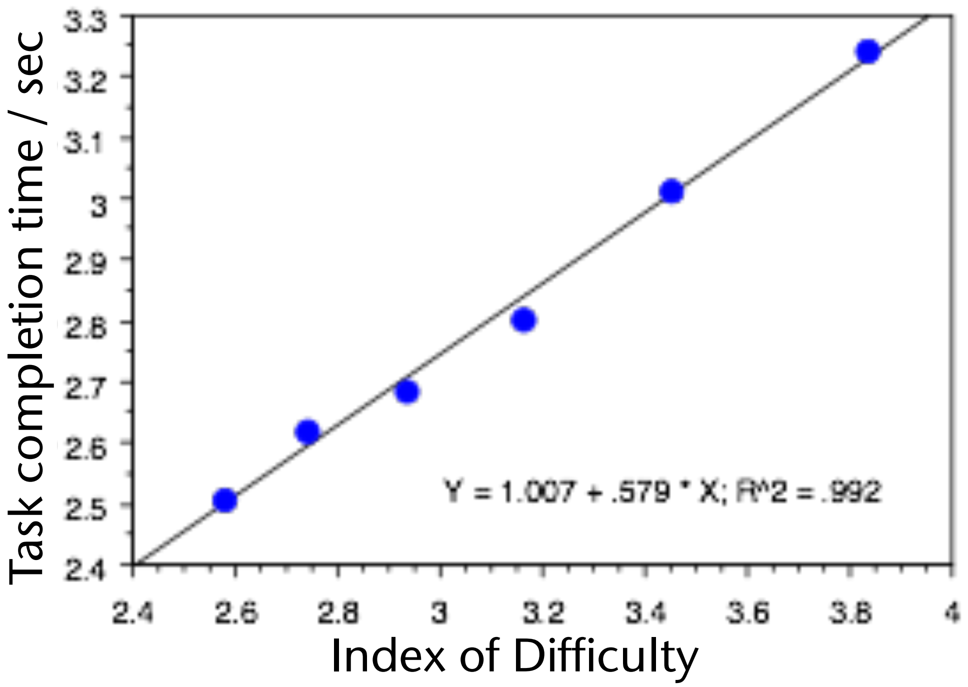
A reasoning error!

But still nice eye candy ;-)

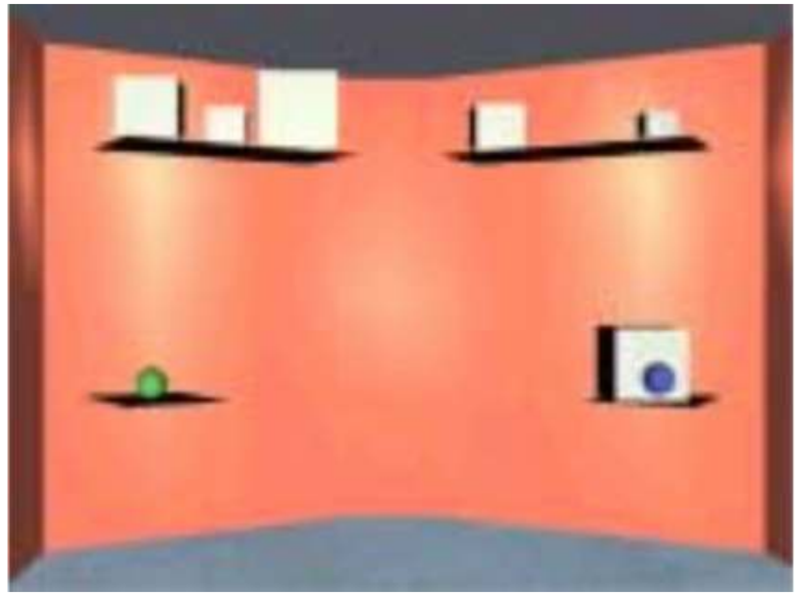


Fitts' Law Also Holds for Hand-Off Tasks (or Does It?)

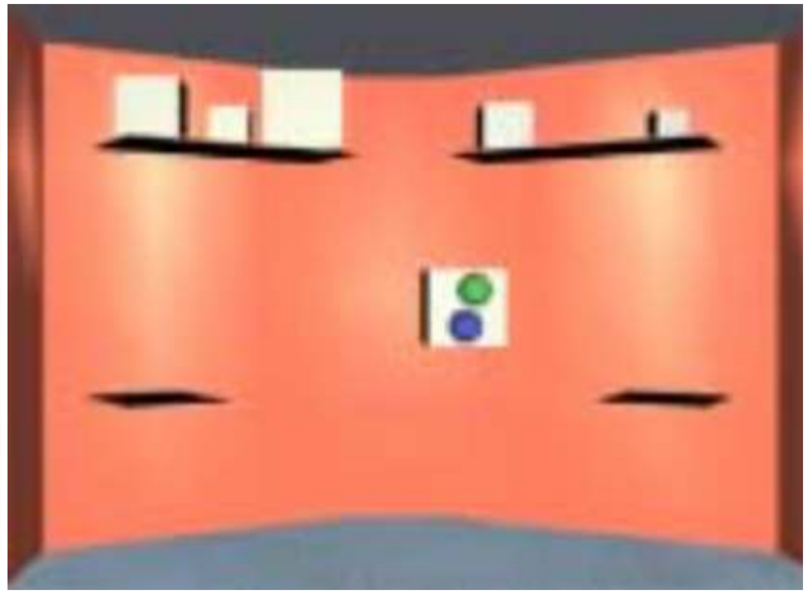
- One user grabs a virtual cube using a point probe, moves it towards the other user's point probe, then the other user tries to grab it
- Result:



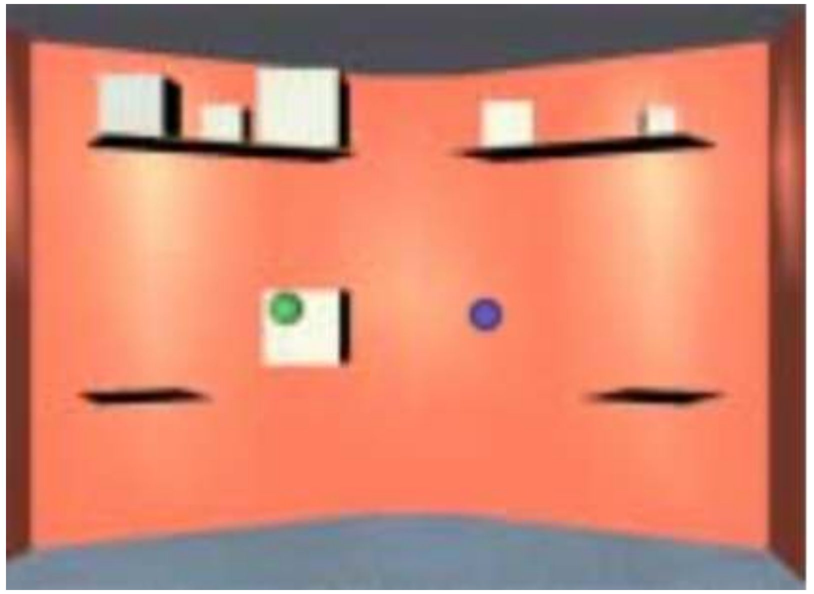
1



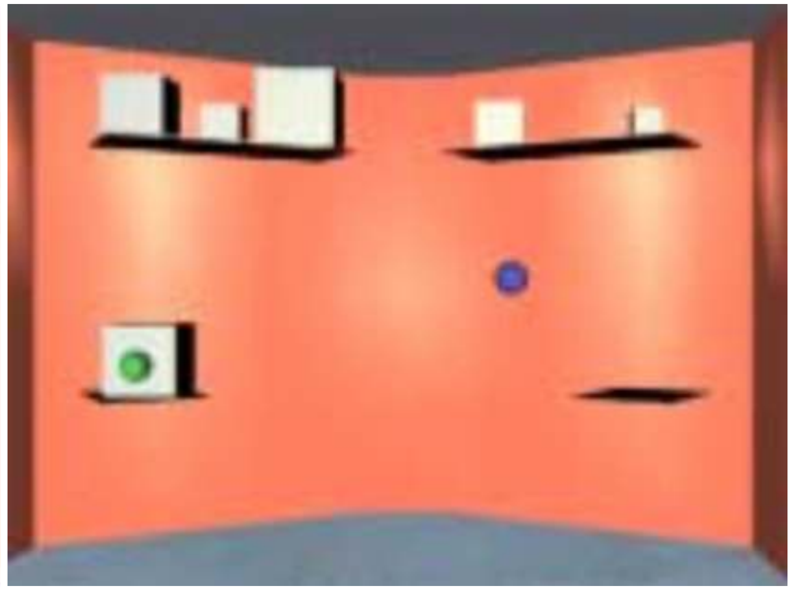
2



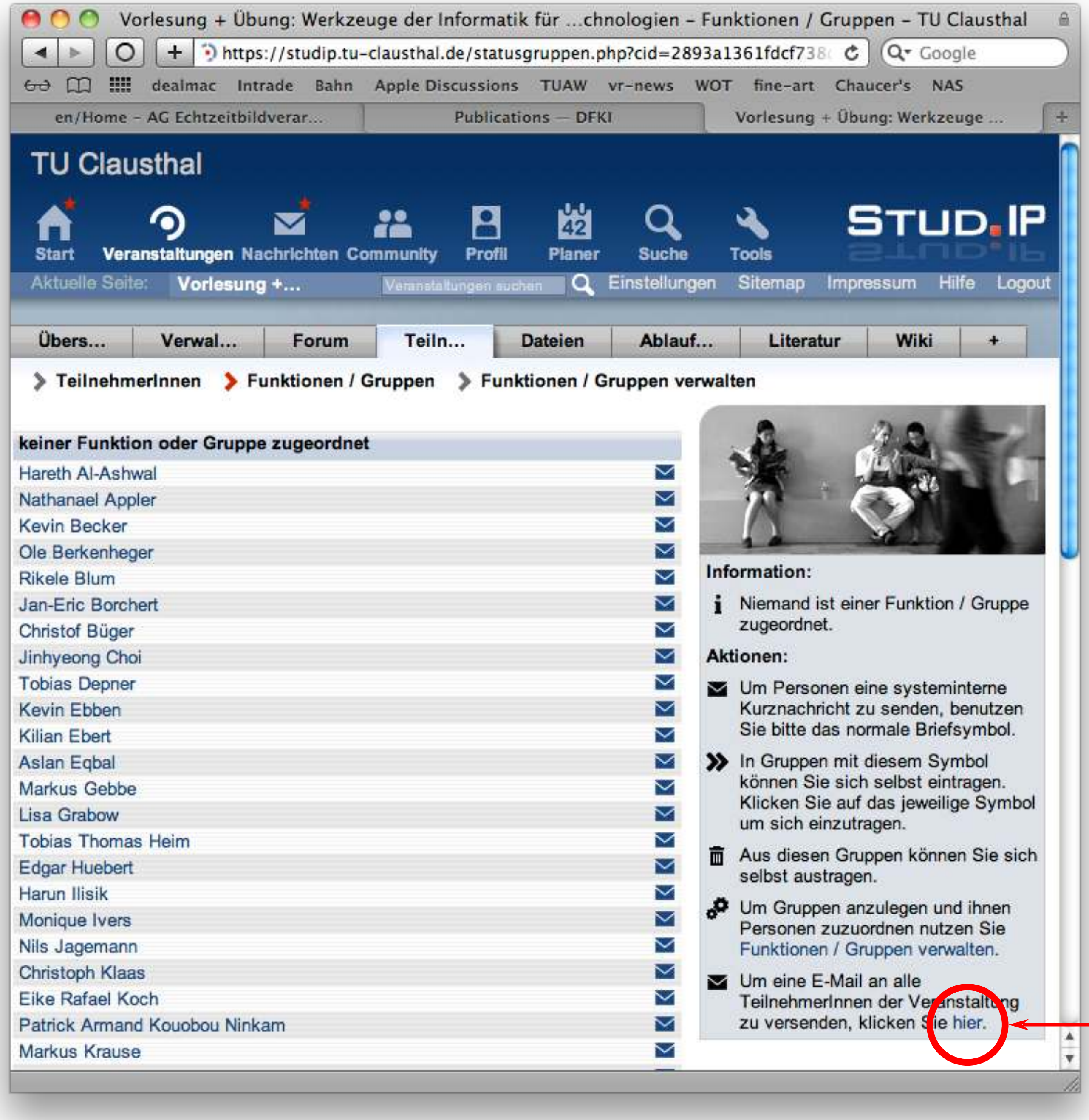
3



4



Bad Example: Studip (Until 2017)



Vorlesung + Übung: Werkzeuge der Informatik für ...chnologien – Funktionen / Gruppen – TU Clausthal

https://studip.tu-clausthal.de/statusgruppen.php?cid=2893a1361fdcf738

keiner Funktion oder Gruppe zugeordnet

- Hareth Al-Ashwal
- Nathanael Appler
- Kevin Becker
- Ole Berkenheger
- Rikele Blum
- Jan-Eric Borchert
- Christof Büger
- Jinhyeong Choi
- Tobias Depner
- Kevin Ebben
- Kilian Ebert
- Aslan Eqbal
- Markus Gebbe
- Lisa Grabow
- Tobias Thomas Heim
- Edgar Huebert
- Harun Ilisik
- Monique Ivers
- Nils Jagemann
- Christoph Klaas
- Eike Rafael Koch
- Patrick Armand Kouobou Ninkam
- Markus Krause

Information:

Niemand ist einer Funktion / Gruppe zugeordnet.

Aktionen:

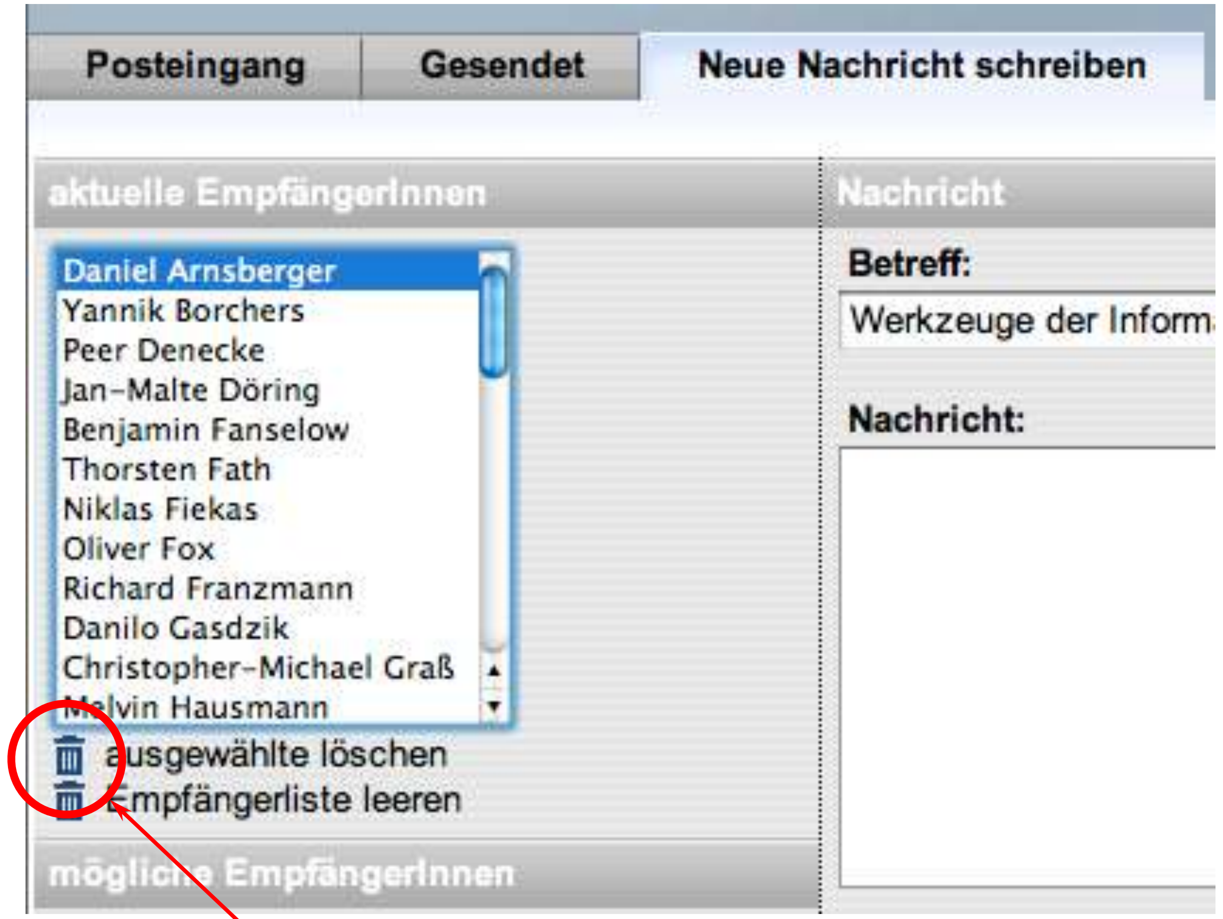
Um Personen eine systeminterne Kurznachricht zu senden, benutzen Sie bitte das normale Briefsymbol.

In Gruppen mit diesem Symbol können Sie sich selbst eintragen. Klicken Sie auf das jeweilige Symbol um sich einzutragen.

Aus diesen Gruppen können Sie sich selbst austragen.

Um Gruppen anzulegen und ihnen Personen zuzuordnen nutzen Sie Funktionen / Gruppen verwalten.

Um eine E-Mail an alle TeilnehmerInnen der Veranstaltung zu versenden, klicken Sie hier.



- Daniel Arnsberger
- Yannik Borchers
- Peer Denecke
- Jan-Malte Döring
- Benjamin Fanselow
- Thorsten Fath
- Niklas Fiekas
- Oliver Fox
- Richard Franzmann
- Danilo Gasdzik
- Christopher-Michael Graß
- Melvin Hausmann

Betreff: Werkzeuge der Inform

Nachricht:

Then, this small symbol is a button!
And it is close to a very "dangerous" button!

This little word is a button!
(and hard to distinguish from the rest
of the text/background!)

Quiz on Fitts

https://www.menti.com/p9rdrjti47

Limitations and Myths of Fitts's Law

- Fitts's Law (in the version given here) cannot capture all aspects of a GUI
  - E.g. moving targets (like scrollable lists)
- Task completion time does **not** depend so much on the device! (by observation)
  - I.e., different users with lots of experience with different devices (e.g., mouse and trackball), same targets ⟶ same task completion times
- There are many other decisions with regards to the design of a UI that are contrary to an application of Fitts's law!

See fun and instructive quiz "A Quiz Designed to Give You Fitts"
For more myth busting, see "A Lecture on Fitts' Law" by Heiko Drewes
Both available on the homepage of this VR course

Fitts' Law with Lag

- In many interactive systems (VR, telepresence, etc.), more or less lag is present
- Fitts' law with lag:

$$T = a + b(l_h + l_s) \cdot \mathrm{ID}$$

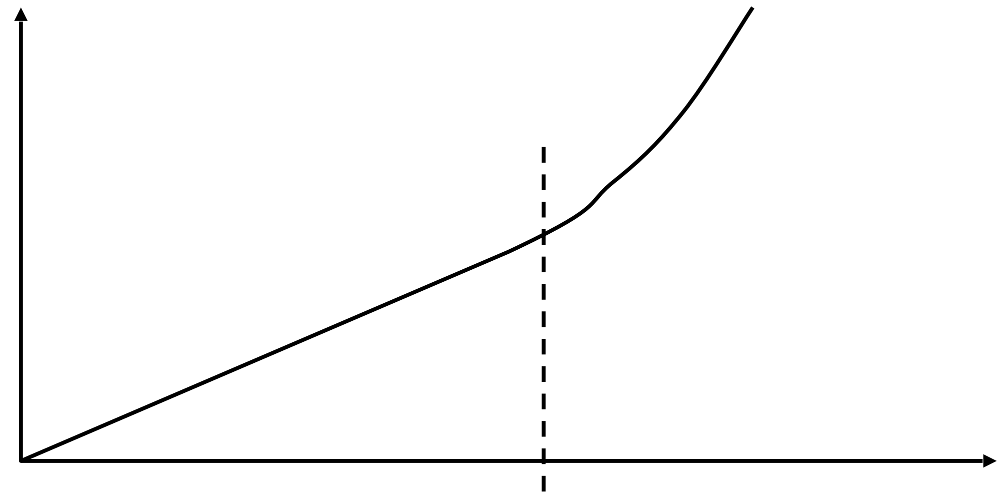
where l_s = system lag ("from motion to photon"),
l_h = "human lag" ("from photon to motion"),
a = time to initialize and terminate task (task dependent),
b = "number of sensory-motor control loop cycles in brain".
Starting values for some parameters by rule of thumb:
b ≈ 1.6
l_h ≈ 0.1,..., 0.3

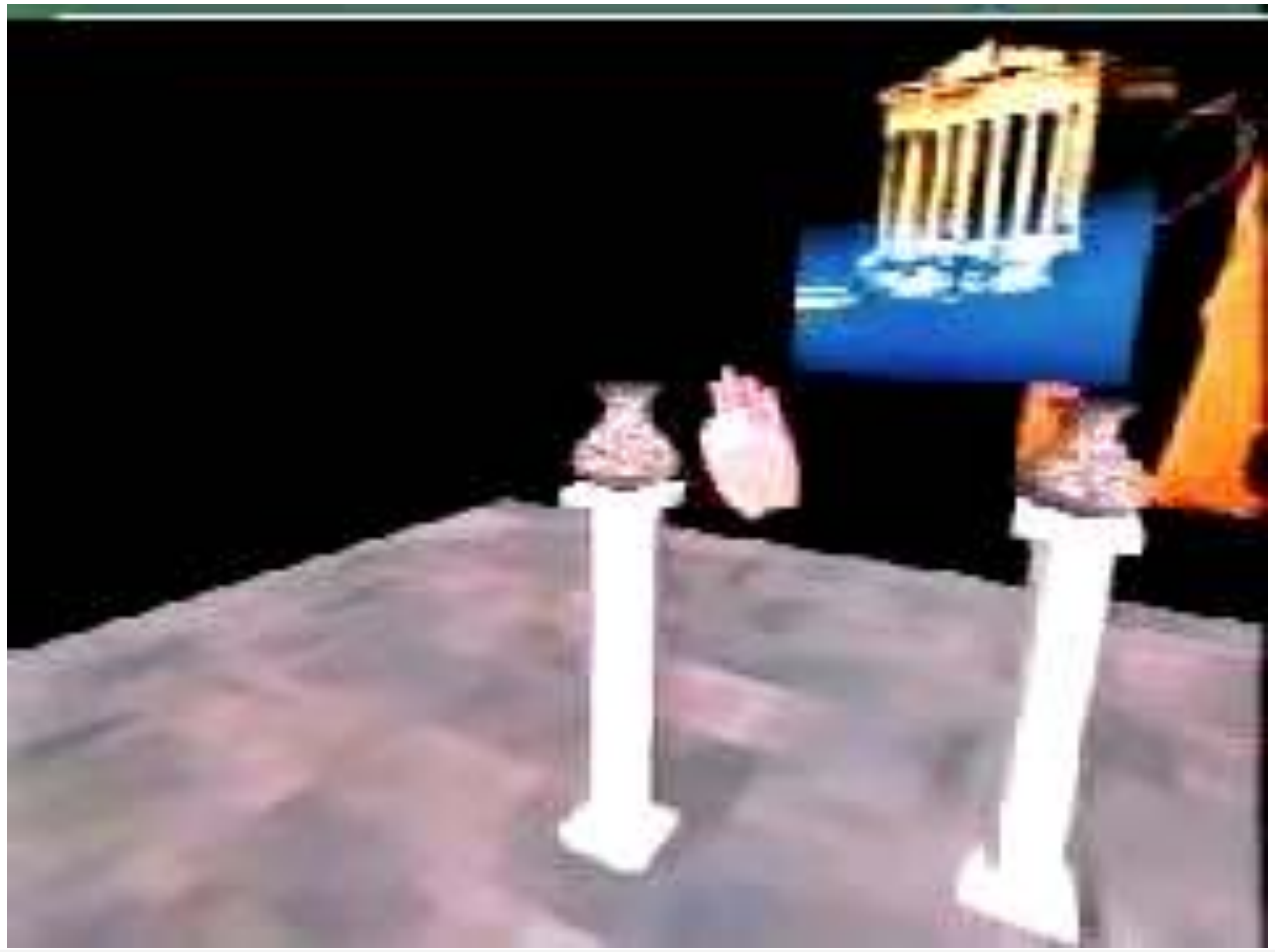
Isomorphic vs Non-Isomorphic Techniques

- Display space = virtual space containing the virtual "pointer"
- Control space = physical space containing the tracker/controller/user
- Control-Display ratio (C-D ratio) = the ratio of the movement in physical space over the resulting movement in display space
- Isomorphic mapping:
  - 1:1 correspondence between physical space and virtual space
  - Better suited for direct interaction techniques
- Non-isomorphic mapping:
  - "Magic" tools (interaction metaphors) expands working volume or precision
  - Better suited for remote interaction techniques
  - Most interaction techniques are non-isomorph

Go-Go Technique: Non-Linear Mapping for Direct Selection/Manipulation

- Direct selection/manipulation requires direct touching & grasping objects with virtual hand
- Goal: increase working volume
- Idea:
  - Scale tracking values non-linearly outside the "near field"
  - Keep linear scaling in near-field for better precision
- Suitable for head and hand tracking
- Works only with isometric input devices
- Disadvantages:
  - Proprioception gets lost
  - No remote precision handling





Interaction Techniques for Selection

- Definition:

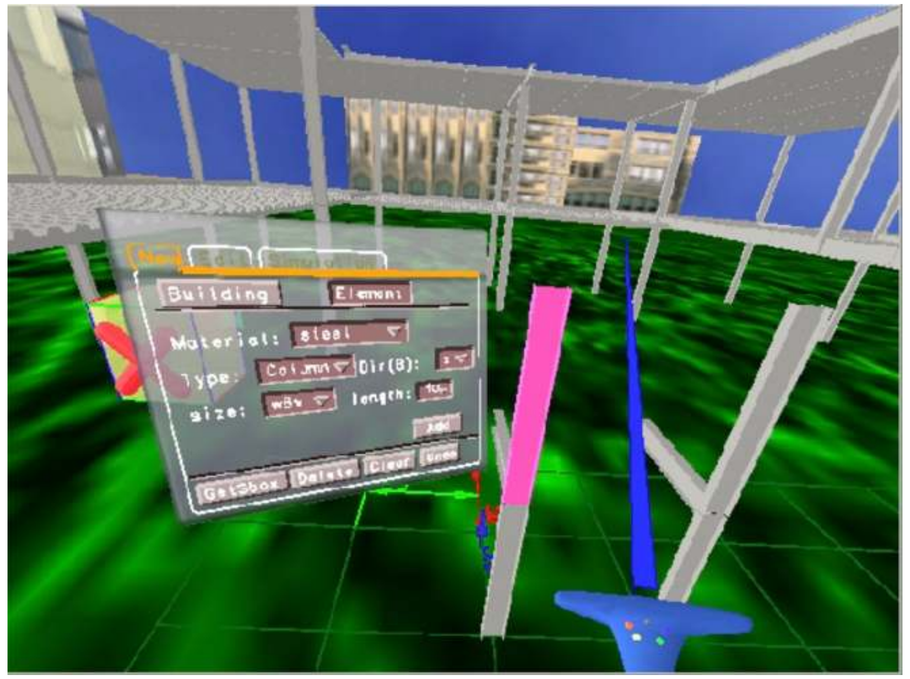
  task decomposition = break the task down into small steps

- One of the standard tools for interaction design
- Task Decomposition for 3D selection in VR:
  1. Switch selection mode on
  2. Identify object(s) to be included in selection (pointing)
     - Often, this is some form of target acquisition (but not necessarily!)
     - Recommendation: always give some kind of feedback during this step
     - Remark: in psychology, this step is often called pointing or, more generally, referential gesture (if performed using gestures)
  3. Confirm/cancel
  4. Feedback: which objects are actually selected

Some Possibilities for Step 2 (Identifying Objects)

- Direct = touch with hand
- Ray-based (ray casting)
  - E.g., shoot "laser pointer" from virtual hand into scene
  - Or: extend a (conceptual) ray from the current viewpoint through finger tip (a.k.a. occlusion technique)
- Volume-based, e.g. using a cone around the ray
- Speech (name objects)
- Menus (how to select items in menu?)
- Mixed techniques:
  - Image plane interaction (later)
  - World-in-Miniature (dito)
  - Etc.

Laser pointer

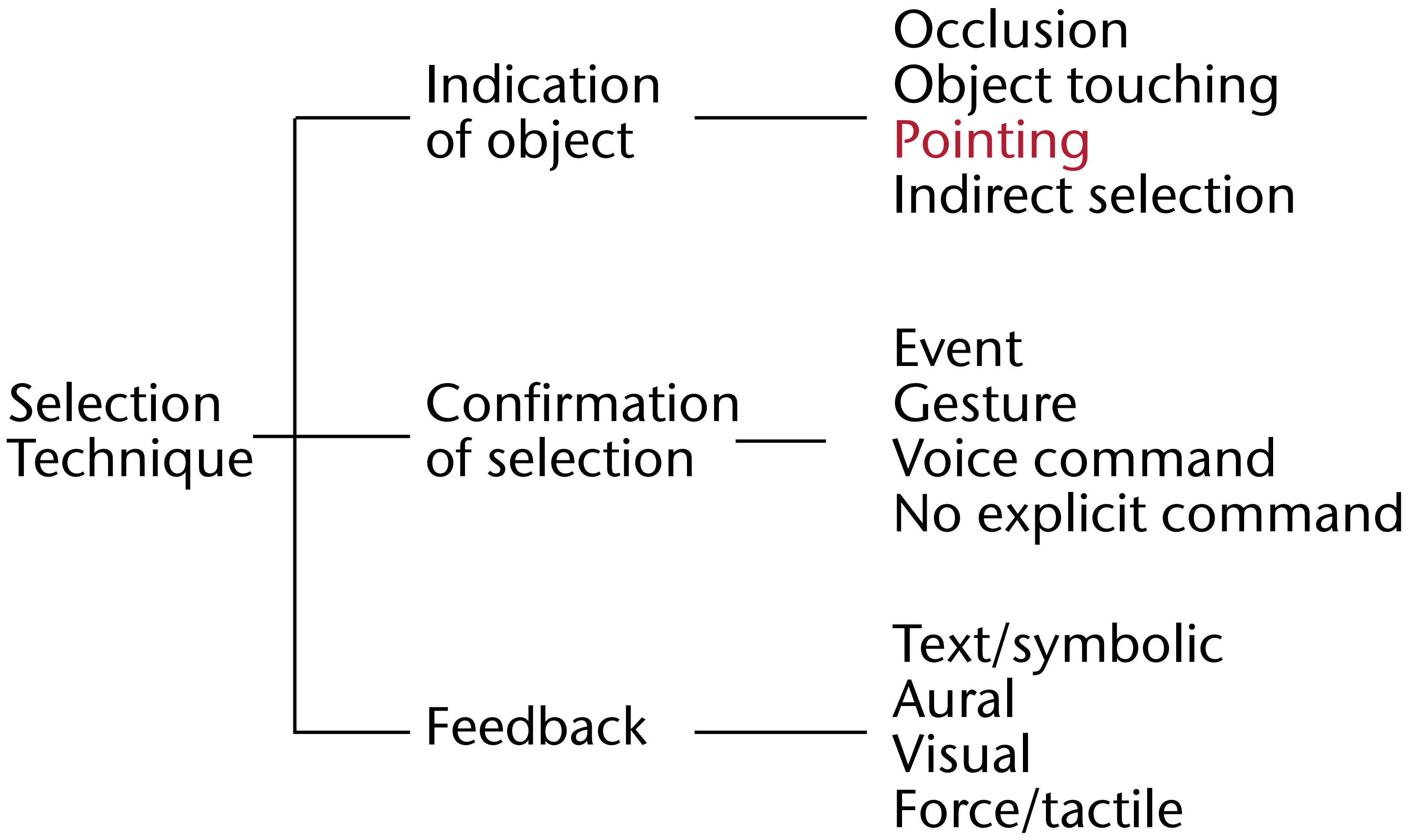


Texture
VTLogo
Polka Dot
Square Dot
Horizonta
Vertical
Color Blast

Menu with ray technique

Overview of Some Ray-Based Techniques

Name of Technique	Volume	Origin	Direction
Raycasting	ray	H	h
Flashlight	cone	H	h
Two-handed pointing	ray	H_2	$H – H_2$
Occlusion selection	ray	E	H - E
Aperture	cone	E	H - E



Selection Technique

Indication of object

Occlusion
Object touching
Pointing
Indirect selection

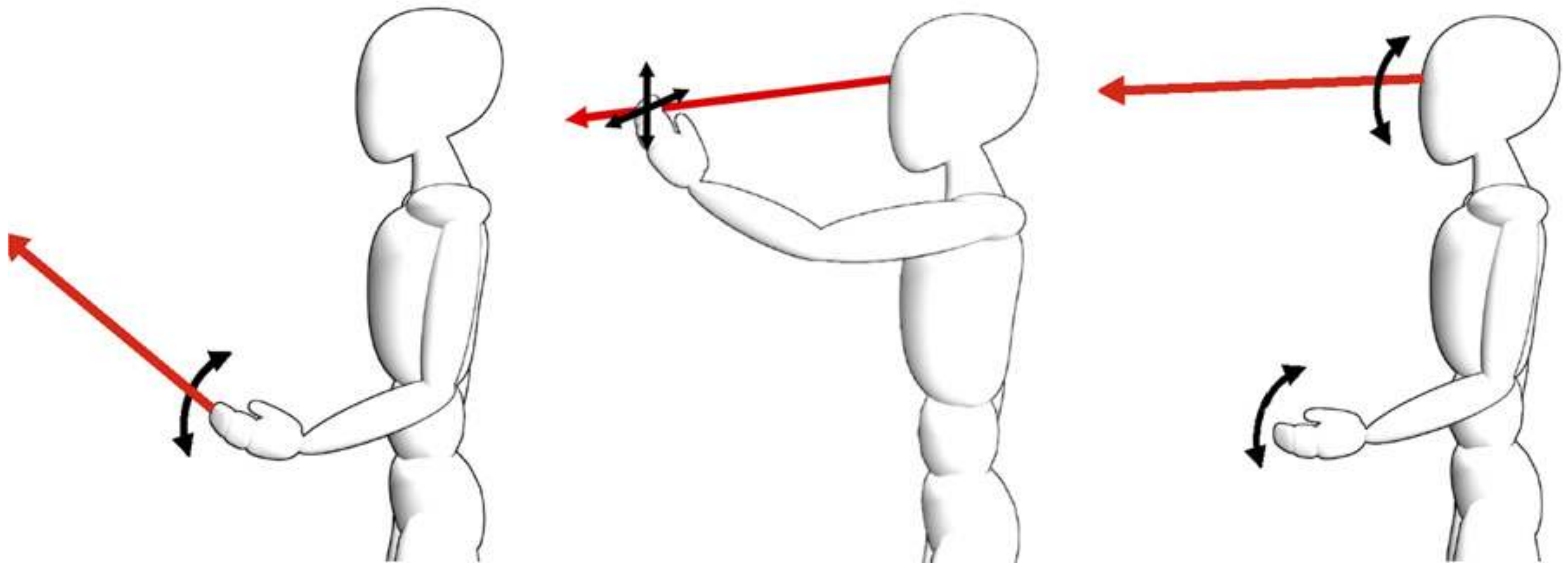
Confirmation of selection

Event
Gesture
Voice command
No explicit command

Feedback

Text/symbolic
Aural
Visual
Force/tactile

Variables:
h = pointing direction of the hand
H = hand position
E = viewpoint
H_2 = position of the left hand



Fitts' Law for Distal Pointing

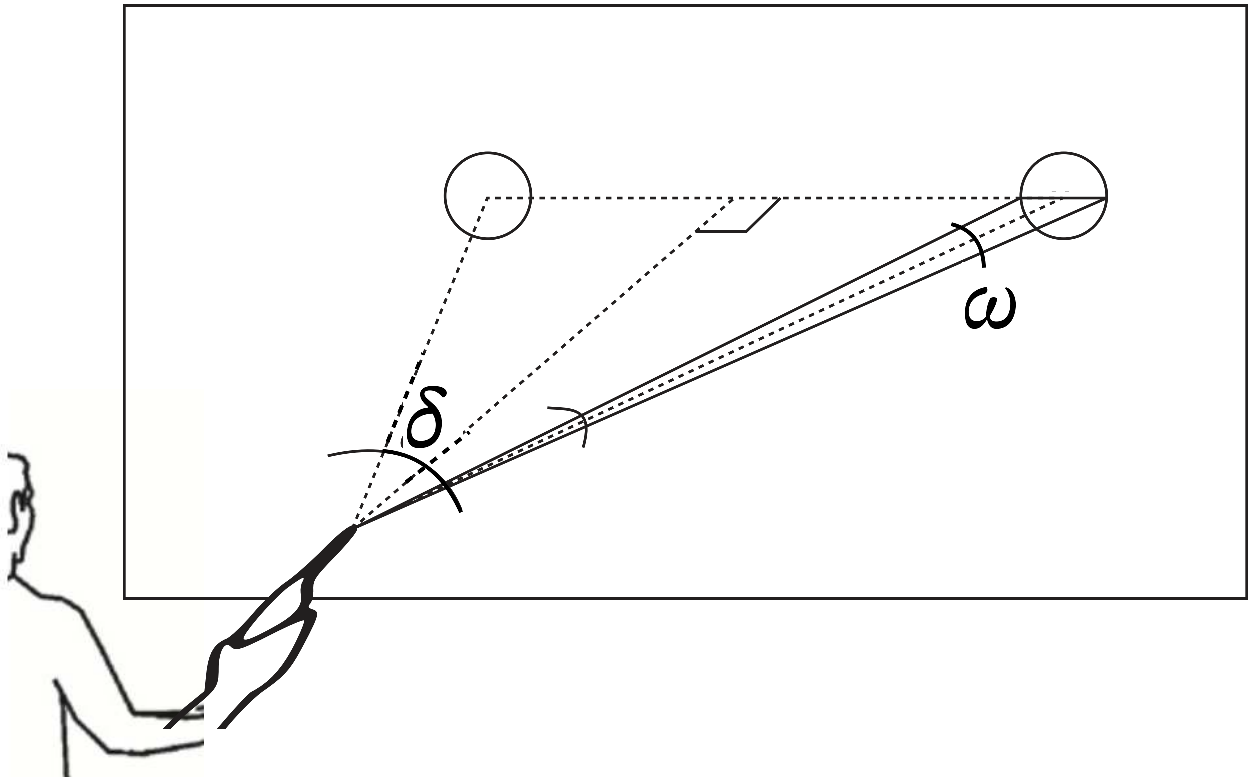
- Analog of "distance" and "width" = respective subtended angles
- Findings:
  - Angles are better than linear distances
    - Hands are rotated, not translated
  - Target size (angular) has more influence than angular distance
- The law for distal pointing:

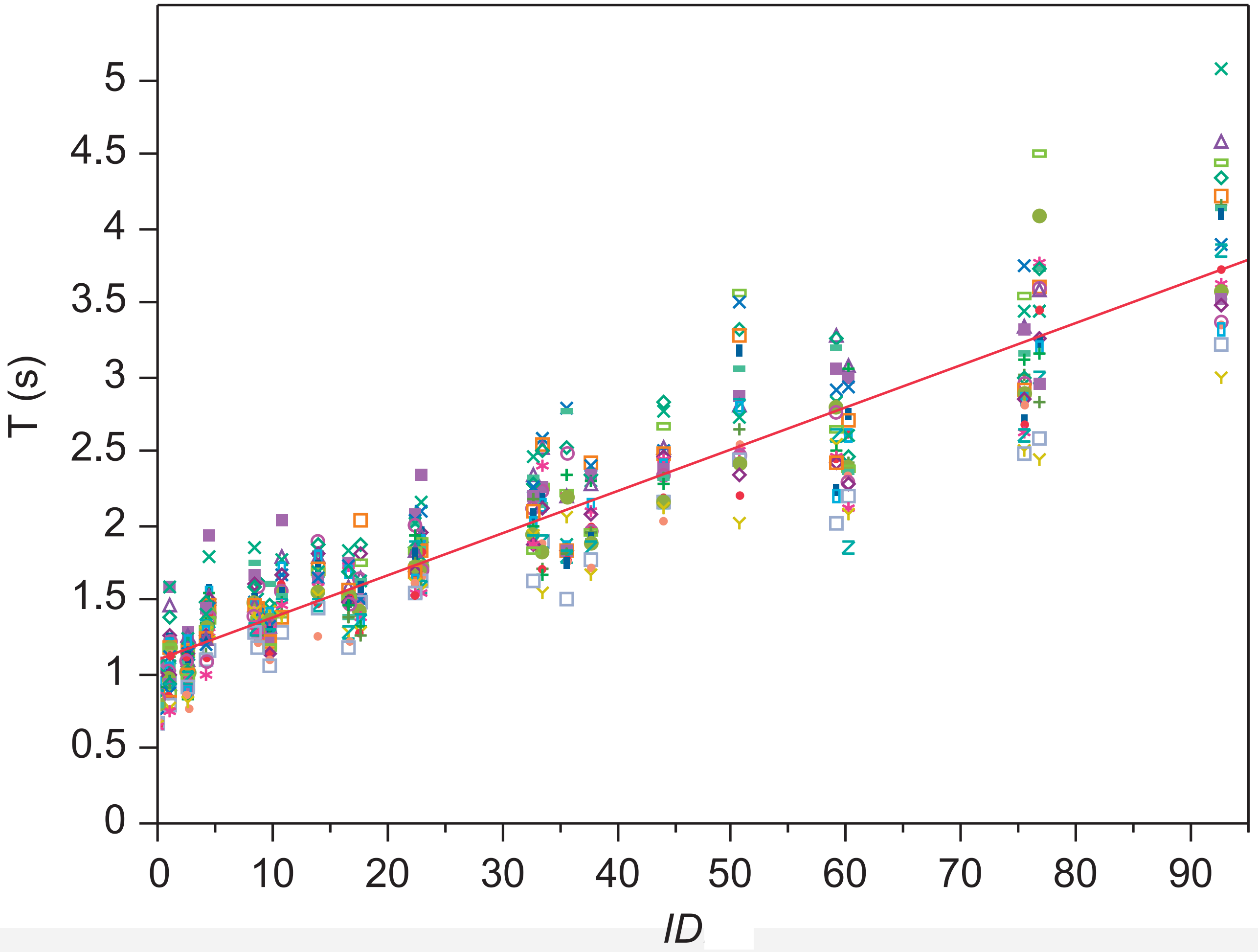
$$T = b \log_2 \left(\frac{\delta}{\omega^k} + 1 \right) + a$$

with a ≈ 1, b ≈ 0.03, k ≈ 3,

δ = angular "distance", start to target direction

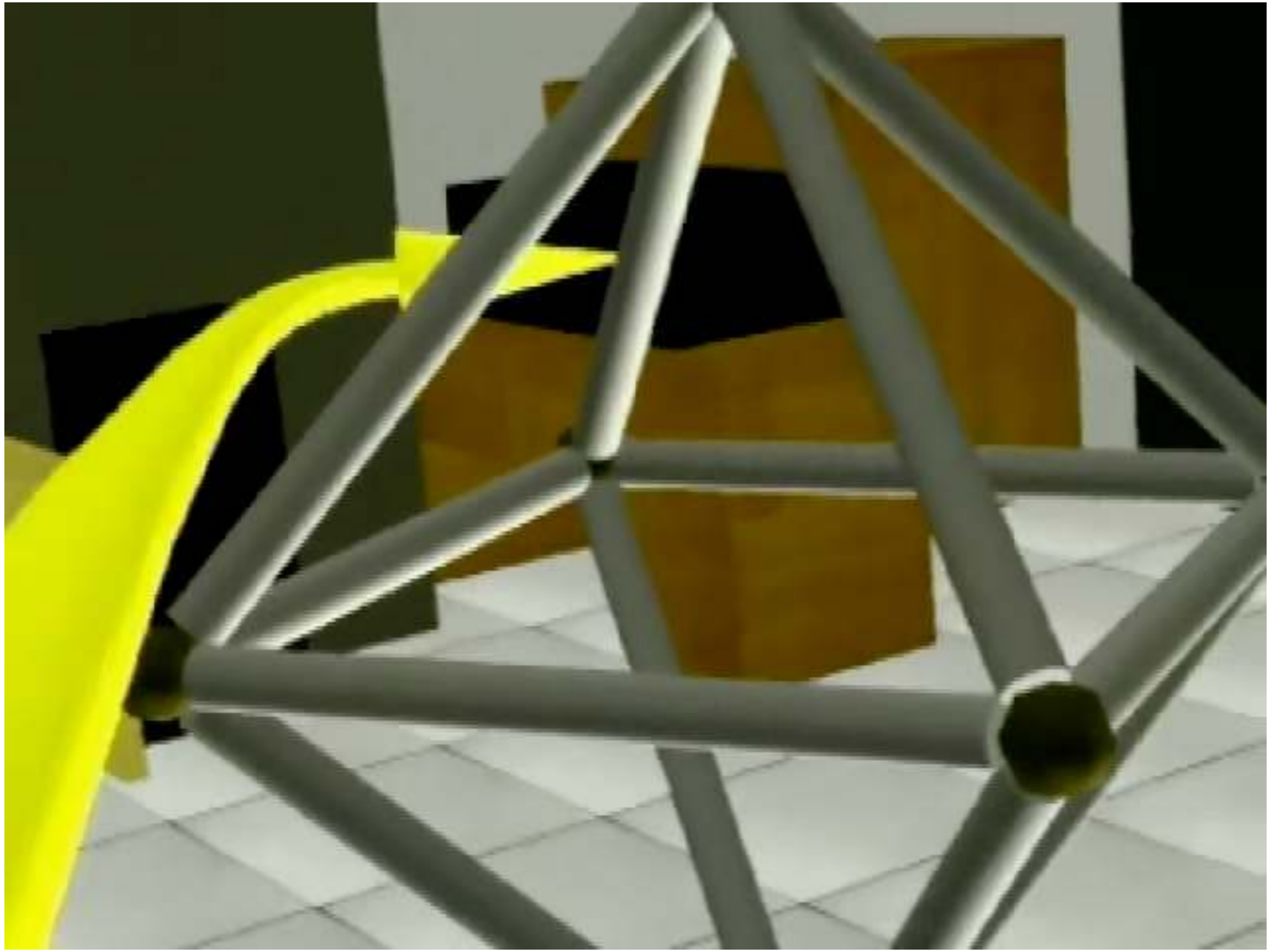
ω = angular "width" of target





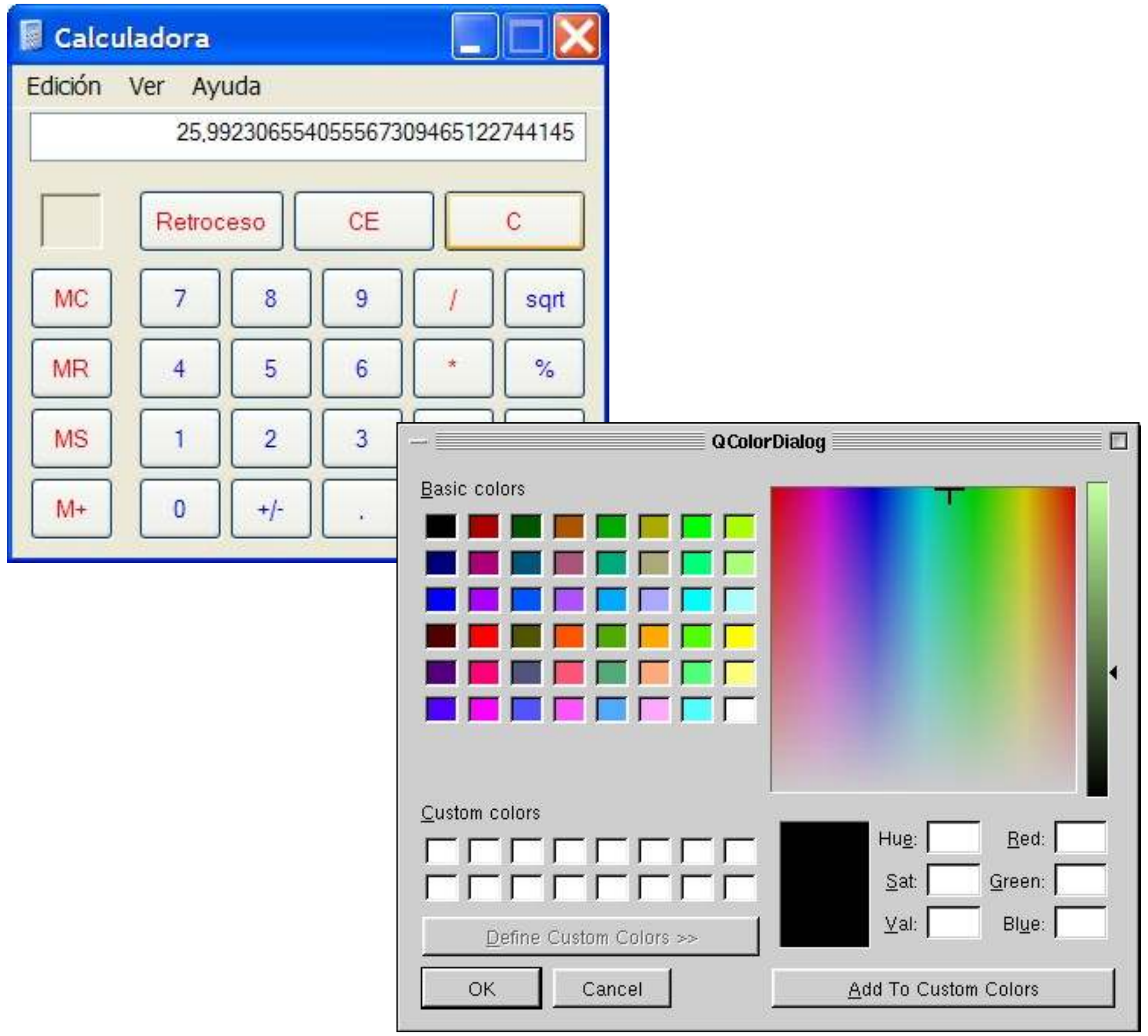
More Selection Techniques: Flexible Pointers

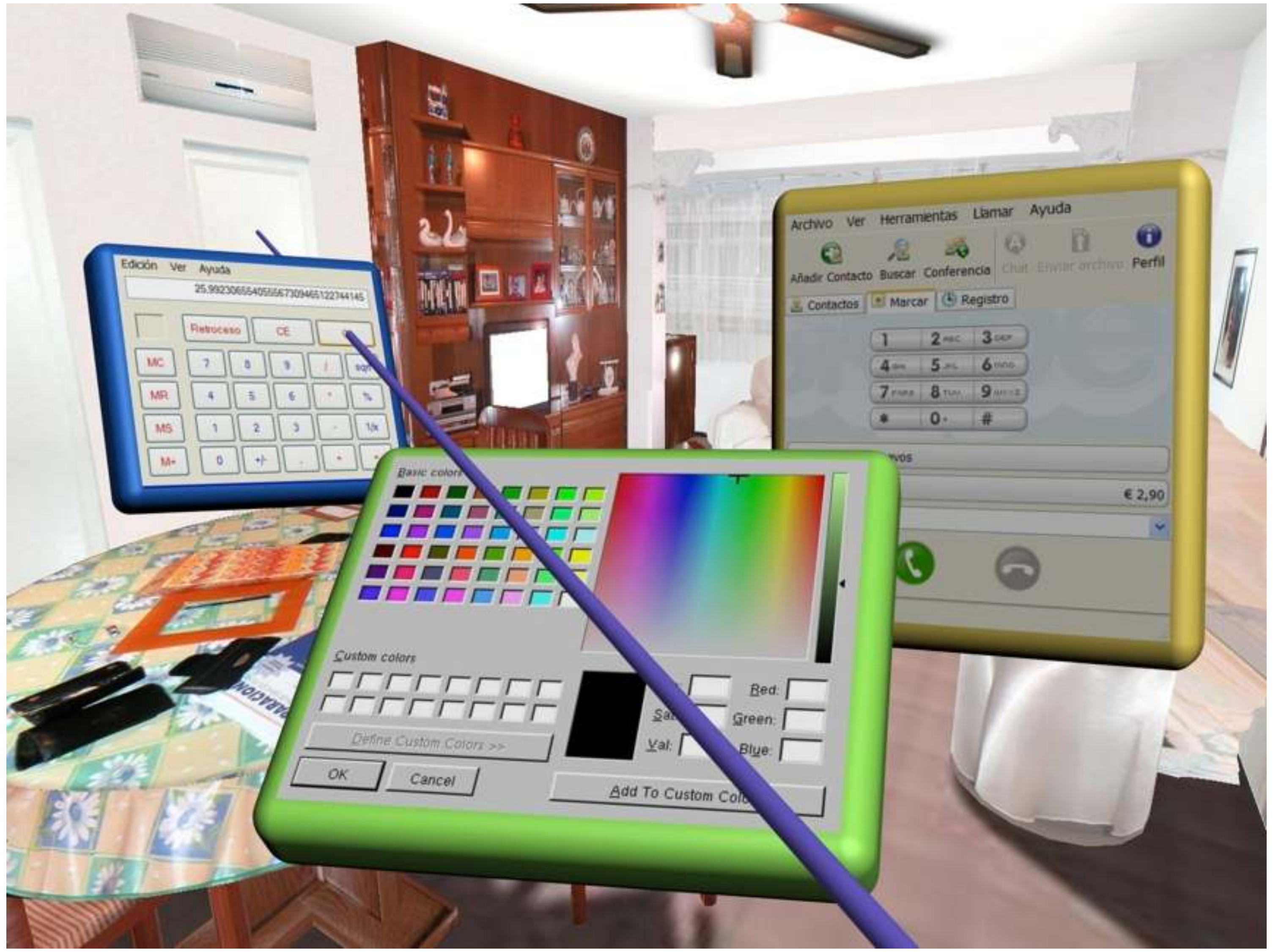
- Observation: when object pointed at is not in line of sight, people try to describe a "curve" with a pointing gestures
- Metaphor in VR: bent selection ray
- Problem: intuitiv and easy specification of a curvature using the available input devices (dataglove, trackers, ...)



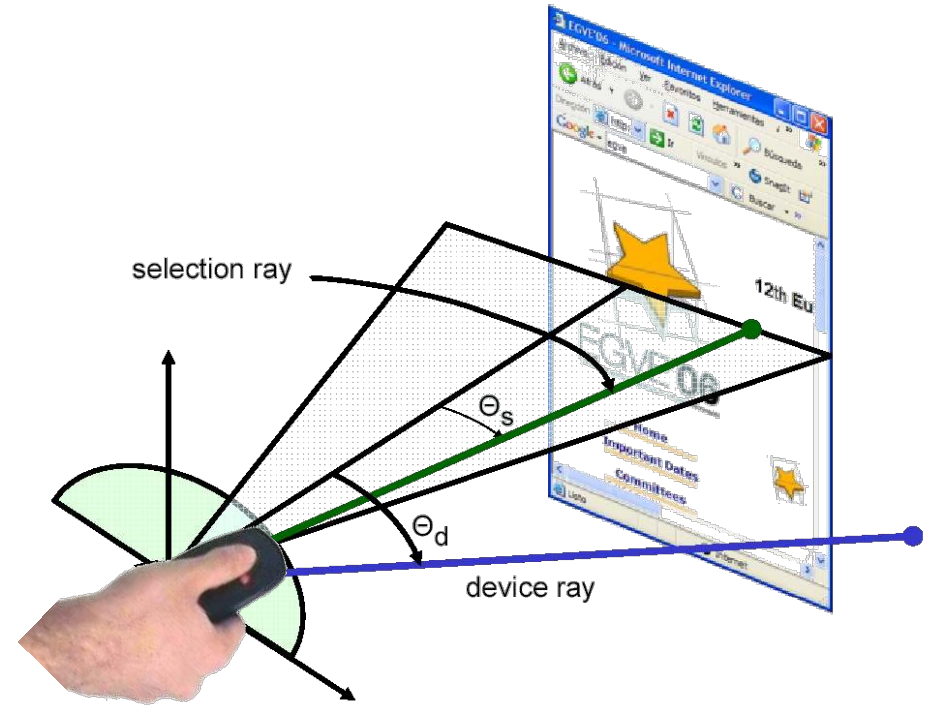
Friction Surfaces

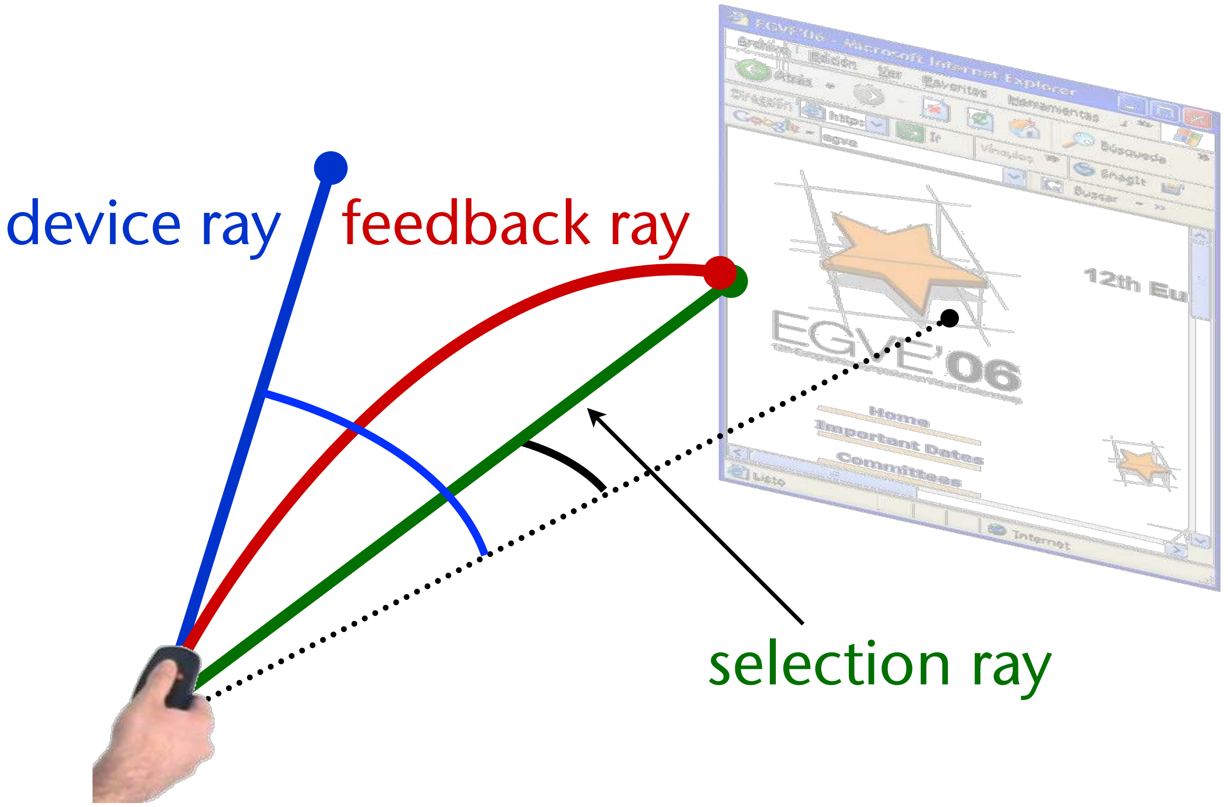
- Example task: control of so-called hybrid interfaces
  - Consider this scenario: control 2D GUIs of desktop apps directly in VR
- Problem: the target width (here: solid angle!) is extremely small





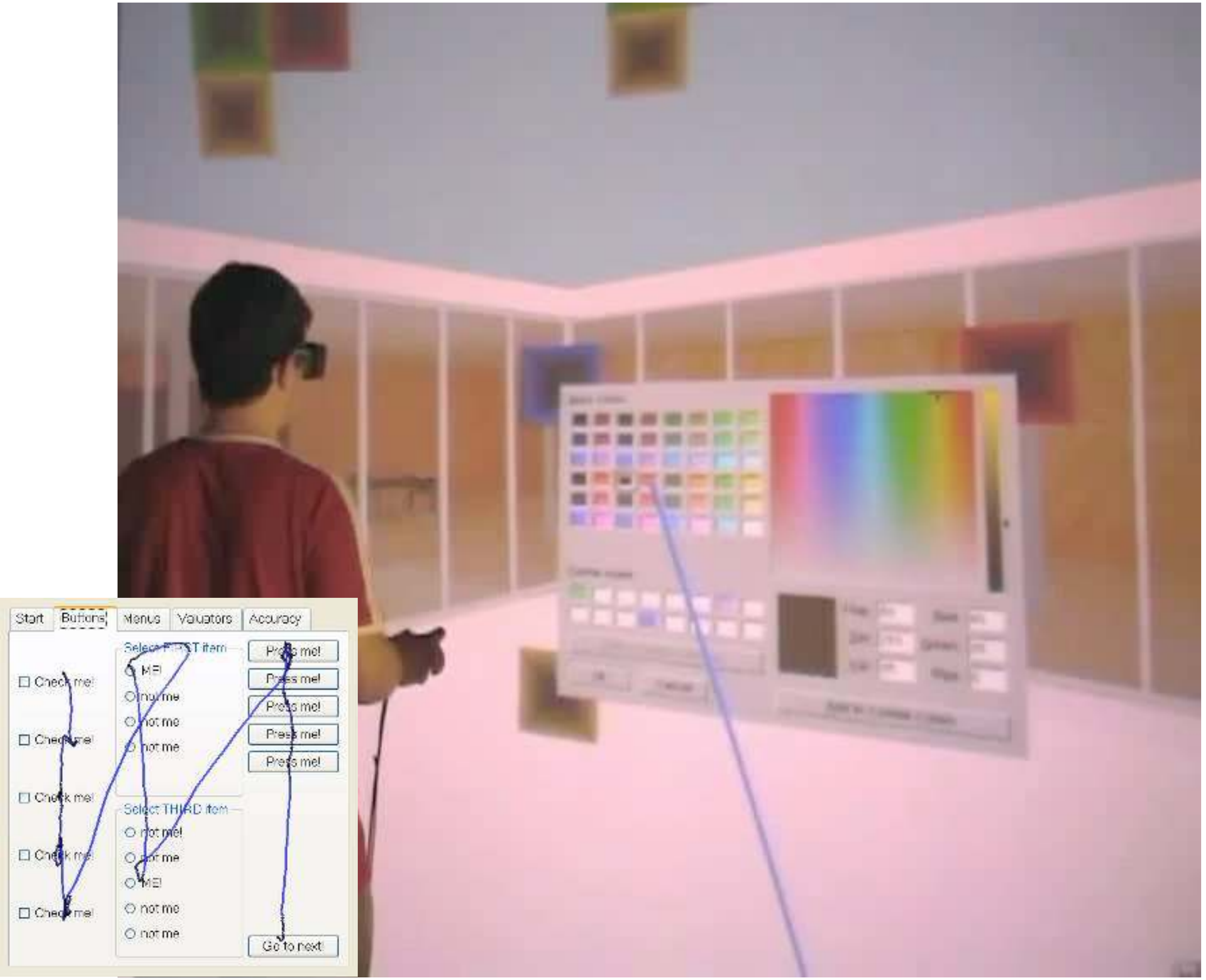
- Idea:
  - Scale the C-D ratio down, as soon as user interacts with a GUI pane (in VE)
  - Problem: how to make the non-isomorphism intuitive, how to bridge the noticeable discrepancy between motor & display space?
- Two rays were irritating to users
- Solution: show just one ray, but make it bend



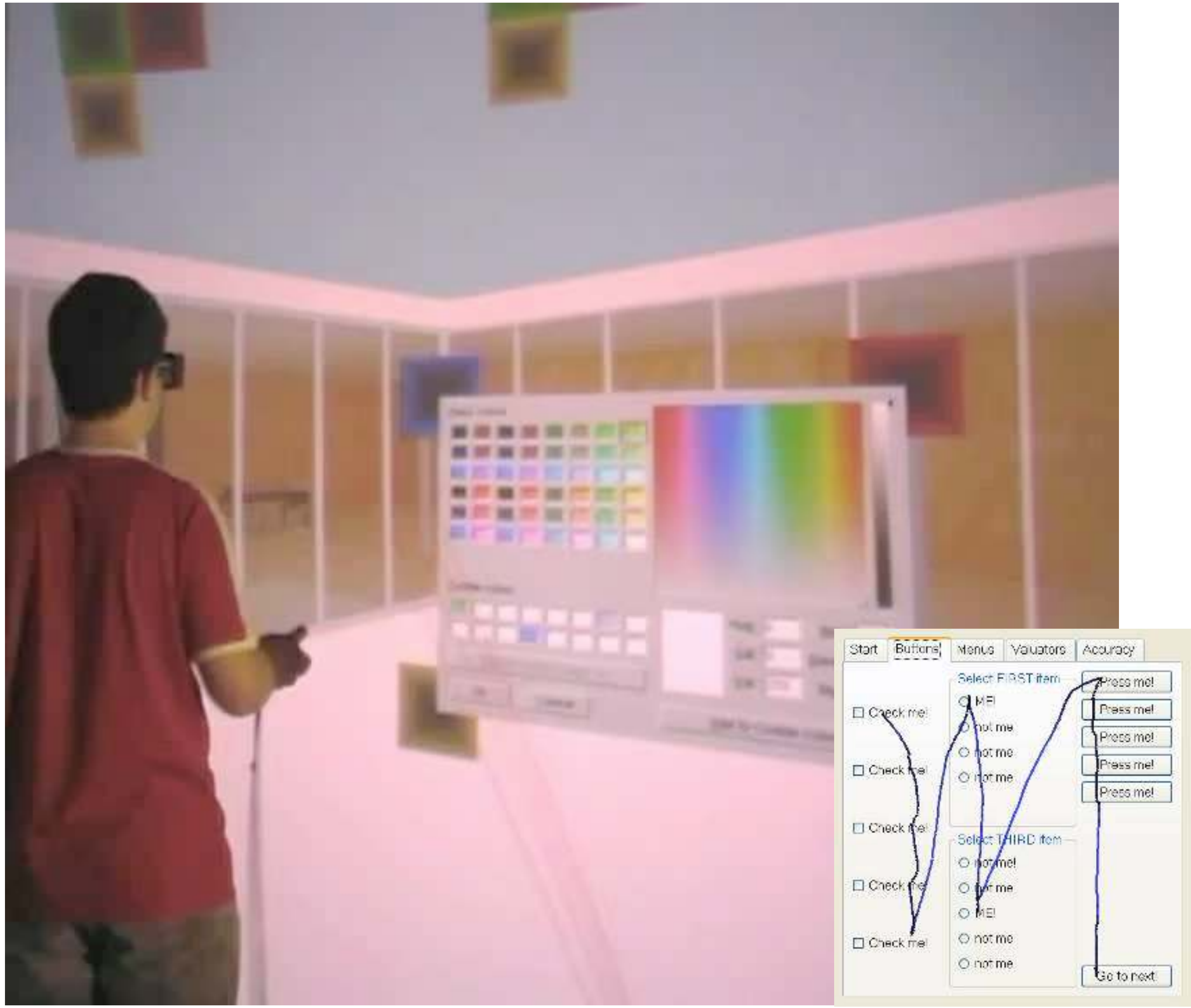


Result: much increased user efficiency

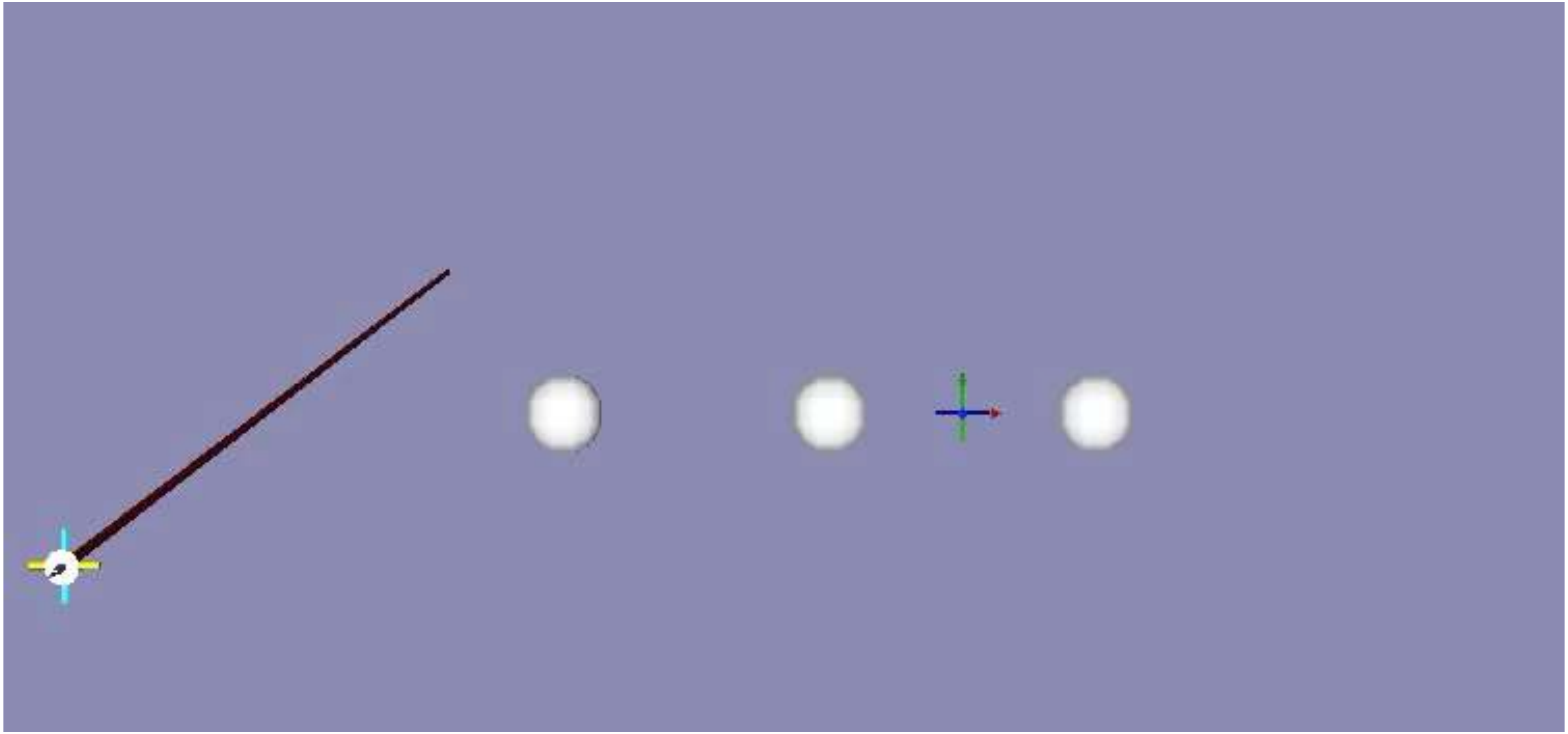
C-D ratio = 1



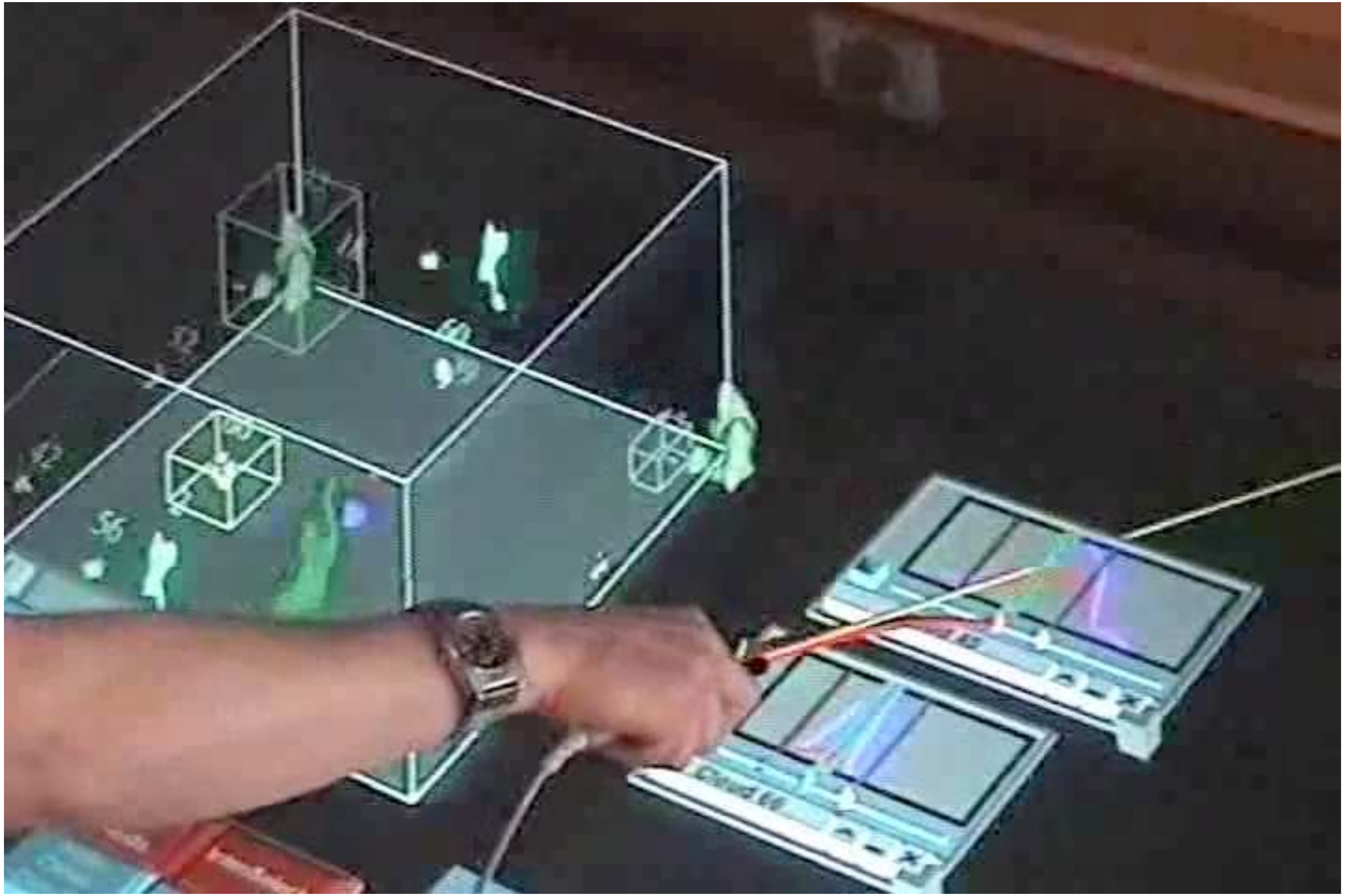
C-D ratio < 1



Video



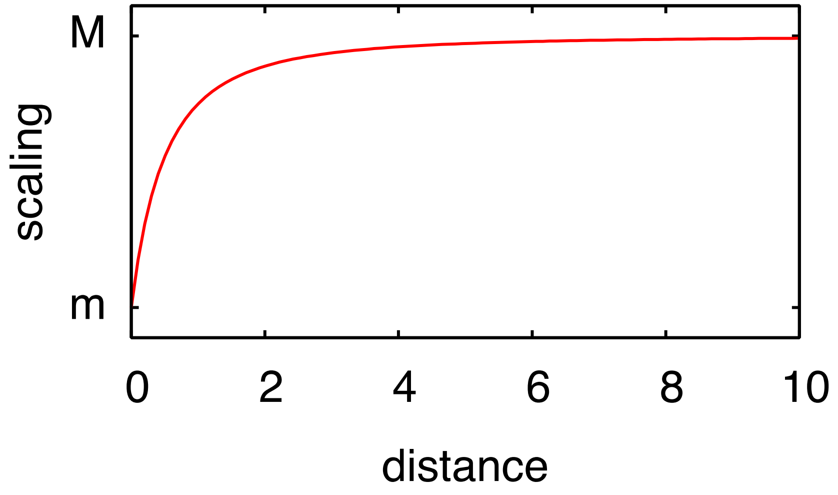
[G. de Haan, M. Koutek, and F. Post]



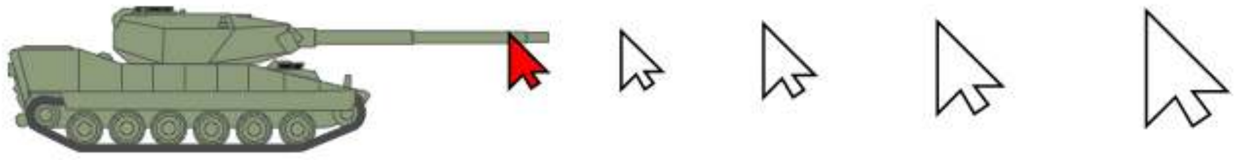
"Semantic" Pointing (Non-Constant C-D Ratio)

- Idea:
  - Modify C-D ratio depending on distance between pointer and closest target
  - Large distance *d* → scale motions from motor space up
  - Small distance *d* → scale motion downs = high precision in display space
  - E.g. with a function like this one:

$$s(d) = M + \frac{m - M}{(1+d)^{\alpha}}$$

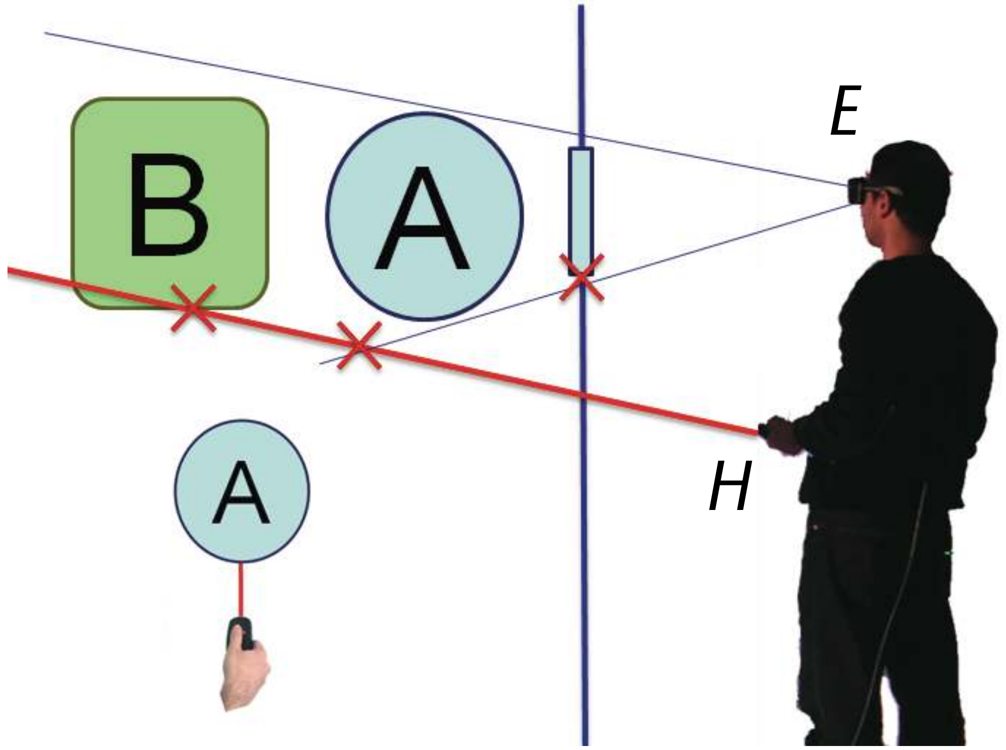


- Visual feedback:
  - Cursor size ~ C-D ratio
  - Color of pointer visualizes distance of target (e.g. "red" = "very close")

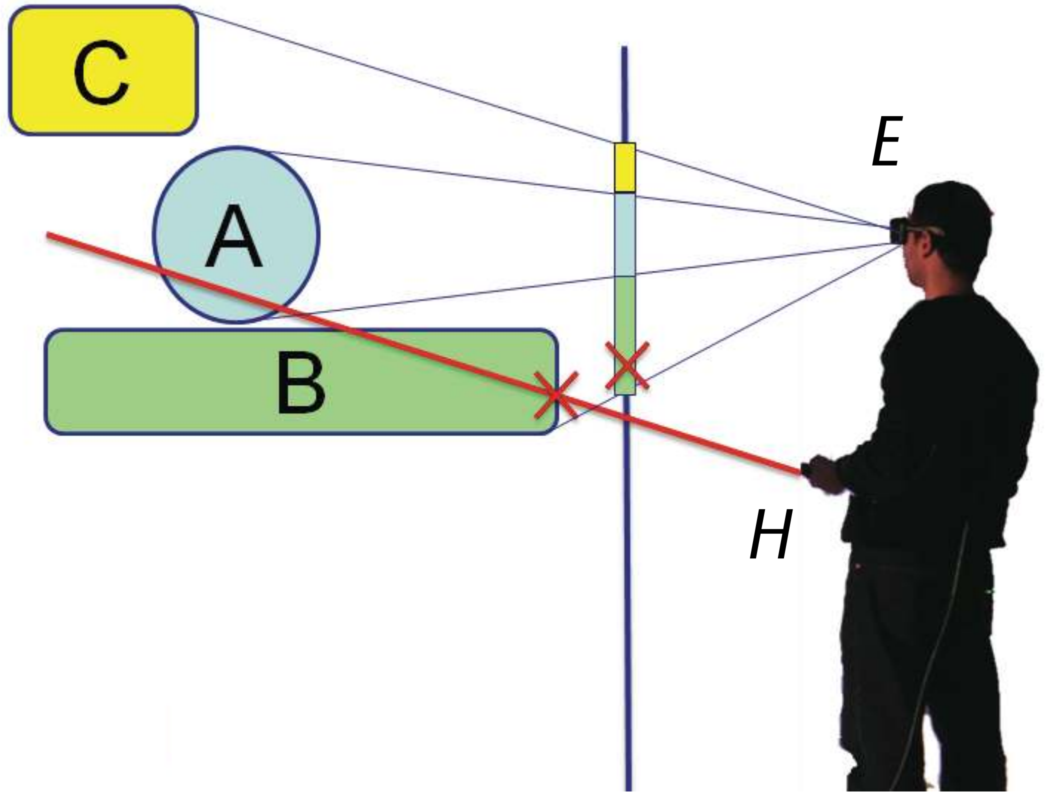


The Eye-Hand Visibility Mismatch

- Two obvious problems of all techniques where a ray emanates from the user's (virtual) hand (any kind of ray-based selection metaphor):
  1. The set of objects visible from viewpoint *E* is different from set of objects "visible" from hand position *H*



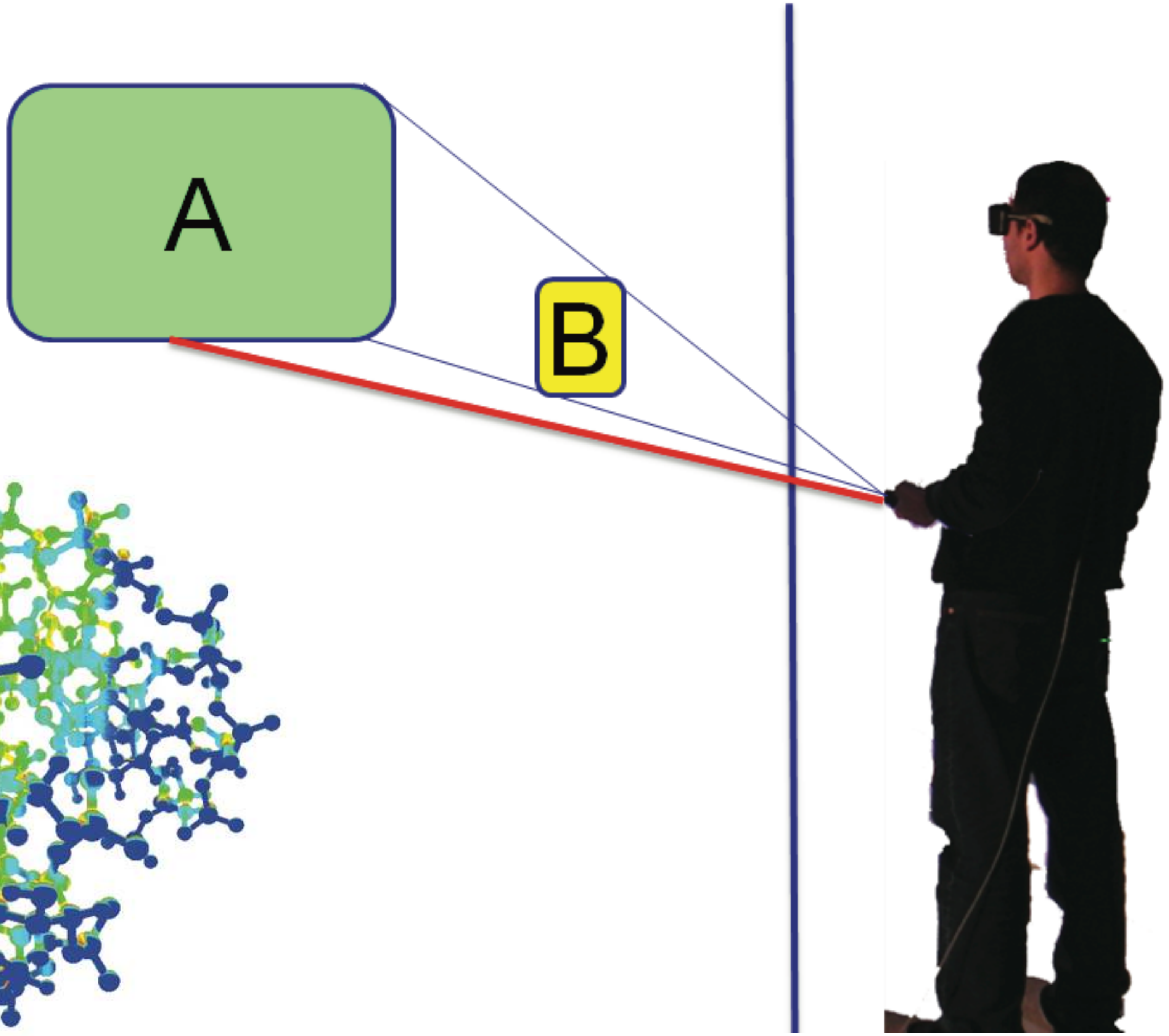
Object B is selectable, but not visible



Object C is visible, but not selectable

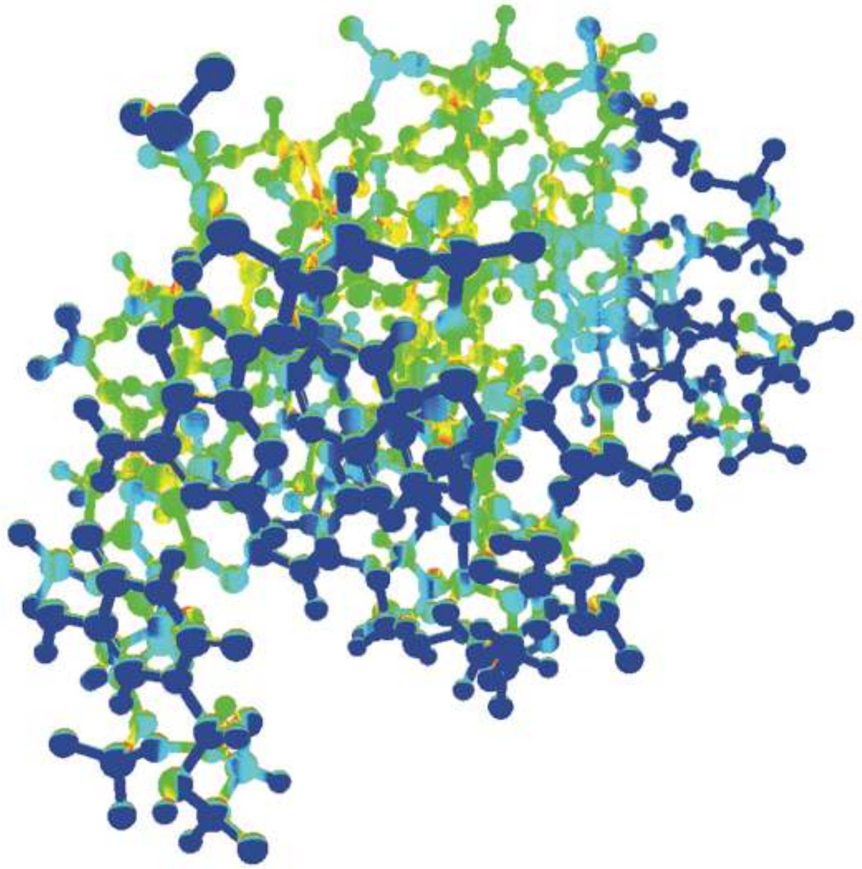
2. The surface of an object "visible" from *H* is different than the surface visible from E; consequences:

- Real target width is different than visible target width
- Perhaps no/insufficient feedback during selection process



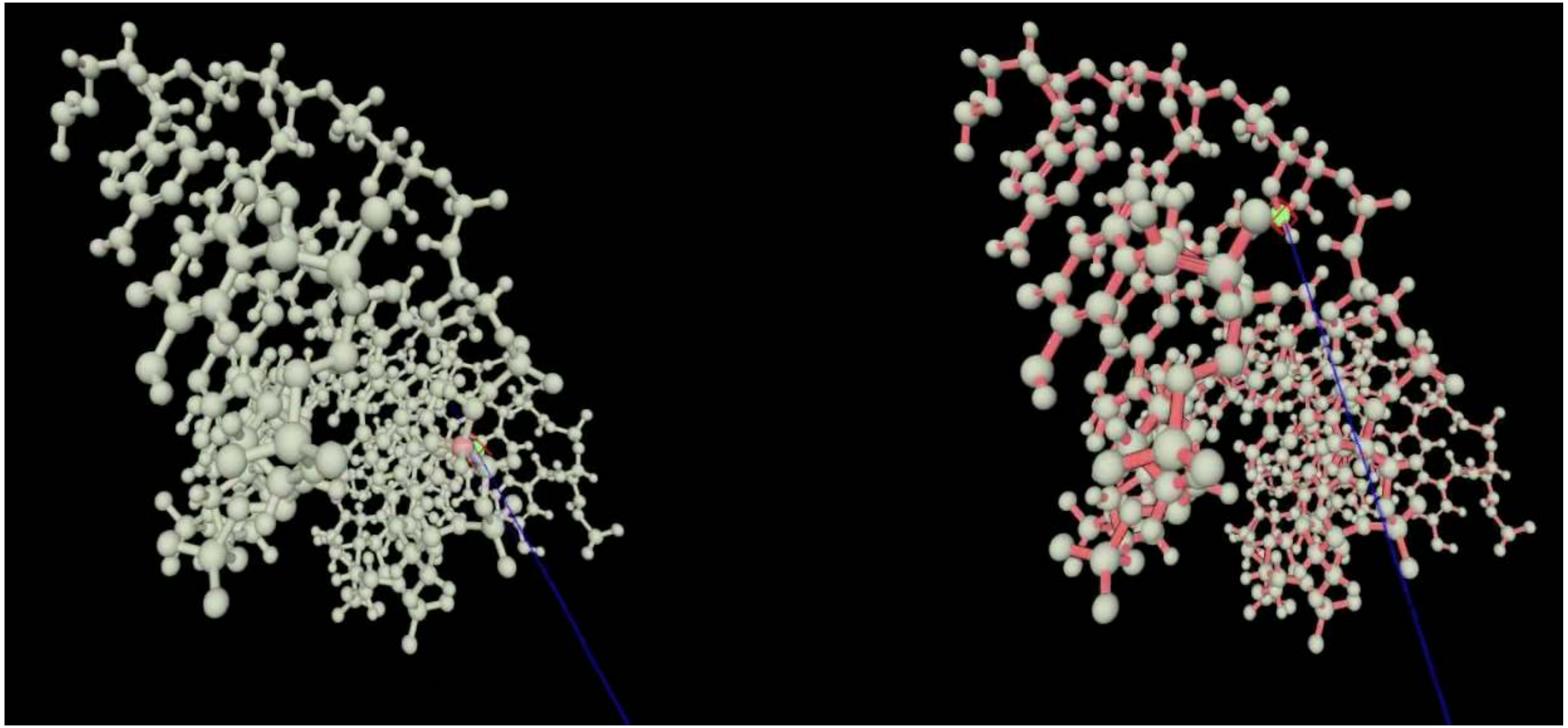
Blue = visible from many different angles

Yellow/red = visible only from small range of angles

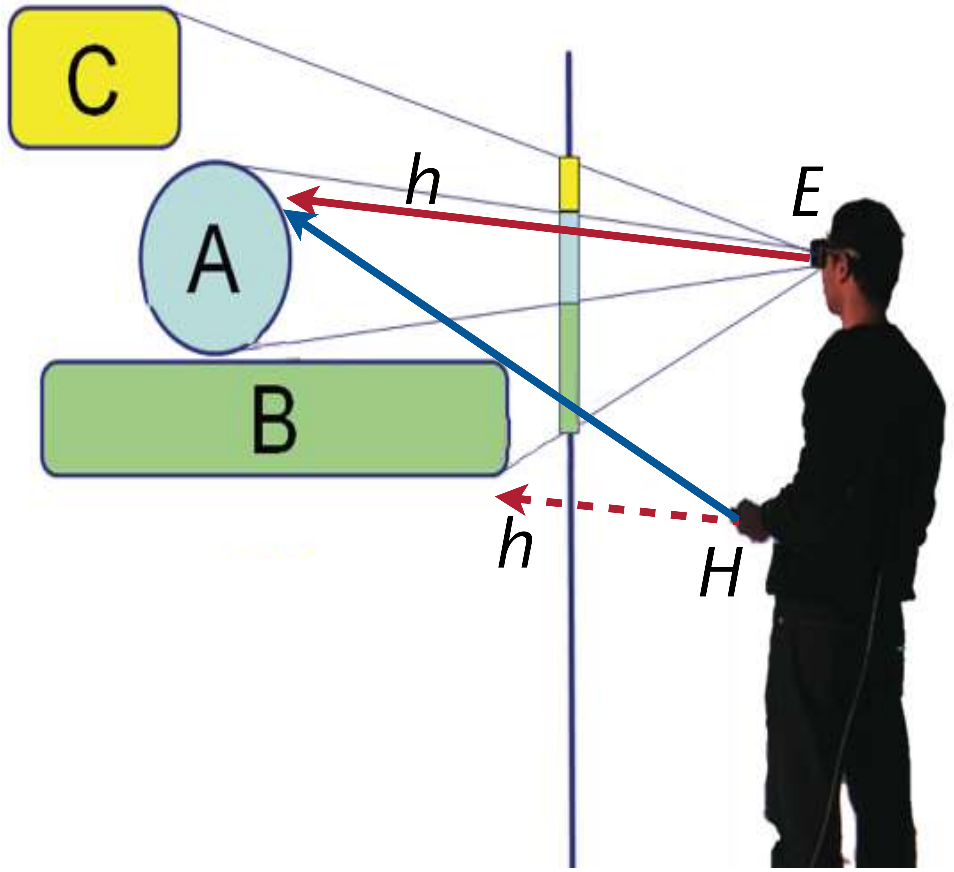


Idea

- Shoot *selection ray* emanating from viewpoint *E* into hand direction *h*
- Visual *feedback ray* emanating from *H* to first intersection of selection ray
- Experiment shows: users are ca. 15-20% faster than with normal ray

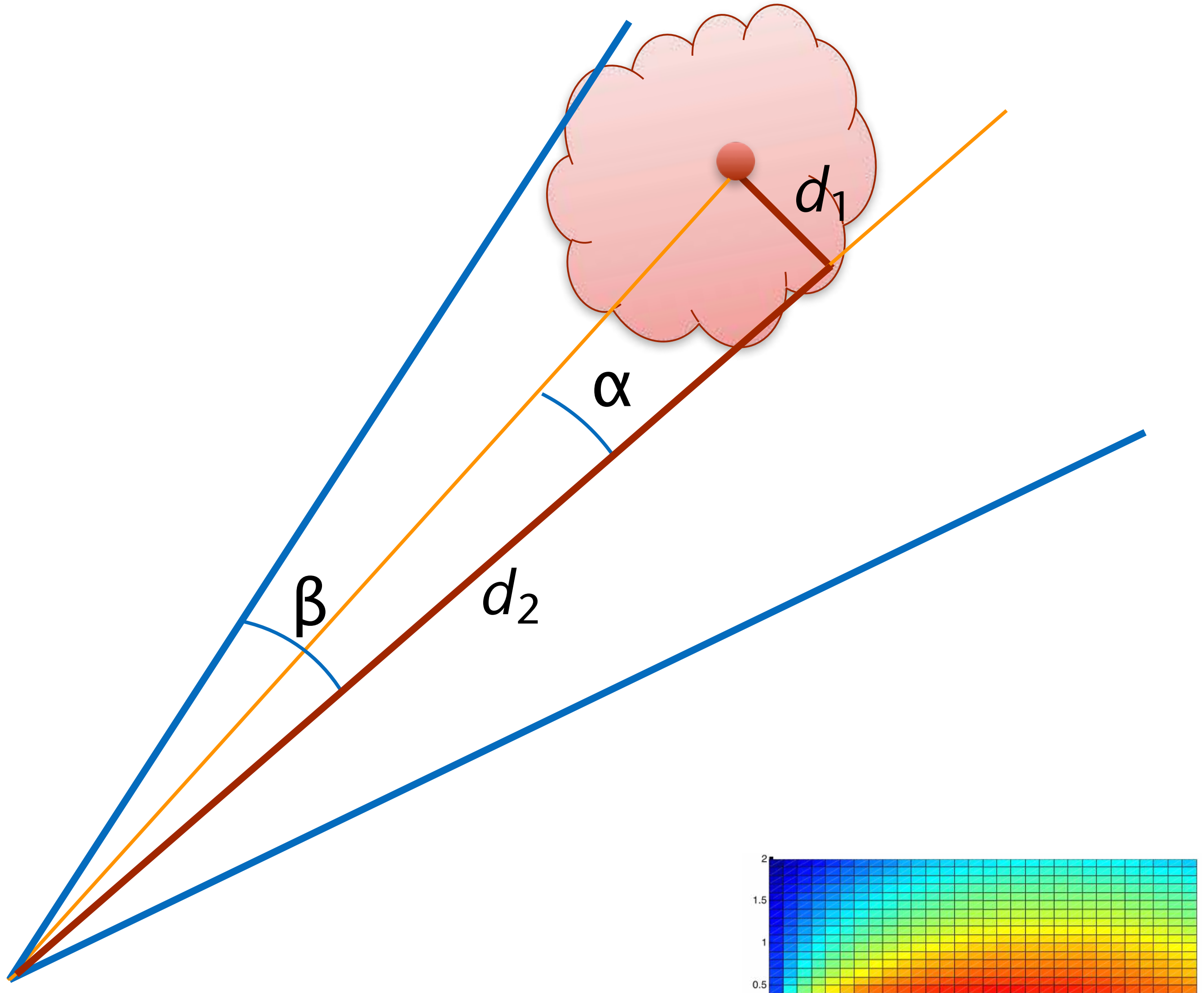


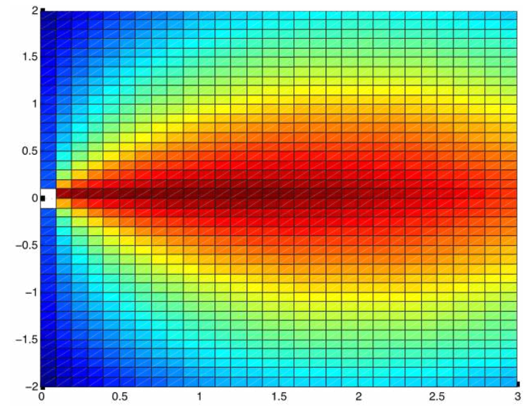
Argelaguet, Andujar, Trueba



Ranking + Filtering for Cone-Based Techniques (IntenSelect)

- Assumptions:
  - Cone is better than ray (smaller I.D.)
  - In general, a lot of objects are in the cone (i.e., dense environment)
- Idea for disambiguation:
  - Compute a "score" for each position inside cone → scalar field
  - Create ranking of objects
- Simple scoring function: $$s = 1 - \frac{\alpha}{\beta}$$
- A scoring function that prefers near objects: $$s = 1 - \frac{1}{\beta} \tan^{-1}\left(\frac{d_1}{(d_2)^k} \right) , \quad k \in [\tfrac{1}{2}, 1]$$





- Problem: jitter in user's hand leads to frequent changes in ranking
- Solution: filtering, i.e., transform time series $s = s(t)$ into filtered series $\hat{s}$

$$\hat{s}(t) = \sigma \hat{s}(t-1) + \tau s(t)$$

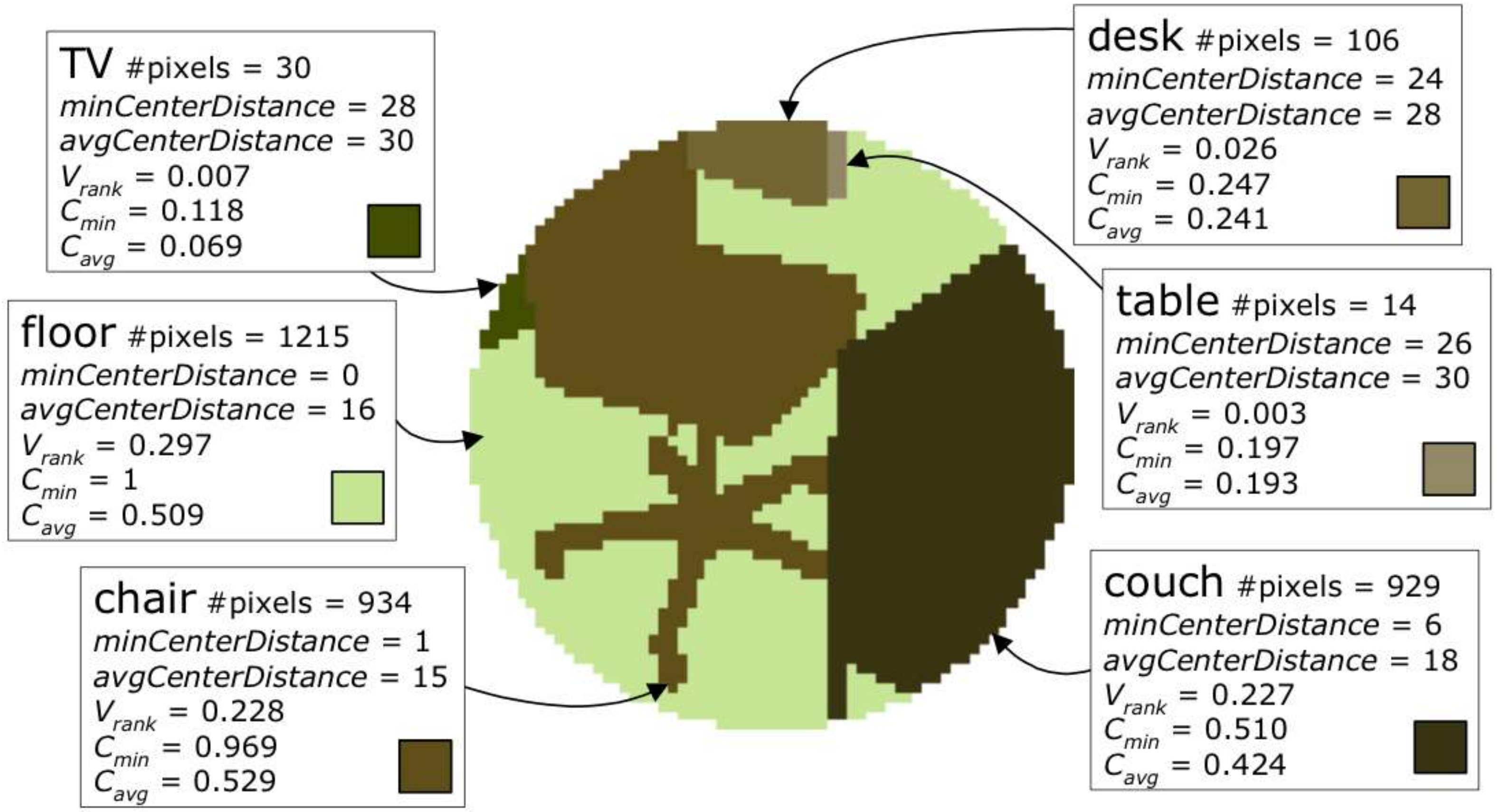
$s(t)$ = score over time, σ = "stickiness", τ = "snappiness"

- Generalization: FIR filter (see signal processing literature)
- Feedback to user:
  - Bend ray towards object with highest ranking
  - Show straight ray for cone's axis

Other Ranking Functions FYI

- Other distance functions, e.g., prefer far-away objects
- Better computation of "distance" from cone's axis:
  - Render object with low resolution into off-screen frame buffer with "viewpoint" = apex of cone, viewing direction = cone's axis
  - Compute average distance of all pixels of object from center (= cone's axis) :

$$s' = 1 - \frac{\frac{1}{n}\sum_{\text{pixel p}} d(p)}{\text{radius}}$$



TV #pixels = 30
minCenterDistance = 28
avgCenterDistance = 30
V_{rank} = 0.007
C_{min} = 0.118
C_{avg} = 0.069 ■

floor #pixels = 1215
minCenterDistance = 0
avgCenterDistance = 16
V_{rank} = 0.297
C_{min} = 1
C_{avg} = 0.509 ■

chair #pixels = 934
minCenterDistance = 1
avgCenterDistance = 15
V_{rank} = 0.228
C_{min} = 0.969
C_{avg} = 0.529 ■

desk #pixels = 106
minCenterDistance = 24
avgCenterDistance = 28
V_{rank} = 0.026
C_{min} = 0.247
C_{avg} = 0.241 ■

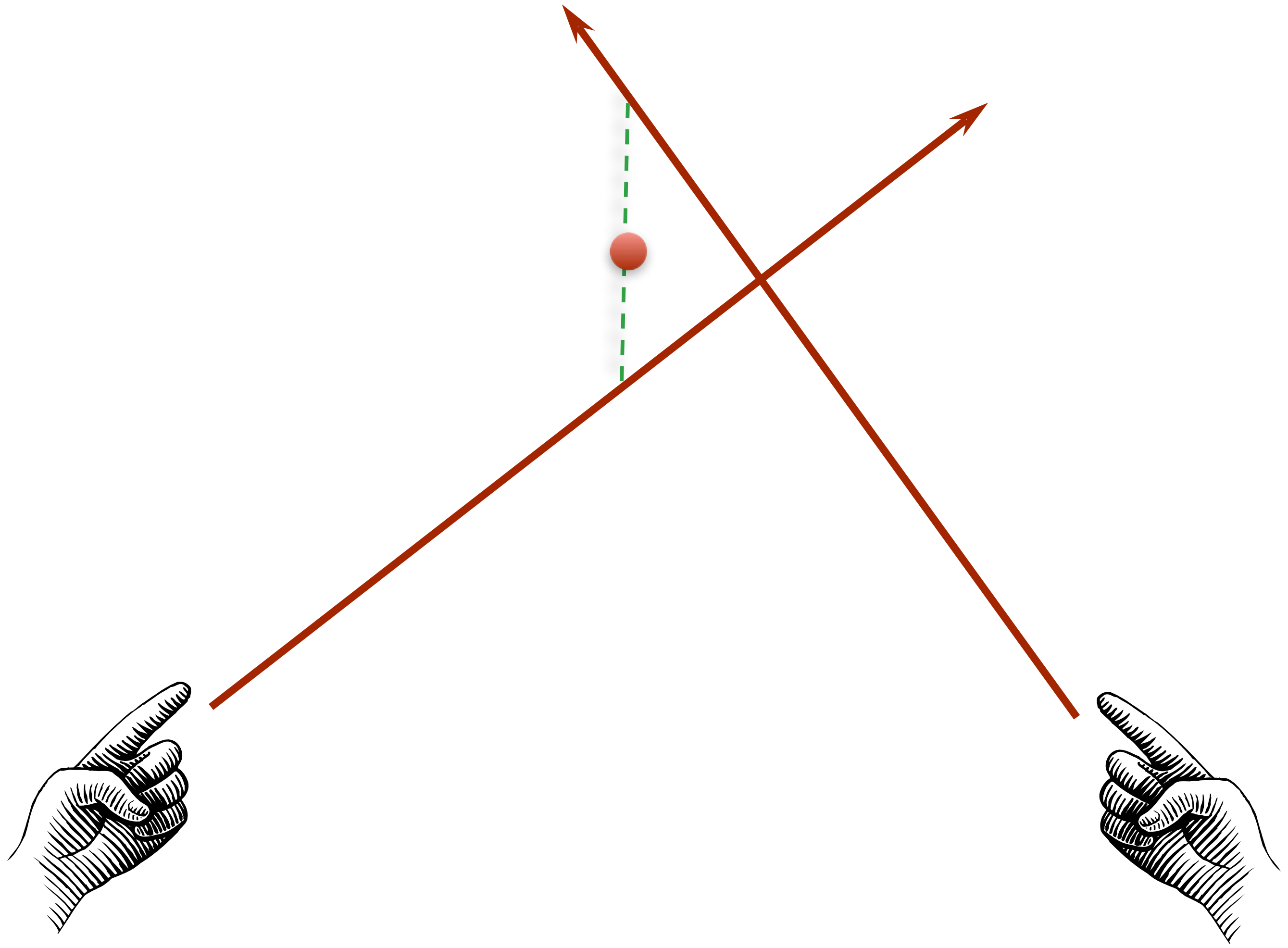
table #pixels = 14
minCenterDistance = 26
avgCenterDistance = 30
V_{rank} = 0.003
C_{min} = 0.197
C_{avg} = 0.193 ■

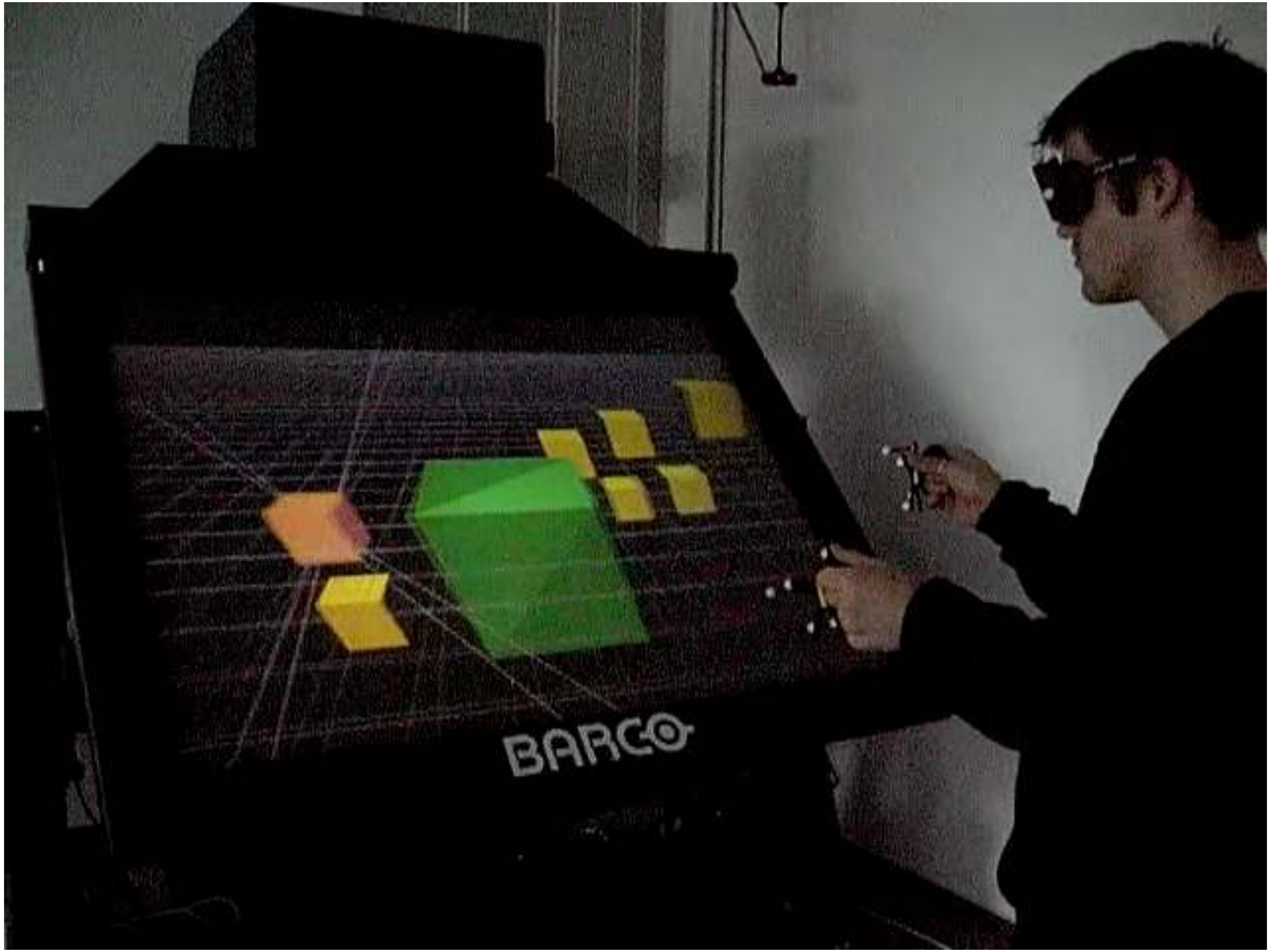
couch #pixels = 929
minCenterDistance = 6
avgCenterDistance = 18
V_{rank} = 0.227
C_{min} = 0.510
C_{avg} = 0.424 ■

Multimodal Selection (Disambiguation Via Speech Commands)

iSith: Two-Handed Selection

- Idea: intersection of two rays defines "selection center"
- Practical implementation:
  - One ray per hand
  - Trigger selection mode as soon as distance between rays < threshold
  - Midpoint of the shortest line between the rays = "selection center"
  - Select all objects "close enough" to this selection center

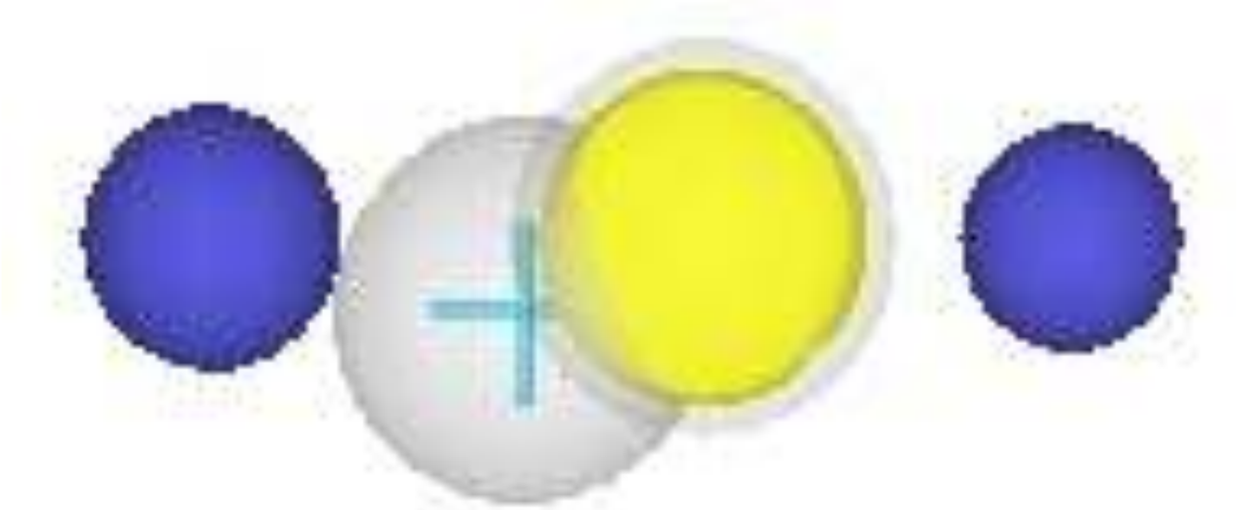


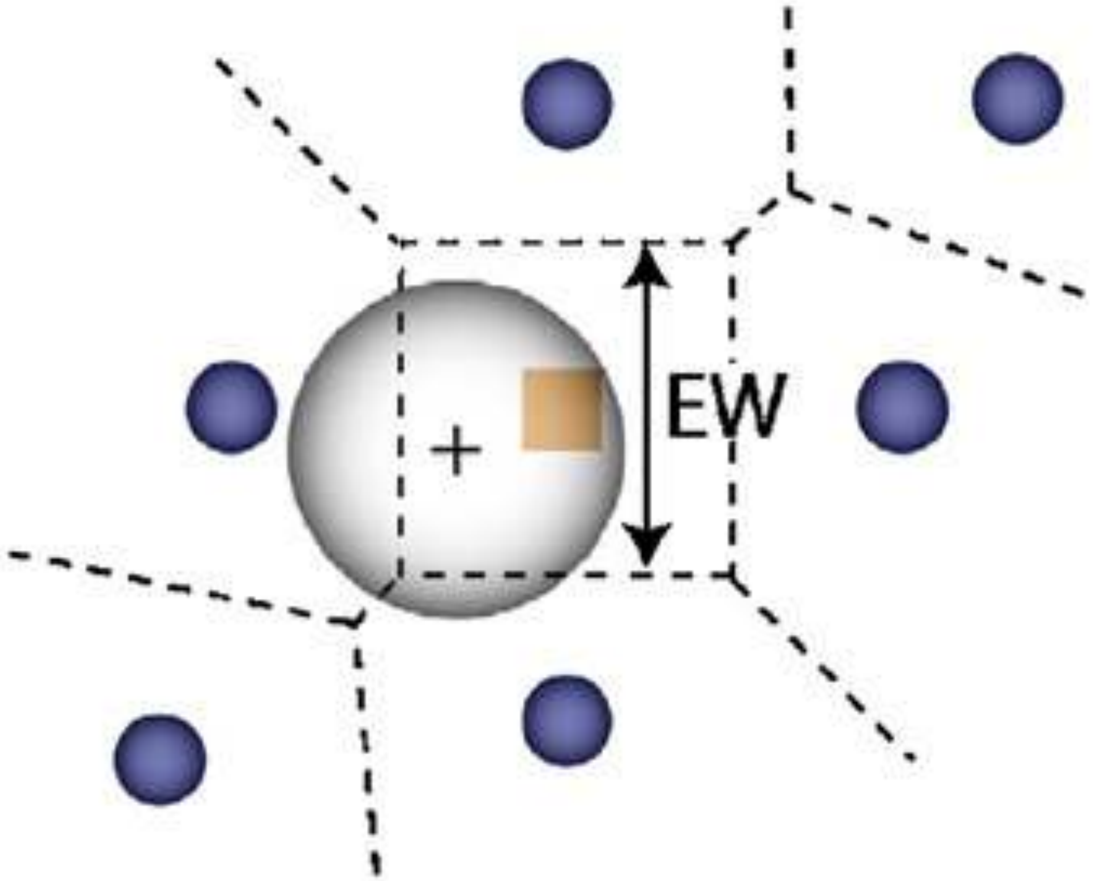


BARCO

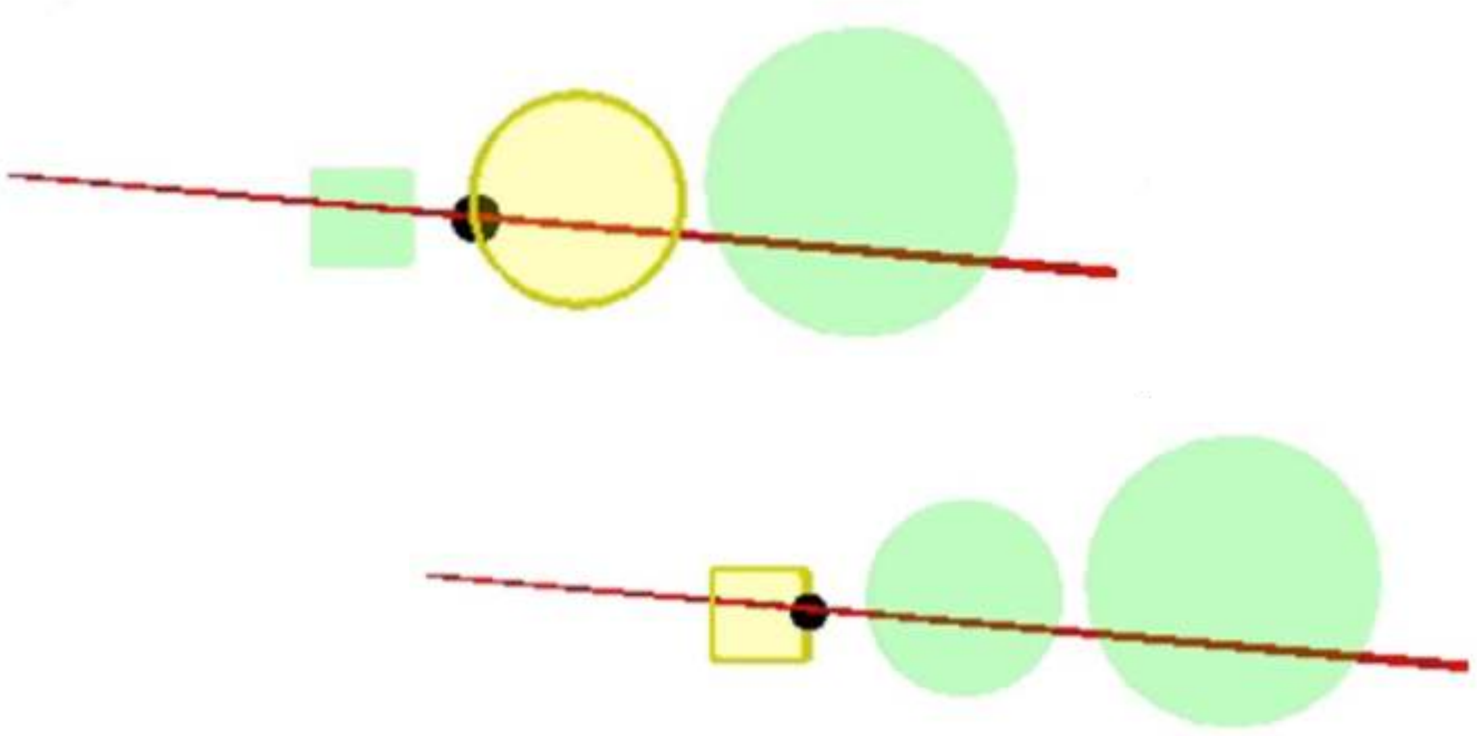
The Bubble Cursor and the Depth Ray

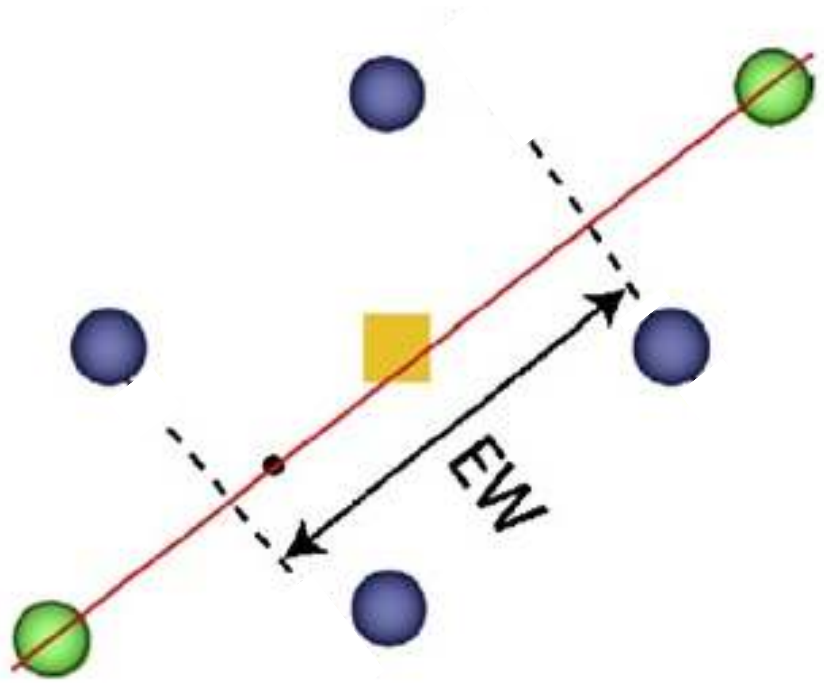
- Another method to increase the effective target width
- Bubble Cursor:
  - 3D cross hairs
  - Select always the object closest to the crosshair (in 3D)
  - Show transparent sphere around 3D crosshairs with radius =
    distance to closest object (feedback for "density")
  - Also, transparent sphere around active obj for feedback
- Effective target size = Voronoi region of object
  - (For Voronoi regions, see course "Computational Geoemtry")



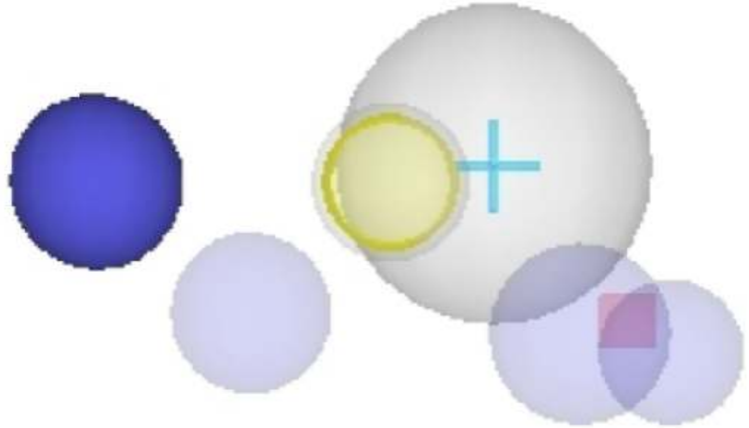


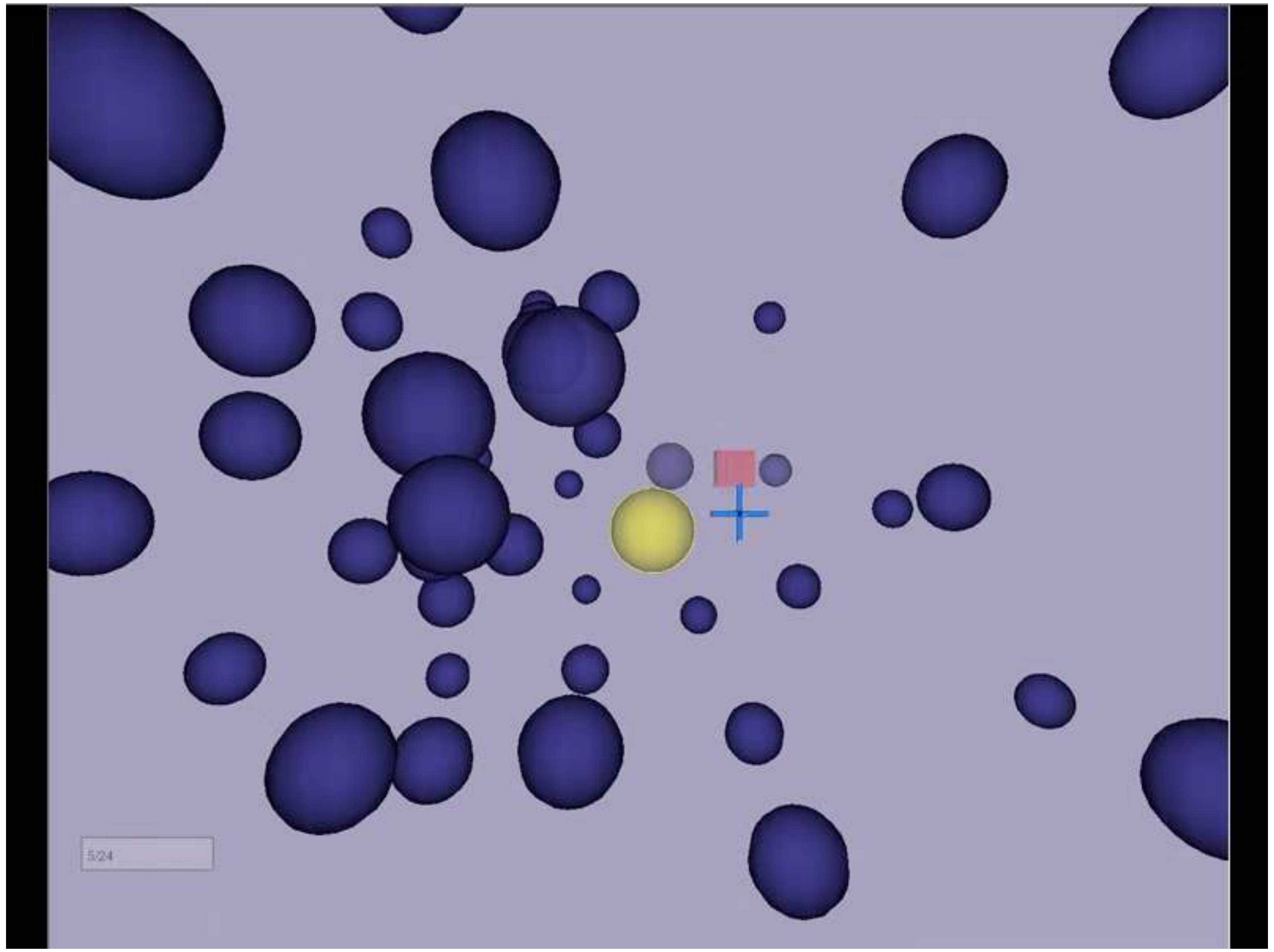
- Depth Ray:
  - Only consider objects being "stabbed" by the ray
  - User moves a "depth marker" along the ray
  - Of all "stabbed" objects, take the one closest to the depth marker
  - Effectively, the ray is partitioned into segments with the objects in the middle
- Effective target size = segment on the ray
- Depth ray = Bubble cursor with DOF decomposition



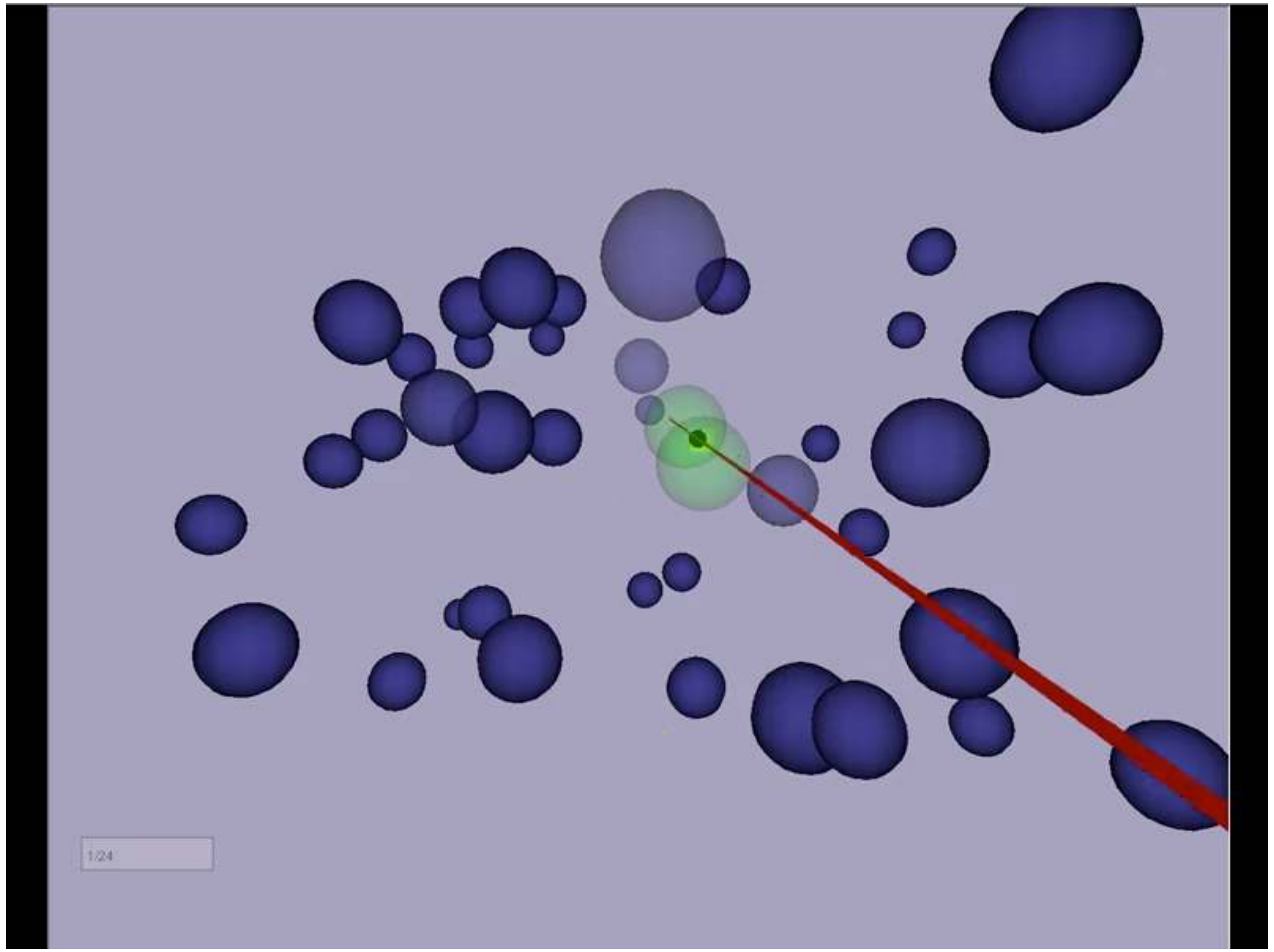


- Handling occlusion: make occluders in front of the depth marker transparent (possibly depending on the distance from the depth marker)





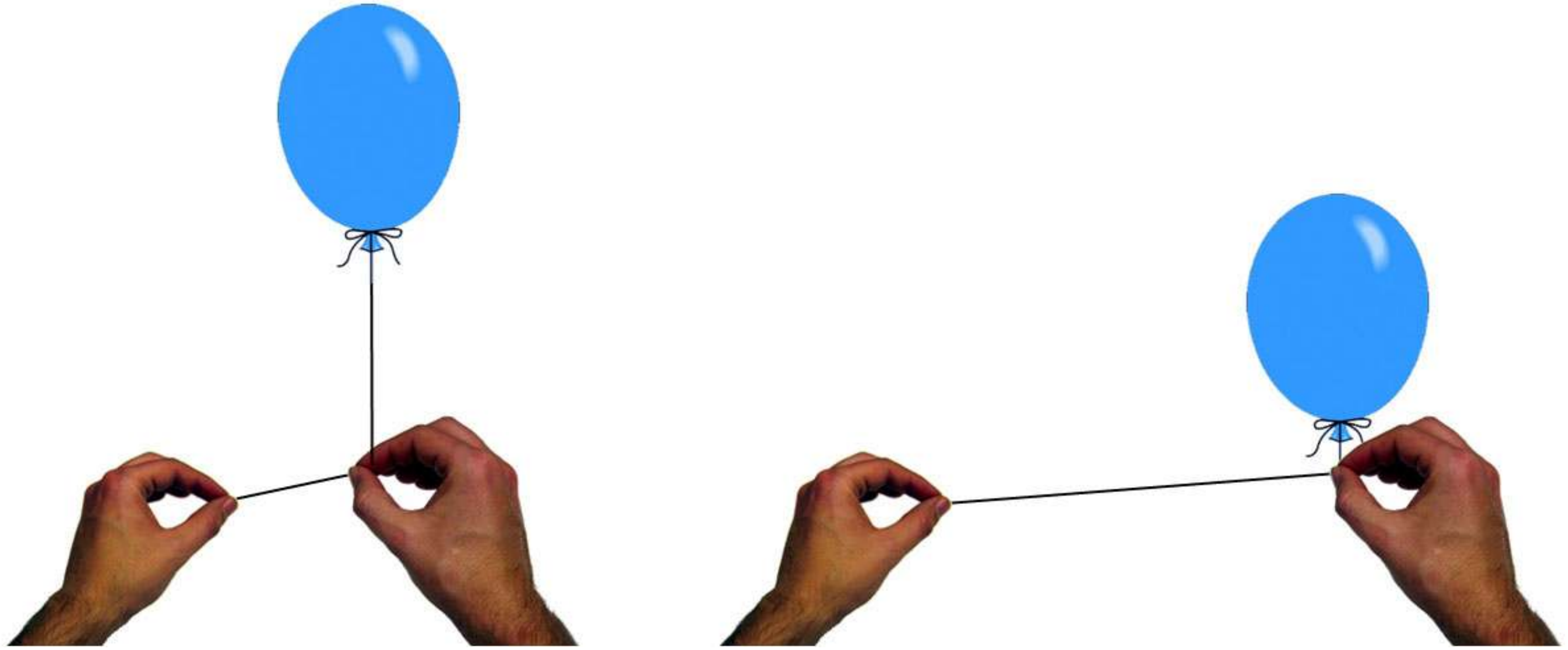
Bubble cursor [Lode van Acken]

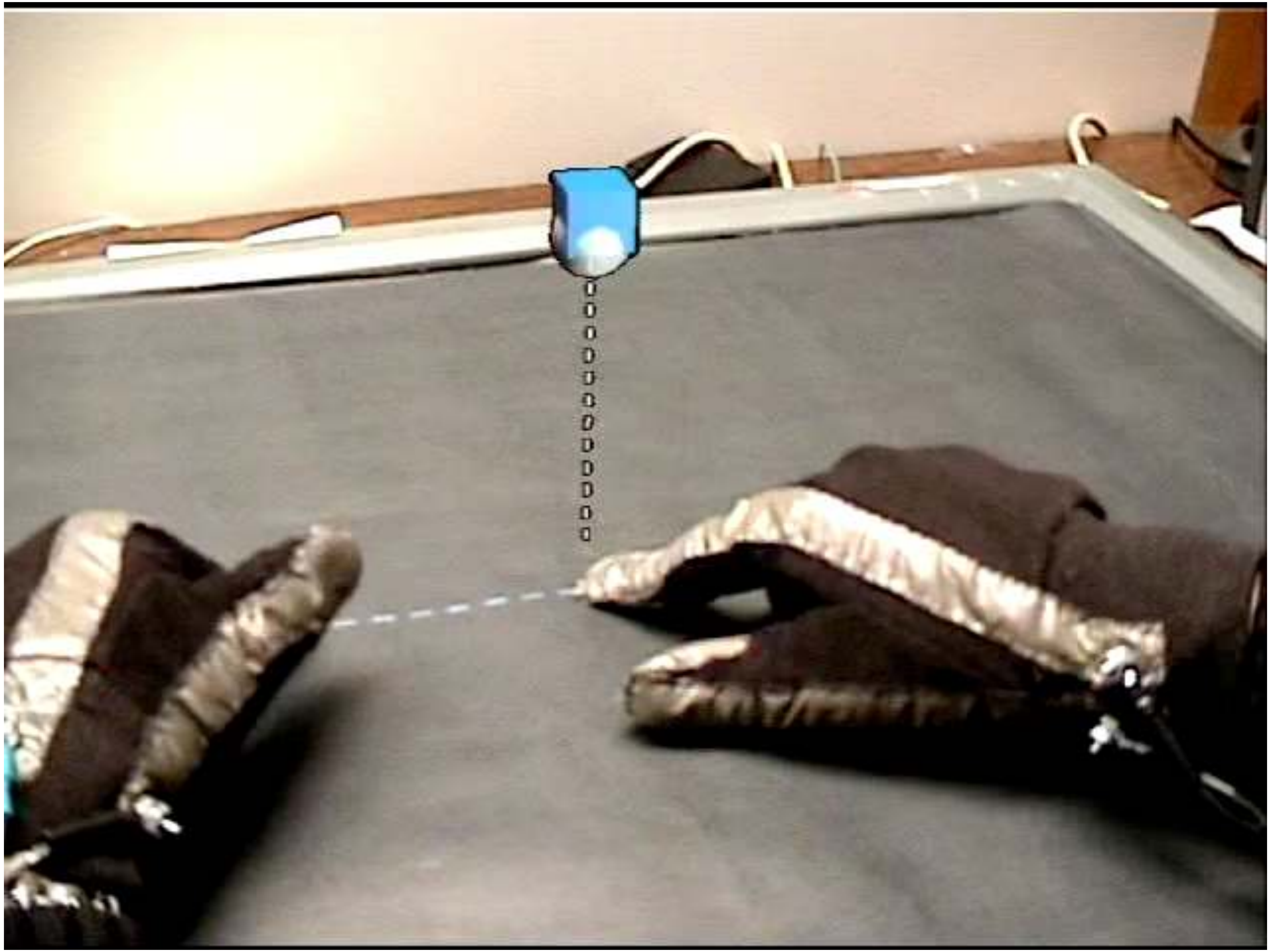


Depth Ray

Balloon Selection

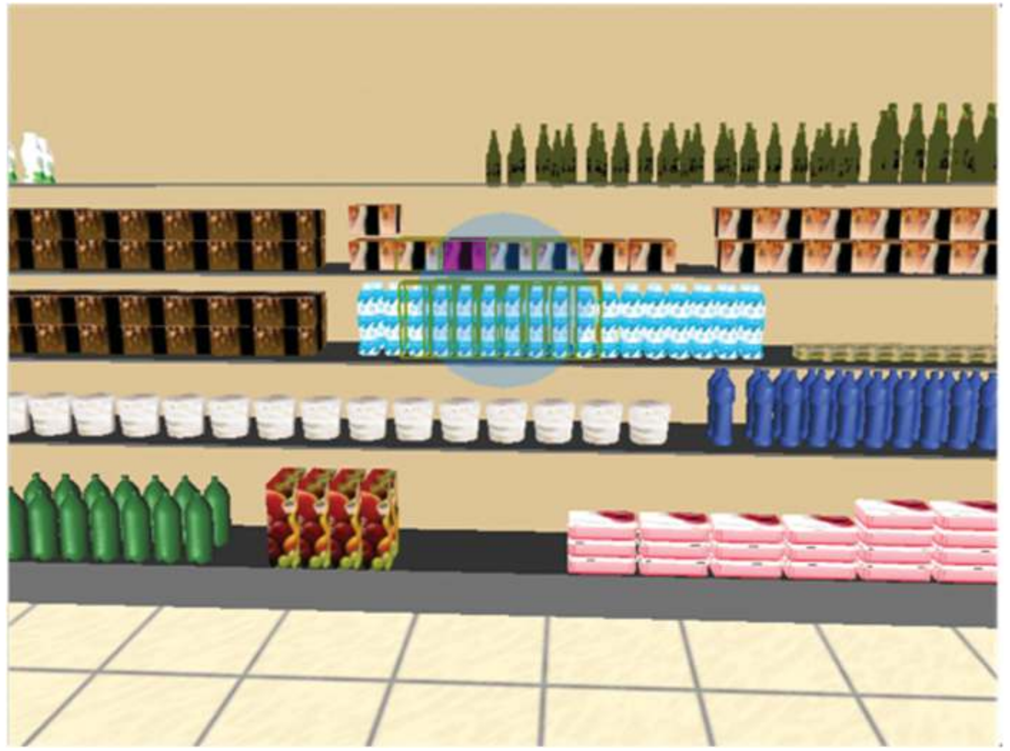
- Metaphor: a helium balloon
  - Dominant hand controls position (2D)
  - Non-dominant hand controls height (1D)
- Meant for use on work bench
- Implementation:
  - Right/left index finger defines position / height, resp.
  - Both index fingers remain on the table
  - *System control* (e.g., triggers) by contacts in data glove
- Advantage:
  - Decomposition of a 3D tasks in two separate low-dimensional tasks
  - Natural constraint (table, a.k.a. *passive haptics*)

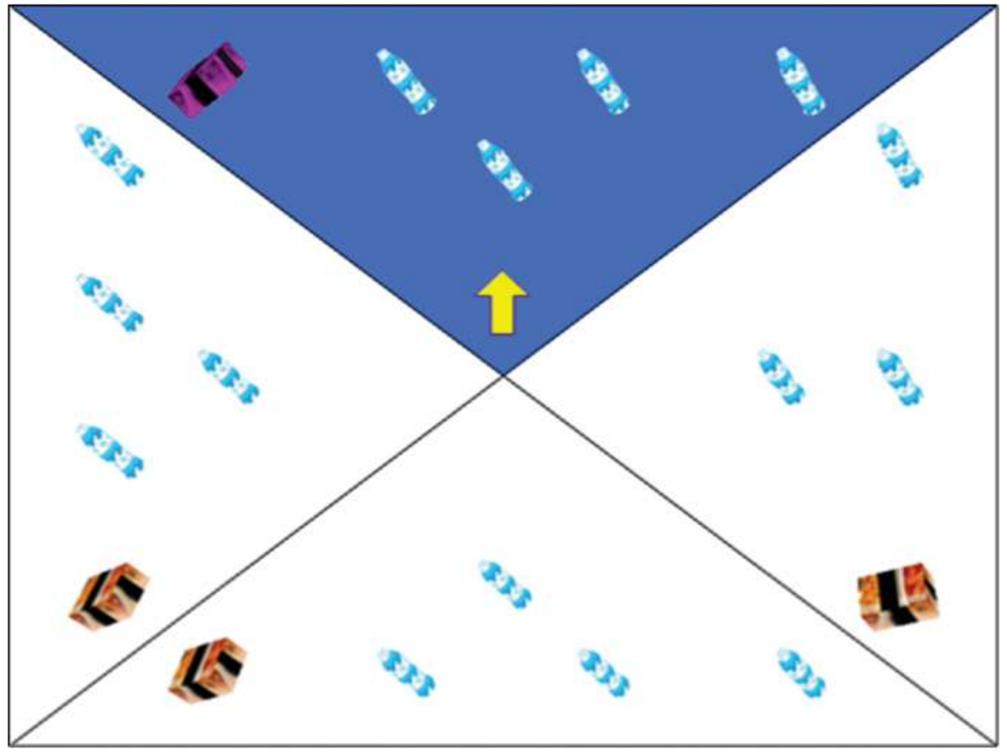




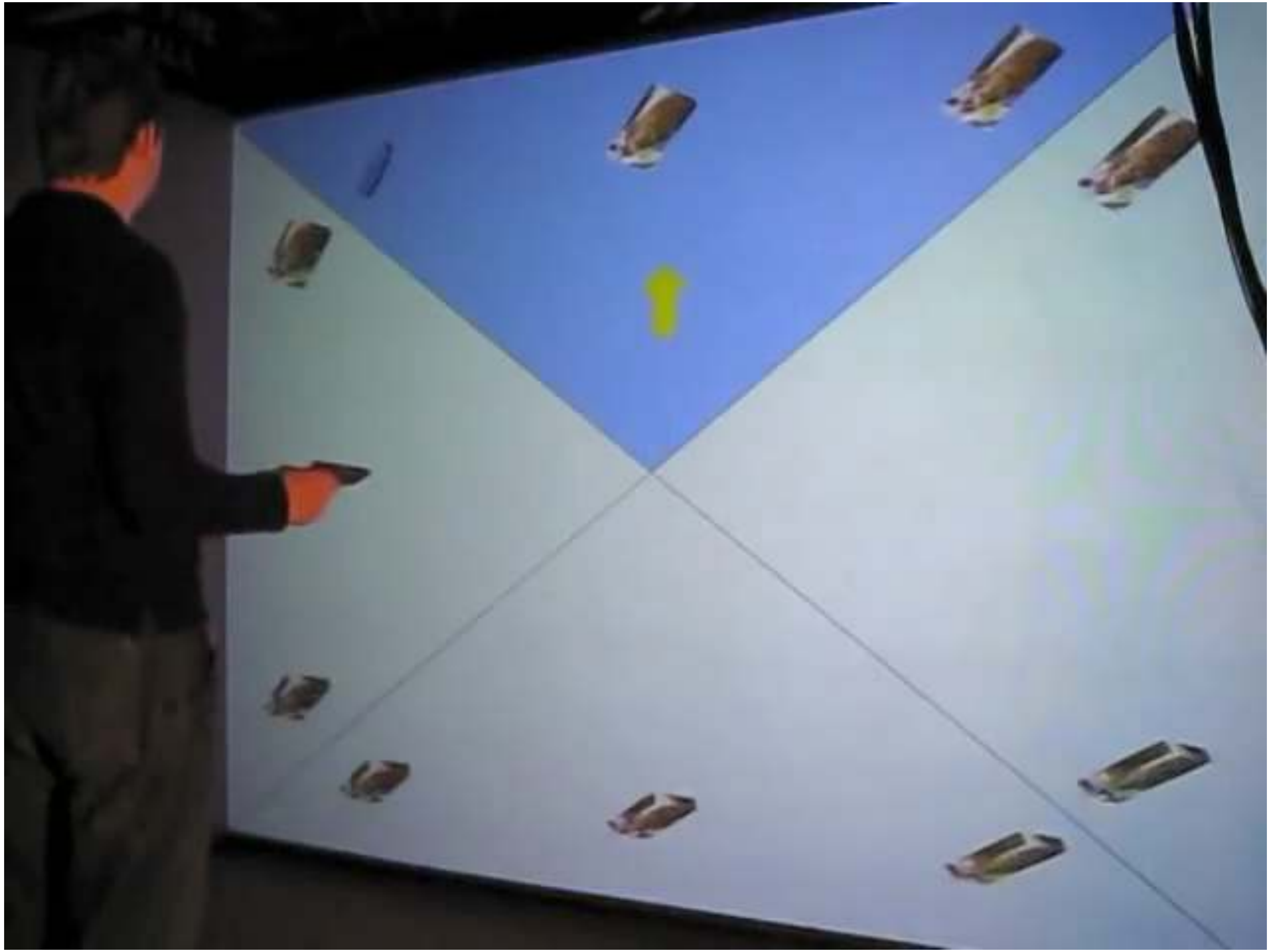
Selection by Progressive Refinement

- Problem: *cluttered scenes*!
- Example: SQUAD
- Works in two phases
  1. Sphere selection
     - Place sphere around first hit point of ray
     - Candidates for second phase = all objects inside sphere
  2. Place candidates on 2D overlay in 4 quadrants
     - User selects one quadrant = ¼ of all candidate objects
     - Repeat ⟶ $N = \lceil \log_4 n \rceil$ steps
- Example: sphere contains 64 targets ⟶ # clicks = 1+3





Movie



Predicted Task Completion Time

- $\mathrm{T}_{\mathrm{SQUAD}} = \mathrm{T}_{\text{sphere sel.}} + \mathrm{T}_{\text{refinement}}$
- $T_{\text{sphere sel.}} = b \cdot \mathrm{ID} + a$
  - Choose sphere large enough so that index of difficulty ≈ 0
- $T_{\text{refinement}} = \sum_1^N (v_i + c)$,

  where v_i = time to complete search for target, c = time to select quadrant
- Assumptions: $c \approx a$, $v = v(n) = \sum_1^N v_i$ = total search time
- In total: $T_{\text{SQUAD}} \approx a(1 + \log_4 n) + v(n)$

Empirical Evaluation and Discussion

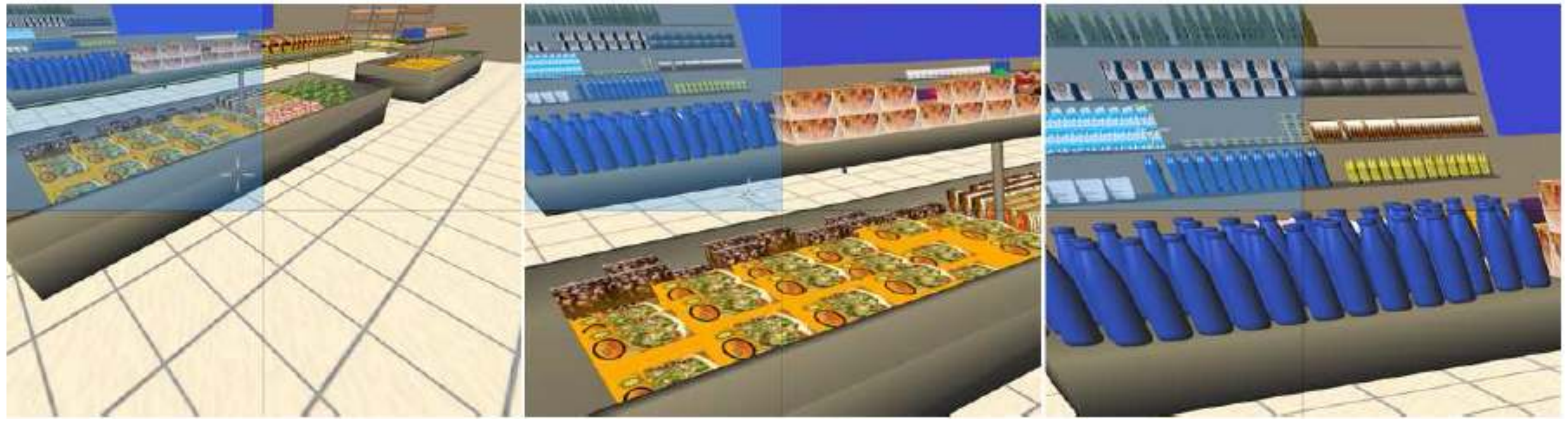
- Empirical findings match predicted task completion time
- SQUAD is better than simple ray-casting
  - for small targets,
  - low density scenes,
  - w.r.t. errors.
- SQUAD is **not** well suited
  - for objects highly distributed in depth (can be remedied)
  - to select objects that are visually similar to their neighboring objects

The Design Space of Selection by Refinement

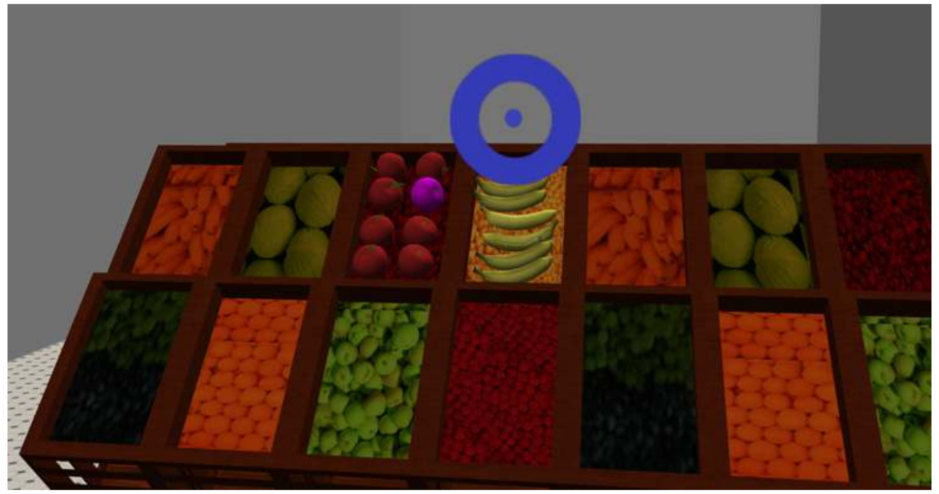
- Three dimensions:
  - Type of progression: discrete, continuous, …
  - Refinement criteria: object attributes, spatial, out of context, …
  - Display of current set of candidates: in context, out of context, …

Examples

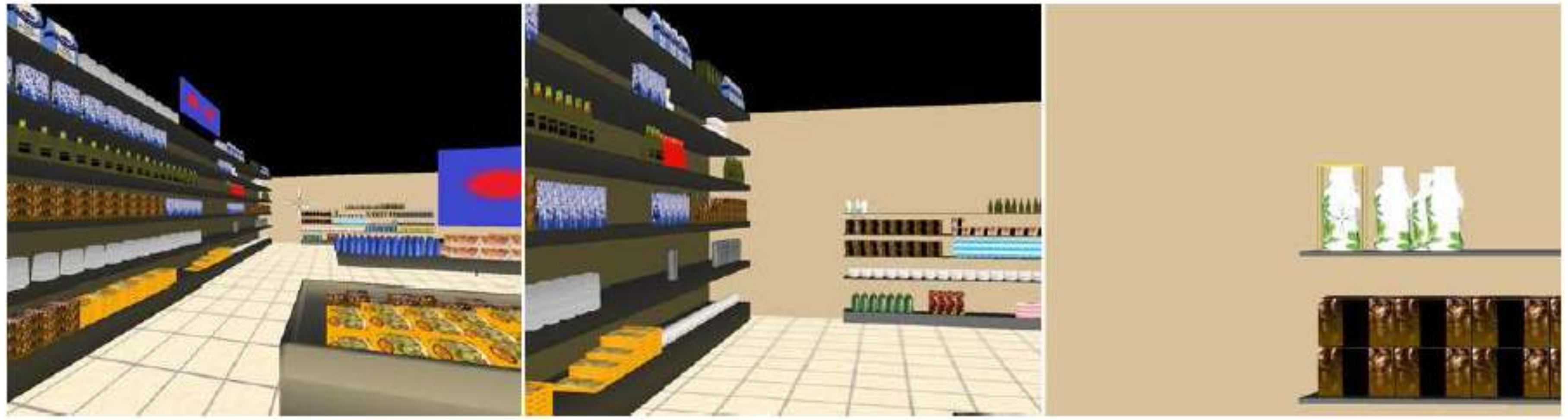
- Discrete zoom, partitioning by quadrants in image space



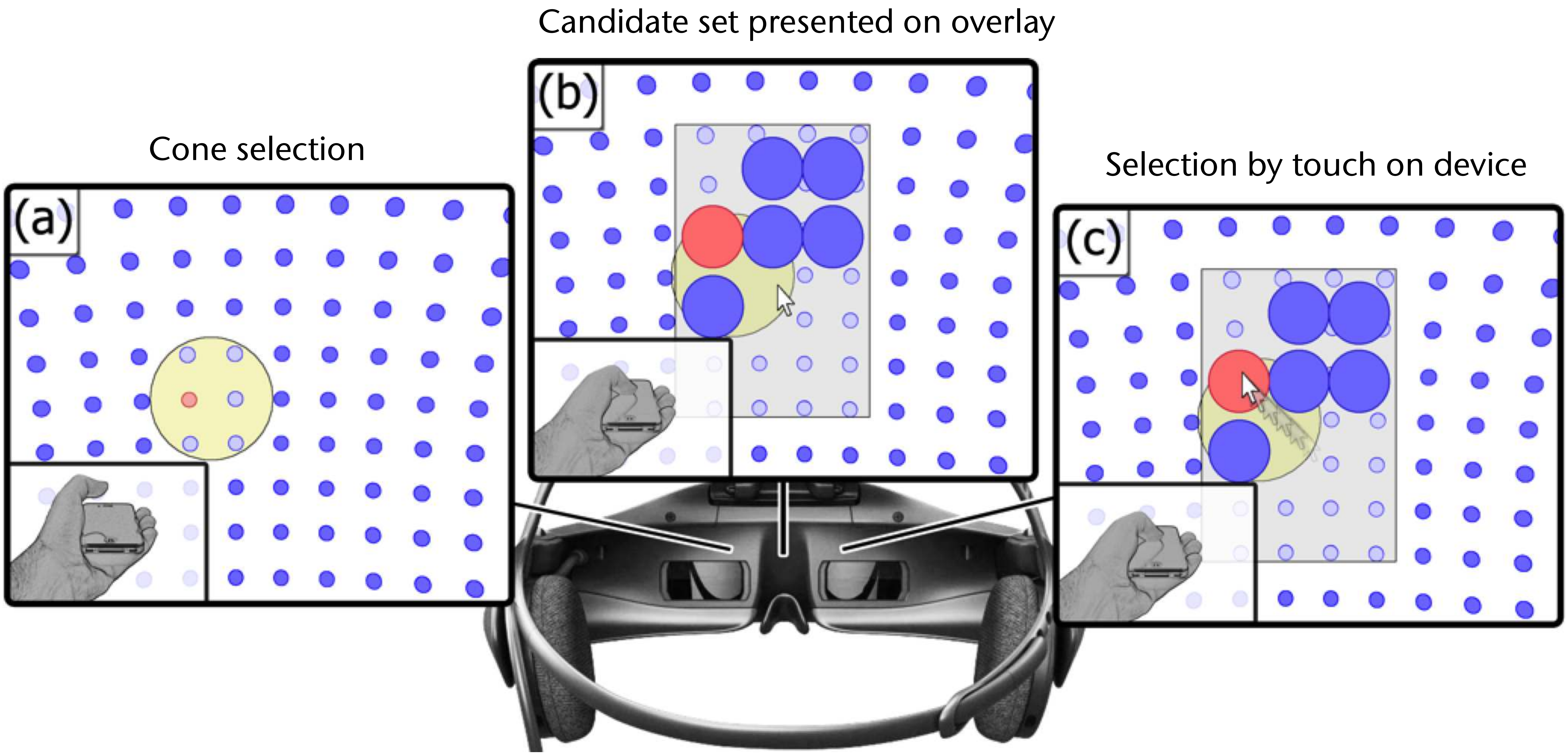
- Point-and-Zoom
  - Trigger initiates zoom-in over touched objects



- Continuous zoom



Disambiguation Canvas

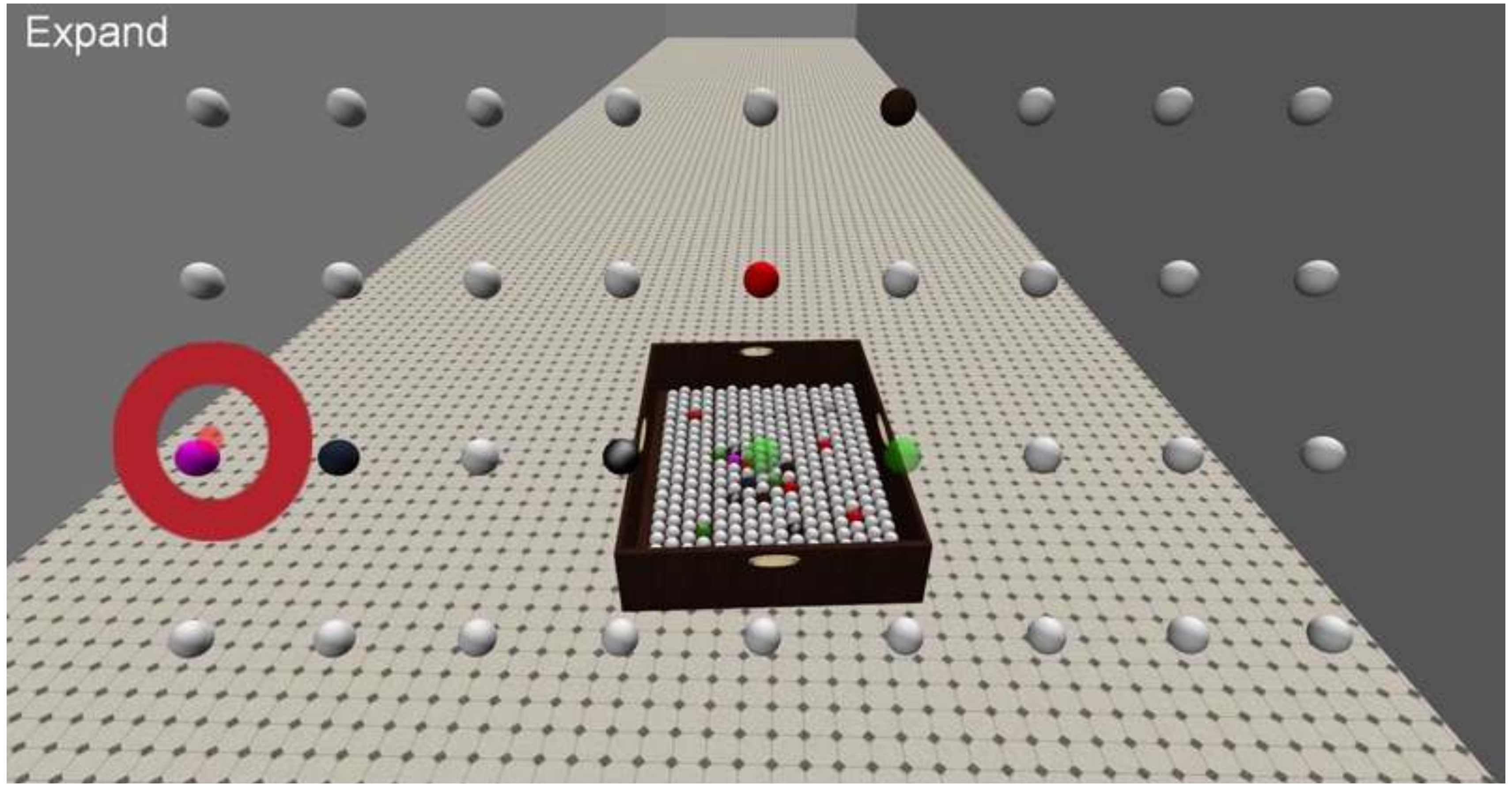

Candidate set presented on overlay
Cone selection
Selection by touch on device
(a)
(b)
(c)

Interlude: How to Describe Experiments for a User Study

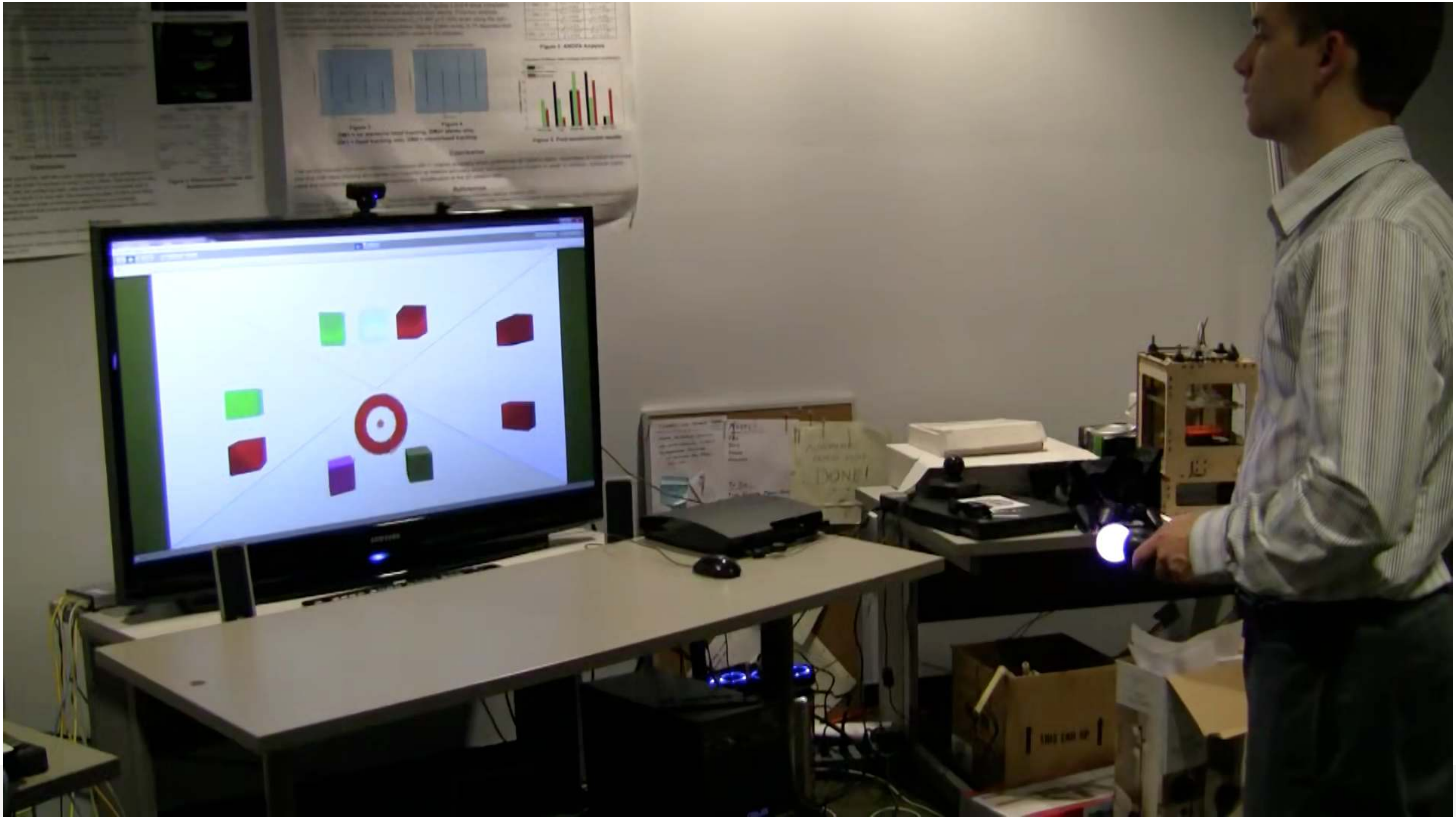
- Precisely describe:
  1. Independent and dependent variables
  2. Participants (number, demographics, pre-existing experience, ...)
  3. Apparatus (devices, computer, ...)
  4. Procedure:
     - what had participants to perform, in which order?
     - did you give them an introduction? which and how?
     - did they get a learning period? how long?
     - when did you have them fill out questionnaires?
     - did you record anything?
- Collect all the necessary data for doing the statistical analysis afterwards
- Do a small pre-study to check your experiment design and data collection
- Do a rigorous, hypothesis-driven, statistical analysis

The "Expand" Selection Technique

- Similar to SQUAD, but with the following modifications:
  - Candidate selection: use circle in image space instead of sphere in 3D
  - Refinement: arrange candidates in grid instead of 4 subsets
  - Transition from phase 1 to phase 2: clone objects, create automatic animation from 3D positions into grid positions
  - Make non-candidate objects in 3D scene transparent

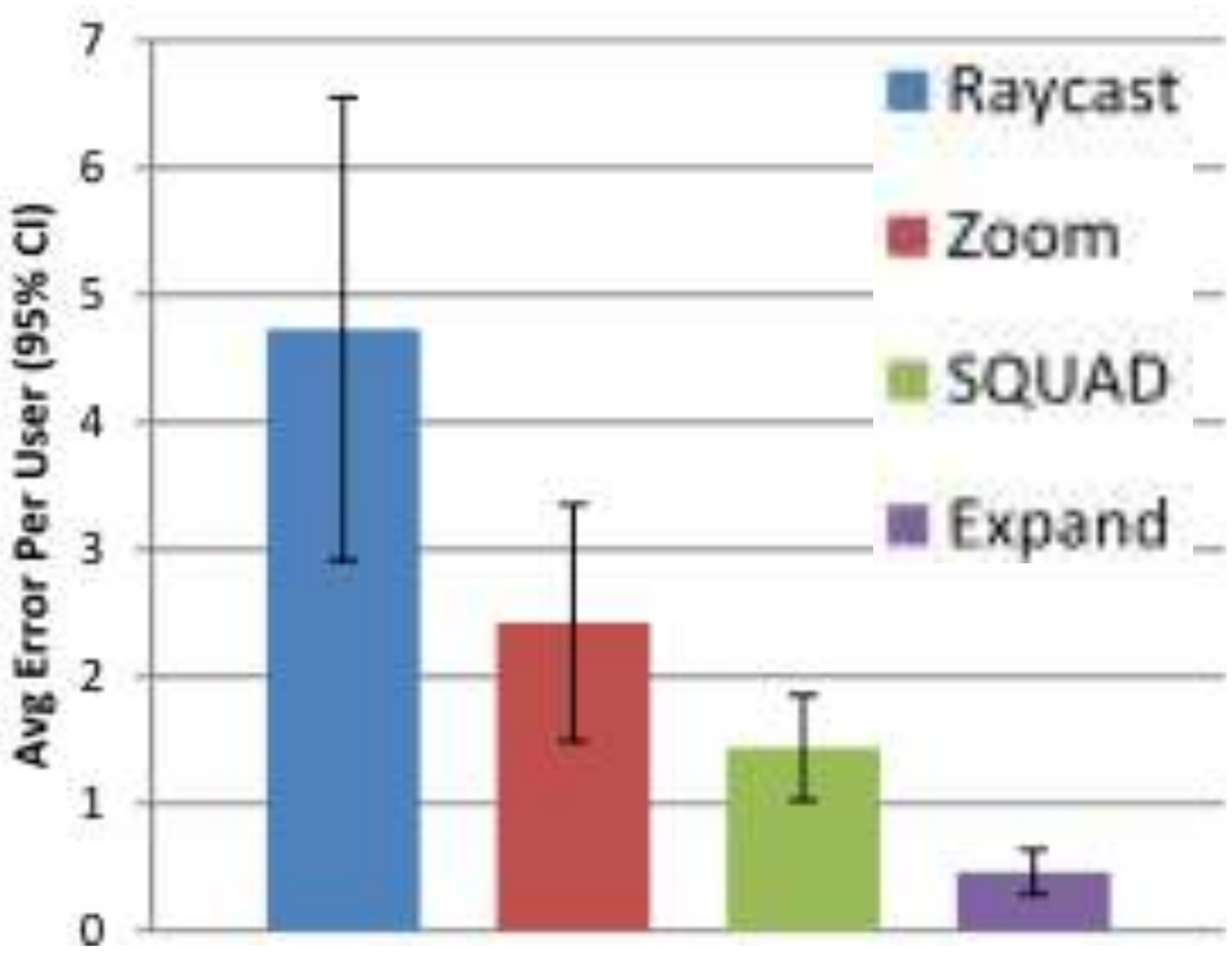


Movie Showing Raycasting, Click-and-Zoom, Expand, SQUAD

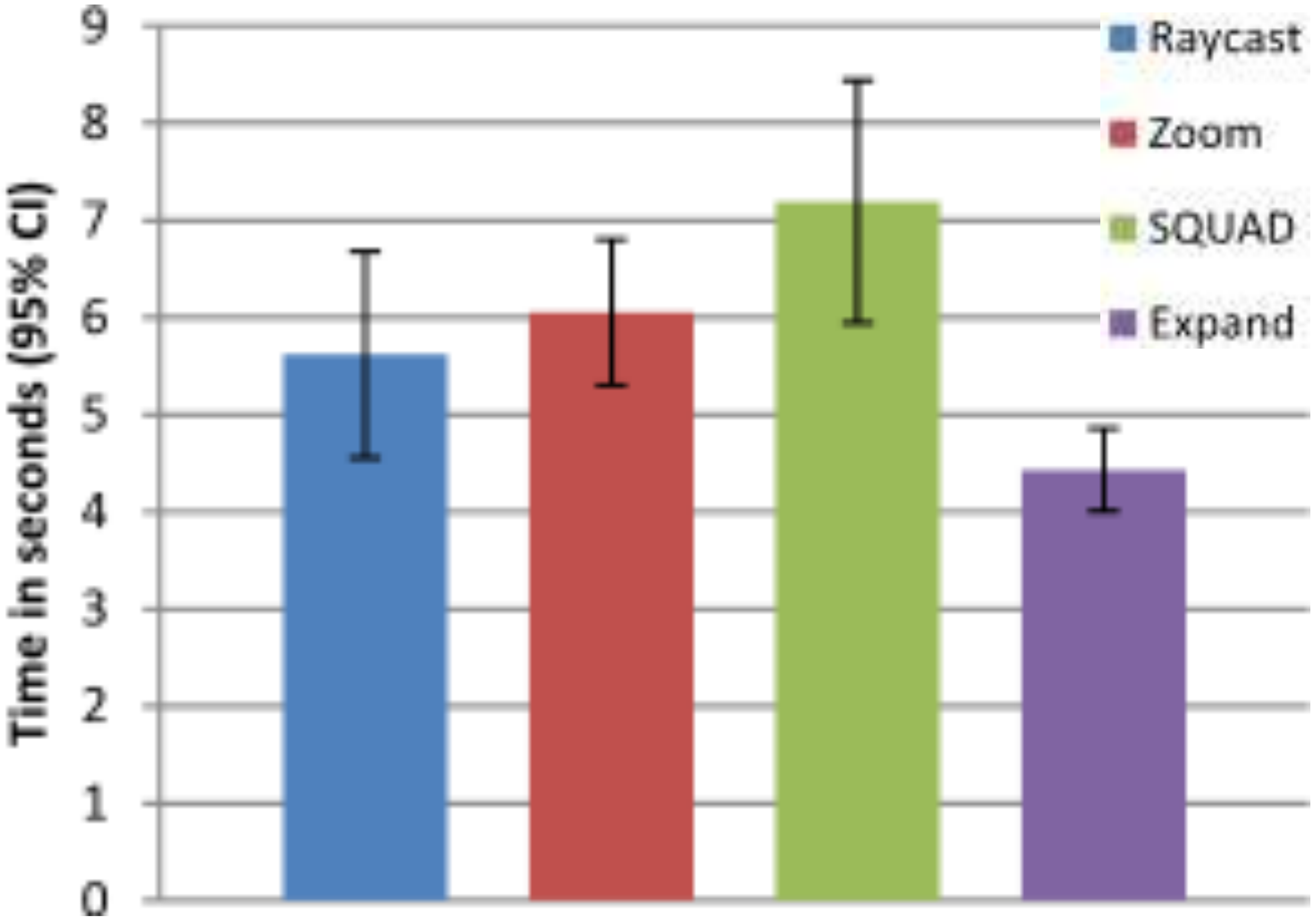


Comparison

Number of errors, averaged over 5 scenarios

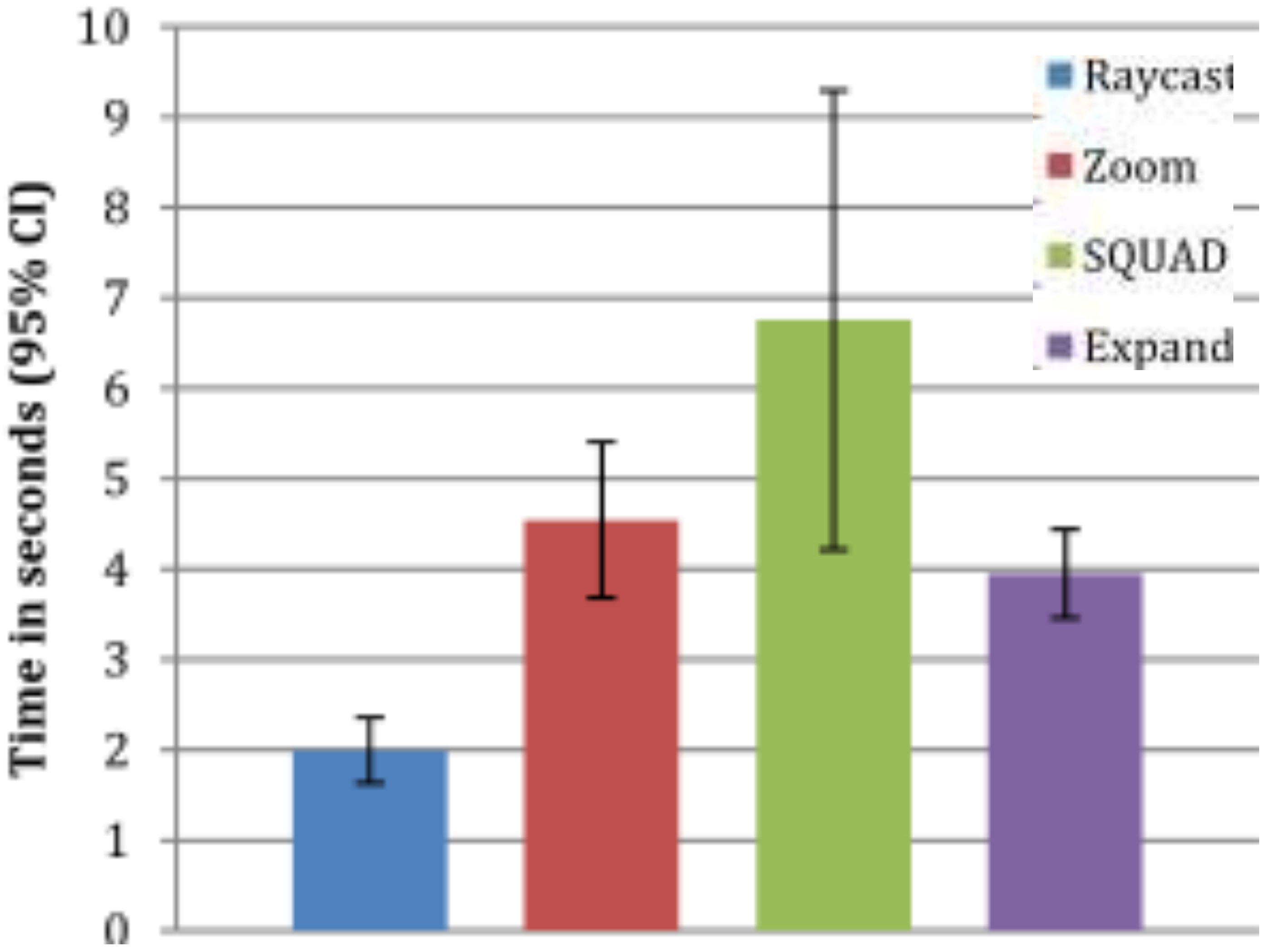


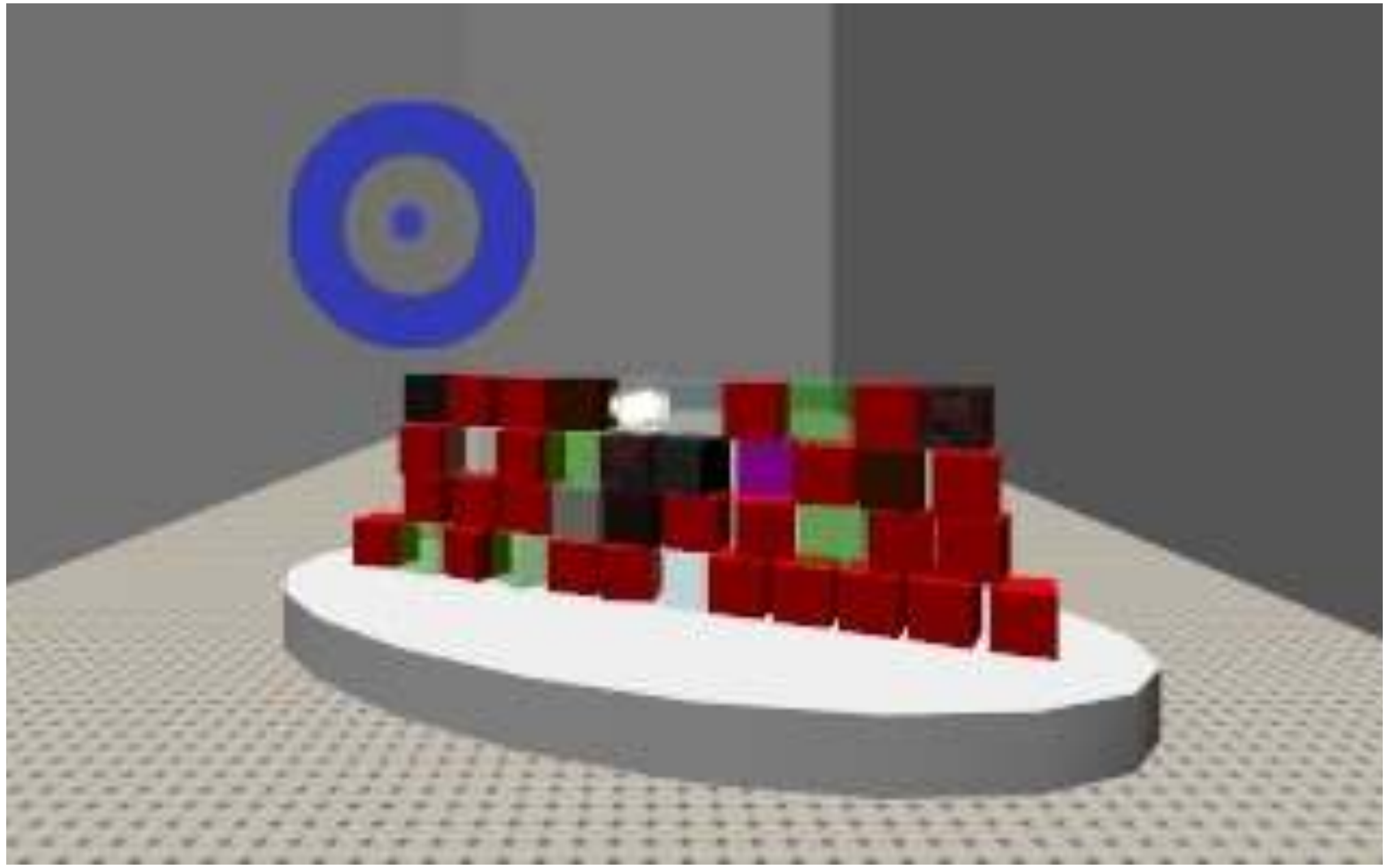
Task completion time, averaged over 5 scenarios



There Seems to be No Silver Bullet

Scenario: high density, low motion



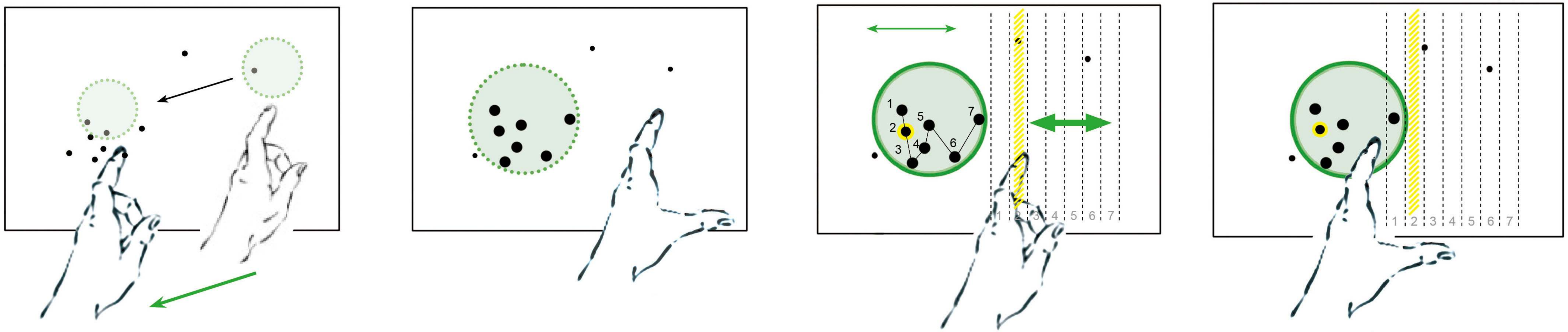


HorizontalDragger

- The technique (here with hand tracking):

1. Select candidate set

2. Move through candidates by horizontal motion



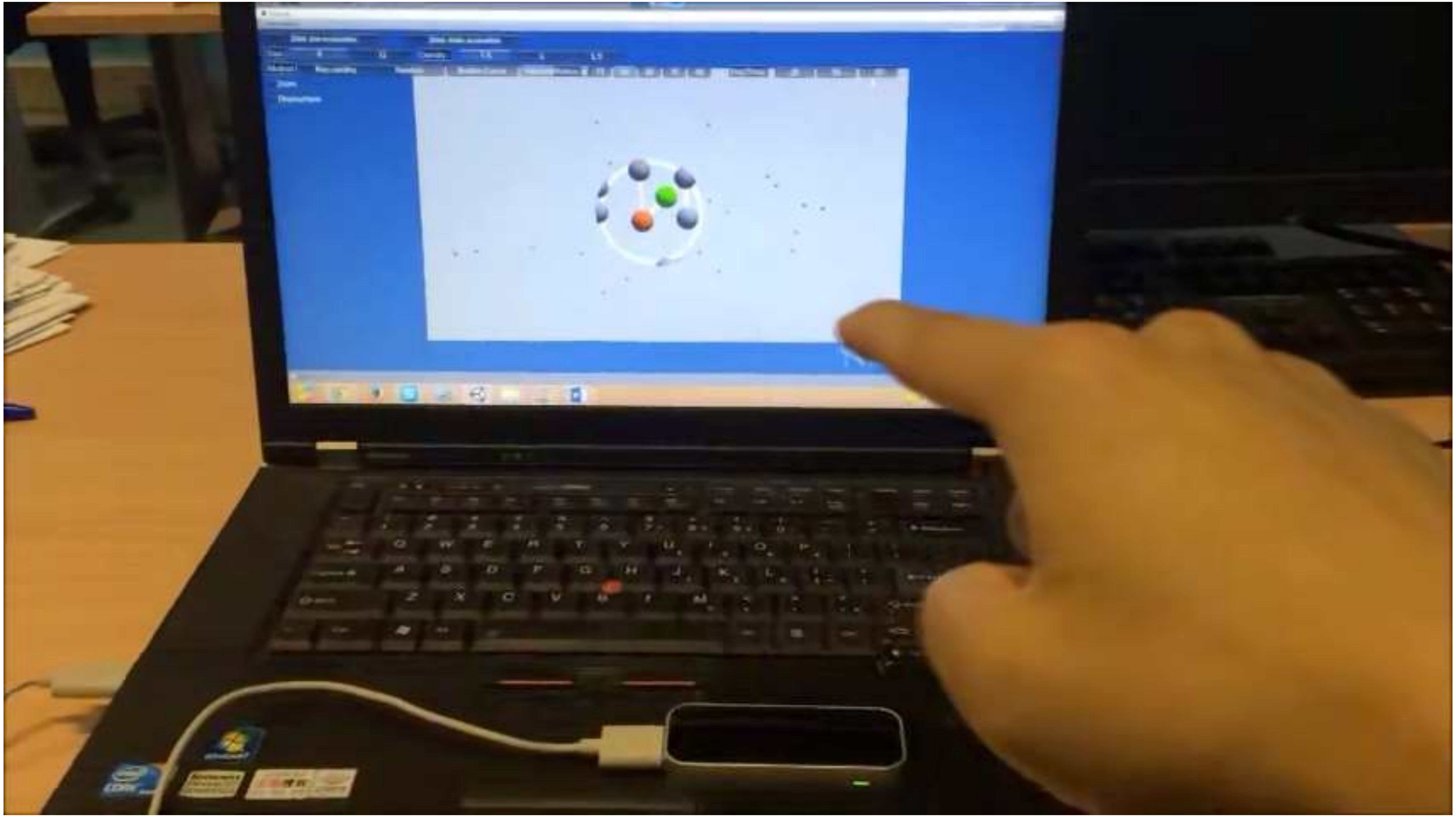
a drag the index to position the circle

b select the ROI

c drag to switch the focus

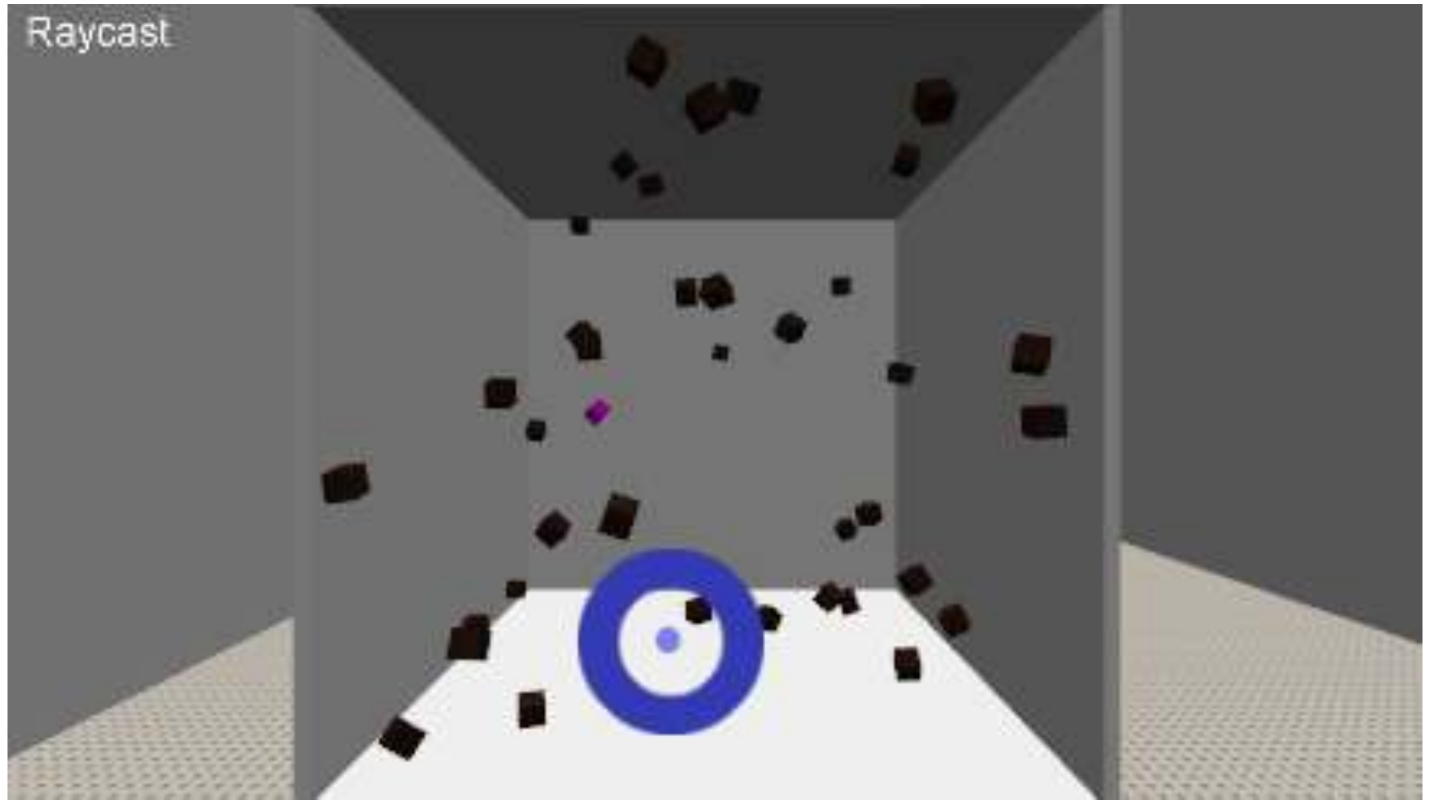
d stretch the thumb to select

- Advantages:
  - Copes relatively well with high densities
  - Second phase is in context

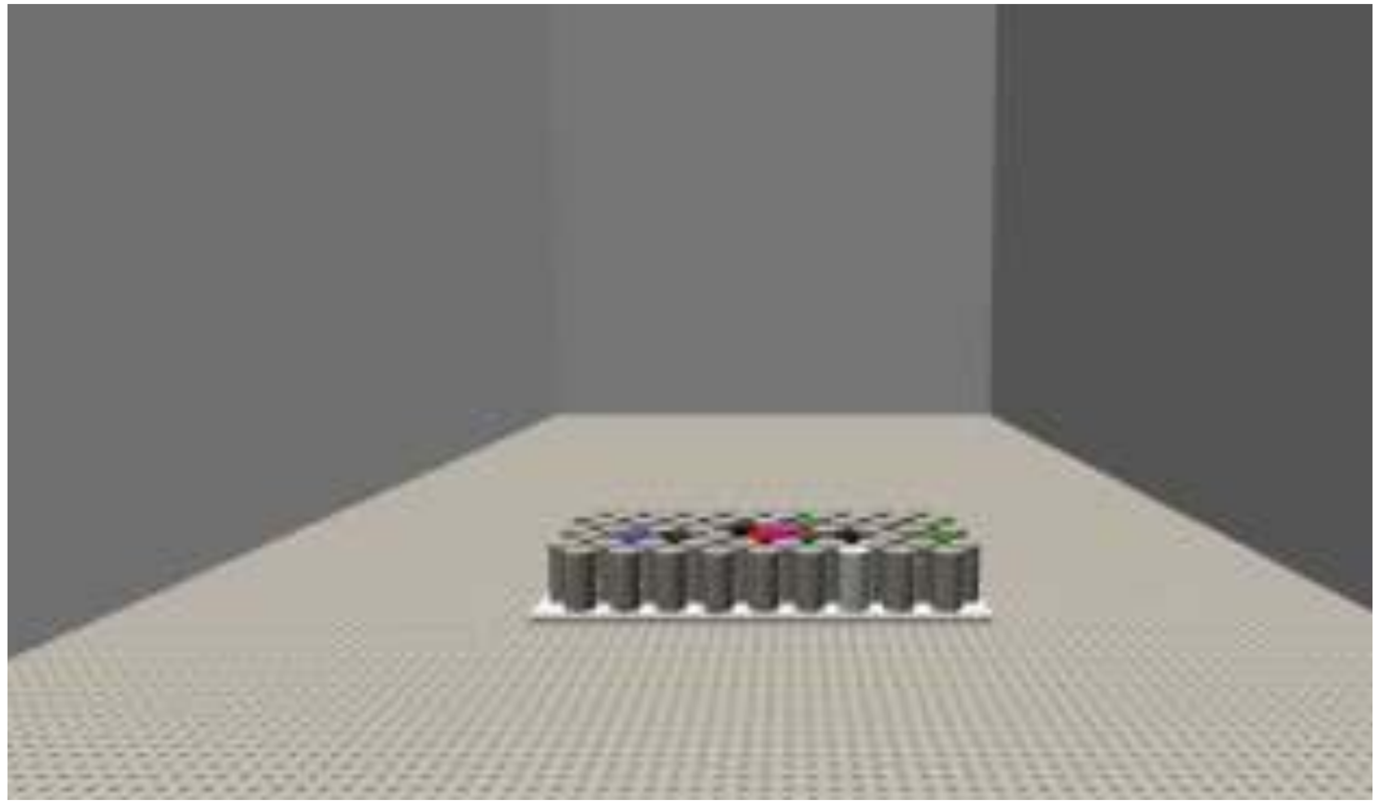


The Design Space of Test Scenarios for Evaluation

- Object density: low … high
- Occlusion: none … high
- Distance/object sizes: same distance & size … highly varying
- Motion: static … fast & randomly moving objects



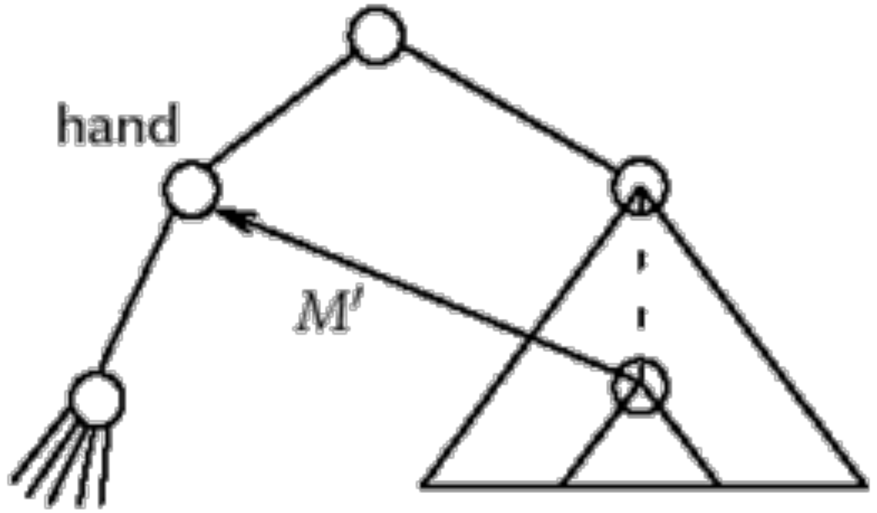
Medium density, random motion



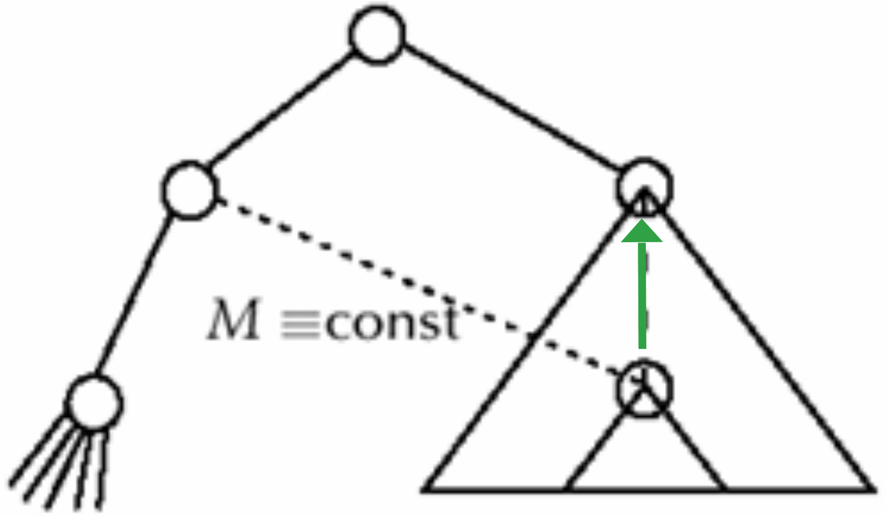
Static, high density, high occlusion

Object Manipulation (Grasping & Moving)

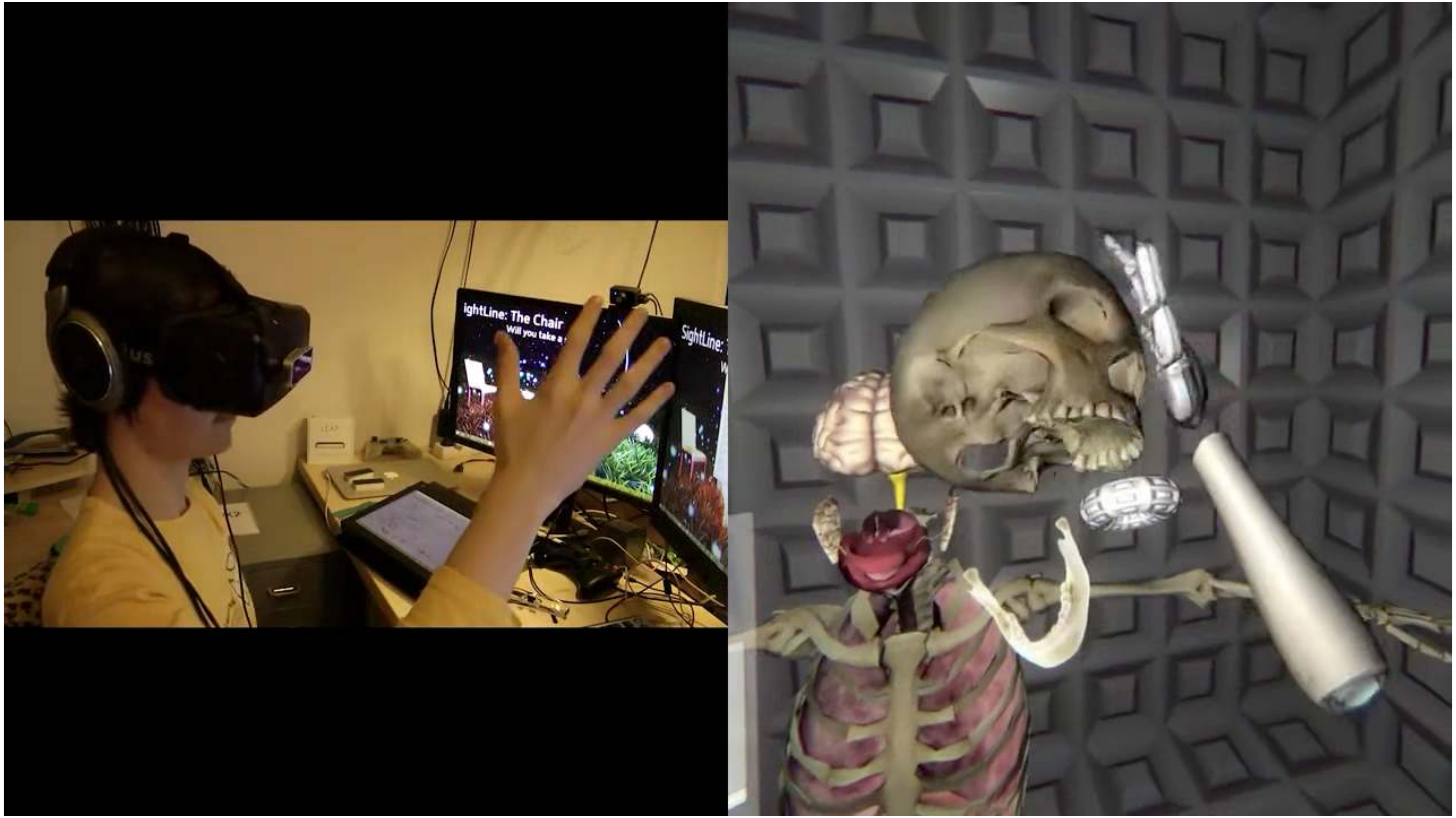
- Another pretty frequent interaction task
- Simple, direct grasping using "sticky metaphor" (not realistic):
  1. Select object
  2. Trigger grasping (via gesture, speech command, ...)
  3. Wait for collision between hand and object
  4. Make object "stick" to hand
  5. Trigger release



- How to implement the "sticking"?
  6. Re-link object to hand node, or
  7. Maintain transformation invariant between hand and object
  - My experience: with non-trivial applications, re-linking causes trouble!



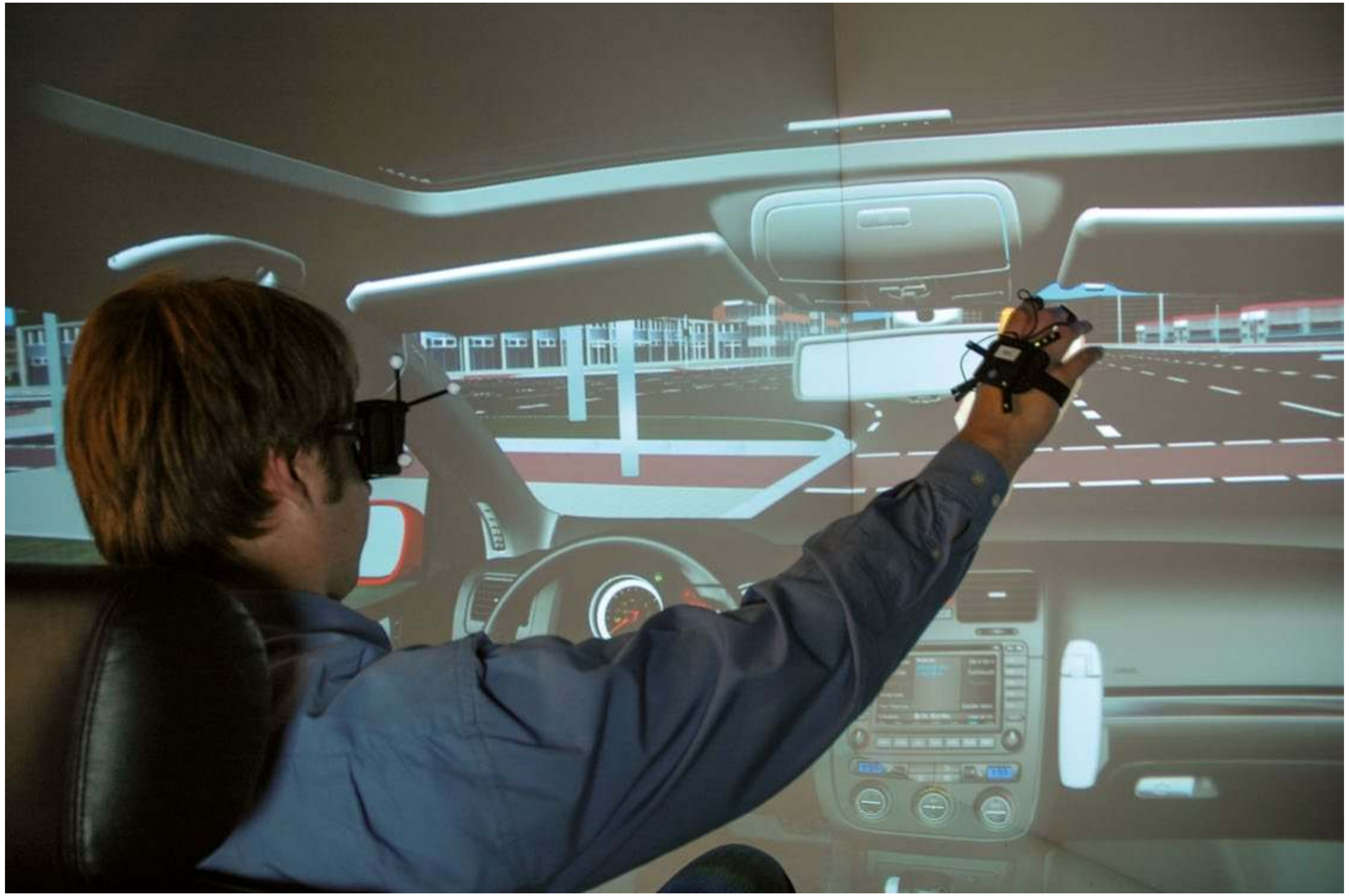
Example Video for "Stick-to-Hand" Grasping



World of Comenius

A Vision: Natural User Interaction (NUI)

- One of the goals: interact with virtual objects using our real hands as if they were real objects, i.e., no interaction metaphor

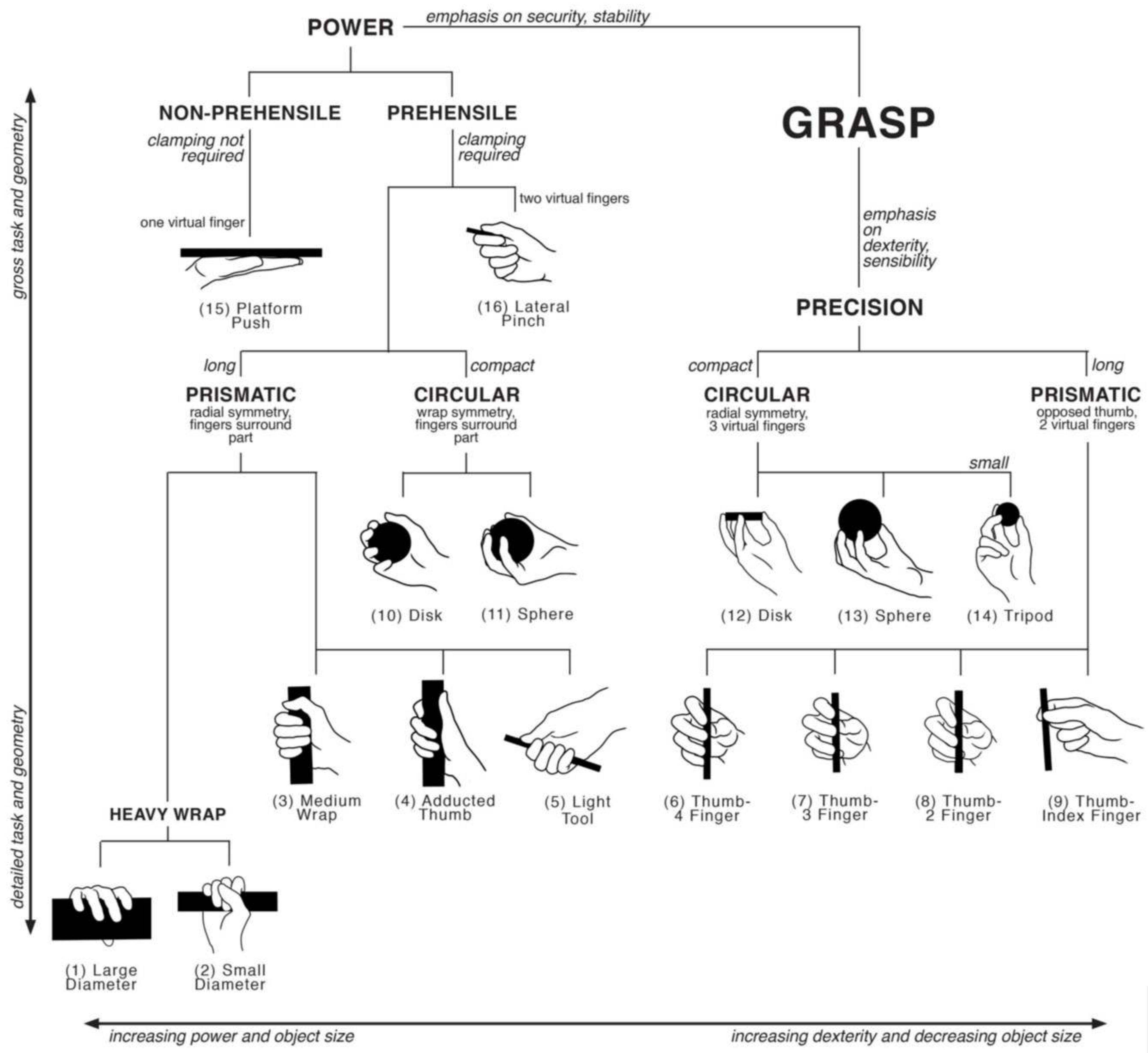


Courtesy of Volkswagen AG

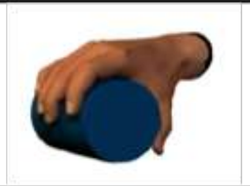
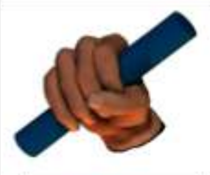
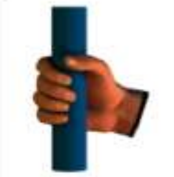
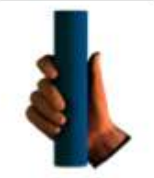
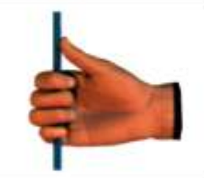
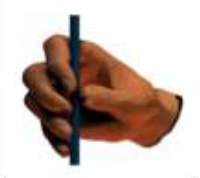
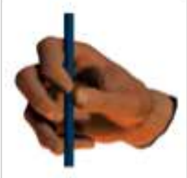
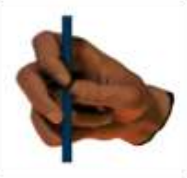
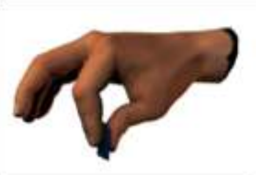
Characterisitcs and History of NUIs

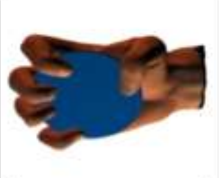
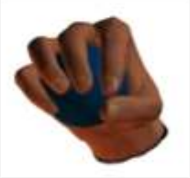
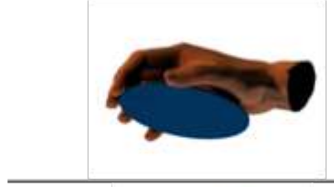
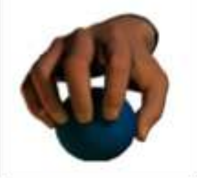
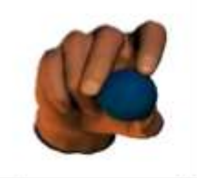
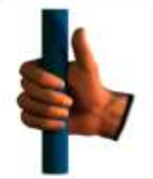
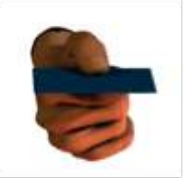
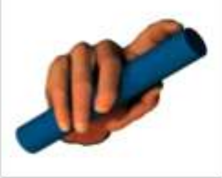
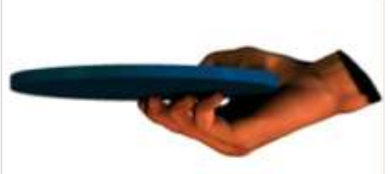
- Direct manipulation benefits:
  - Novices can learn basic functionality quickly, usually through a demonstration by a more experienced user
  - Experts can work extremely rapidly to carry out a wide range of tasks, even defining new functions and features
  - Error messages are rarely needed
  - Feedback: users can see immediately, if their actions are furthering their goals, and if not, they can simply change the direction of their activity
- Originally proposed by Ben Shneiderman for GUIs as Direct Manipulation (in 1980)
  - But the same principles and benefits apply to today's NUIs

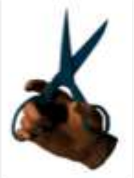
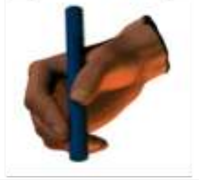
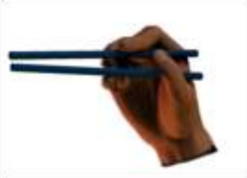
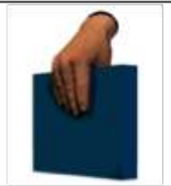
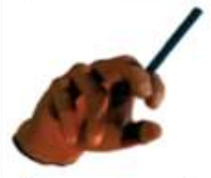
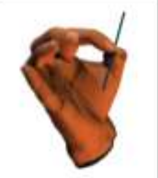
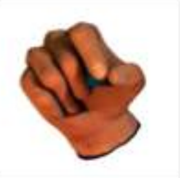
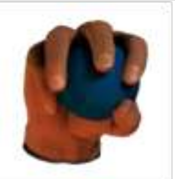
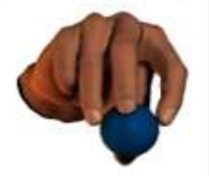
Taxonomies of Natural Grasping Poses

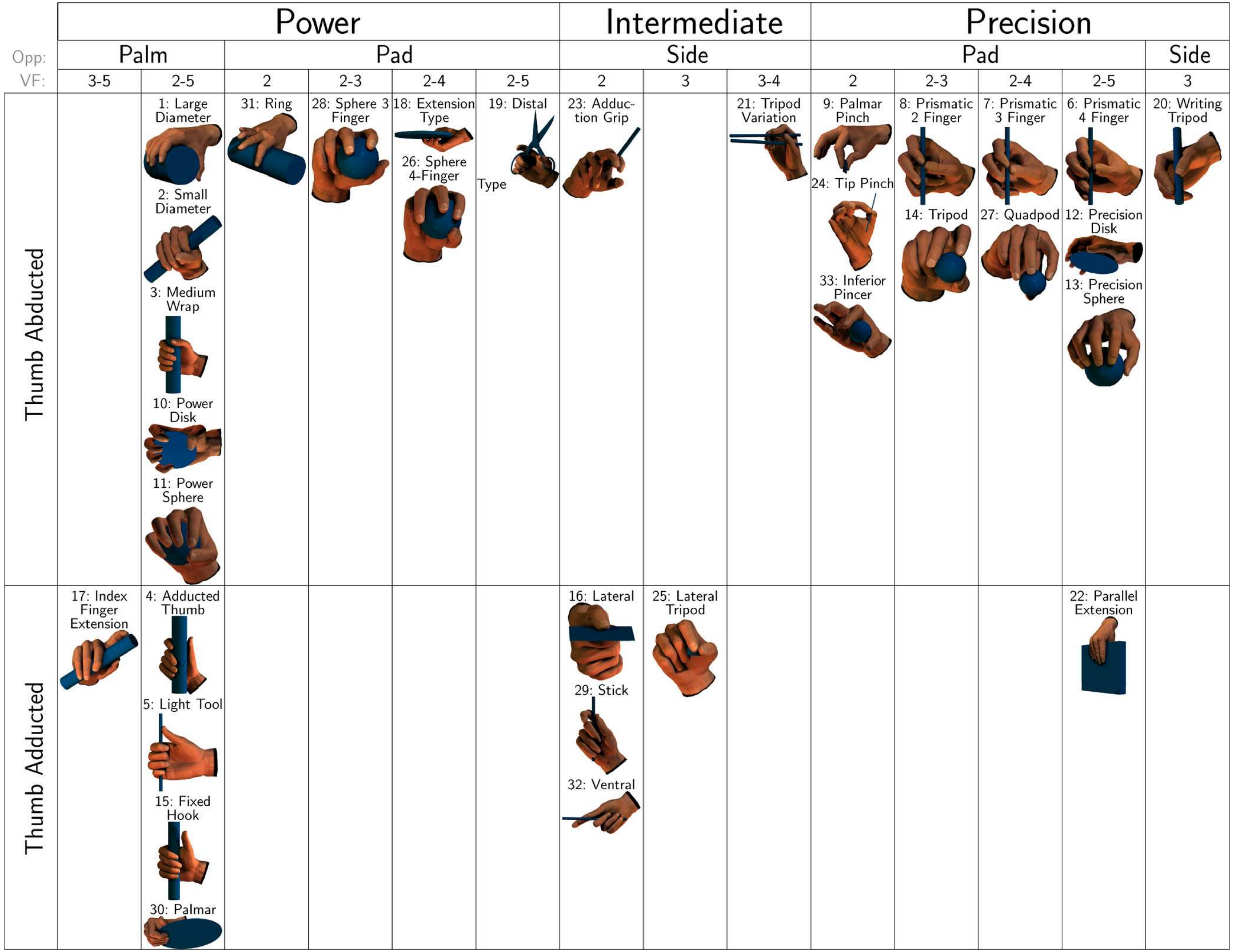


[Cutkosky & Howe, 1990; and Zheng et al, 2011]

Picture	Type	Opp. Type	Thumb Pos.
	Power	Palm	Abd
	Power	Palm	Abd
	Power	Palm	Abd
	Power	Palm	Add
	Power	Palm	Add
	Precision	Pad	Abd
	Precision	Pad	Abd
	Precision	Pad	Abd
	Precision	Pad	Abd

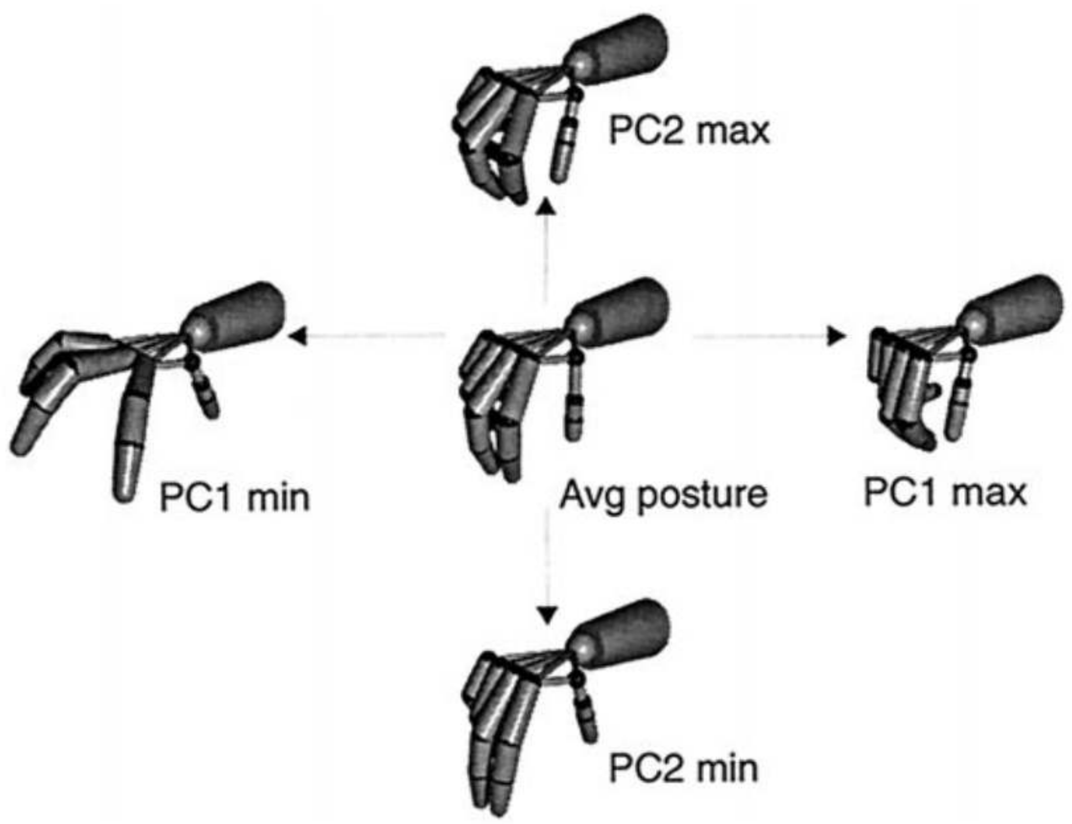
Picture	Type	Opp. Type	Thumb Pos.
	Power	Palm	Abd
	Power	Palm	Abd
	Precision	Pad	Abd
	Precision	Pad	Abd
	Precision	Pad	Abd
	Power	Palm	Add
	Inter-mediate	Side	Add
	Power	Palm	Add
	Power	Pad	Abd

Picture	Type	Opp. Type	Thumb Pos.
	Power	Pad	Abd
	Precision	Side	Abd
	Inter-mediate	Side	Abd
	Precision	Pad	Add
	Inter-mediate	Side	Abd
	Precision	Pad	Abd
	Inter-mediate	Side	Add
	Power	Pad	Abd
	Precision	Pad	Abd



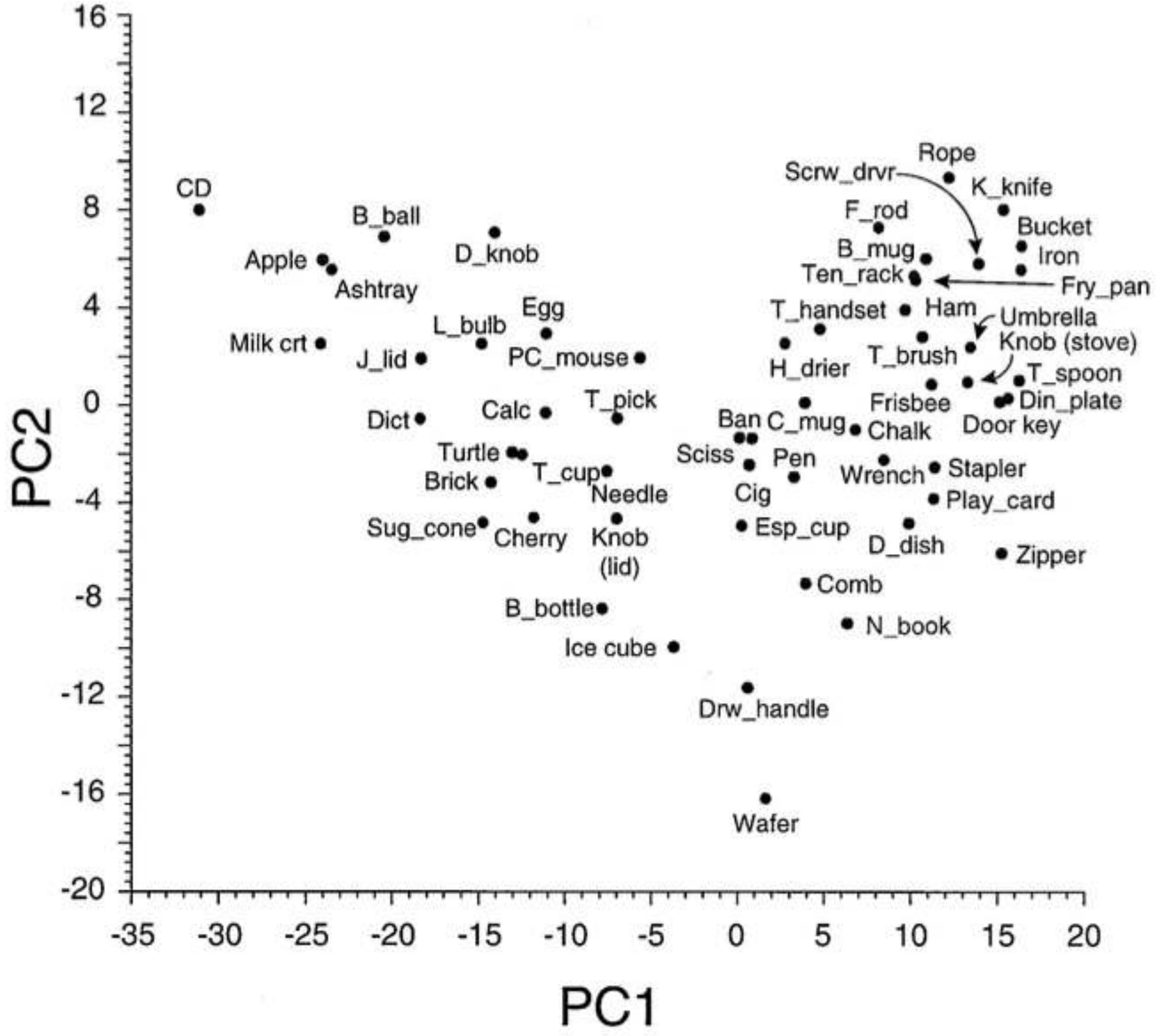
		Power						Intermediate			Precision				
Opp:		Palm		Pad				Side			Pad				Side
VF:		3-5	2-5	2	2-3	2-4	2-5	2	3	3-4	2	2-3	2-4	2-5	3
	Thumb Abducted		1: Large Diameter 2: Small Diameter 3: Medium Wrap 10: Power Disk 11: Power Sphere	31: Ring	28: Sphere 3 Finger	18: Extension Type 26: Sphere 4-Finger	19: Distal Type	23: Adduction Grip		21: Tripod Variation	9: Palmar Pinch 24: Tip Pinch 33: Inferior Pincer	8: Prismatic 2 Finger 14: Tripod	7: Prismatic 3 Finger 27: Quadpod	6: Prismatic 4 Finger 12: Precision Disk 13: Precision Sphere	20: Writing Tripod
	Thumb Adducted	17: Index Finger Extension	4: Adducted Thumb 5: Light Tool 15: Fixed Hook 30: Palmar					16: Lateral 29: Stick 32: Ventral	25: Lateral Tripod					22: Parallel Extension	

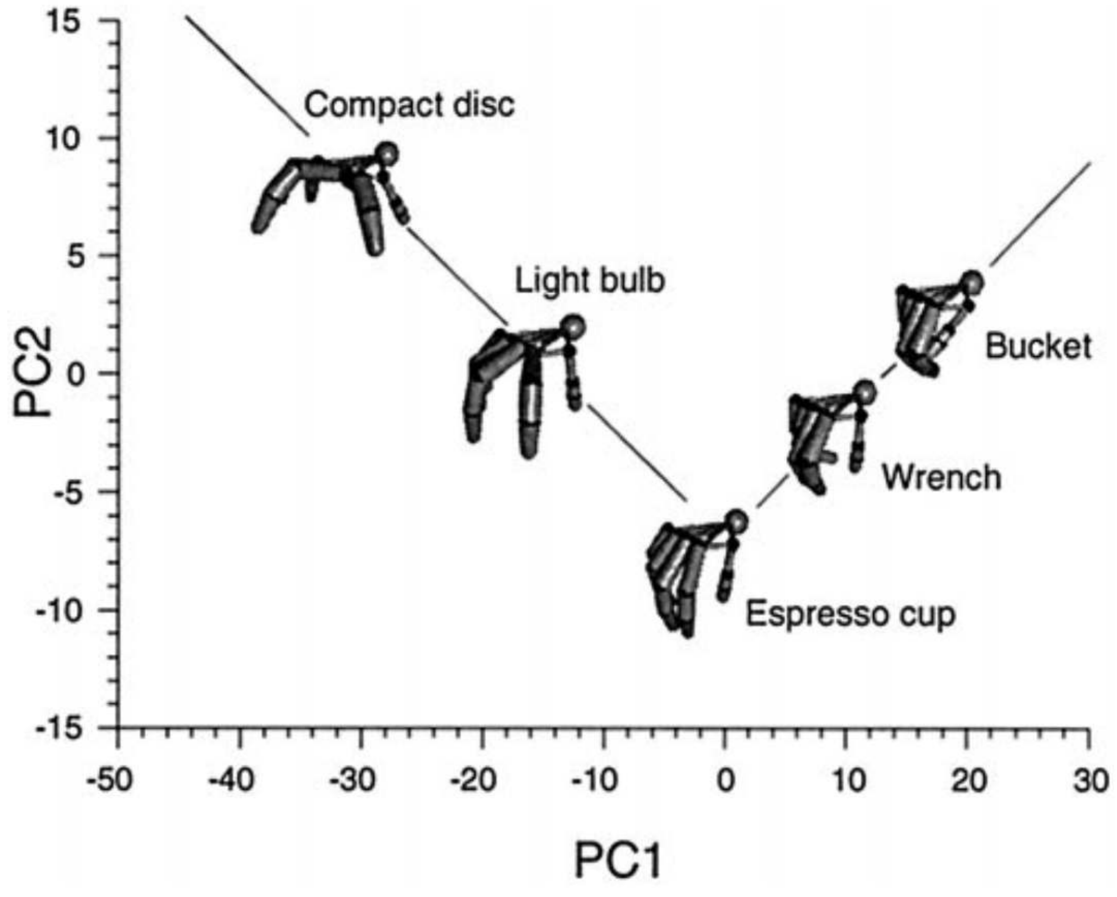
The Manifold of Hand Postures



Human grasping postures
are a low-dimensional manifold
in posture space

PCA

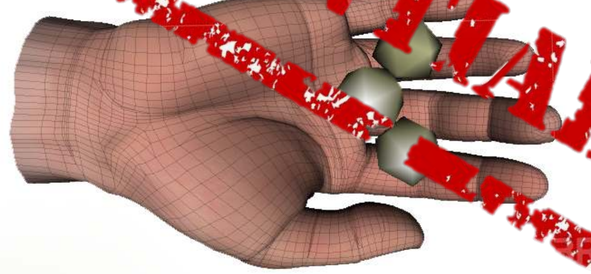




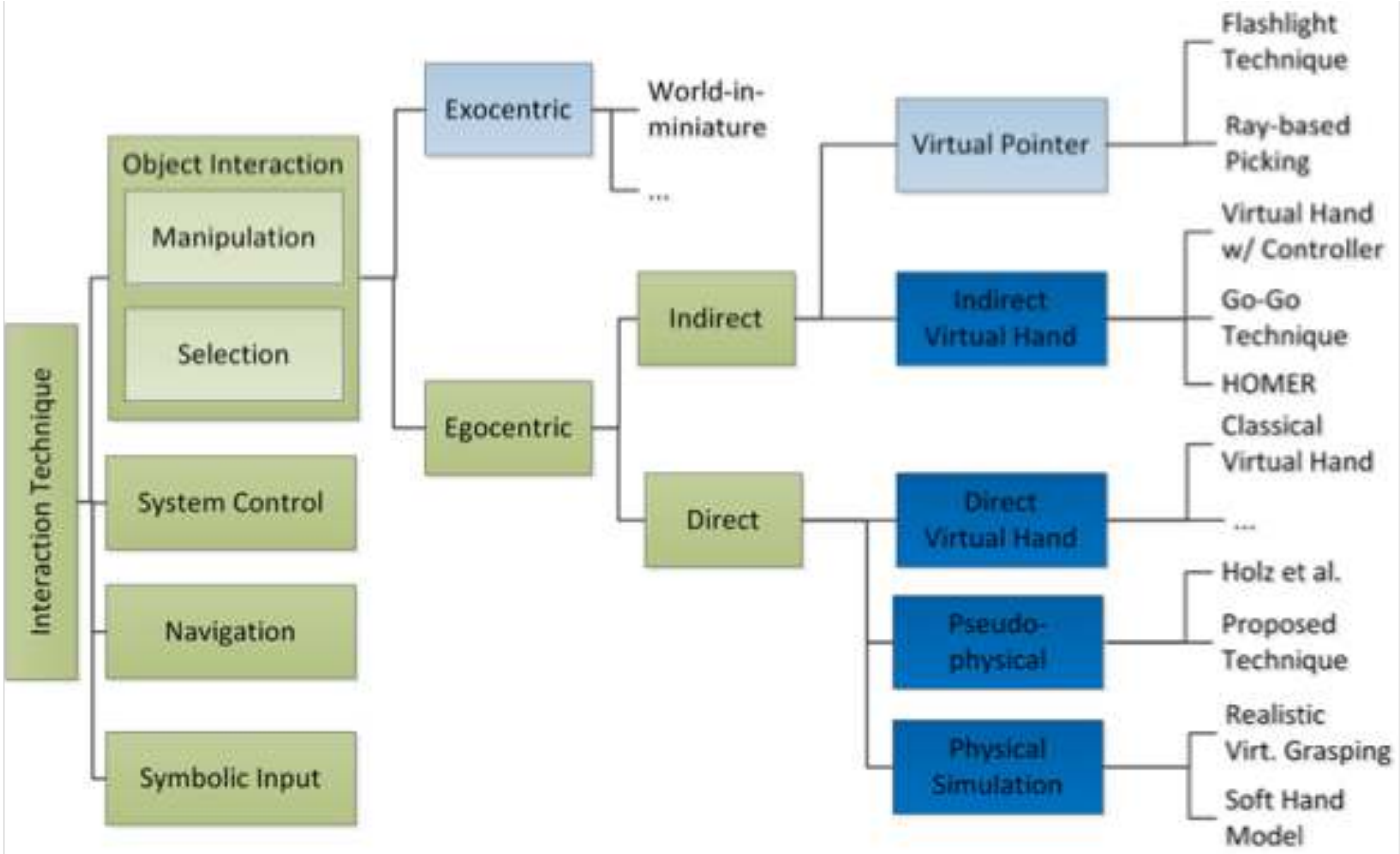
Call for Thesis

CONFIDENTIAL

- Idea:
  - Consider grasping/manipulating objects as an optimization problem
  - I.e., at each frame, find a position/orientation of the object that minimizes penetration / maximizes distance to the fingers
- Building blocks:
  - We have software that calculates penetration volumes / minimal distances between objects in ~1 millisecond
  - Use non-linear optimization software (lots out there)
- Requirements:
  - Programming skills in C/C++
  - Mathematical thinking/understanding,
    ideally a bit of knowledge in optimization methods
    (no theorem proving needed)



A Taxonomy of Object Interaction Techniques



Mathias Moehring: Realistic Interaction with Virtual Objects Within Arm's Reach

Quiz on 3D Interaction

https://www.menti.com/p9rdrjti47

A Few General Design Principles

- Two-handed interaction
- Multimodal interaction
- Physical props (e.g., specimen box, tangible UI's)
- Action-at-a-distance
- Proprioception
- "Redirecting" the user
- Task decomposition
- Dimension decomposition

Two-Handed Interaction

- Users have 2 hands: a dominant one (usually right) and a non-dominant one (left)
- The roles of each hand:
  - Non-dominant hand = reference coordinate system, positioning of context
  - Dominant hand = fine-skilled motor tasks within that context
- Good metaphors = metaphors that utilize both hands within their respective roles
- Examples already seen: iSith, balloon, WIM, …

Multimodal Interaction Principles

- Single input:
  - Specialized:
    - Single modality that is clearly ideal and sufficient
    - Ex.: Big red emergency button
  - Different but equivalent:
    - User chooses one of several modalities, each gives same result
    - Ex.: trigger an action by menu or voice command
- Sequential input:
  - User uses one modality to trigger next modality
  - Ex.: Push-to-talk
- Simultaneous input:
  - Concurrent:
    - User issues different commands
    - Ex.: point-to-fly while requesting into using voice command
  - Complementary:
    - Combined info from different modalities gives one command
    - Ex.: intersect object by ray while saying "select"
  - Redundant:
    - Different modalities convey same info
    - Ex.: Intersect tube with virtual hand while saying "grab cube"

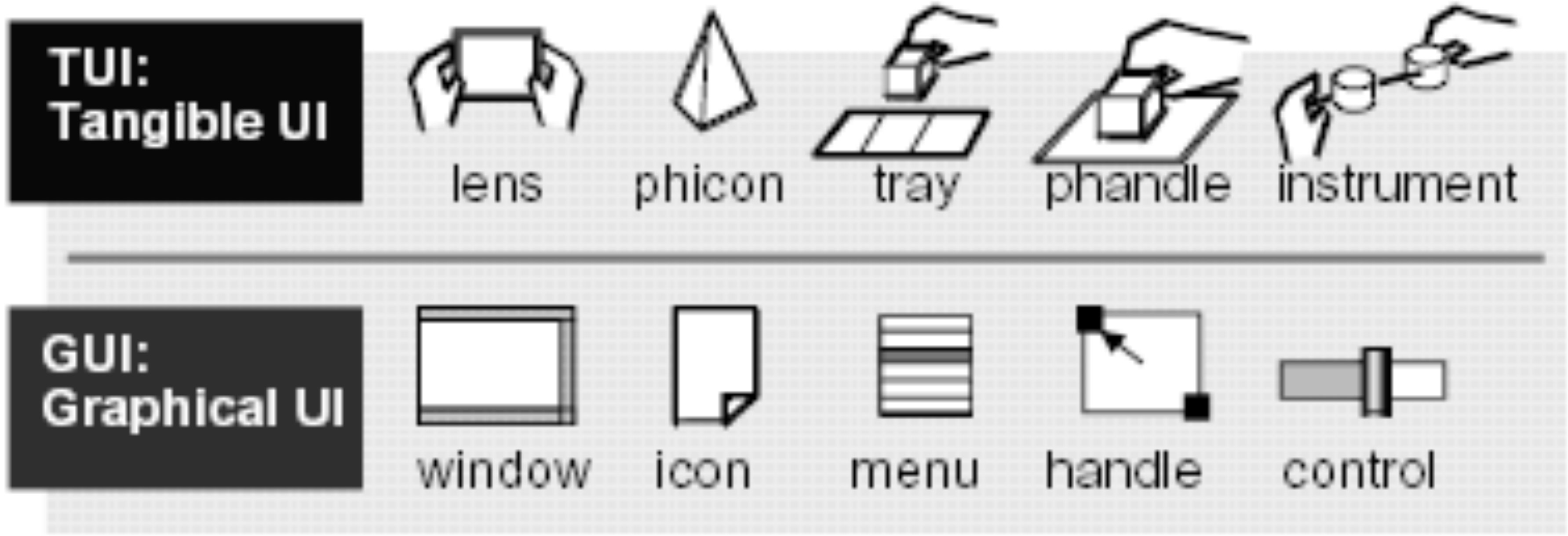
Design Principle: Tangible User-Interfaces (Props)

- Idea: instantiate virtual/abstract interaction metaphors (handles, icons, sliders, ...) by physical objects
- Definition Tangible User Interface (TUI):
  An attempt to give physical form to digital information, making "bits" directly manipulatable and perceptible by people.

  Tangible Interfaces will make bits accessible through

  - augmented physical surfaces (e.g. walls, desktops, ceilings, windows),
  - graspable physical objects (e.g. building blocks, models, instruments),
  - ambient physical media (e.g. light, sound, airflow, water-flow, kinetic sculpture).

- Analogies between GUIs and TUIs:



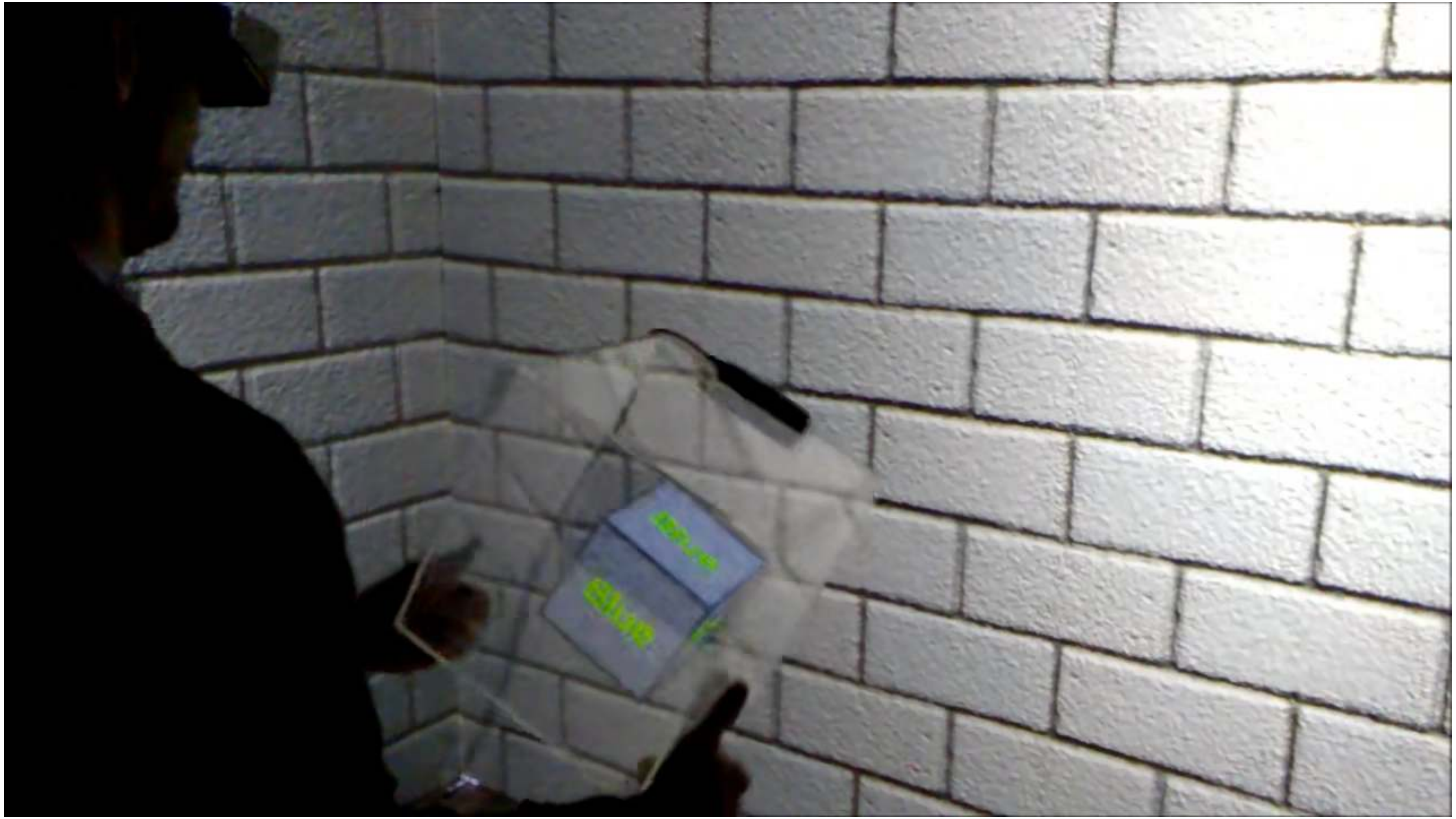
- Examples:

Tangible Magic Lens

http://tangible.media.mit.edu/

Example for Physical Props: the Specimen Box

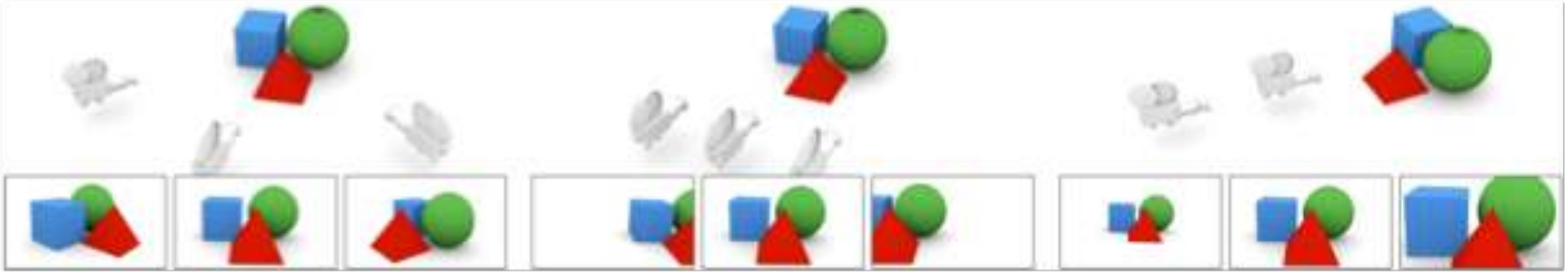
- Uses a tracked, translucent box as a prop with a world-fixed display
- The manipulated object is rendered on the wall *behind* the box, such that it looks like it is *inside* the box



- Advantages:
  - Direct, intuitive object manipulation
  - Passive haptic feedback
- Disadvantages:
  - Tethering of the box
  - Reflections on the box
  - Users felt the object to be *less present*
- Unclear which technique is faster (object manipulation using specimen box or one-handed "sticky" object grabbing)

An Interaction Design Principle: Separating DOFs

- Here, by the example of the Rotate-Pan-Zoom technique (a.k.a. Rotate-Scale-Translate):



Rotate Pan Zoom

- Other examples:
  - Balloon Selection
  - 3D manipulation widgets
  - Navidget

DOF Separation/Reduction by the PRISM Metaphor

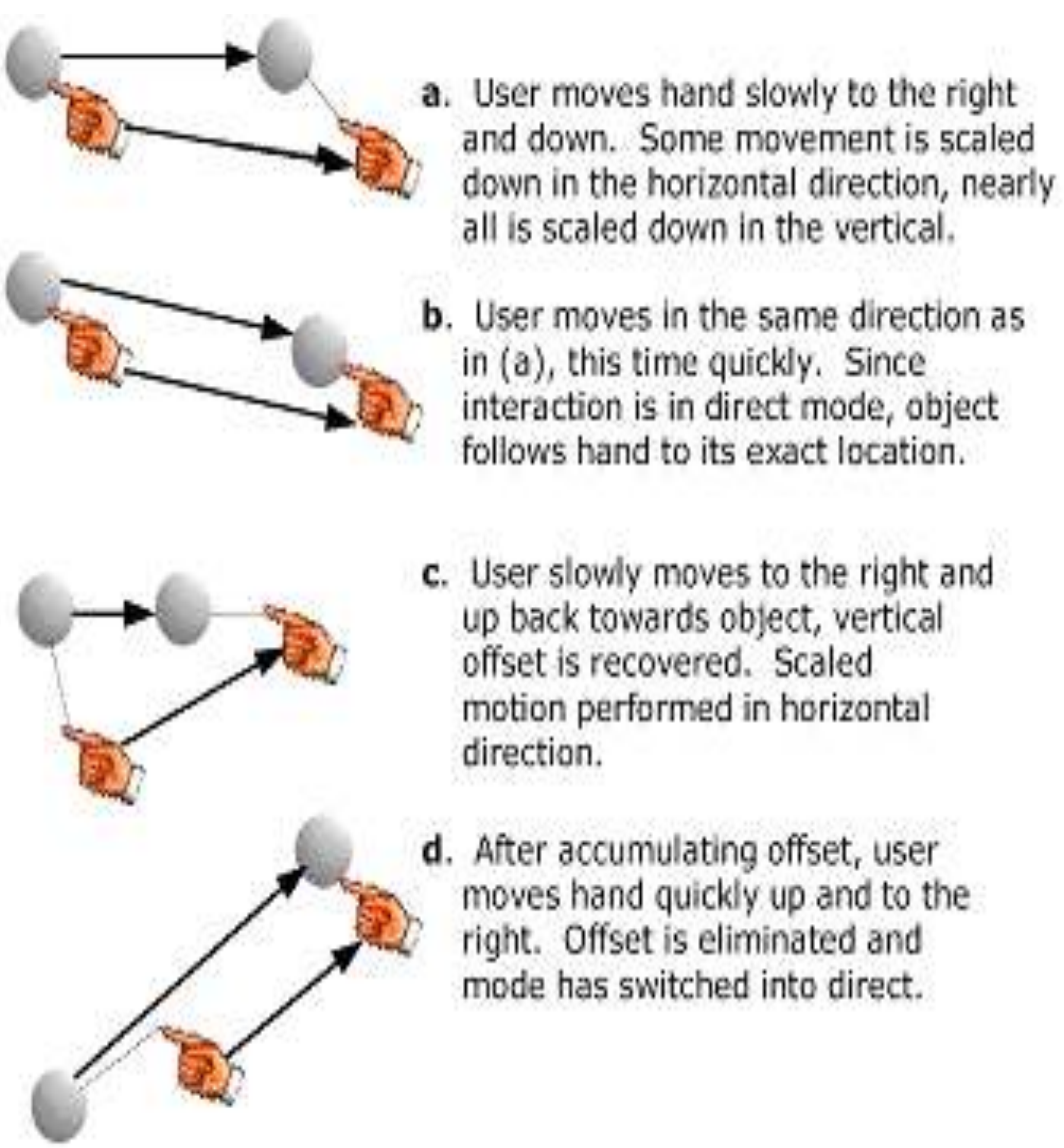
- Idea:
  - Rubberband metaphor with automatic adjustment of precision
  - System detects when the user is trying to be precise and when not
  - Adjust C-D ratio (*k*) accordingly
- More specifically:
  - Let D_O = translation distance of manipulated object,
    D_H = translation distance of user's hand,
  - Set distance

$$D_O = k \cdot D_H \qquad k = \begin{cases} 1 & , V_H > S \\ V_H/S & , \min < V_H < S \\ 0 & , V_H \leq \min \end{cases}$$

with V_H = average speed of hand during past ½ second,
S = some threshold

- Additional idea:
  - Do the scaling independently for each coordinate axis
  - Advantage: helps to move objects exactly along an axis of the coord system
- Recovery from offsets/drifts:
  - Problem: the positions of hand and object drift apart over time
  - Solution: reduce the offset while the user moves the hand very fast
    - Make the object move even faster
    - During fast movement, the user doesn't notice
- Remark: this technique works almost exactly analogously for rotations
  - Just convert the orientation of the user's hand to axis + angle, scale the angle, then convert back to rotation matrix
  - In the original PRISM paper, it was implemented differently

Videos

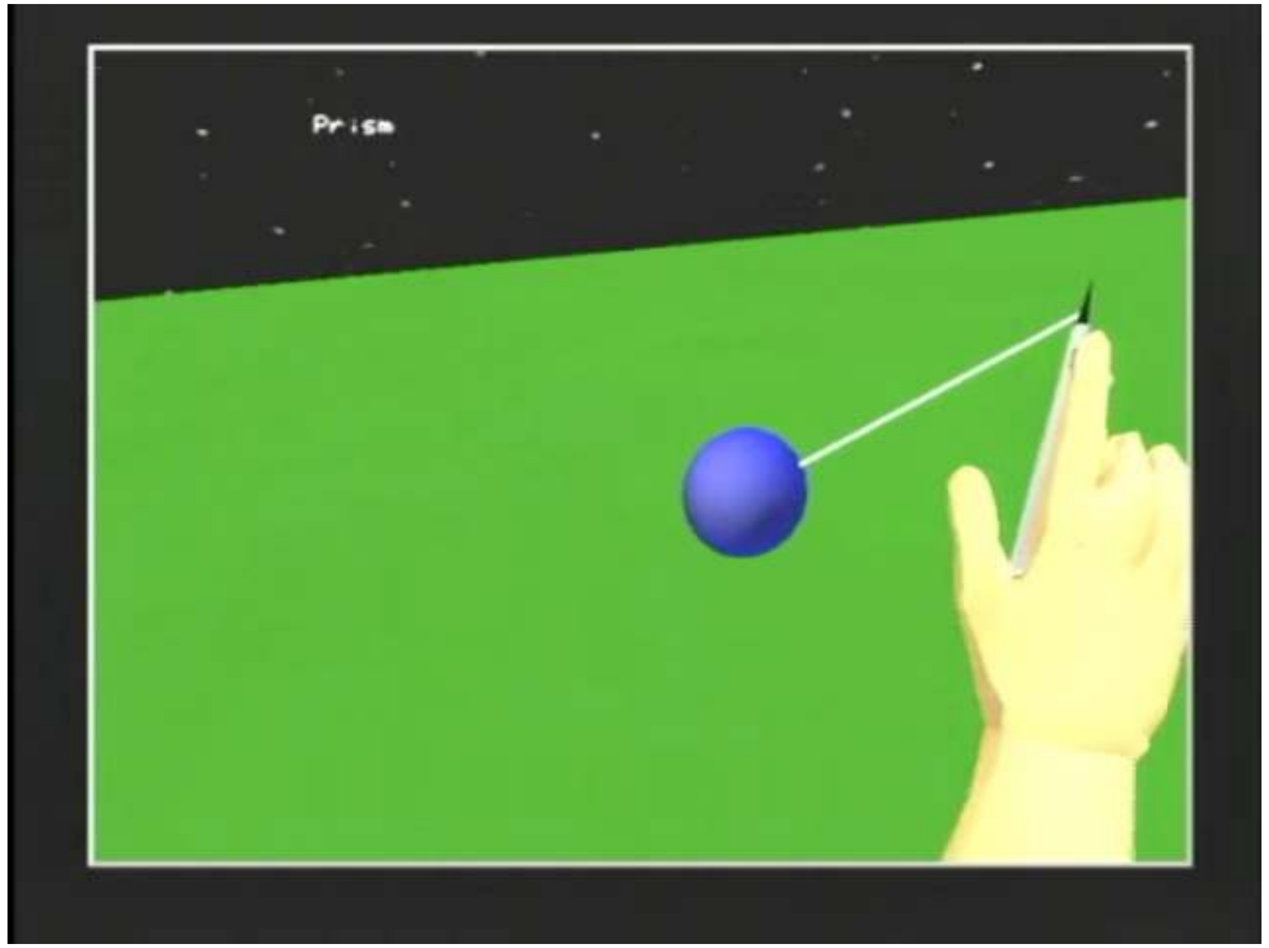


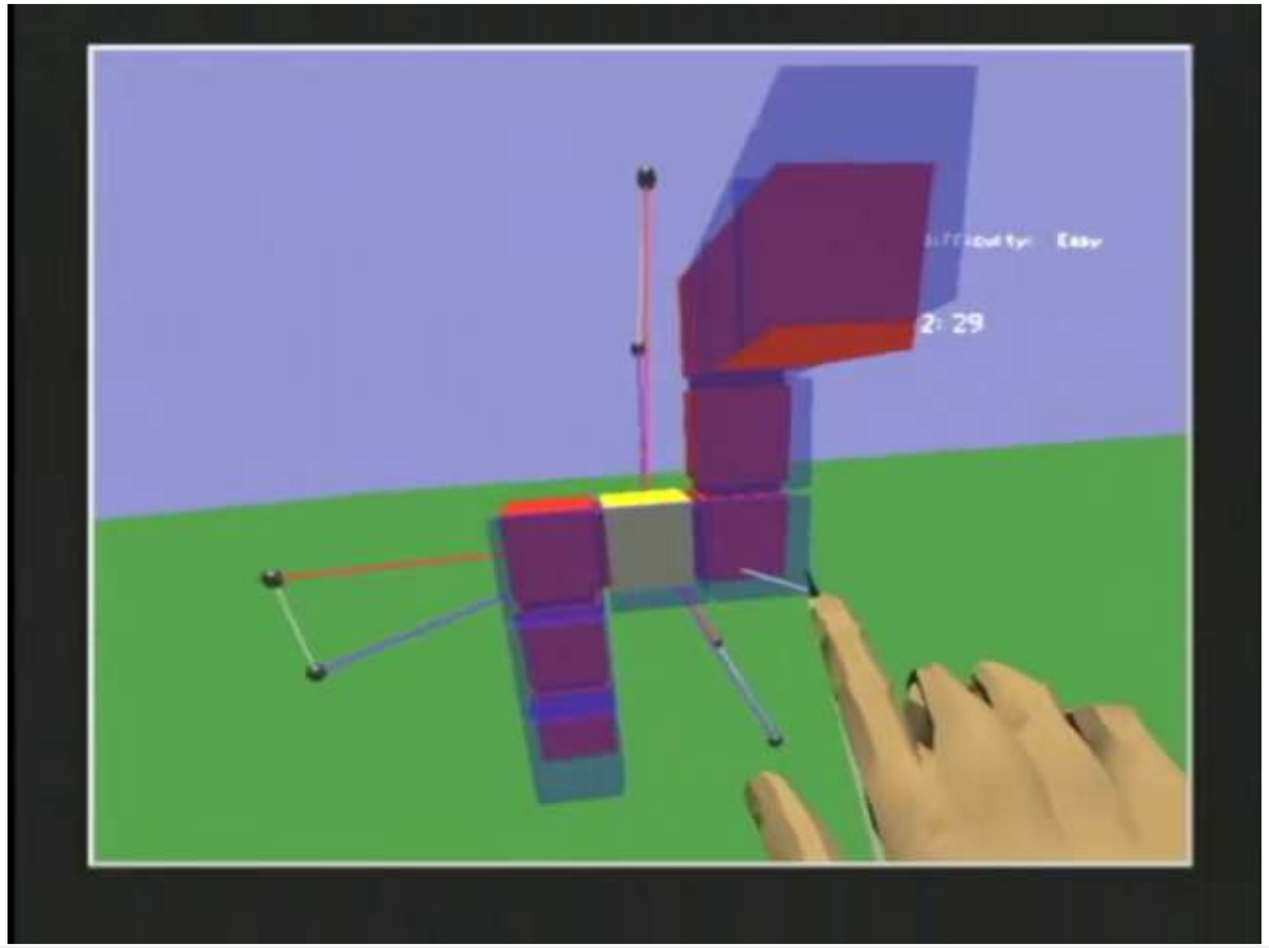
a. User moves hand slowly to the right and down. Some movement is scaled down in the horizontal direction, nearly all is scaled down in the vertical.

b. User moves in the same direction as in (a), this time quickly. Since interaction is in direct mode, object follows hand to its exact location.

c. User slowly moves to the right and up back towards object, vertical offset is recovered. Scaled motion performed in horizontal direction.

d. After accumulating offset, user moves hand quickly up and to the right. Offset is eliminated and mode has switched into direct.



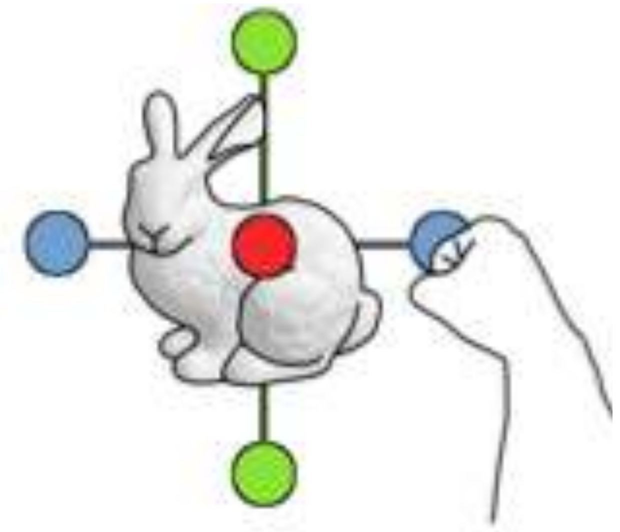


Frees, Kessler, Kay (http://give.ramapo.edu/prism/prism.html)

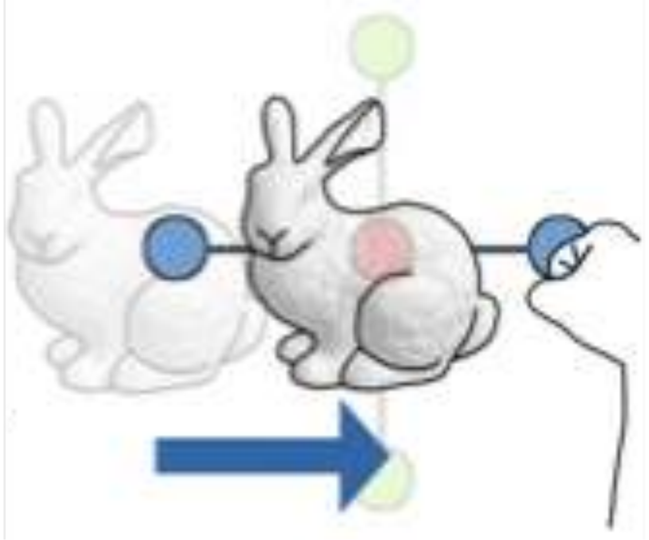
DOF Separation/Reduction by Widgets

- Idea: provide 3D manipulation widgets (with handles) so users know which DOF they are manipulating, and each handle is associated with one DOF only
- Example:
- Results for this example:
  - Comparison for time-to-completion (TTC) between widgets technique (1 DOF) and 6 DOF manipulation (unconstrained, unseparated)
  - For translational tasks: TTC(widget) < TTC(6 DOFs)
  - For rotational tasks: TTC(widget) > TTC(6 DOFs)
  - For both tasks: error(widget) << error(6 DOFs)

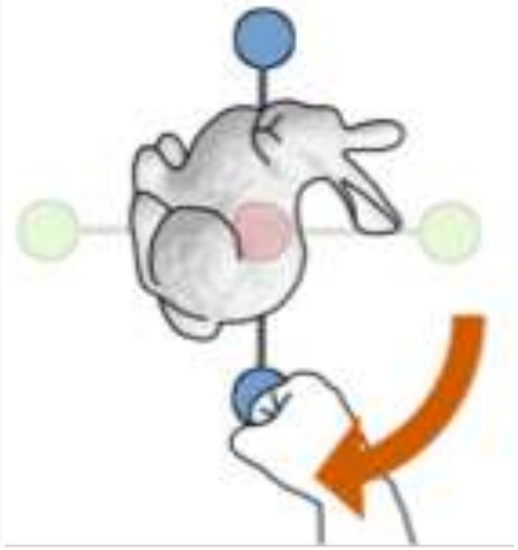
Initial pose



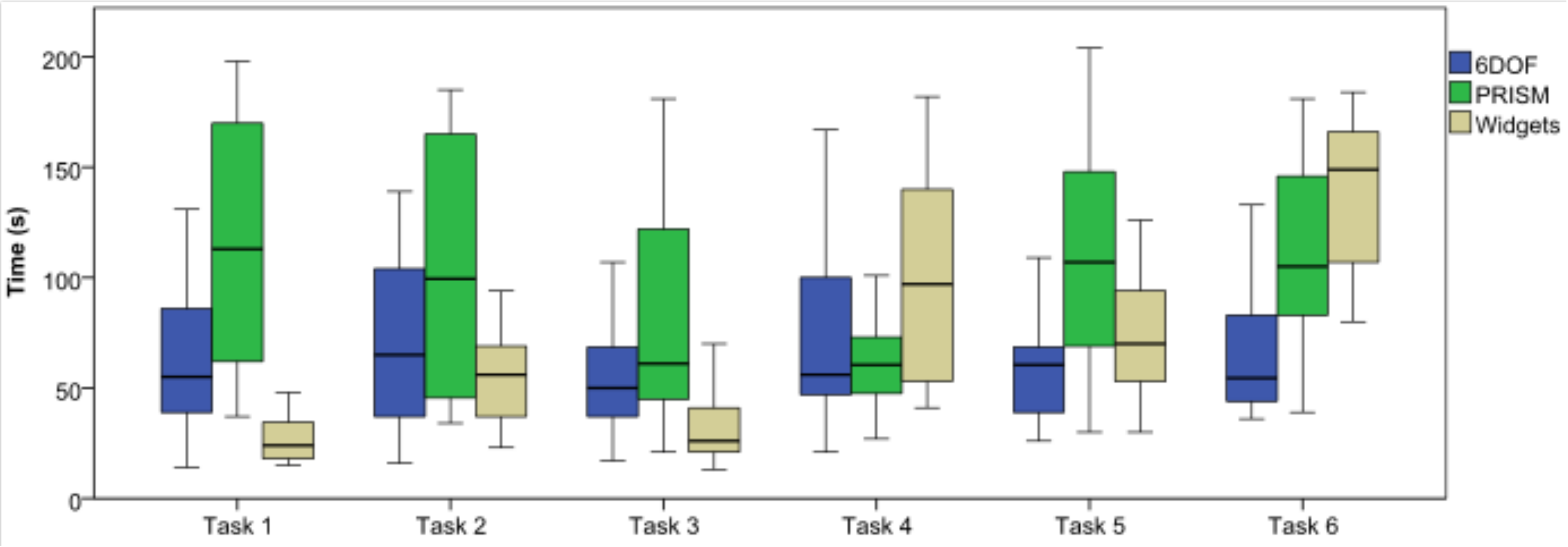
Translation



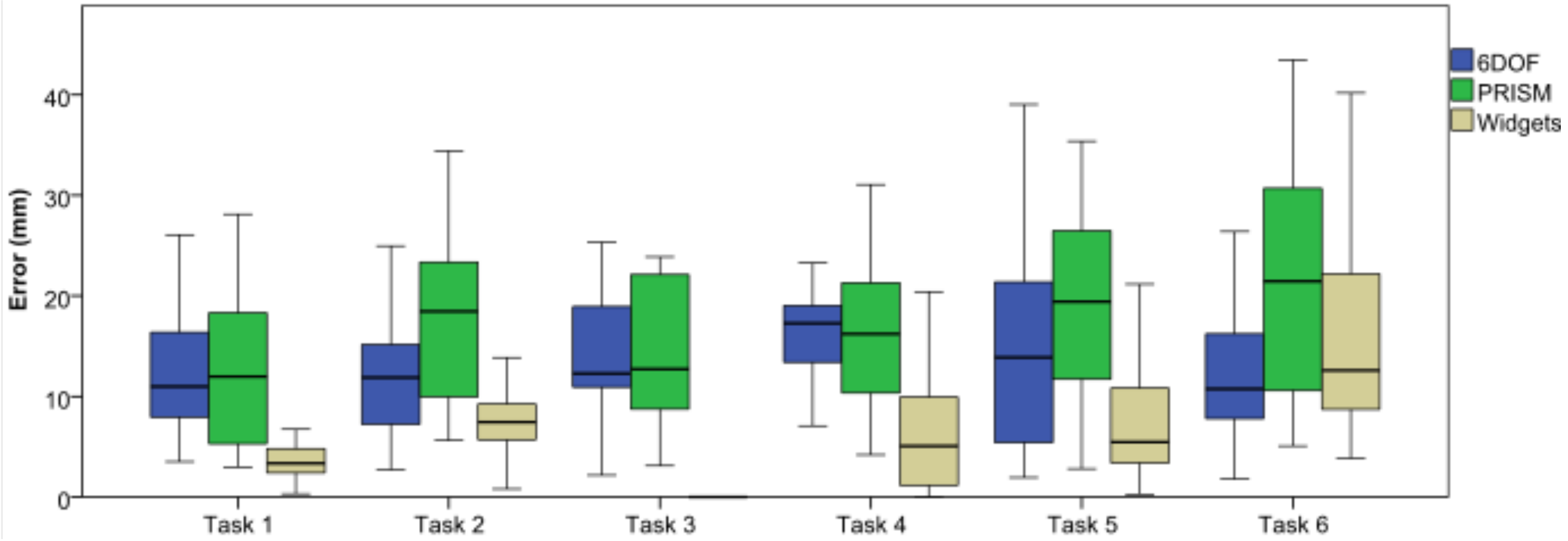
Rotation



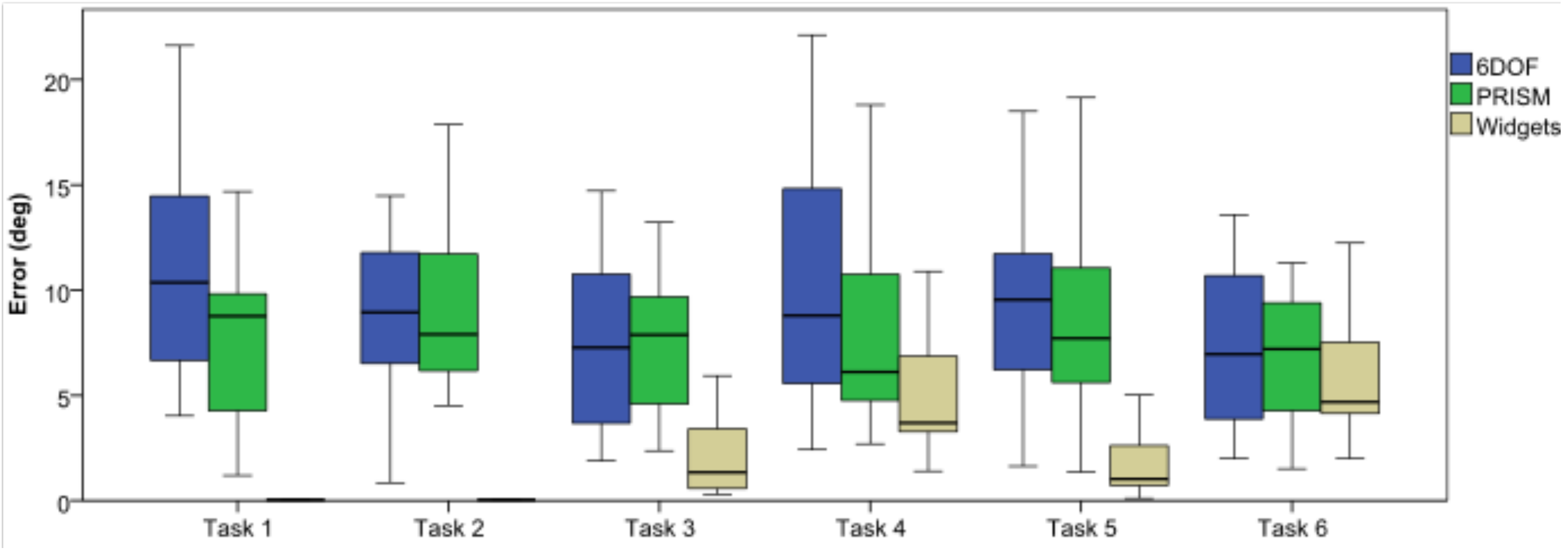
Time-to-completion

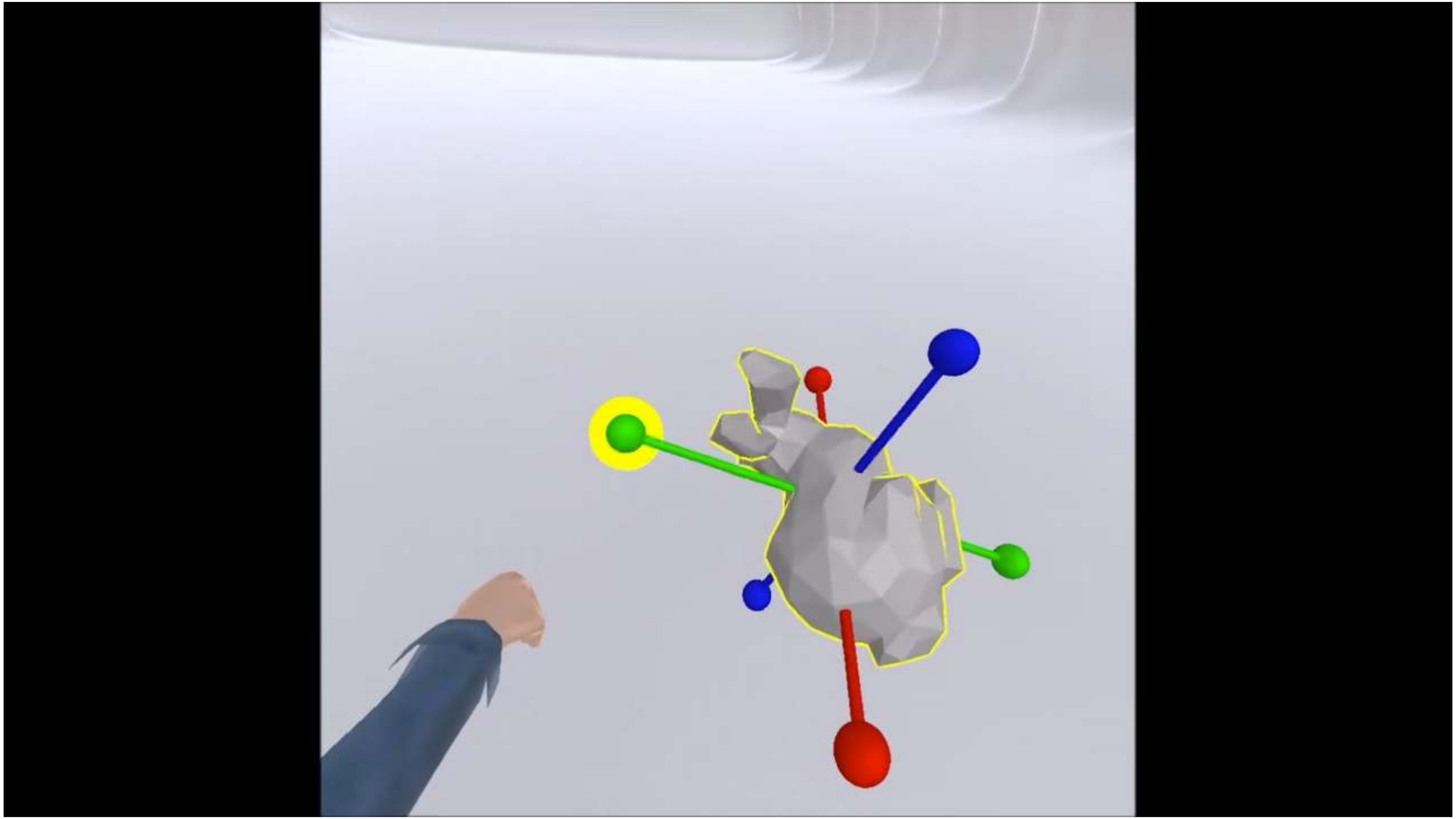


Translational error



Rotation error

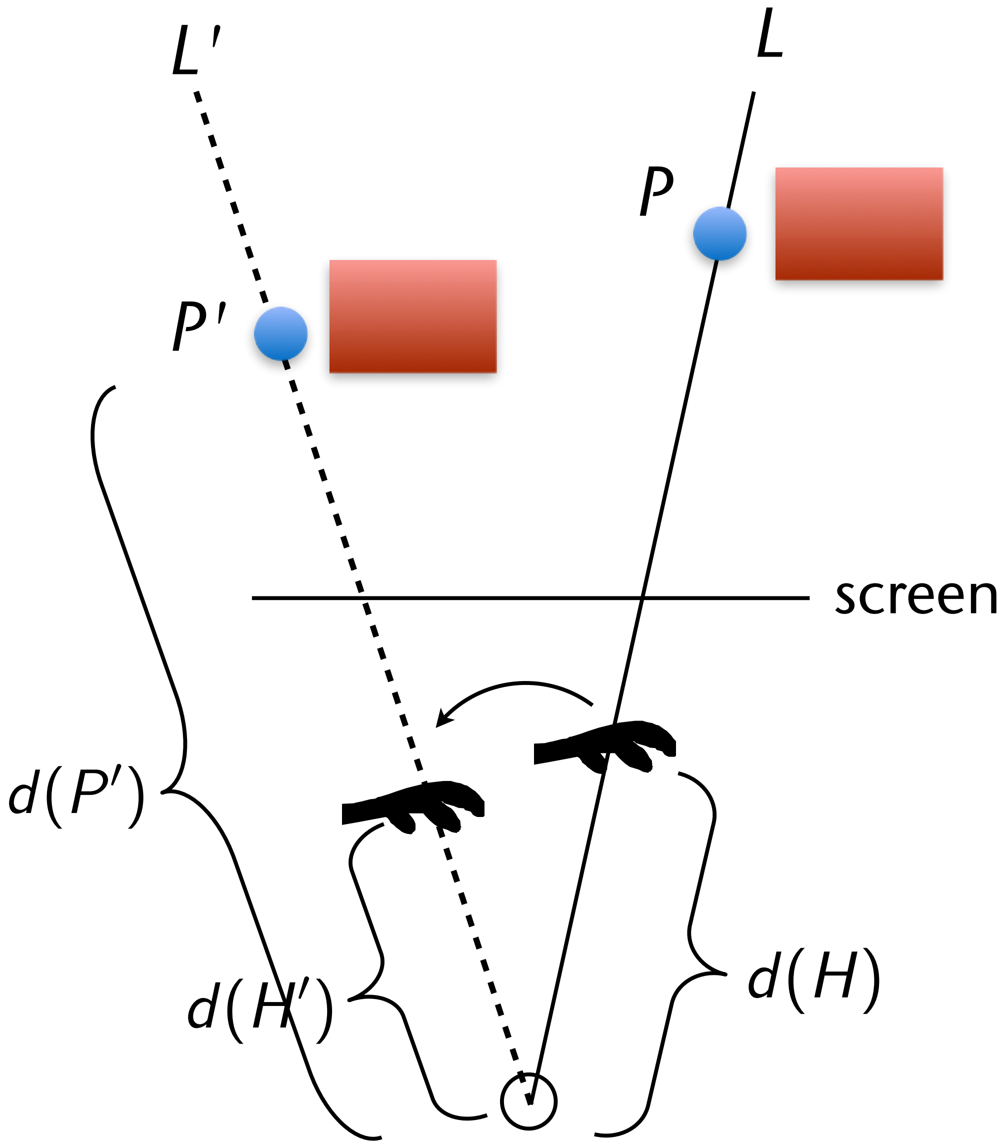




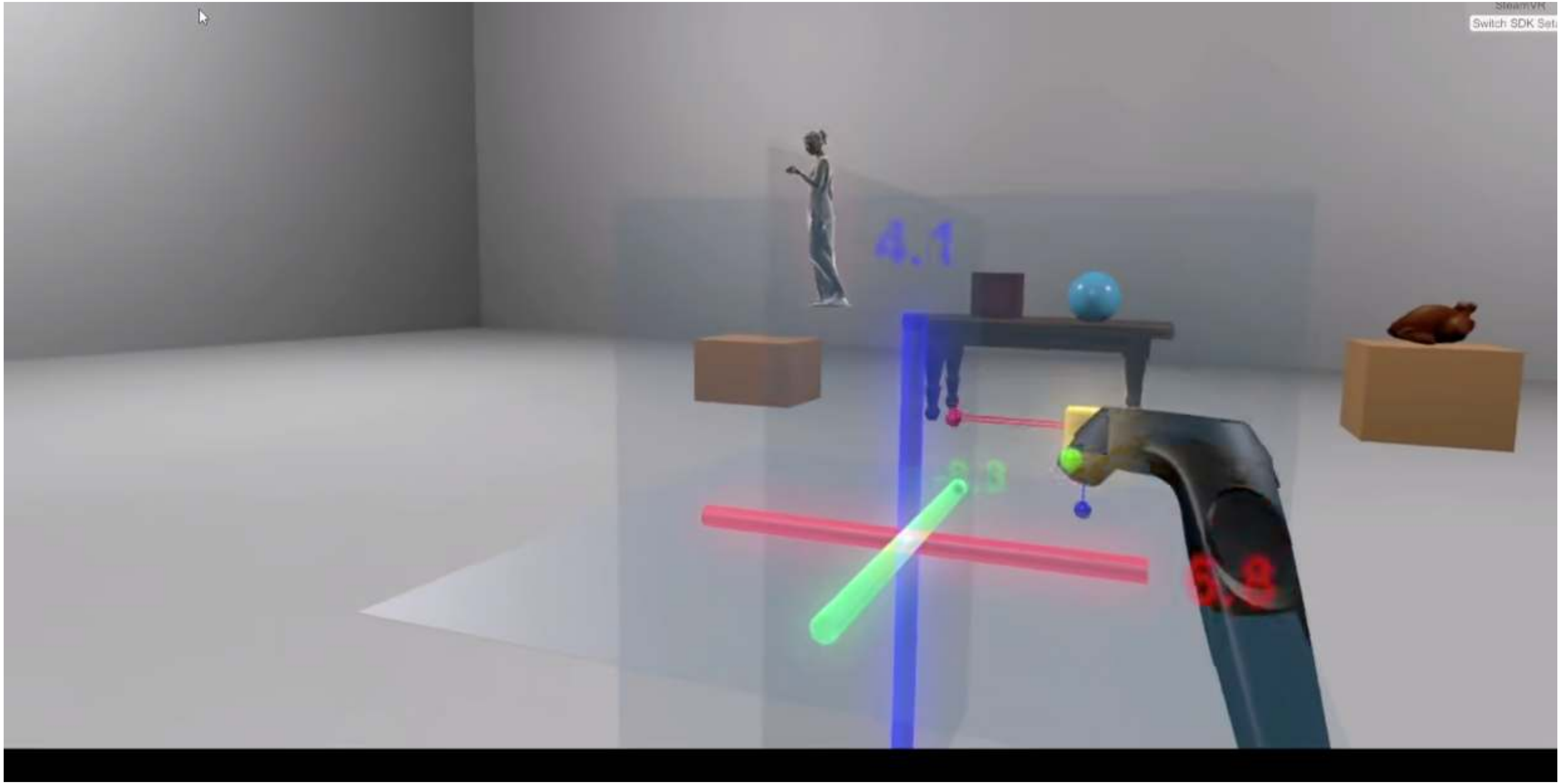
The Action-at-a-Distance Principle

- General VR interaction design principle: perform action in peri personal space, transfer to object at a distance
- Example: move objects in the distance
  - Idea: scale motion of hand such that in screen space (2D) the relative position between the hand and the object does not change

1. At trigger time, finde line *L* through hand
2. Compute point *P* closest to object
3. In the following, compute current line *L'* through hand
4. Compute point *P'*, such that $\frac{d(P')}{d(H')} = \frac{d(P)}{d(H)}$
5. Translate object by $(P' - P)$

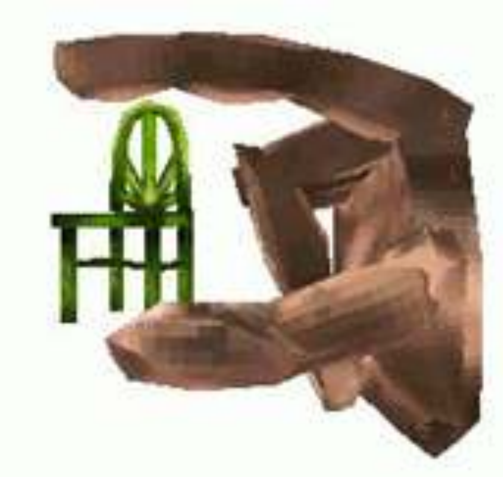


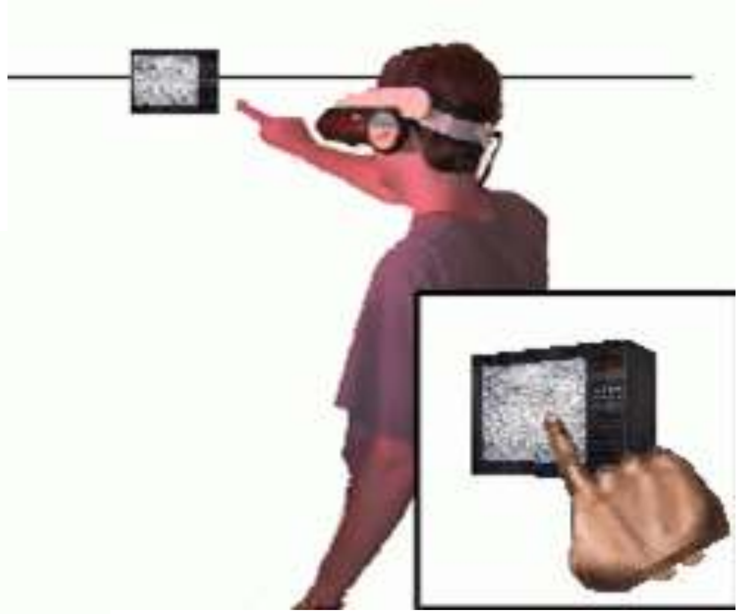
Example of DoF Separation and Action-at-a-Distance for Obj Manipul.

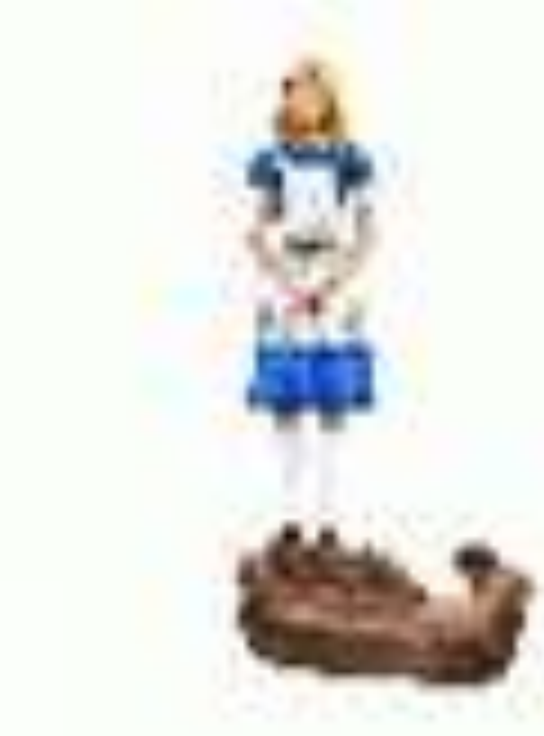


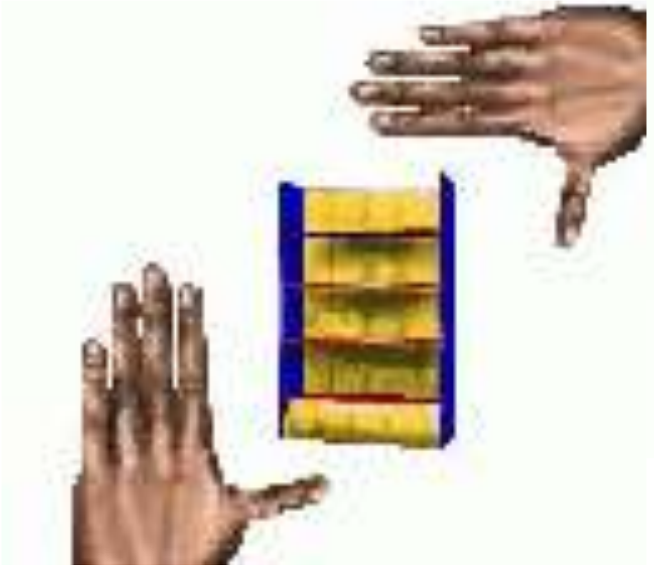
Example: Image Plane Interaction

- General idea: user does not interact with 3D objects, but instead with their 2D image
- "Image plane" selection metaphors:
  - Shoot a ray between thumb and forefinger
  - Shoot ray from eye through fingertip
  - "*Lifting palm*"
  - Frame the object with both hands



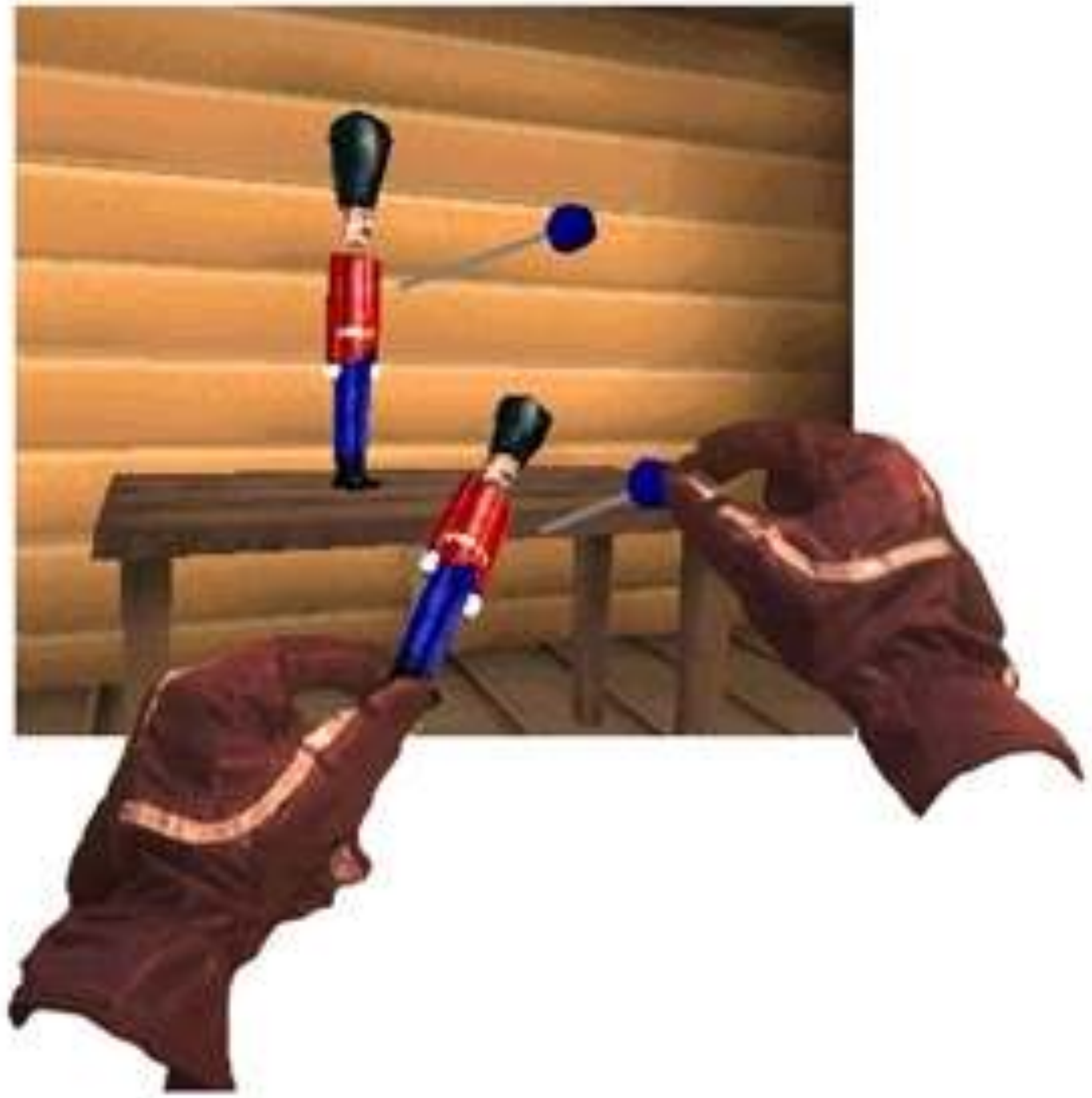






Example: Voodoo Dolls

- Technique for manipulation/positioning of remote objects
- Idea: create a temporary copy (= voodoo doll) of the remote objects
- Task decomposition:
  - Create copy of the reference object (voodoo doll), attach it to the left hand
    - The original of that object will *not* be moved
  - Create copy of the object to be manipulated (here, needle), attach it to the right hand
    - The original of that object *will* be moved
  - Movement of the manipulated object — relative to the *copy* of the reference object(!) — is mapped to the original



- How to create the copy of the object to be manipulated:
  - Use image-plane technique: make pinch gesture "in front" of the object
  - Size of the copy ≈ size of the virtual hand
- How to create a copy of the reference object(s):
  - Use some framing technique (= image-plane technique again)
  - Make copy of all objects within the frame

pinch gesture

copy attached to hand

framing by circle

framing by rectangle

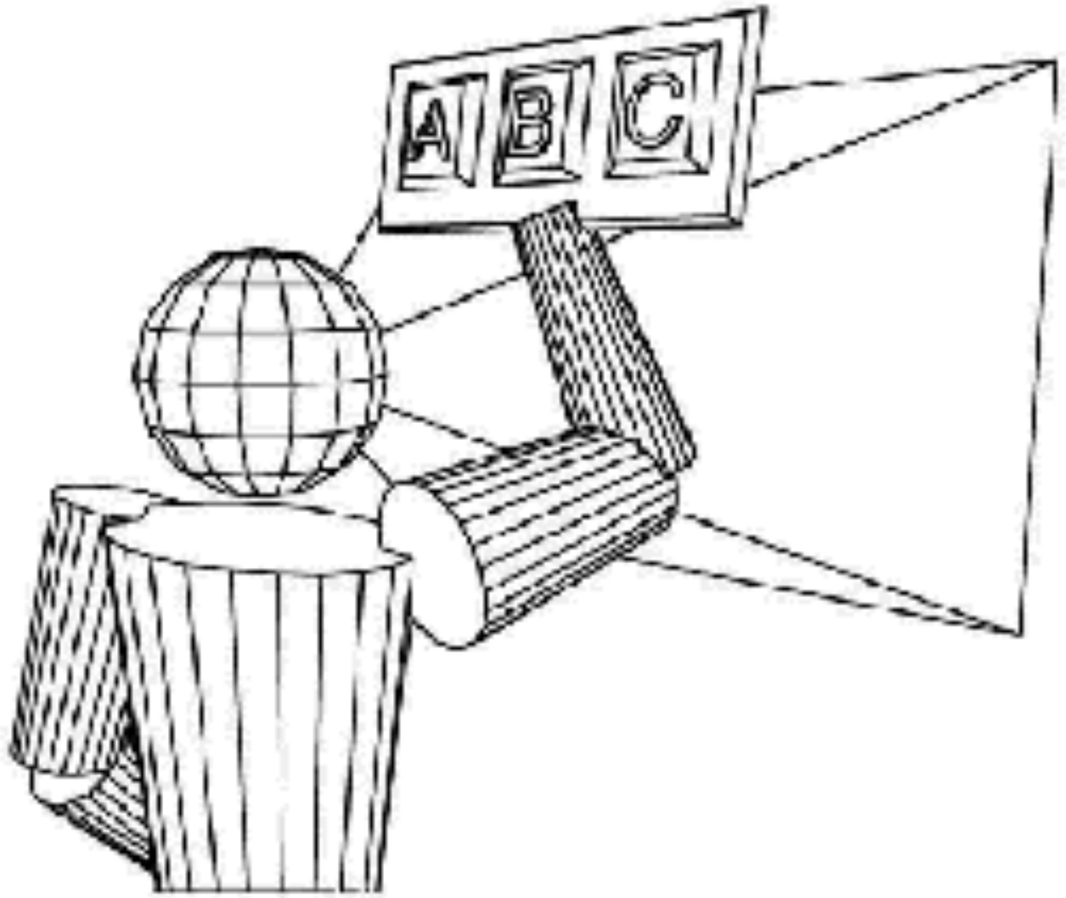
Example of the process of a manipulation

1. User "grasps" table with pinch gesture of the left hand
2. System creates copy, attaches it to left hand, plus some other "context" objects in the surroundings of that object (e.g., telephone and monitor)
3. User "grasps" telephone with right hand (pinch)
4. System creates copy of telephone and attaches it to right hand
5. User puts *copy* of telephone at some other place on the *copy* of the table
6. System maps the translation to the original telephone

- Advantages:
  - Left hand is being used exactly for what it was "designed"
  - User can work on different scales, without having to specify the scaling explicitly
    - The scaling happens implicitly by selection of the reference objects

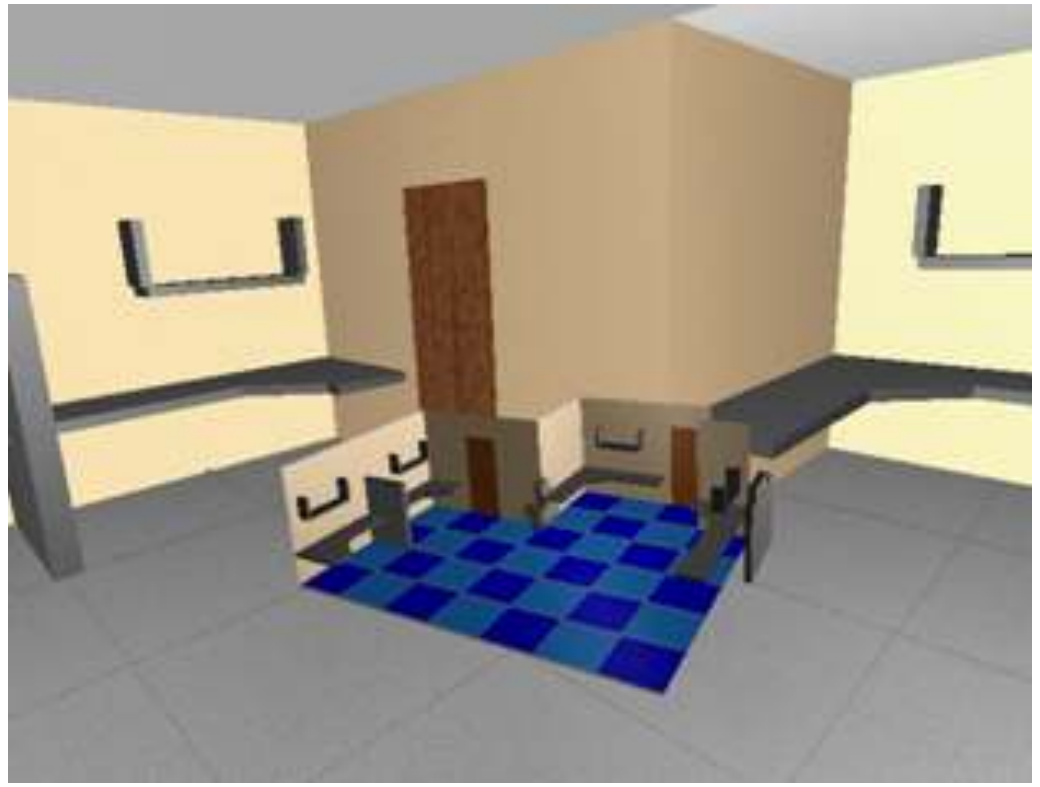
Design Principle: Proprioceptive Interaction

- *proprius* (latin) = *self* (adjective)
- Idea: utilize the fact that humans know exactly where their limbs are, even with closed eyes
- Metaphors derived from that:
  - "Real pulldown" menus: user reaches up, makes grasping gesture, then pulls her hand down → menu appears
  - Deleting objects: grasp object, throw over the shoulder



Design Principle: World-in-Miniature

- The metaphor:
  - Use a 3D miniature "map" (analogous to 2D mini-maps) ⟶ World-in-Miniature (WIM)
  - All interactions in and with the WIM are mapped to the "real" VE
  - Attach the WIM to the non-dominant hand
- Object manipulation = grasp & move the miniature object in the WIM
- Navigation = move the frustum in the WIM, or select a point in the WIM

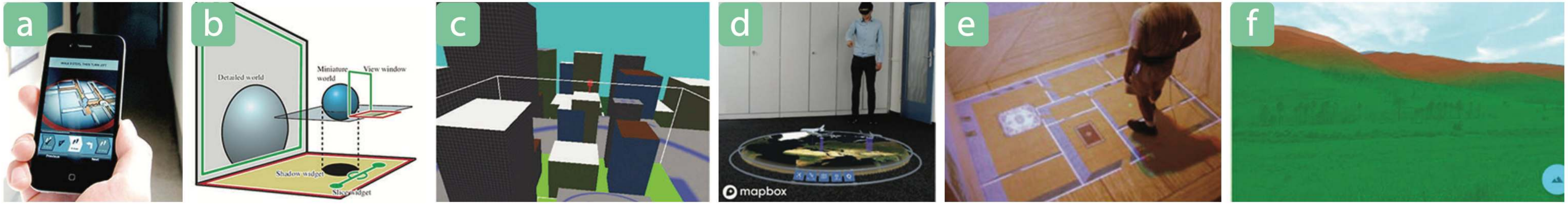


Video

Doug Bowman

The Design Space of WiM's

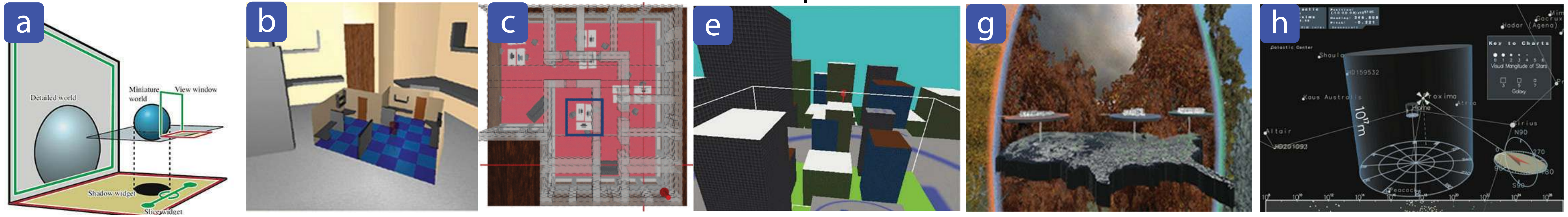
Size:



Very small
< 10 cm

Huge
> 2 m

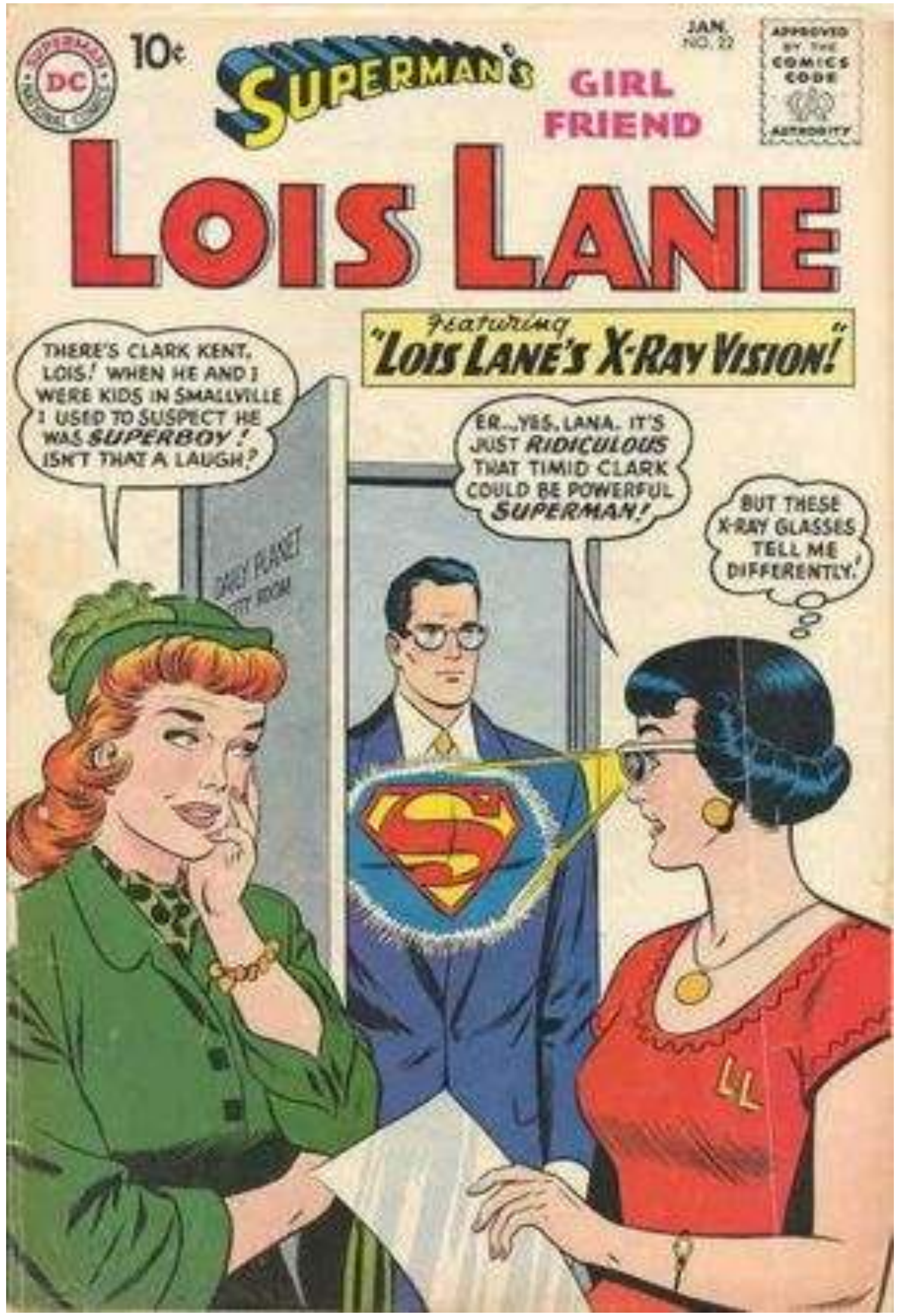
Scope:



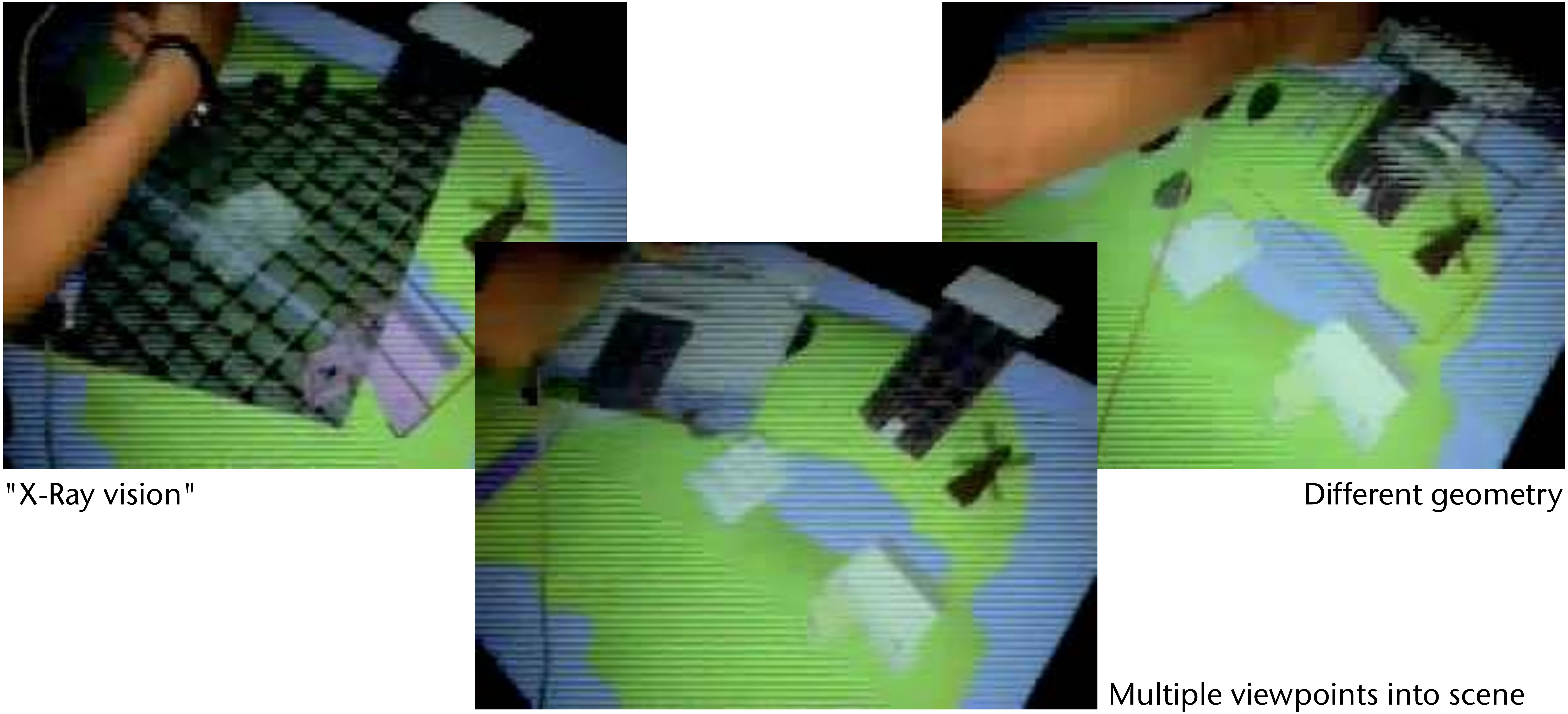
Other dimensions include: abstraction level, placement in VR/AR relative to user, single/multiple, ...

Design Principle: Magic Lenses

- Idea: user sees a different version of the VE through the magic lens
- Where "different" could mean:
  - Other rendering parameters
  - Other geometry
  - Another viewpoint, …
- Examples:
  - Wireframe rendering
  - Magnification
  - Additional viewpoints (like magic mirror)
  - Geometry beneath the surface
  - Preview window for *eyeball-in-hand* or *scene-in-hand* navigation
  - "X-Ray vision"


10¢
SUPERMAN'S GIRL FRIEND
JAN. NO. 22
APPROVED BY THE COMICS CODE AUTHORITY
LOIS LANE
featuring "LOIS LANE'S X-RAY VISION!"
THERE'S CLARK KENT, LOIS! WHEN HE AND I WERE KIDS IN SMALLVILLE I USED TO SUSPECT HE WAS SUPERBOY! ISN'T THAT A LAUGH?
ER...YES, LANA. IT'S JUST RIDICULOUS THAT TIMID CLARK COULD BE POWERFUL SUPERMAN!
BUT THESE X-RAY GLASSES TELL ME DIFFERENTLY!
DAILY PLANET CITY ROOM

Examples



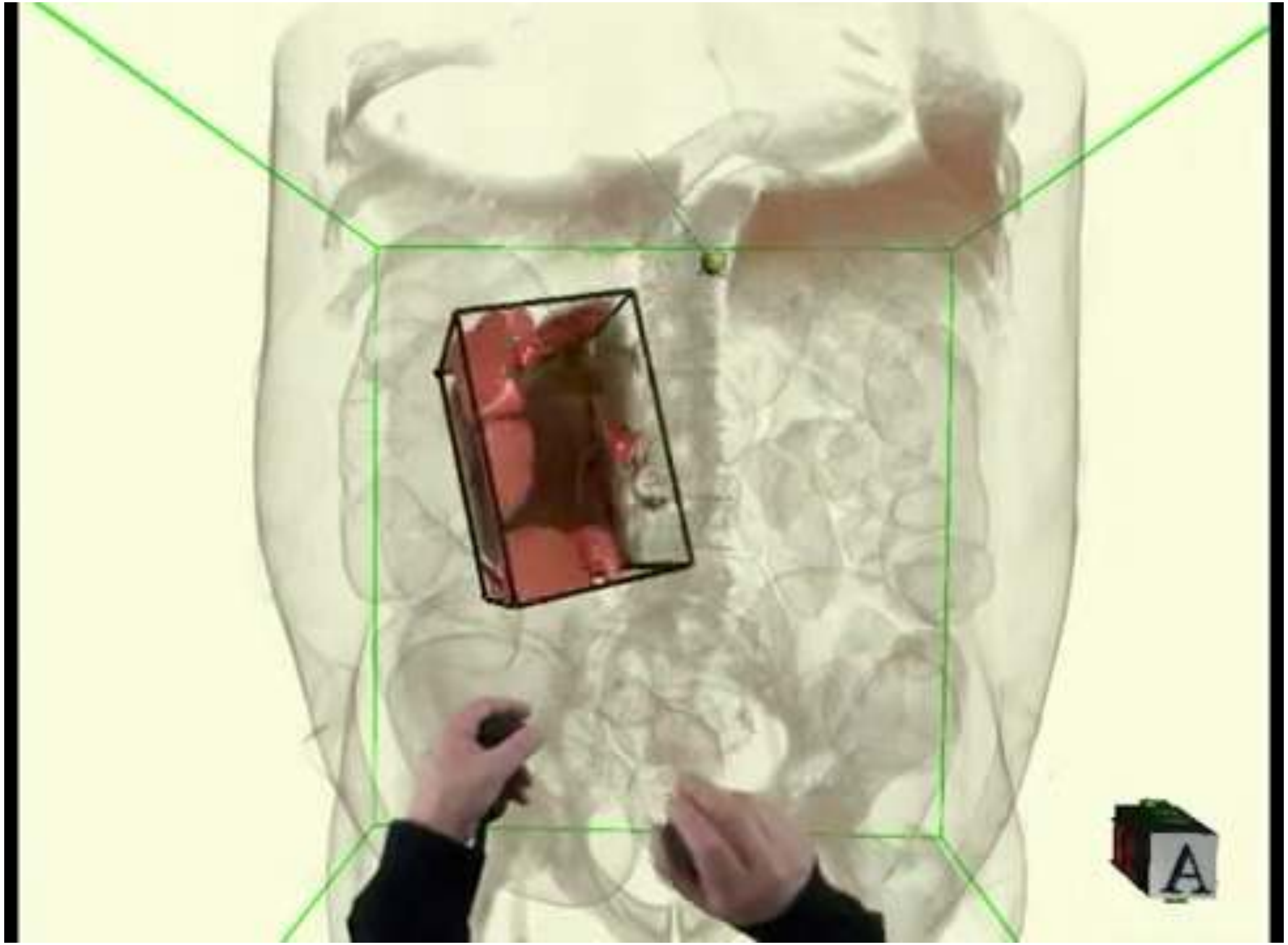
"X-Ray vision"

Different geometry

Multiple viewpoints into scene

Magic Lens for Medical Visualization

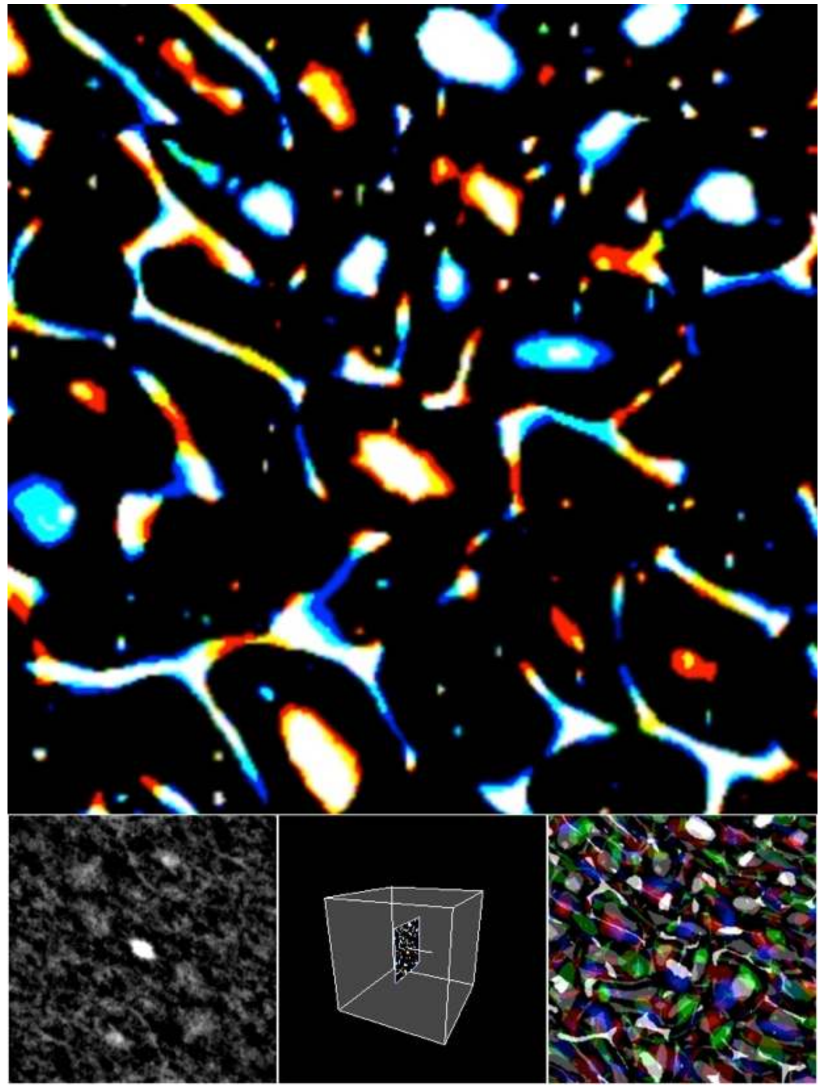
Magic lenses can also be specified as a volume (called a "filter box" here)

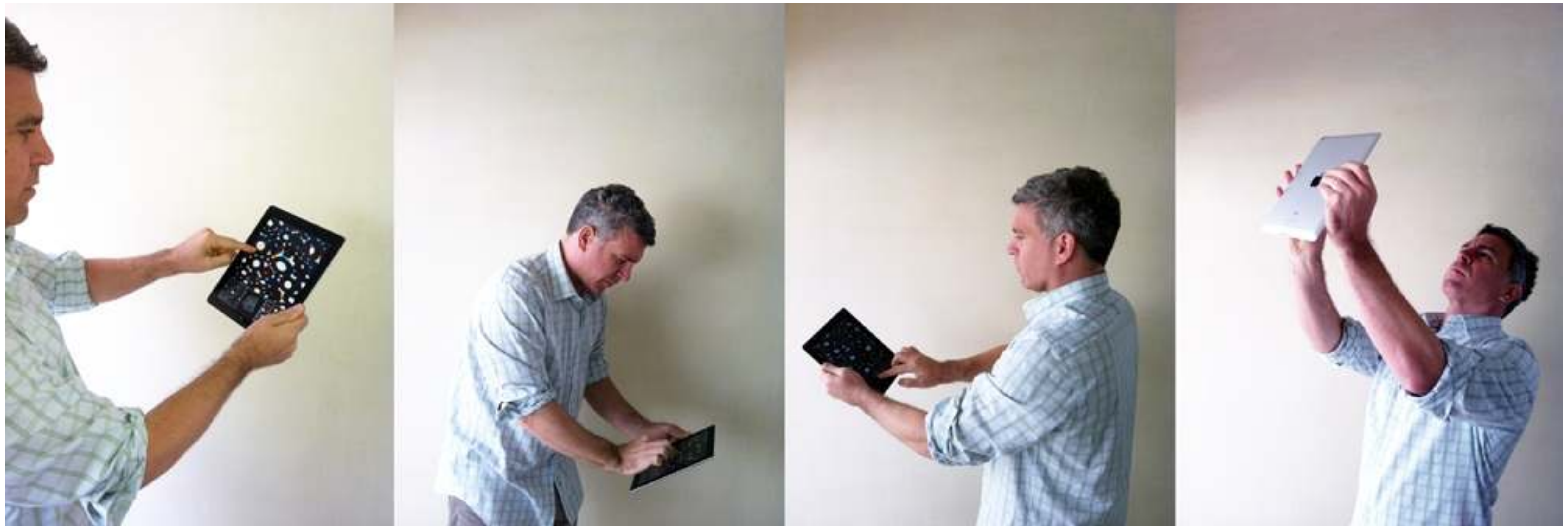


Digital ArtForms' iMedic

An Application in Scientific Visualization

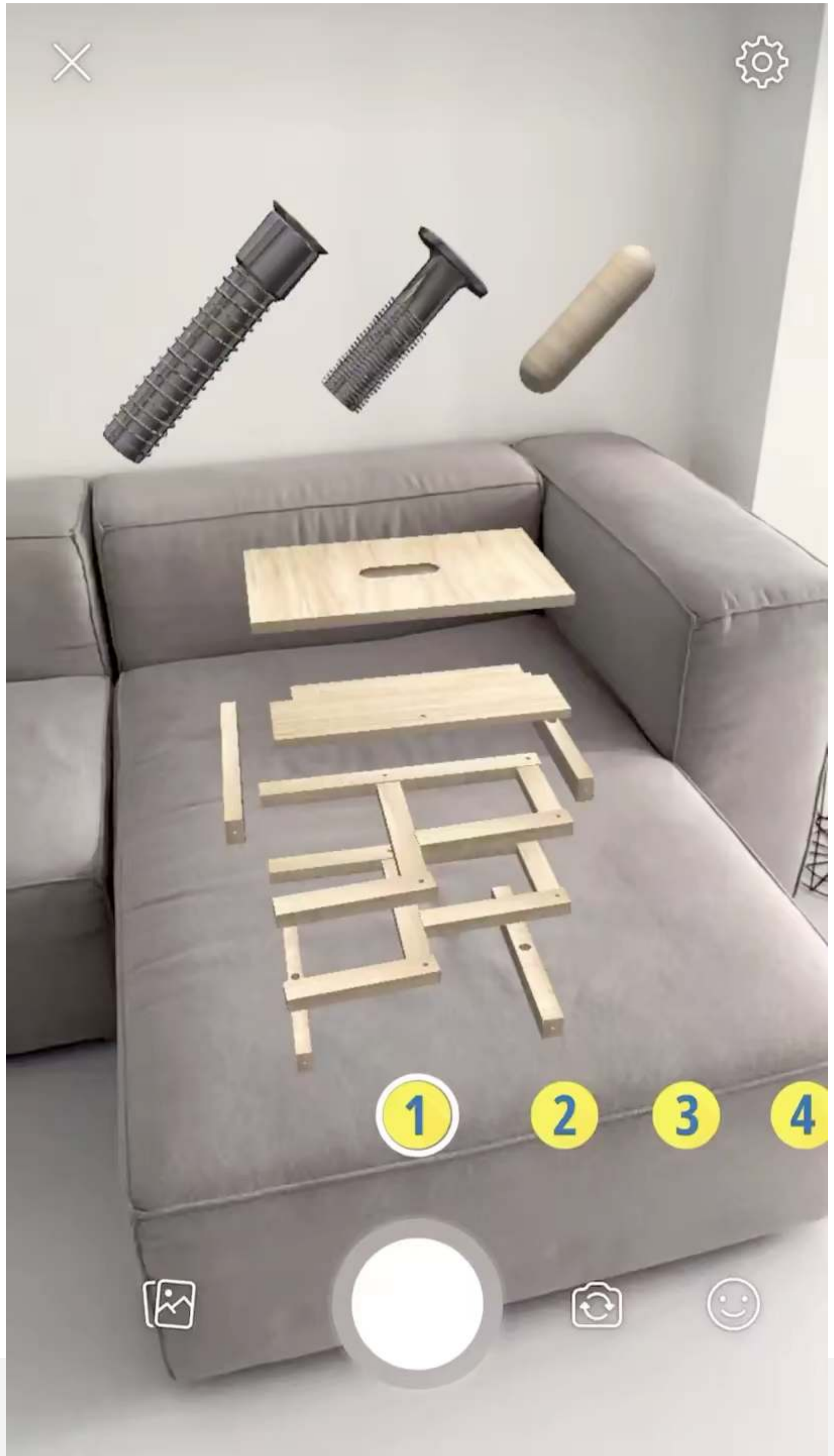
- Goals:
  - Visualization of volume data on an iPad (here, CT's)
  - Intuitive navigation (= specification of the viewpoint)
- Solution: regard the iPad as a "magic lens" into the VE





The Relationship with Augmented Reality

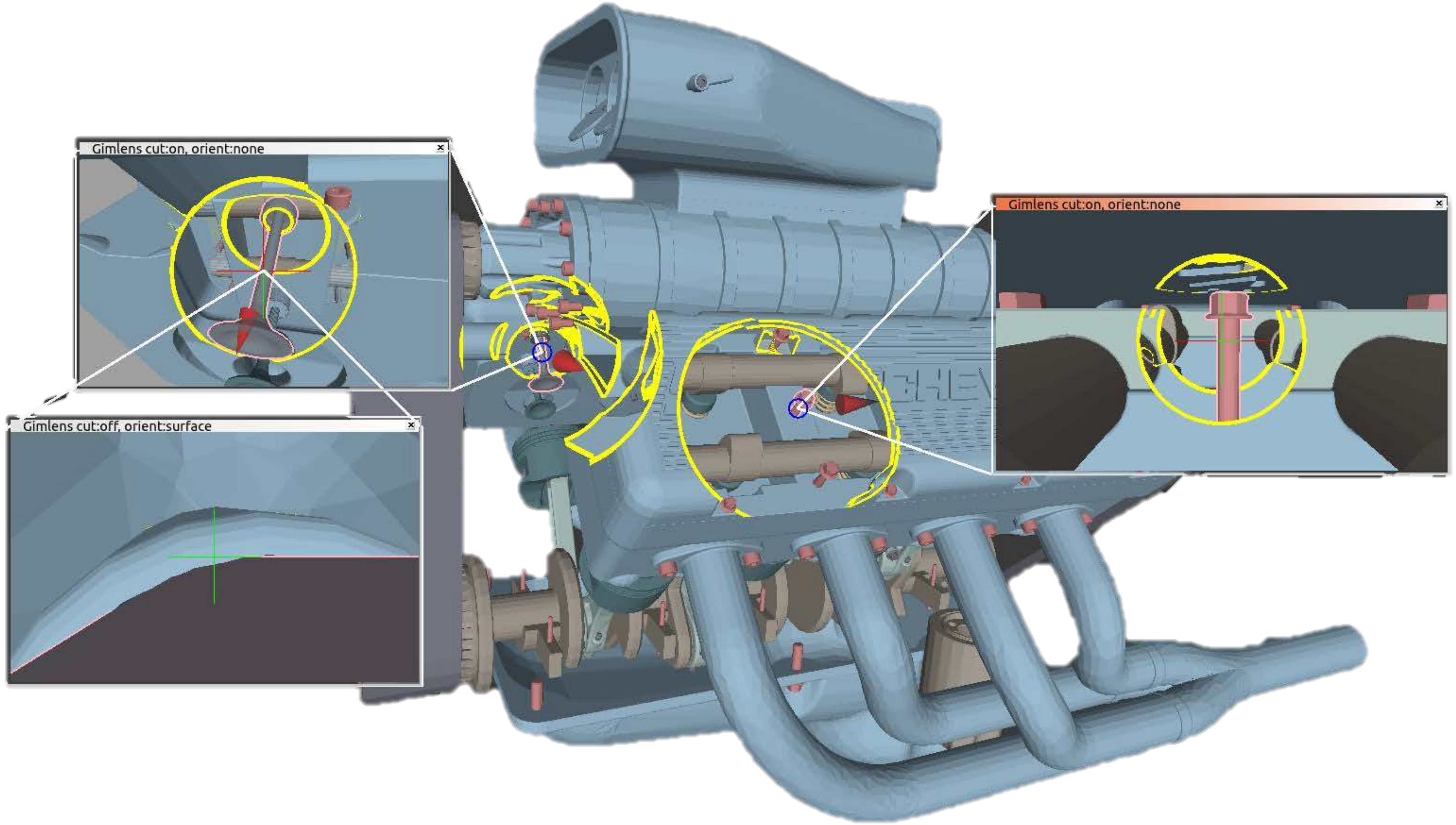
- In a sense, a magic lens is an interaction metaphor that works just like some of the AR apps on tablets

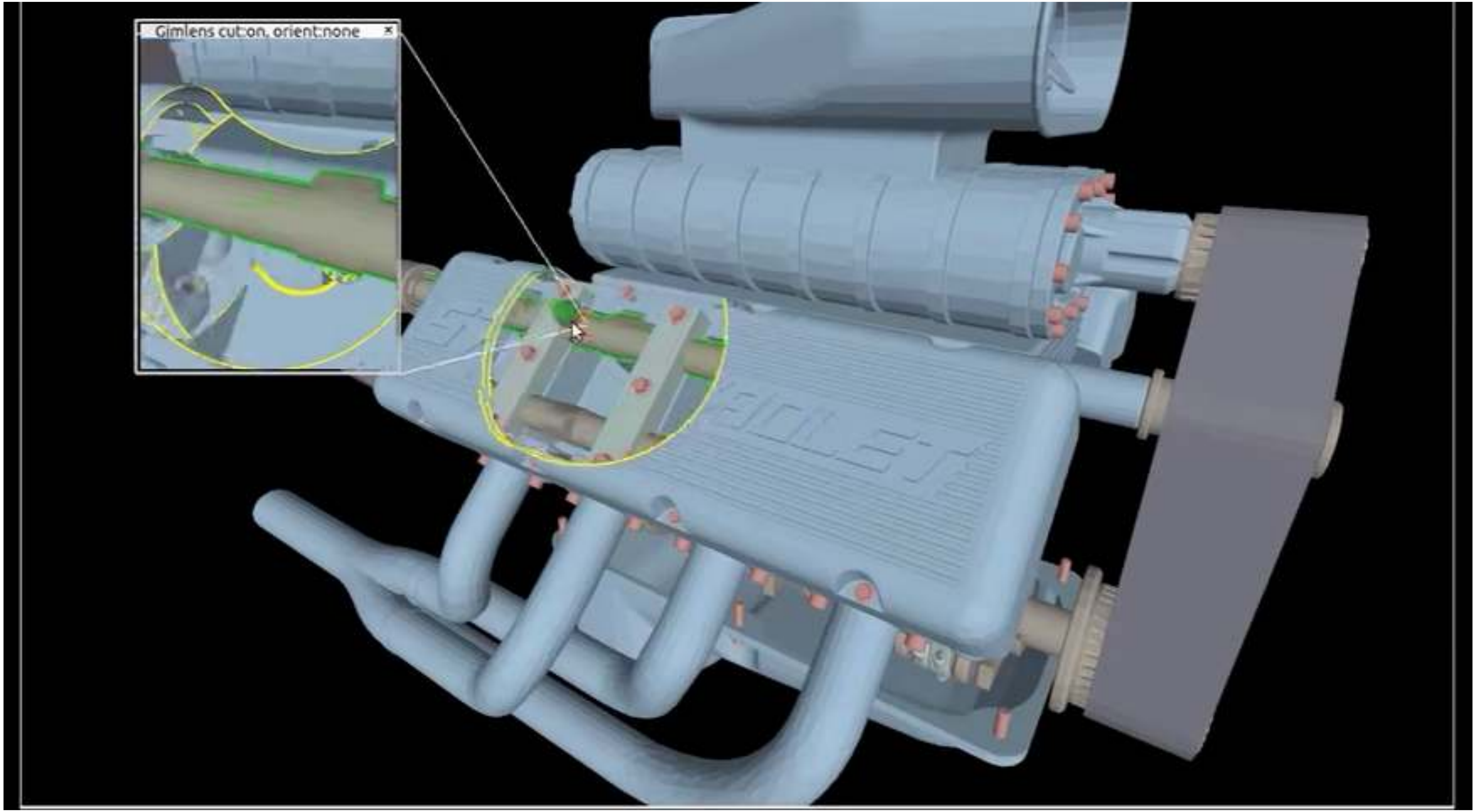


IKEA

Gimlenses: an Application of Magic Lenses in CAD Visualization

- *Gimlens* = magic lens to specify cut-aways
- Cut-away = truncated cone
- Positioning by cone-shaped proxies
- Implementation: fragment shader tests fragments against the cone


Gimlens cut:on, orient:none
Gimlens cut:off, orient:surface
Gimlens cut:on, orient:none



Gimlens cut:on, orient:none
×

Redirecting Users

- General idea: introduce non-isomorphic mapping between user's walking/ reaching/head tracking ind physical space and in virtual space
- The trick is to make any mapping or change in position/orientation unnoticeable to users!

1. Continuous redirection: choose unnoticeable gains that create only small deviations between virtual/real
2. Perform a small rotational/translational "jump" during a saccade (needs eye gaze tracking in HMD)
3. Discrete redirection by making the user reorient through specific events, e.g., barriers or "distractors", e.g., by suddenly appearing objects (sign posts, virtual barriers, birds, ...)
4. Redirection by changing the geometry of the VE

Redirected Walking and Related Techniques

- Walking in real space = very natural, intuitive navigation in VEs
- Problem: virtual space can be much larger than the physical space, in which the user is confined in the real world
- Challenge: how to make large VE accessible without additional navigation metaphors?
- One solution:
  show "fence" ("guard")
  in the VE around the user
  before they hit
  any real obstacles

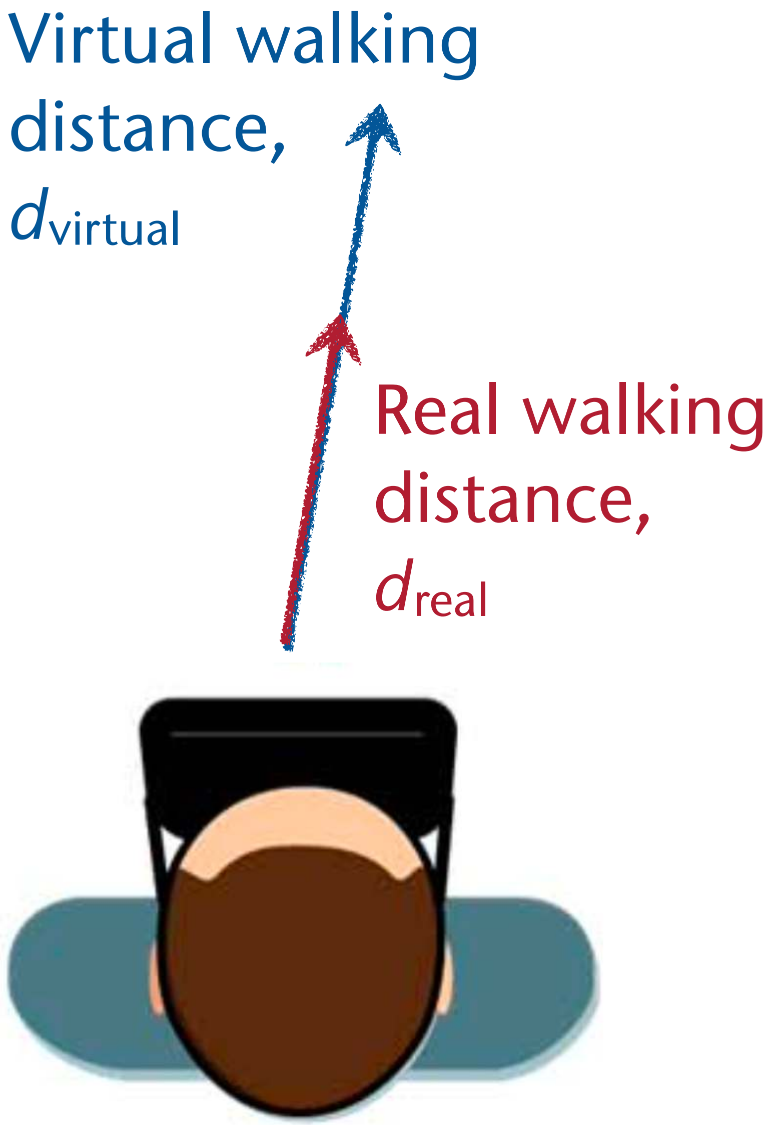
Azmandian, Grechkin, Phan, Bolas, Suma

Re-Direct the User ("cheating" on the user)

- Map *distance* of user's walking in real space to smaller/larger distance of avatar in virtual space ⟶ translation gain

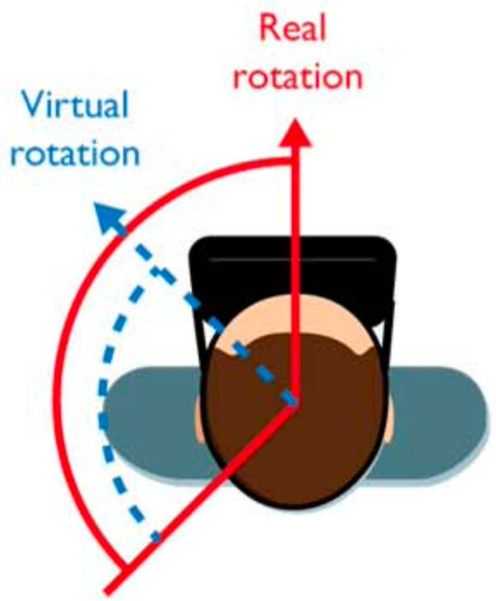
$$g_T = \frac{d_{\text{virtual}}}{d_{\text{real}}}$$

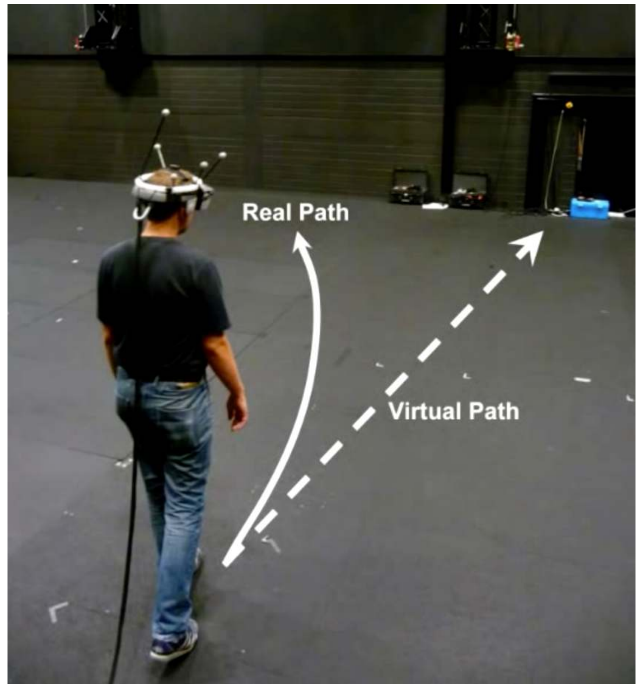
  - (Notice similarity to the C/D ratio)


Virtual walking distance, d_{virtual}
Real walking distance, d_{real}

- Map *angles* of user's turning in real space to smaller/larger turning angles of avatar in virtual space ⟶ rotation gain

$$g_R = \frac{\theta_{\mathrm{VR}}}{\theta_{\mathrm{real}}}$$



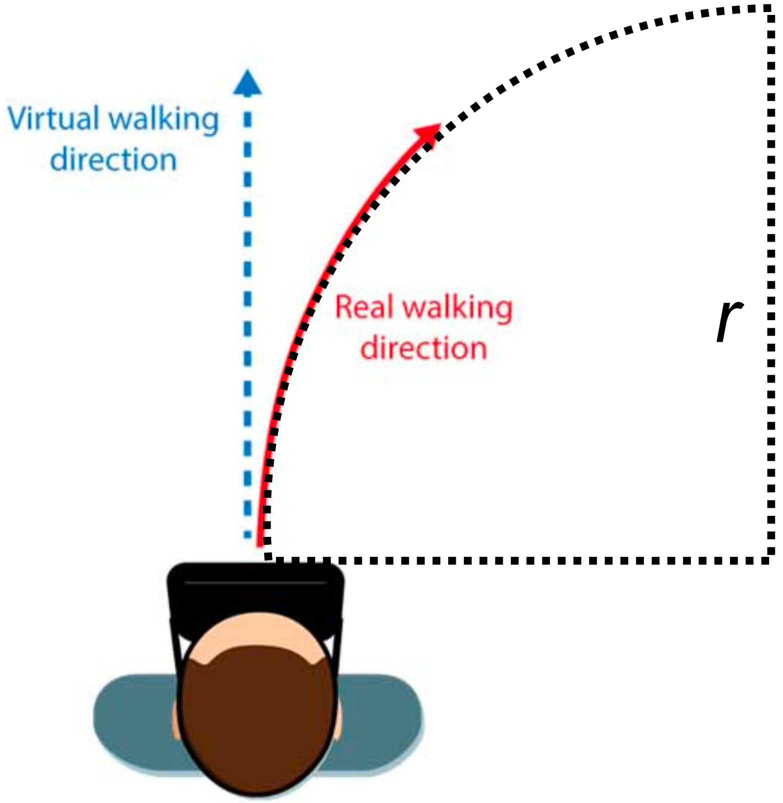


where $\theta_{\mathrm{VR}} / \theta_{\mathrm{real}}$ = rotation angle in VR / real world, resp., and the angle is derived from axis+angle representation of

$$M_t \cdot (M_{t-1})^{-1}$$

where M_t / $M_{t\text{-}1}$ = coordinate frame of the user's head at frame time t / t-1, resp.

- Sometimes, rotational gain is given as curvature gain = curvature of real path, $\frac{1}{r}$, when walking a virtual straight path



Detection Thresholds

- A *translational gain* got detected by most users when

$$g_T \lesssim 0.78 \qquad \text{or} \qquad g_T \gtrsim 1.22$$

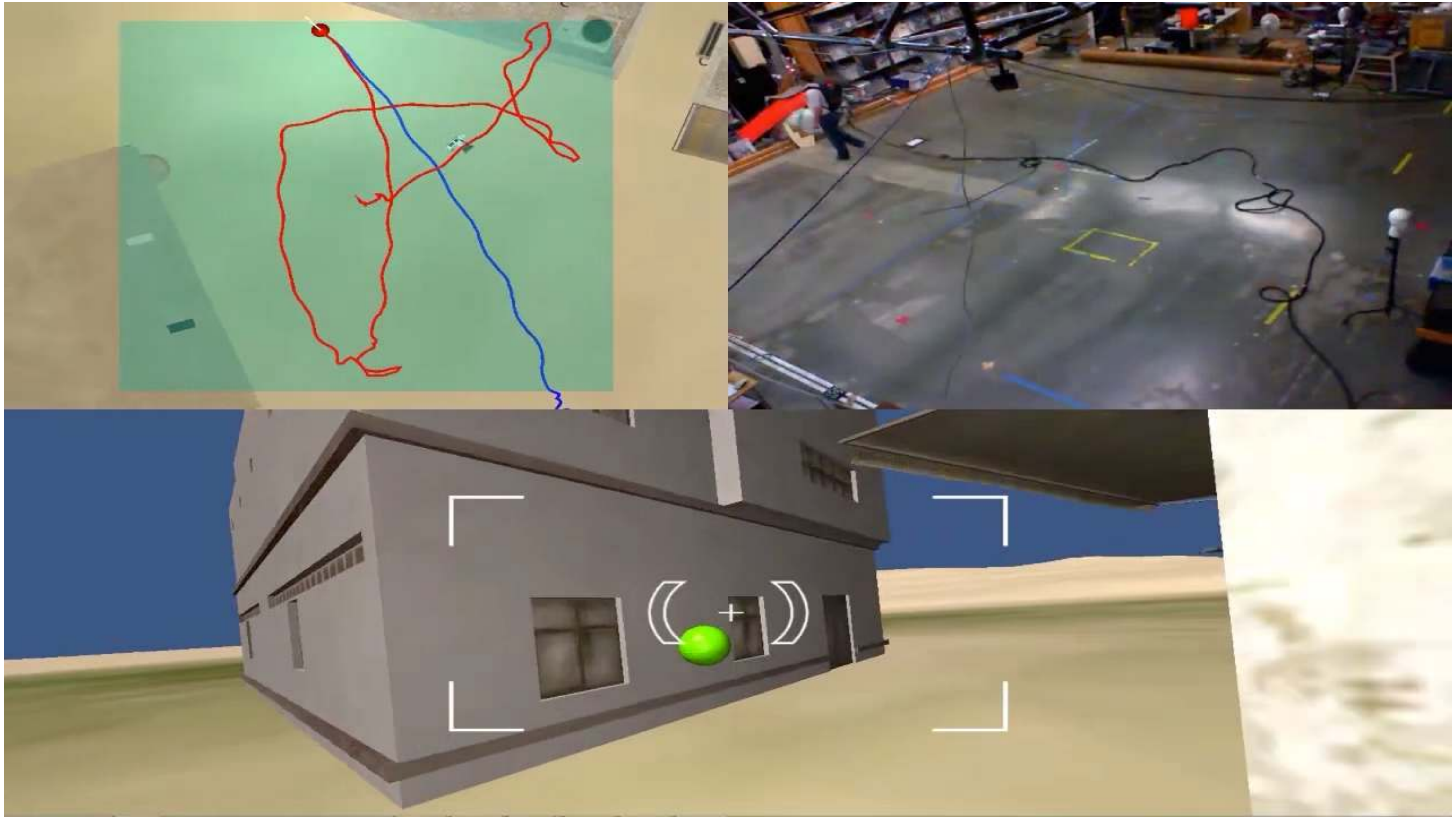
- A *rotational gain* got detected when

$$g_R \lesssim 0.67 \qquad \text{or} \qquad g_R \gtrsim 1.24$$

- A *curvature gain* got detected when

$$r \gtrsim 22\ m$$

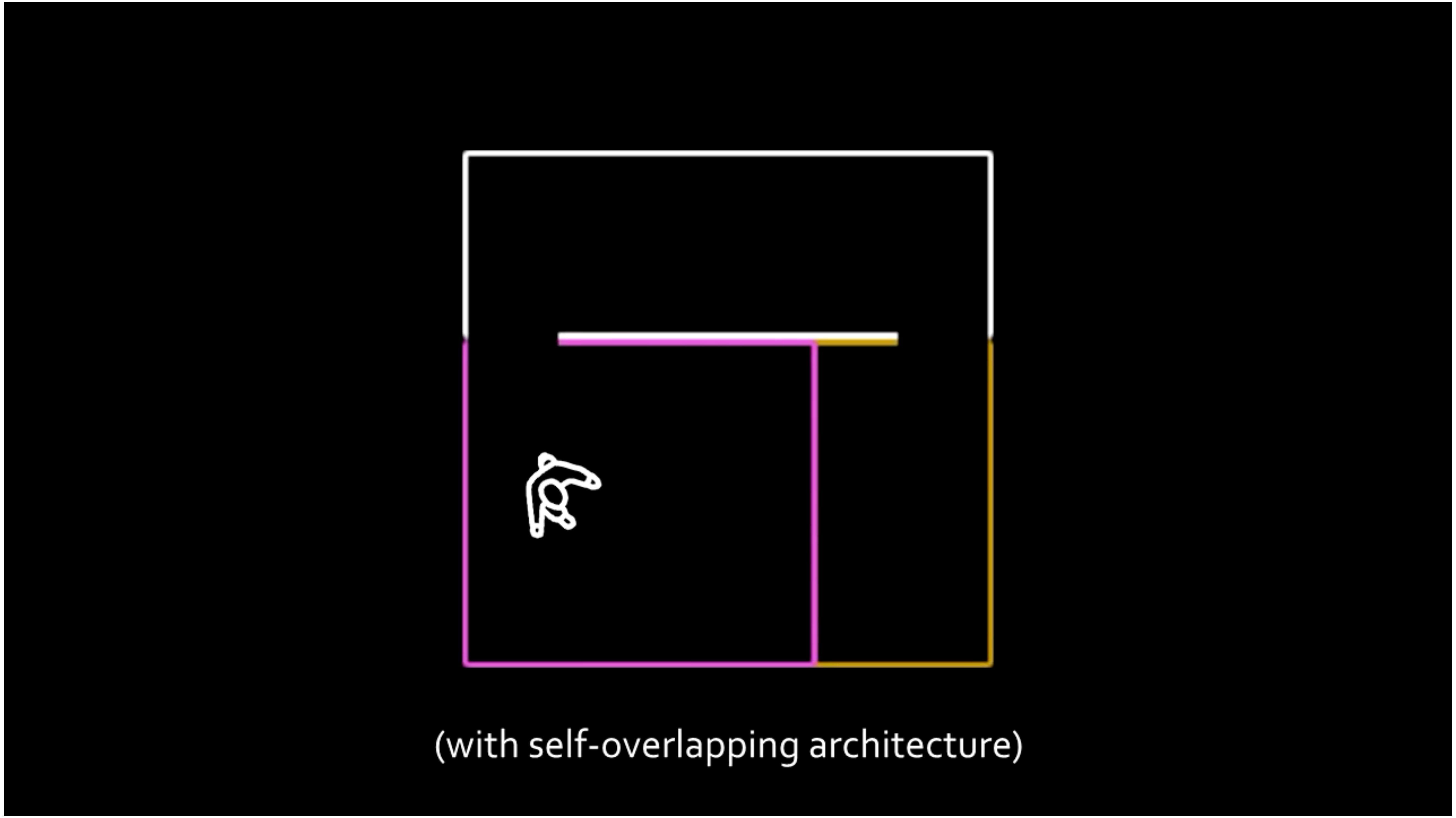
Video



Azmandian, Grechkin, Phan, Bolas, Suma

Related Technique: Spatial Illusions

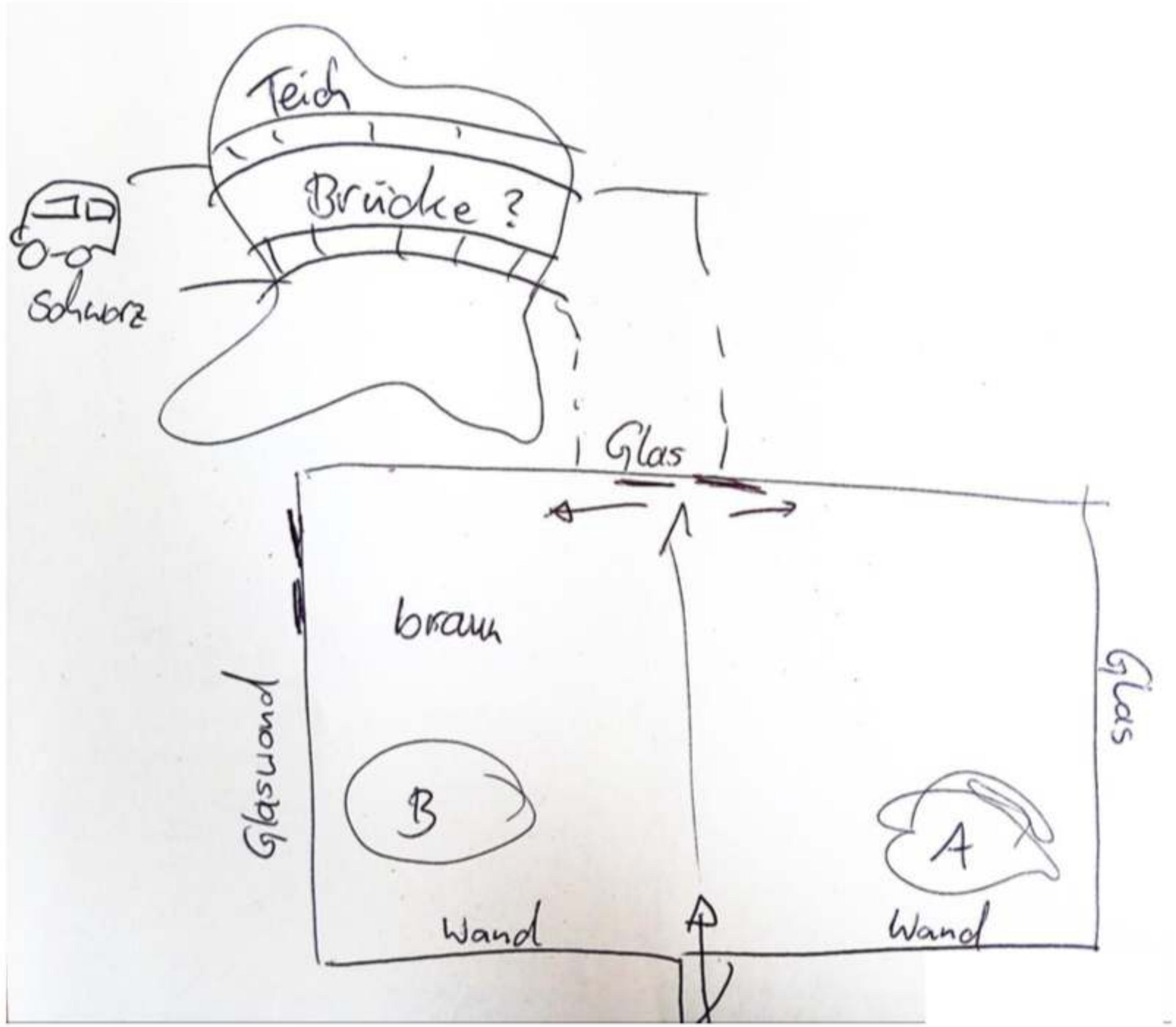
- One possibility: self-overlapping architectural spaces

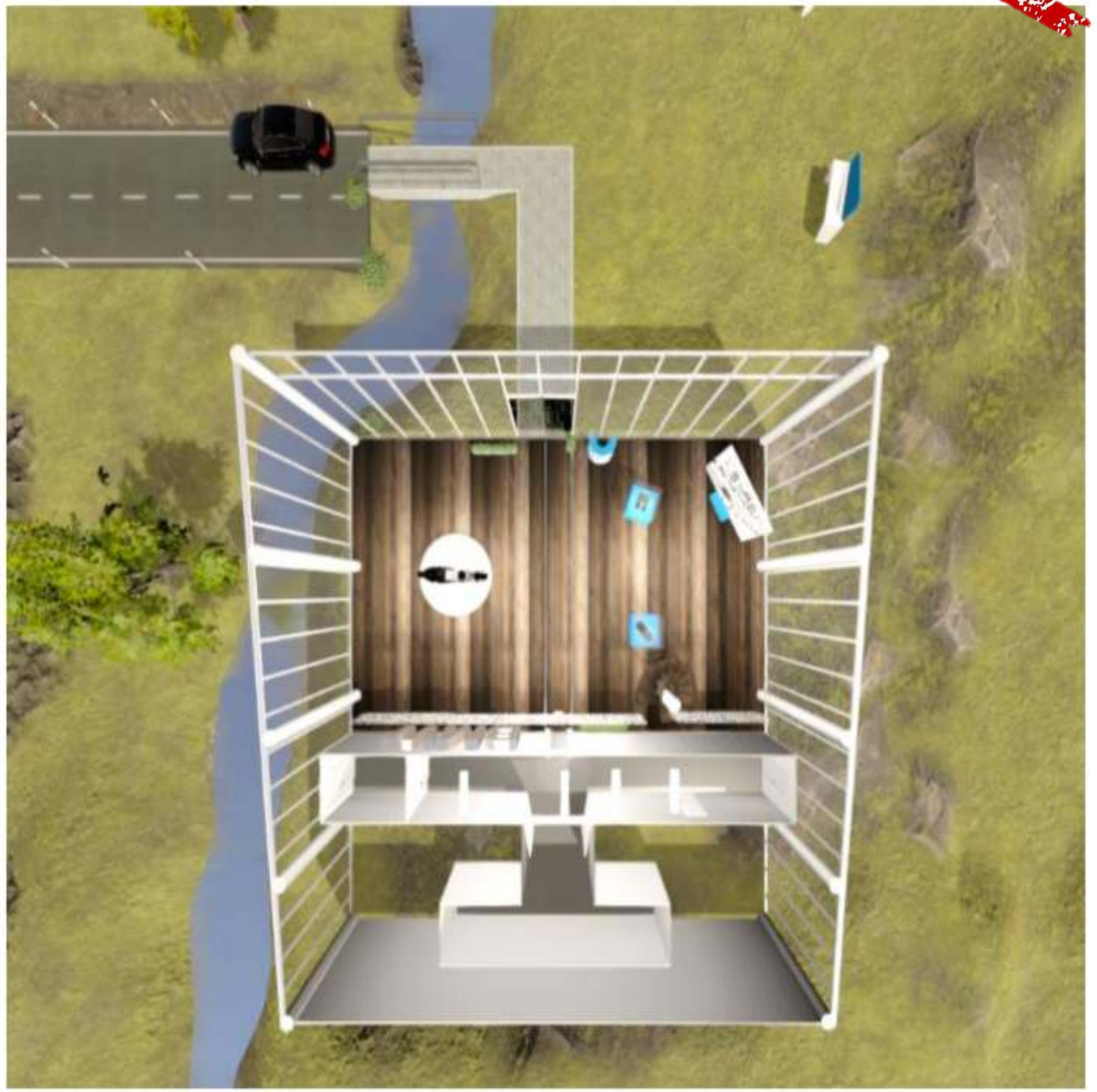


An Idea for Evaluating Redirected Walking

CONFIDENTIAL

- Instead of measuring a JND, make users draw the VE afterwards

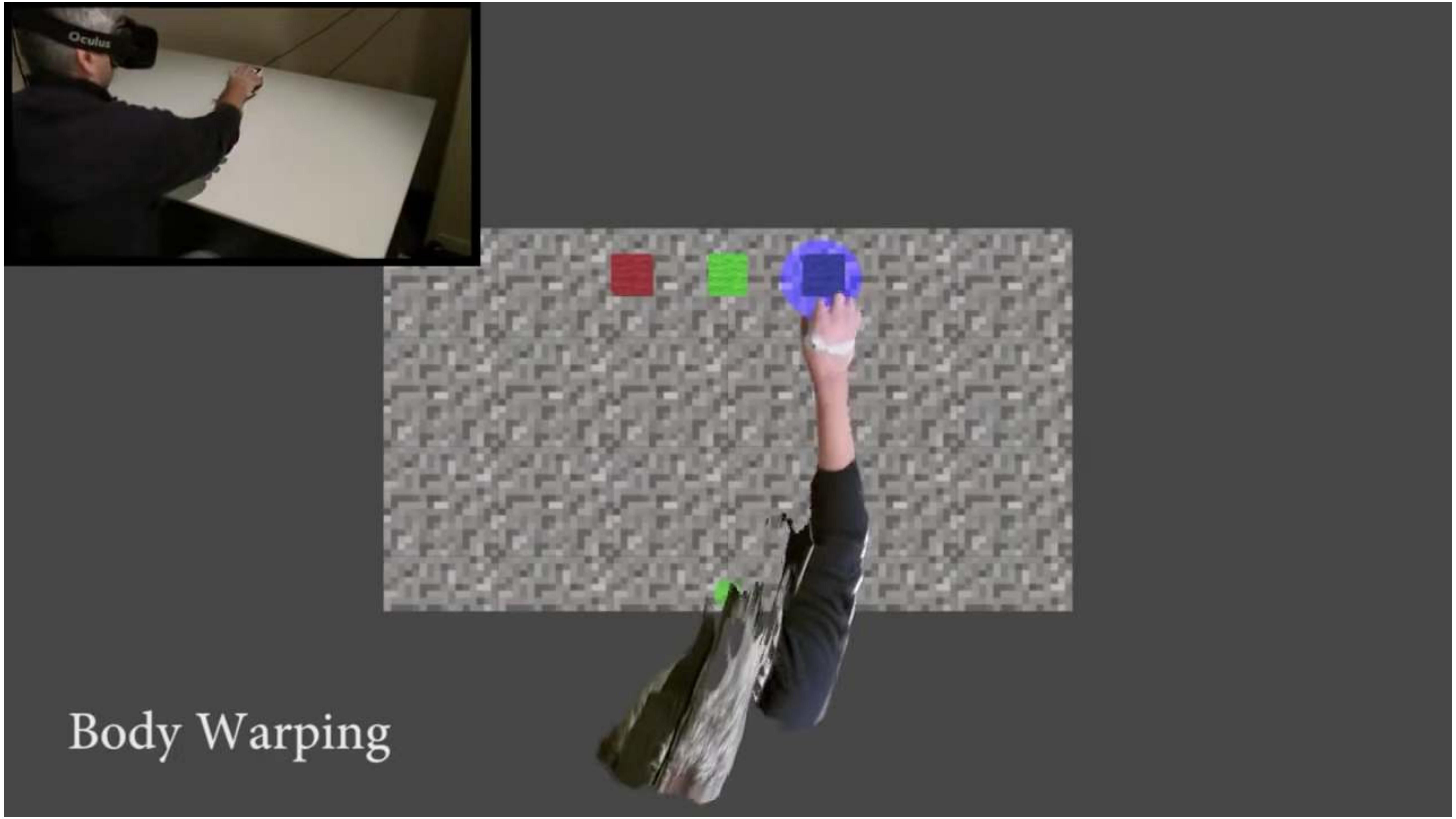




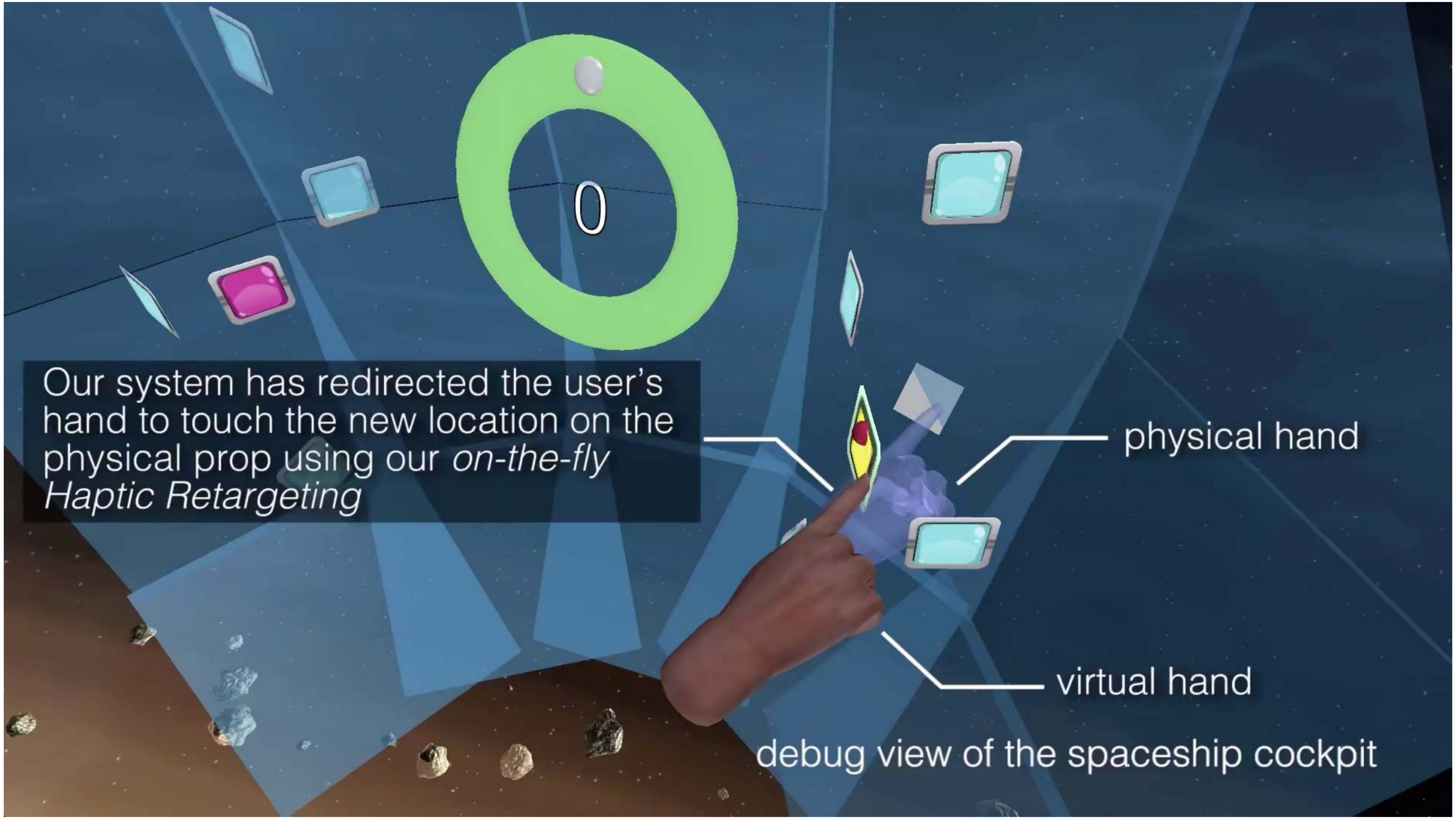
Body Redirection: Passive Haptic Retargeting

- Goal: provide rich passive haptic feedback with minimal physical haptic props
- Setting: HMD, tracked hand, seated user
- Idea: apply redirection techniques to hand location and user orientation
  - "Body warping" = redirect user's hand + slight avatar deformation
  - "World warping" = rotational gain
  - Proposed terminology: redirected reaching

Video



Potential Use: Passive Haptic Feedback



Our system has redirected the user's hand to touch the new location on the physical prop using our *on-the-fly Haptic Retargeting*

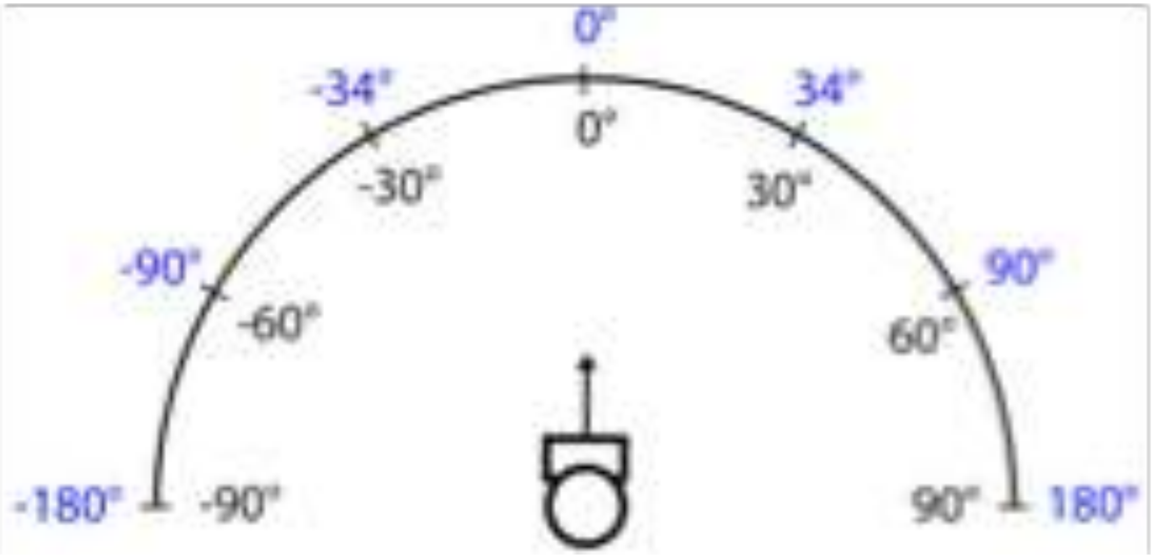
Sparse Haptic Proxy Touch Feedback … – Andrew Wilson et al., Microsoft Research, HPI, CHI 2017

Amplified Head Rotation

- Manipulates C/D ratio for user's head rotation, i.e. rotational gain
  - Remember the Go-Go technique for the user's hands?
- Especially useful for seated VR experience
- The technique:
  - User defines preferred forward direction ⟶ yaw angle = 0
  - Only yaw angle (rotation about vertical axis) is modified
  - Yaw angle (a.k.a. heading) is multiplied by factor (nonlinear):

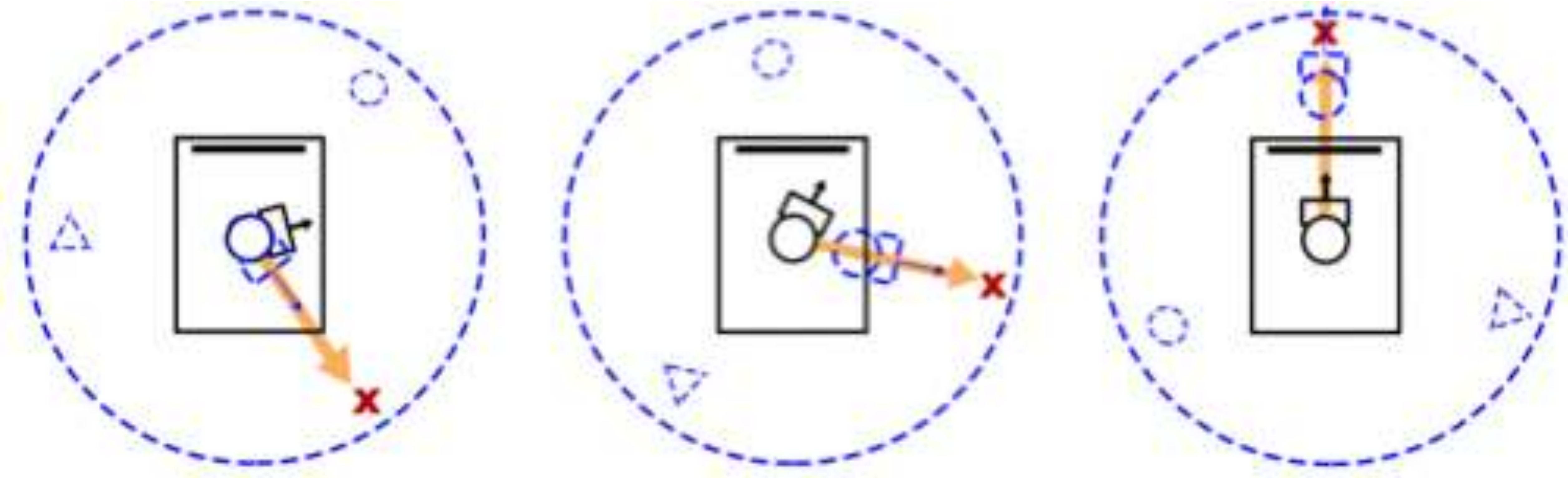
$$\theta_v = (2 - \cos(\theta_r)) \cdot \theta_r$$

where $\theta_{r/v}$ = yaw angle in real/virtual space



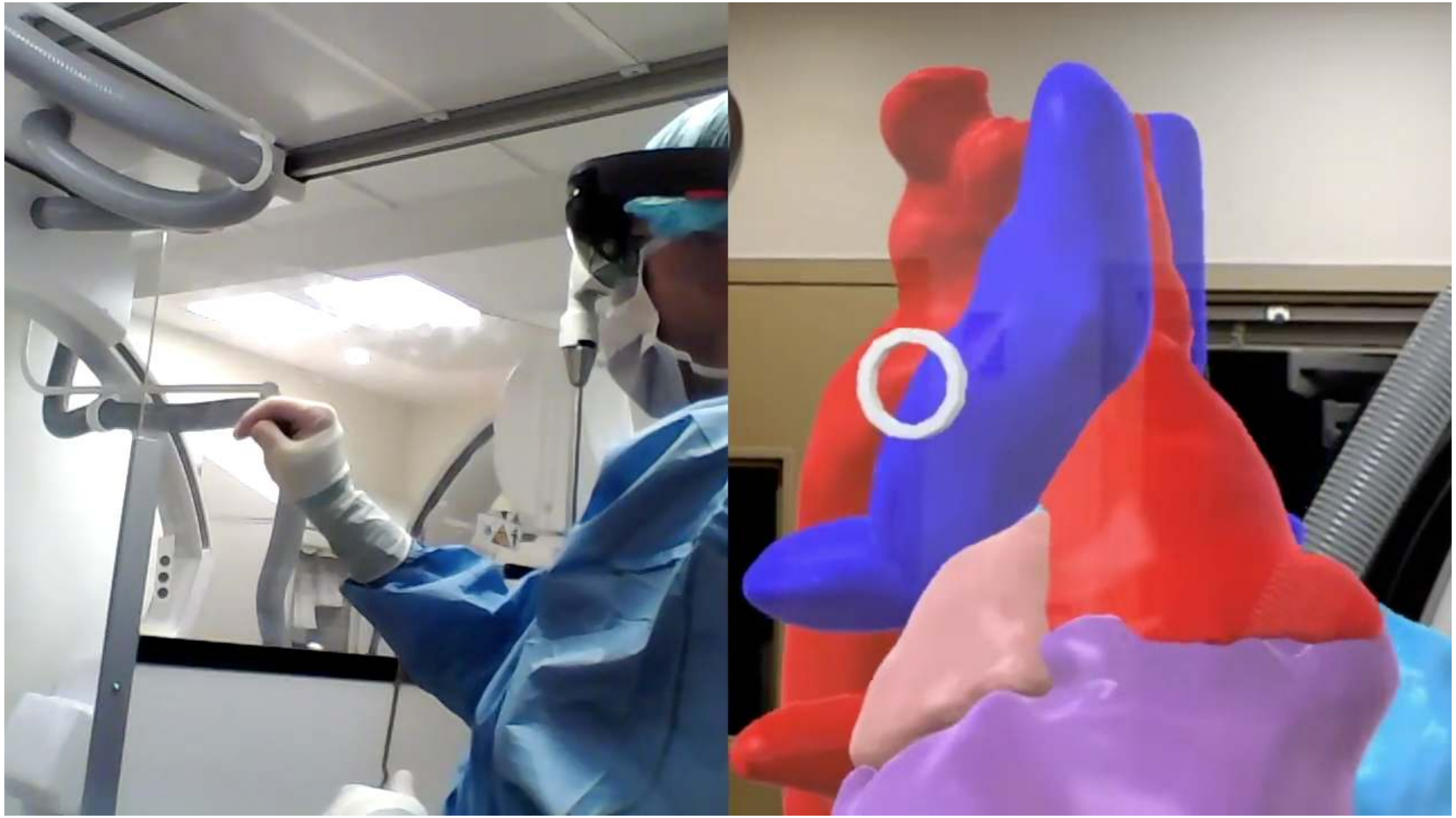
Including Redirection / Wash-Out

- Gradually realign a user's head orientation with the preferred forward direction as they virtually move (translate) through the VE



Design-Principle: Hands-Free Interaction

- Hot topic currently
- Especially for AR applications (hololens)
- Example: hands-free rotation of a visualization in an HMD
- Rotate object by small head rotations (first-order control)
- System commands using voice control



Alon Grinshpoon et al., Columbia University, 2018

System Control

- The 3rd big category of interaction tasks in VR:
    - The somewhat unbeloved child in the VR interaction community
    - The general task of these interactions: change the system's state
        - and everything else that doesn't fit anywhere else
- Consequence: a taxonomy is almost impossible
- Typical techniques:
    - Menus
    - Speech recognition
    - A set of gestures (for 1-out-of-n triggers)
    - Physical devices

FYI: Classification of Widgets for 3D UIs

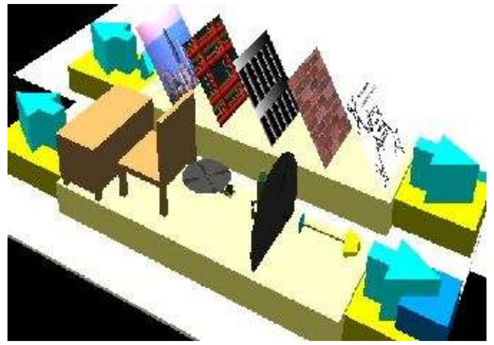
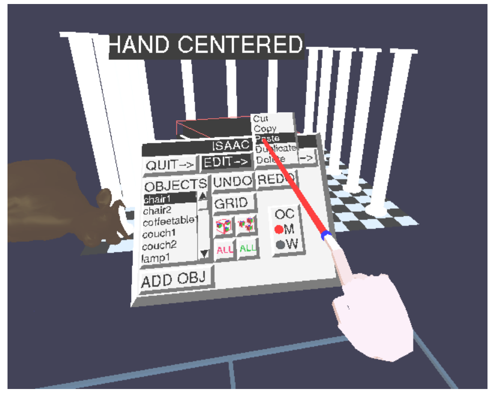
Direct 3D Object Interaction
- Object Selection
- Geometric Manipulation

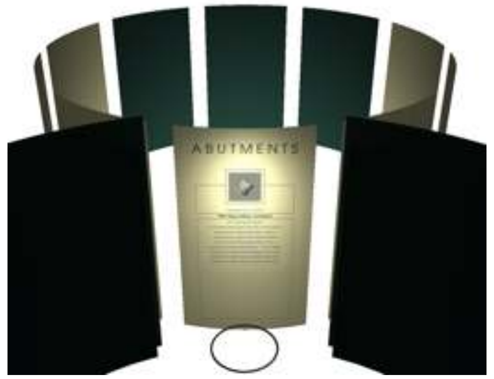
3D-Scene Manipulation
- Orientation and Navigation
- Scene Presentation Control

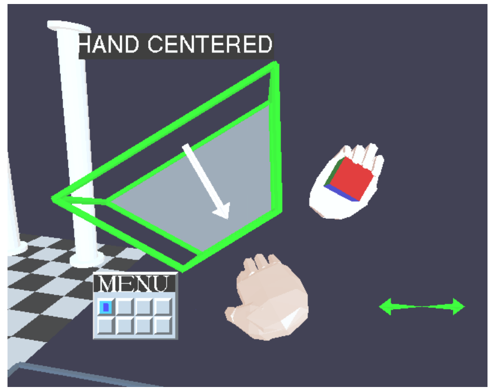
Exploration and Visualization
- Geometric Exploration
- Hierarchy Visualization
- 3D Graph Visualization
- 2D-Data and Document Visualization
- Scientific Visualization

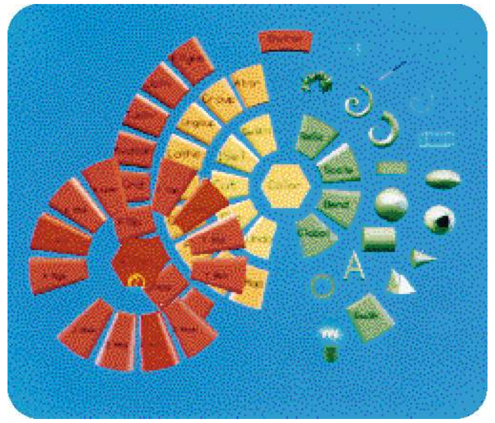
System / Application Control
- State Control / Discrete Valuators
- Continuous Valuators
- Special Value Input
- Menu Selection
- Containers

Menu Selection
- Temporary Option Menus
  - *Rotary Tool Chooser*
  - *Menu Ball*
  - *Command & Control Cube*
  - *Popup Menu*
  - *Tool Finger*
  - *TULIP*
- **Single Menus**
  - *Ring menu*
  - *Floating Menu*
  - *Drop-Down-Menu*
  - *Revolving Stage*
  - *Chooser Widget*
  - *3D-Palette, Primitive Box etc.*
- **Menu Hierarchies**
  - *Hands-off Menu*
  - *Hierarchical Pop-Up Menus*
  - *Tool Rack*
  - *3D Pie Menu*
  - → Hierarchy Visualizations

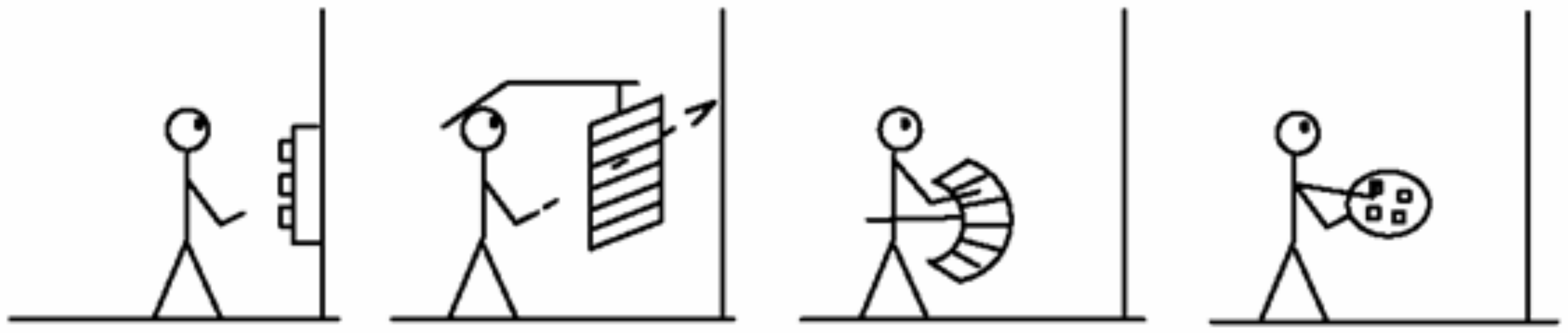






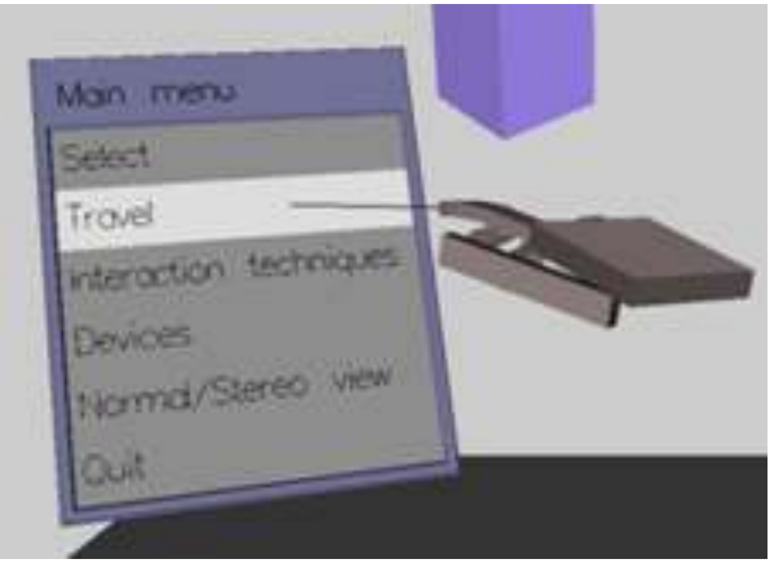
Menus in VR

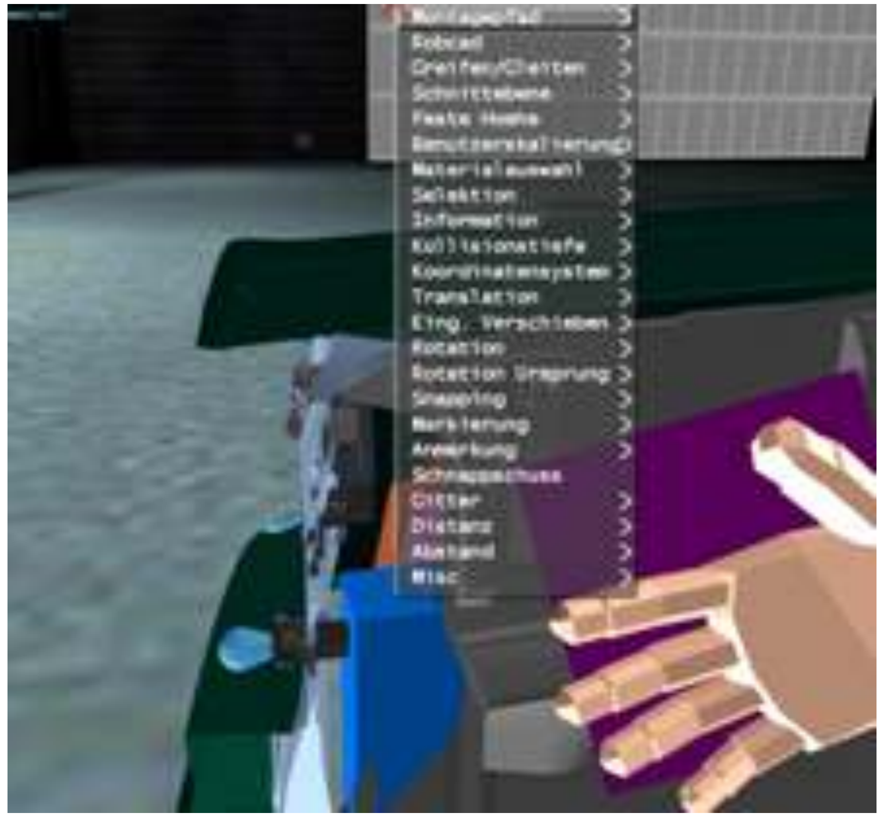
- Task decomposition:
  1. Bring up menu
  2. Navigate the menu
  3. Select an item
- A possible taxonomy should contain:
  - Input modalities: gestures, speech, buttons, ...
  - Positioning of the menu
  - Selection of items
  - Dimension and shape of the menu
- Examples for positioning the menu:

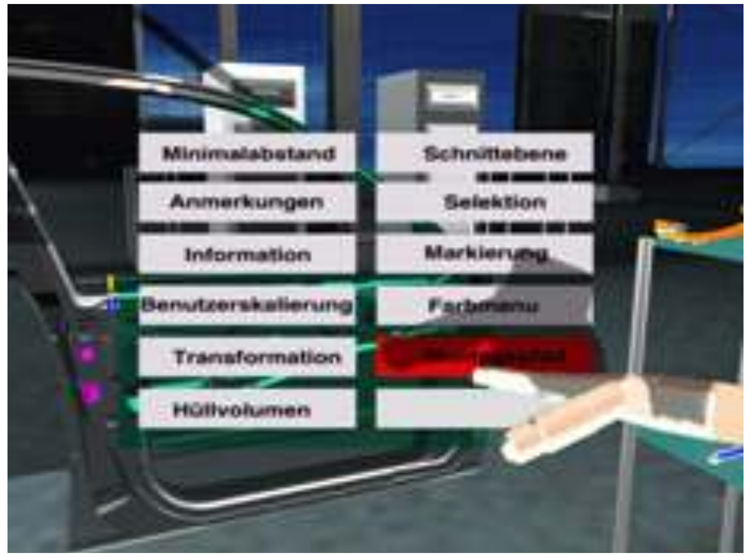


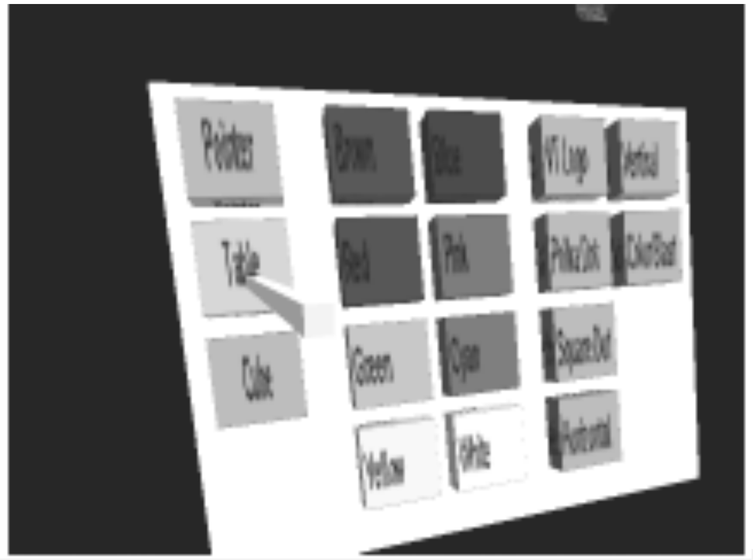
Examples

- Embedded in 3D or 2D overlay (heads-up)
- Item selection: one of the earlier selection techniques, e.g., ray casting or occlusion technique, or map relative hand motion to "active" menu item
- Positioning:
  - Fixed in 3D,
  - Heads-up (moves with head),
  - Attached to left hand, ...



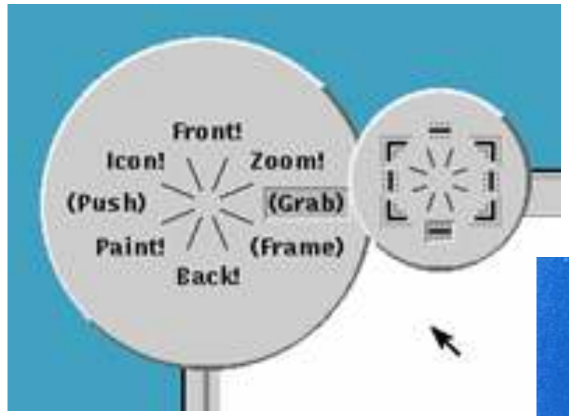


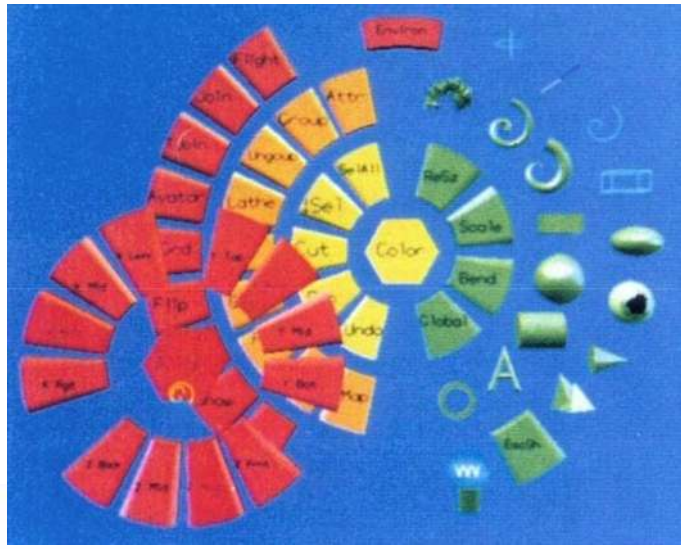


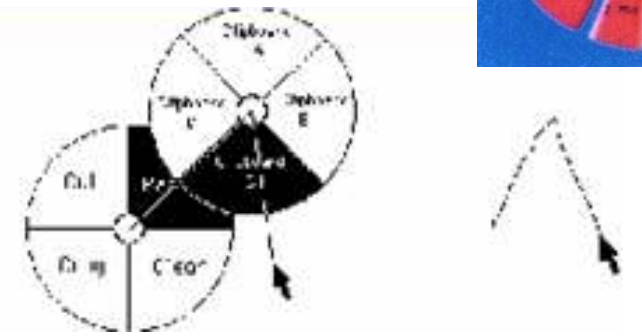


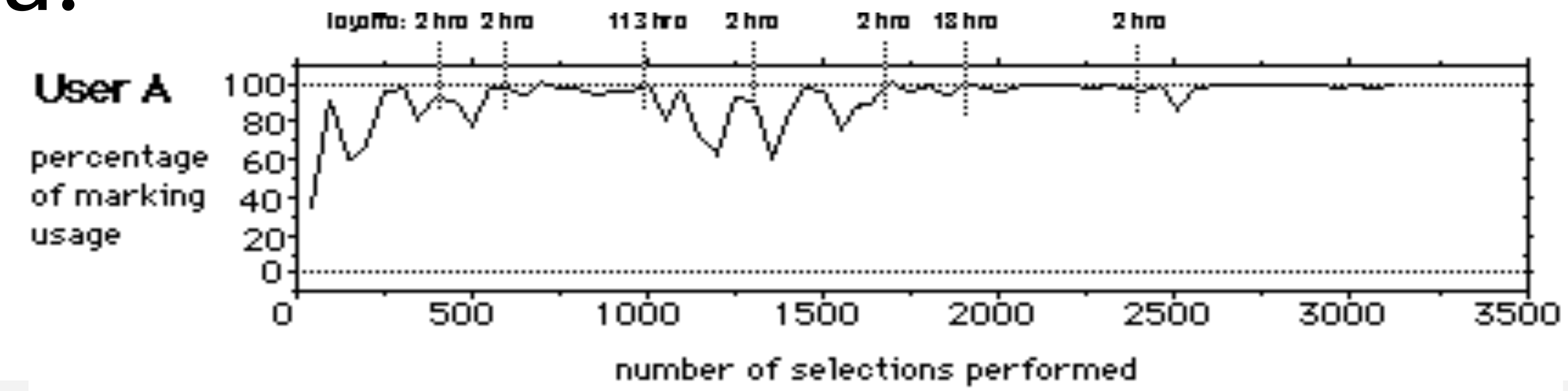
2D Digression: "Marking Menus" (aka. Pie Menu)

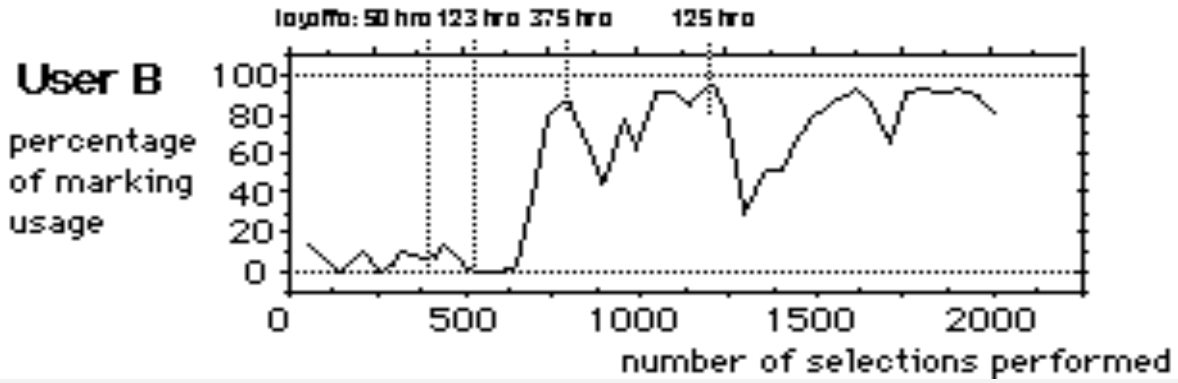
- Idea: arrange menu items around a center (in a circle, square, ...)
- With menu trigger: position the menu center at the current pointer position
- Advantage:
  - Smooth transition from novice mode to expert mode
  - Experts can navigate the menu "blindfolded"
- Mouse gestures (marks) are much more efficient than a menu:



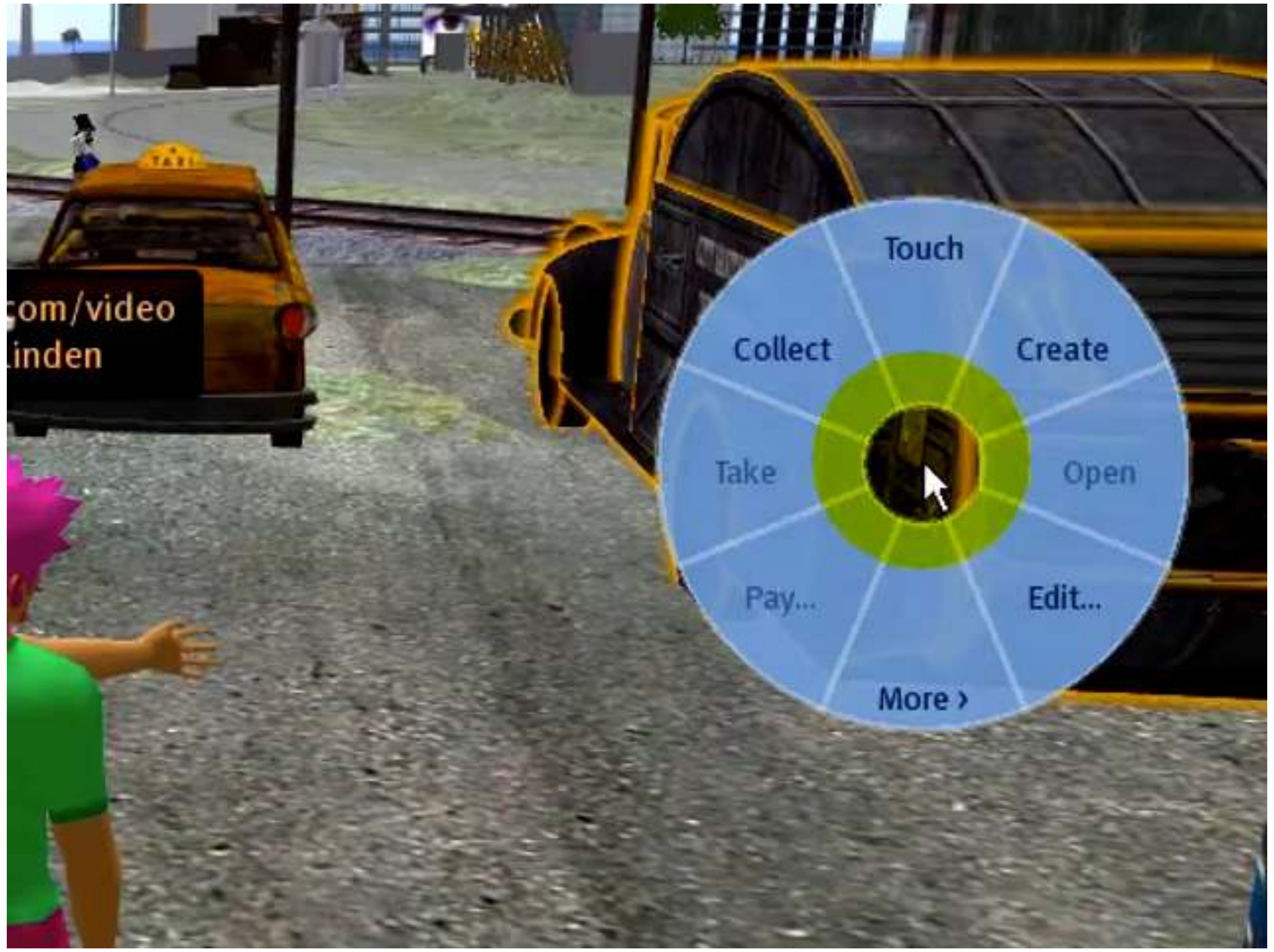


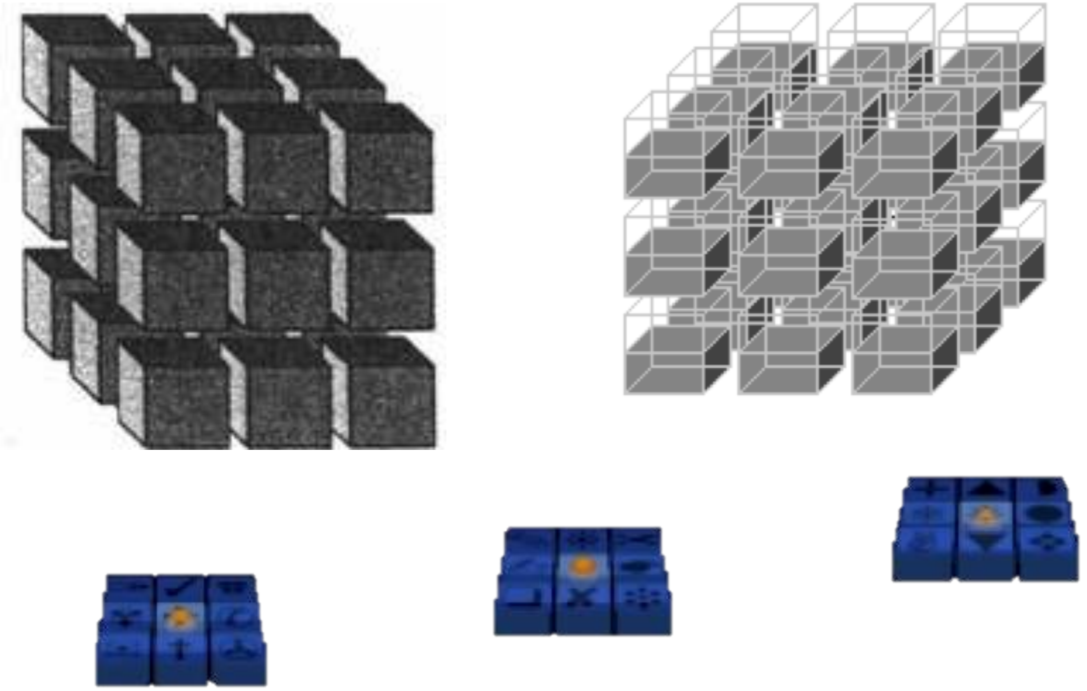



layoffs: 2 hrs 2 hrs 113 hrs 2 hrs 2 hrs 18 hrs 2 hrs
User A
percentage of marking usage
100
80
60
40
20
0
0
500
1000
1500
2000
2500
3000
3500
number of selections performed



- Example in 2D (Second Life):
- In 3D?
    - Direct "translation" → "control cube"
    - Not too successful
    - Can you do it better?





Usability Heuristics for User Interface Design by Jakob Nielsen

1. Visibility of system status: always keep user informed
2. Users should always have full control and freedom: e.g., always provide an emergency exit, undo, and redo
3. Recognition rather than recall: users should not have to remember information from one part of the dialogue to another ⟶ always make actions and options visible
4. Cater to both novice and expert users, e.g., by accelerators that are unobtrusive to the novice user
5. Aesthetic and minimalist design: don't show information that is irrelevant or rarely needed (it competes with the relevant info)

- ...