CSE 165: 3D User Interaction

Lecture #6:
Selection – Part 2

Announcements

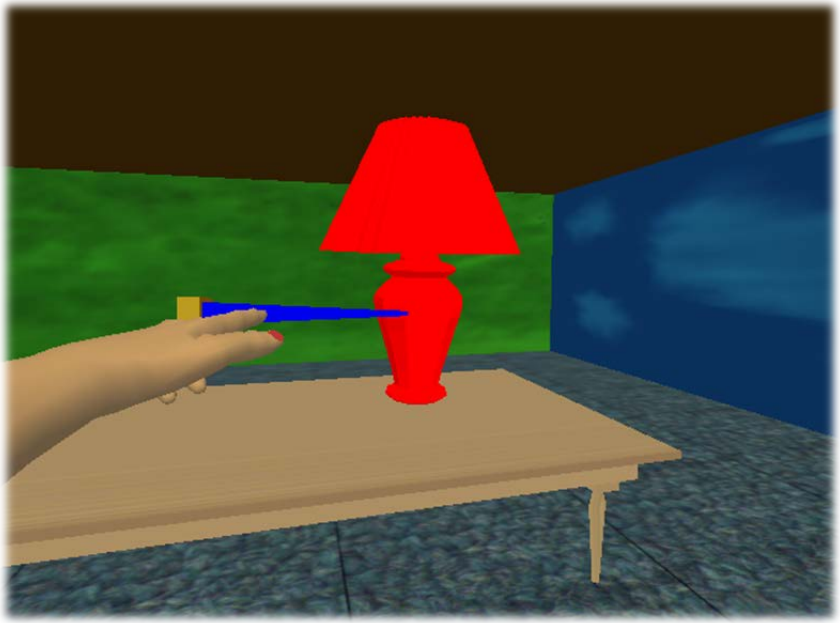
- Project 1 due this Friday at 2pm
- Grading in VR lab B210 2-3:30pm
  - Two groups:
    - even hours start at 2pm
    - odd hours at 3pm
- Homework submission:
  - Due tomorrow at 2pm
  - Need to upload scripts and assets to private Git repository
  - Add user: cse165 to your repo as a collaborator
  - Copy and paste your repository in the comment section when you are submitting your script to TritonEd

Video Presentations

Zhiyao Yan	EEG based BCI Navigation in VR MindMaze MindLeap VR Demo	https://youtu.be/6rj8tb0hrsw?t=16s https://www.youtube.com/watch?v=NoXhfHFeyPE
Alvin Ho	Da vinci robot surgery	https://www.youtube.com/watch?v=ATpxart8KoQ https://www.youtube.com/watch?v=XSMPNNkAmfE
Ran Tao	Google Earth VR	https://www.youtube.com/watch?v=VDM9q2ydhM8&feature=youtu.be&t=5m35s
Ka Chan	Rstyle 3D: VR for Real Estate Sales and Architecture (Demo v1.8)	https://www.youtube.com/watch?v=HIkaaP40xEs
Nathan Nguyen	VR Extreme Multiplayer Climbing - Climbey	https://www.youtube.com/watch?v=AZU0Vy7Ugl8

Ray-Casting

- User points at objects with virtual ray
- Ray defines and visualizes pointing direction
- First intersected object is selected



$$\mathbf{p}(\alpha) = \mathbf{h} + \alpha \cdot \vec{\mathbf{p}}$$

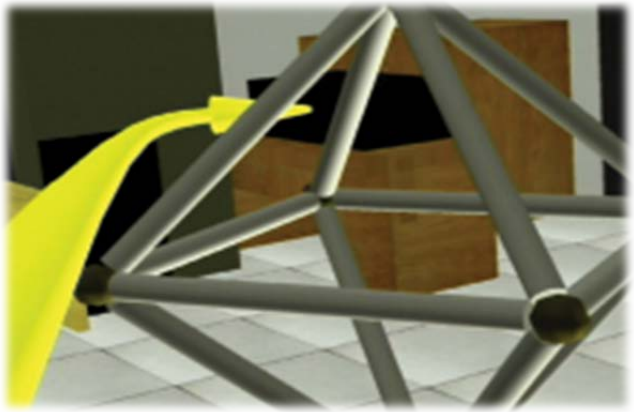
$\mathbf{h} = 3\text{D position of virtual hand}$

$\vec{\mathbf{p}} = \text{ray attached to } \mathbf{h}$

$0 < \alpha < \infty$ determined by first object intersection

Two-Handed Pointing

- Ray casting with 2 hands
- More control
  - Distance between hands controls length
  - Allows pointing at things behind other things



$$\mathbf{p}(\alpha) = \mathbf{h}_\mathbf{l} + \alpha \cdot (\mathbf{h}_\mathbf{r} - \mathbf{h}_\mathbf{l})$$

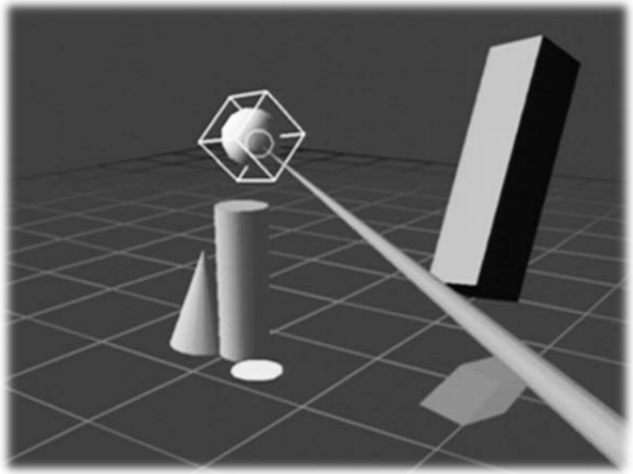
$0 < \alpha < \infty$ is fixed

$\mathbf{h}_\mathbf{l}$ = 3D position of left hand

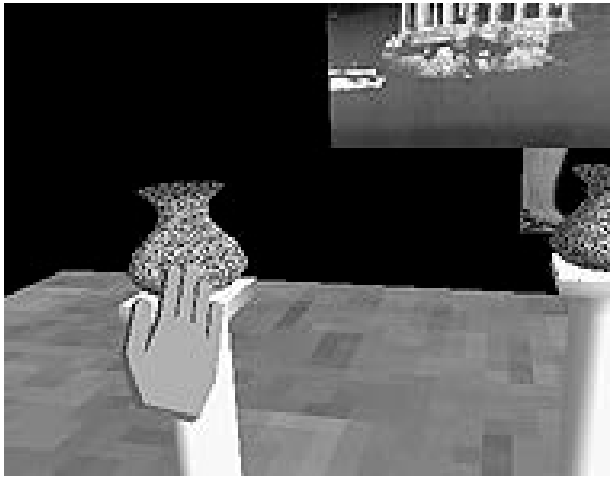
$\mathbf{h}_\mathbf{r}$ = 3D position of right hand

Flashlight

- Does not need precision
- Conic selection volume
  - Tip of cone in wand
  - Cone direction determined by wand direction
  - Fixed cone size
- If multiple objects in cone
  - Object closer to center line of cone is selected
  - If multiple objects are equally close to center line: select object closer to device



Virtual Hand



- Select and manipulate directly with hand
- Hand represented as 3D cursor
- Intersection between cursor and object indicates selection

$$\mathbf{p}_v = \alpha \cdot \mathbf{p}_r, \mathbf{R}_v = \mathbf{R}_r$$

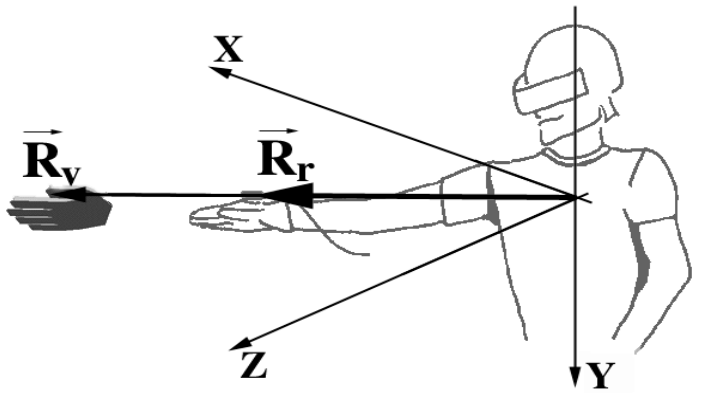
$\mathbf{p}_r, \mathbf{R}_r$ = position and orientation of real hand

$\mathbf{p}_v, \mathbf{R}_v$ = position and orientation of hand in VE

α = fixed scaling factor

Go-Go

- By Ivan Poupyrev, 1996
- Arm-extension technique
- Touch objects to select, like simple virtual hand
- Non-linear mapping between physical and virtual hand position
- Requires torso position
- Local and distant regions

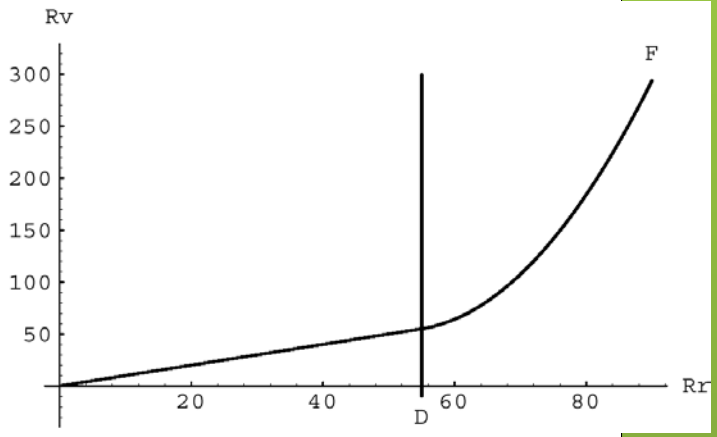


$$r_v = F(r_r) = \begin{cases} r_r & if\ r_r \le D \\ r_r + \alpha(r_r - D)^2 & otherwise \end{cases}$$

where r_r = length of $\vec{\mathbf{R}}_\mathbf{r}$

r_v = length of $\vec{\mathbf{R}}_\mathbf{v}$

D, α are constants



World-in-Miniature (WIM)

- By Stoakley, 1995
- “Dollhouse” world held in user’s hand
- Miniature objects can be manipulated directly
- Moving miniature objects affects full-scale objects
- Can also be used for navigation

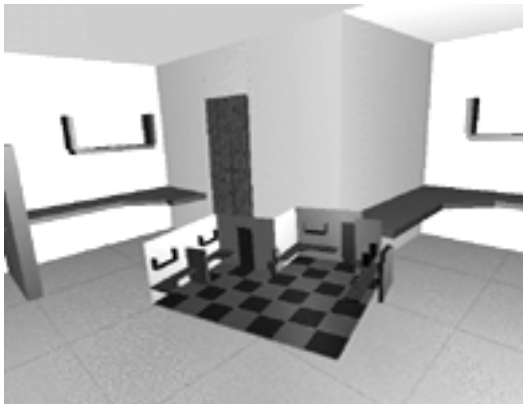
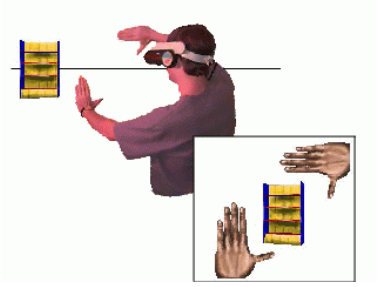


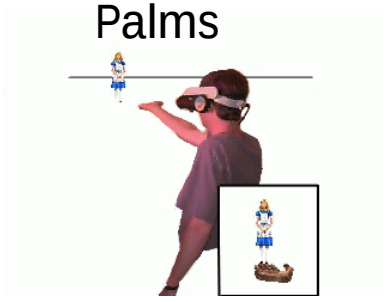
Image Plane Techniques

- Require only 2 DOF
  - Selection based on 2D projections
  - Use virtual image plane in front of user
    - Dependent on head/eye position

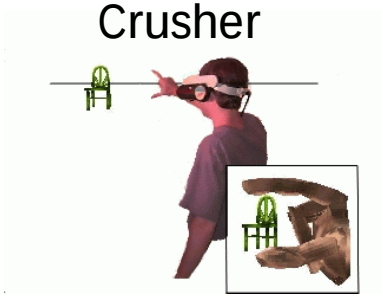
Framing



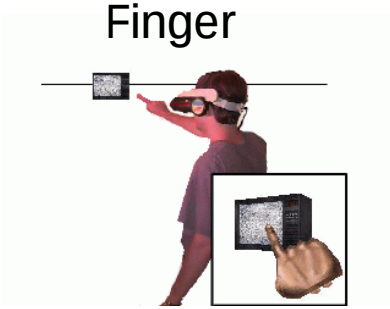
Lifting Palms



Head-Crusher

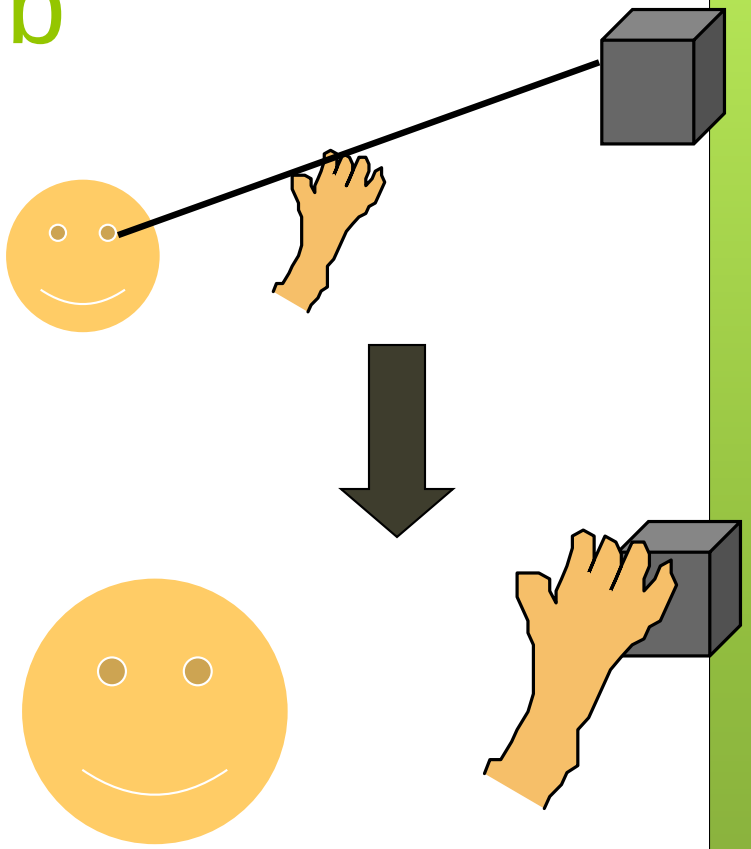


Sticky Finger



Scaled-World Grab

- By Mine et al., 1997
- Often used with occlusion
- At selection, scale world down so that virtual hand touches selected object
- User initially does not notice a change in the image, until head or hand is moved

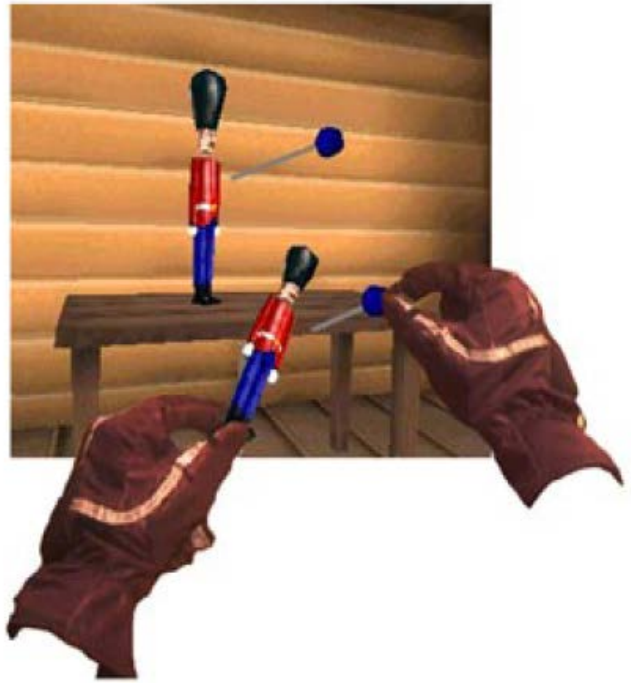


Forced Perspective

- Museum of Simulation Technology
  - http://www.youtube.com/watch?v=HOfll06X16c

Voodoo Dolls

- Pierce et al. 1999
- Two-handed technique
- Builds upon image plane and WIM techniques
- Developed for pinch gloves
  - Requires finger pose tracking
- Creates copies of objects (dolls) for manipulation
- Non-dominant hand: stationary frame of reference
- Dominant hand: defines position and orientation



Selection by Dwell Time

- User points at object with any technique
  - Virtual pointer
  - Eye gaze
- Action is triggered after dwell time threshold is exceeded
- Works without physical buttons
- Frequently used in controller-less VR: Google Cardboard, Samsung Gear VR

3D UI With the Leap

- Selection
  - Hover w/timeout (dwell)
  - Trigger with non-dominant hand gesture
  - Two finger near-pinch
- Manipulation
  - Hand orientation
  - 3-finger orientation
  - 2-finger orientation (2 DOF)

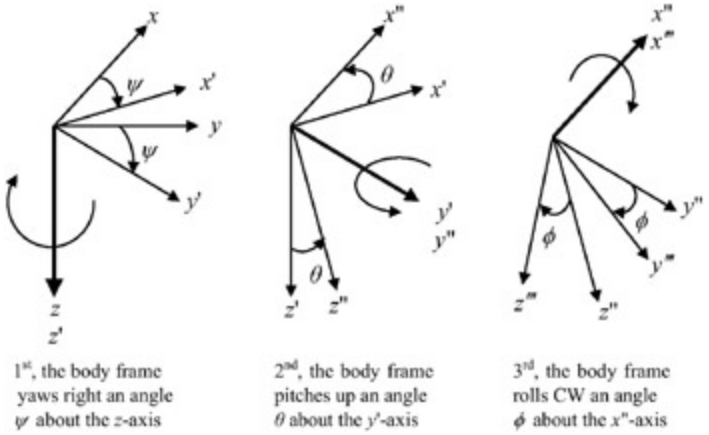
General Tips for the Leap

- Finger pinches hard to detect
- More than 3 fingers hard to distinguish
- Fingers hard to distinguish when hand not viewed well from head
- Hand detection (left/right): need to carefully bring hands into FOV from bottom edge

Quaternions

Rotation Calculations

- Intuitive approach: Euler Angles:
  - Simplest way to calculate rotations
  - Defines rotation by 3 sequential rotations about coordinate axes
  - Example Z-Y-X:



1st, the body frame yaws right an angle ψ about the z-axis

2nd, the body frame pitches up an angle θ about the y'-axis

3rd, the body frame rolls CW an angle ϕ about the x''-axis

http://www.globalspec.com/reference/49379/203279/3-3-euler-angles

Problems With Euler Angles

- Problems with Euler angles:
  - No standard for order of rotations
  - Gimbal Lock, occurs in certain object orientations
    - Video (0:20-1:12)
      - http://www.youtube.com/watch?v=zc8b2Jo7mno
- Better: rotation about arbitrary axis (no Gimbal lock)
  - Can be done with 4x4 matrix
  - But: smoothly interpolating between two orientations is difficult

➔ Quaternions

Quaternion Definition

- Given angle and axis of rotation:
  - a: rotation angle
  - {nx,ny,nz}: normalized rotation axis
- Calculation of quaternion coefficients w, x, y, z:
  - w = cos(a/2)
  - x = sin(a/2) * nx
  - y = sin(a/2) * ny
  - z = sin(a/2) * nz

Useful Quaternions

w	x	y	z	Description
1	0	0	0	Identity quaternion, no rotation
0	1	0	0	180° turn around X axis
0	0	1	0	180° turn around Y axis
0	0	0	1	180° turn around Z axis
sqrt(0.5)	sqrt(0.5)	0	0	90° rotation around X axis
sqrt(0.5)	0	sqrt(0.5)	0	90° rotation around Y axis
sqrt(0.5)	0	0	sqrt(0.5)	90° rotation around Z axis
sqrt(0.5)	-sqrt(0.5)	0	0	-90° rotation around X axis
sqrt(0.5)	0	-sqrt(0.5)	0	-90° rotation around Y axis
sqrt(0.5)	0	0	-sqrt(0.5)	-90° rotation around Z axis

Quaternions: Further Reading

- Rotating Objects Using Quaternions
  - http://www.gamasutra.com/view/feature/131686/rotating_objects_using_quaternions.php
- Quaternions in Unity 3D:
  - https://docs.unity3d.com/ScriptReference/Quaternion.html
- Quaternions in OpenSceneGraph:
  - http://www.openscenegraph.org/projects/osg/wiki/Support/Maths/QuaternionMaths