

3D User Interface Techniques for Selection and Manipulation

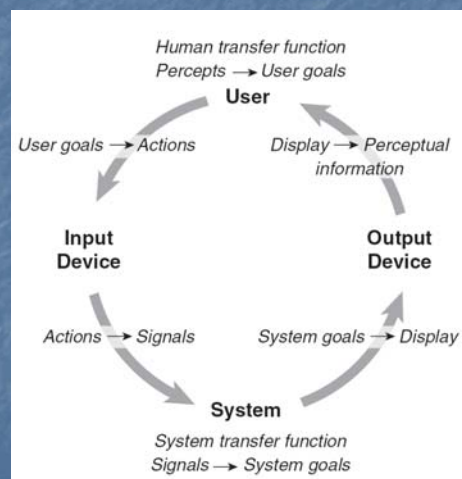
Lecture #8: Selection and Manipulation
Spring 2017
Joseph J. LaViola Jr.

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Interaction Workflow



Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Universal 3D Interaction Tasks

- Navigation
 - Travel – motor component
 - Wayfinding – cognitive component
- Selection
- Manipulation
- System control
- Symbolic input

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Why Selection and Manipulation?

- Major method of interaction with physical environments
- Major method of interaction with virtual environments
- Affects the quality of entire 3D interface
- Design of 3D manipulation techniques is difficult

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Lecture Outline

- What is 3D selection and manipulation?
- Relationship between IT and input device
- Manipulation technique classification
- Techniques
 - selection
 - manipulation
 - hybrid
- Isomorphism vs. Non-isomorphism

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Selection & Manipulation

- Selection: specifying one or more objects from a set
- Manipulation: modifying object properties (position, orientation, scale, shape, color, texture, behavior, etc.)

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Goals of Selection

- Indicate action on object
- Query object
- Make object active
- Travel to object location
- Set up manipulation

Selection Performance

- Variables affecting user performance
 - object distance from user
 - object size
 - density of objects in area
 - occluders

Canonical Parameters

- Selection
 - distance and direction to target
 - target size
 - density of objects around the target
 - number of targets to be selected
 - target occlusion.
- Positioning
 - distance/direction to initial position
 - distance/direction to target position
 - translation distance
 - required precision of positioning
- Rotation
 - distance to target
 - initial orientation
 - final orientation
 - amount of rotation

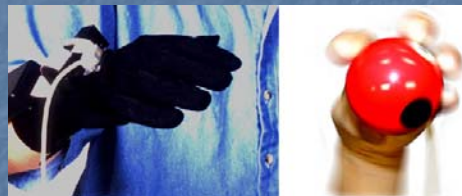
Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

3D Interaction Techniques and the Input Device

- Number of control dimensions
- Control Integration
- Force vs. Position control
- Device placement
- Form Factor



Attached to Hand

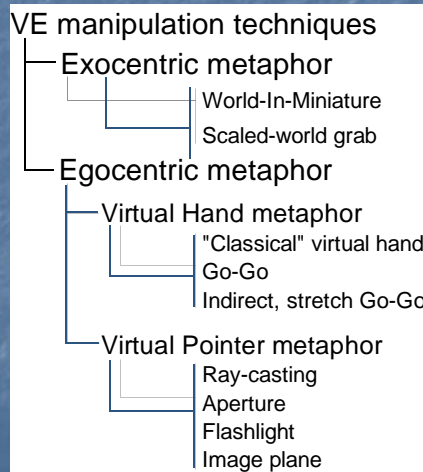
Rolled with fingers

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Technique Classification by Metaphor

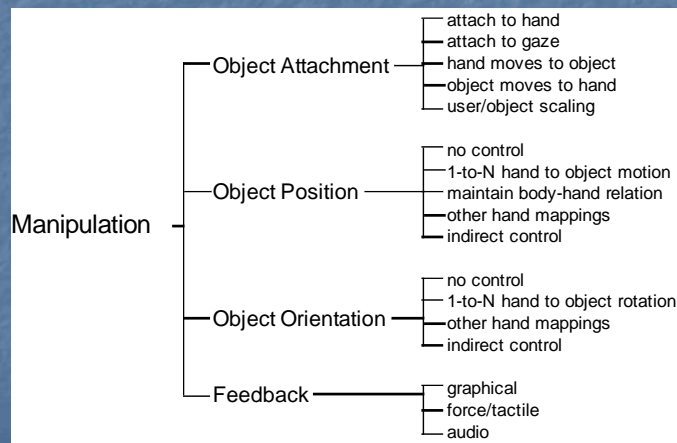


Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Technique Classification by Components



Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

3D Selection and Manipulation Techniques

- Pointing
 - ray-casting
 - two-handed pointing
 - flashlight & aperture
 - image plane
- Direct manipulation
 - simple virtual hand
 - Go-Go
 - WIM
- Hybrids
 - Homer
 - Scaled-World Grab
 - Voodoo Dolls



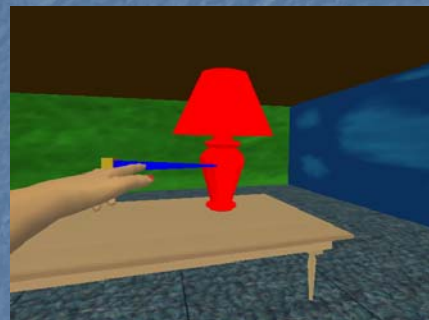
Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Pointing – Ray-Casting

- User points at objects with virtual ray
- Ray defines and visualizes pointing direction



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

where $0 < \alpha < \infty$

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

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

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Pointing – Two-Handed Pointing

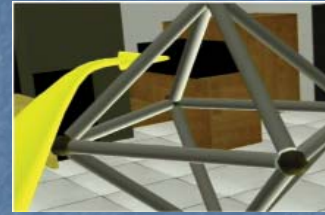
- Ray casting with 2 hands
- More control
 - distance between hands controls length
 - twisting curves pointer

$$\mathbf{p}(\alpha) = \mathbf{h}_l + \alpha \cdot (\mathbf{h}_r - \mathbf{h}_l)$$

where $0 < \alpha < \infty$

\mathbf{h}_l = 3D position of left hand

\mathbf{h}_r = 3D position of right hand



Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Pointing – Flashlight and Aperture

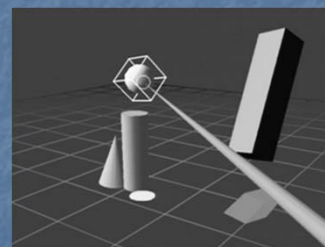
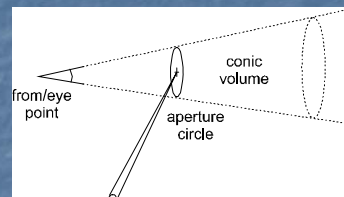
- Flashlight – soft selection technique
 - does not need precision
 - conic volume constant
- Aperture – extension to Flashlight
 - adjustable volume

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

where $0 < \alpha < \infty$

\mathbf{h} = 3D position of hand

\mathbf{e} = 3D coordinates of viewport



Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Pointing – Image Plane Family

- Requires only 2 DOF
 - selection based on 2D projections
 - virtual image plane in front of user



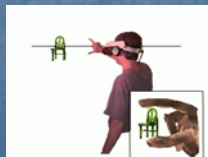
Framing



Lifting Palms



Head-Crusher



Sticky Finger



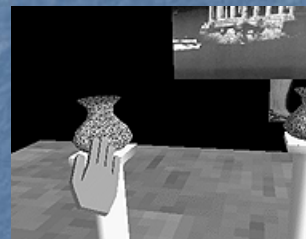
Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Direct Manipulation – Virtual Hand

- Select and manipulate directly with hands
- 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 = a scaling factor

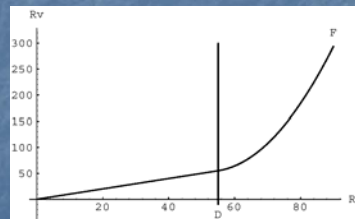
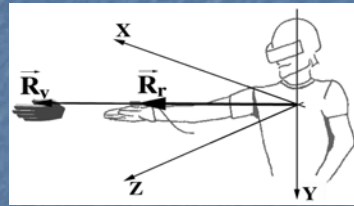
Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Direction Manipulation – Go-Go

- Arm-extension technique
- Like simple v. hand, touch objects to select them
- Non-linear mapping between physical and virtual hand position
- Local and distant regions



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

where r_r = length of \vec{R}_r

r_v = length of \vec{R}_v

D, α are constants

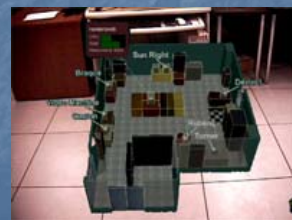
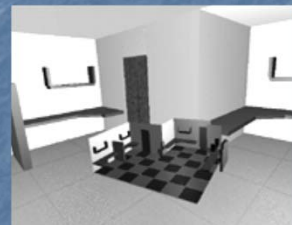
Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Direct Manipulation – WIM

- “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



Spring 2017

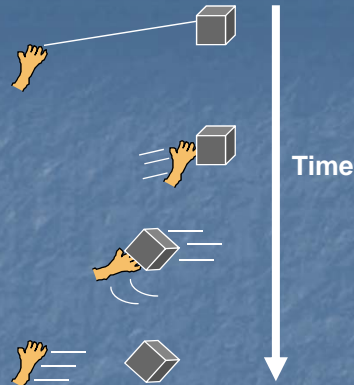
CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Hybrids – HOMER

Hand-Centered
Object
Manipulation
Extending
Ray-Casting

- Select: ray-casting
- Manipulate: hand



Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

HOMER Implementation

- Requires torso position t
- Upon selection, detach virtual hand from tracker, move v. hand to object position in world CS, and attach object to v. hand (w/out moving object)
- Get physical hand position h and distance $d_h = \text{dist}(h, t)$
- Get object position o and distance $d_o = \text{dist}(o, t)$

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

HOMER Implementation (cont.)

- Each frame:
 - Copy hand tracker matrix to v. hand matrix (to set orientation)
 - Get physical hand position h_{curr} and distance:
 $d_{h-curr} = dist(h_{curr}, t)$
 - V. hand distance $d_{vh} = d_{h-curr} \times \left(\frac{d_o}{d_h}\right)$
 - Normalize torso-hand vector $th_{curr} = \frac{h_{curr} - t}{\|h_{curr} - t\|}$
 - V. hand position $vh = t + d_{vh} * (th_{curr})$

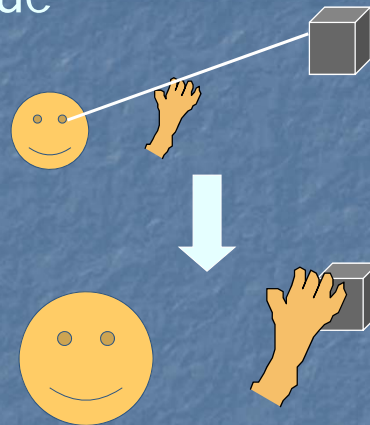
Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Hybrids – Scaled-World Grab Technique

- Often used w/ occlusion
- At selection, scale user up (or world down) so that v. hand is actually touching selected object
- User doesn't notice a change in the image until he moves



Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Scaled-World Grab Implementation

- At selection:
 - Get world CS distance from eye to hand d_{eh}
 - Get world CS distance from eye to object d_{eo}
 - Scale user (entire user subtree) uniformly by d_{eo} / d_{eh}
 - Ensure that eye remains in same position
 - Attach selected object to v. hand (w/out moving object)
- At release:
 - Re-attach object to world (w/out moving object)
 - Scale user uniformly by d_{eh} / d_{eo}
 - Ensure that eye remains in same position

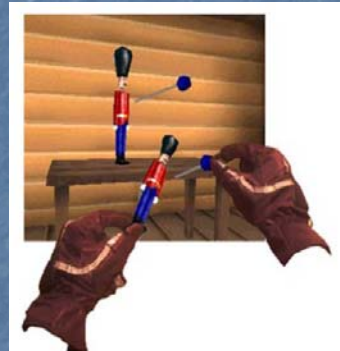
Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Hybrids – Voodoo Dolls

- Two handed technique
- Builds upon image plane and WIM techniques
- Creates copies of objects (dolls) for manipulation
- Non-dominant hand – stationary frame of reference
- Dominant hand – defines position and orientation



Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Isomorphic vs. Non-Isomorphic Philosophies

- Human-Machine interaction
 - input device
 - display device
 - transfer function (control to display mapping)
- Isomorphic – one-to-one mapping
- Non-isomorphic – scaled linear/non-linear mapping

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Non-Isomorphic 3D Spatial Rotation

- Important advantages
 - manual control constrained by human anatomy
 - more effective use of limited tracking range (i.e. vision-based tracking)
 - additional tools for fine tuning interaction techniques
- Questions
 - faster?
 - more accurate?

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Rotational Space

- Rotations in 3D space are a little tricky
 - do not follow laws of Euclidian geometry
- Space of rotations is not a vector space
- Represented as a closed and curved surface
 - 4D sphere or manifold
- Quaternions provide a tool for describing this surface

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Quaternions

- Four-dimensional vector (\mathbf{v}, w) where \mathbf{v} is a 3D vector and w is a real number
- A quaternion of unit length can be used to represent a single rotation about a unit axis \hat{u}

and angle θ as

$$q = \left(\sin\left(\frac{\theta}{2}\hat{u}\right), \cos\left(\frac{\theta}{2}\right) \right) = e^{\frac{\theta}{2}\hat{u}}$$

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Linear 0th Order 3D Rotation

- Let q_c be the orientation of the input device and q_d be the displayed orientation then

$$(1) \quad q_c = (\sin(\frac{\theta_c}{2} \hat{u}_c), \cos(\frac{\theta_c}{2})) = e^{\frac{\theta_c}{2} \hat{u}_c}$$

$$(2) \quad q_d = (\sin(\frac{k\theta_c}{2} \hat{u}_c), \cos(\frac{k\theta_c}{2})) = e^{\frac{k\theta_c}{2} \hat{u}_c} = q_c^k$$

- Final equations w.r.t. identity or reference orientation q_o are

$$(3) \quad q_d = q_c^k \quad (4) \quad q_d = (q_c q_o^{-1})^k q_o, \quad k = \text{CD gain coefficient}$$

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Non-Linear 0th Order 3D Rotation

- Consider

$$(3) \quad q_d = q_c^k \quad (4) \quad q_d = (q_c q_o^{-1})^k q_o$$

- Let k be a non-linear function as in

$$\omega = 2 \arccos(q_c \cdot q_o) \quad \text{or} \quad \omega = 2 \arccos(w)$$

$$k = F(\omega) = \begin{cases} 1 & \text{if } \omega < \omega_o \\ f(\omega) = 1 + c(\omega - \omega_o)^2 & \text{otherwise} \end{cases}$$

where c is a coefficient and ω_o is the threshold angle

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Design Considerations

- Absolute mapping – taken on *i*-th cycle of the simulation loop

$$q_{d_i} = q_{c_i}^k$$

- Relative mapping – taken between the *i*-th and *i-1*th cycle of the simulation loop

$$q_{d_i} = (q_{c_i} q_{c_{i-1}}^{-1})^k q_{d_{i-1}}$$

Absolute Non-Isomorphic Mapping

- Generally do not preserve directional compliance
- Strictly preserves nulling compliance

Relative Non-Isomorphic Mapping

- Always maintain directional compliance
- Do not generally preserve nulling compliance

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Amplified Non-Linear Rotation for VE Navigation (1)

- Users expect the virtual world to exist in any direction
 - 3-walled Cave does not allow this
 - adapt expected UI to work in restricted environment
- Amplified rotation allows users to see a full 360 degrees in a 3-walled display
- A number of approaches were tested
 - important to take cybersickness into account

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Amplified Non-Linear Rotation for VE Navigation (2)

- Apply a non-linear mapping function to the user's waist orientation θ and his or her distance d from the back of the Cave
- Calculate the rotation factor using a scaled 2D Gaussian function

$$\phi = f(\theta, d) = \frac{1}{\sqrt{2\pi\sigma_1}} e^{-\frac{(|\theta| - \pi(1-d/L))^2}{2\sigma_2^2}}$$

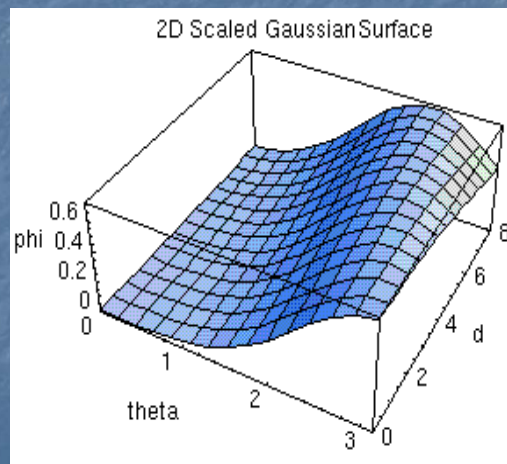
- The new viewing angle is $\theta_{new} = \theta(1 - \phi)$

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Amplified Non-Linear Rotation for VE Navigation (3)



$$\sigma_1 = 0.57$$

$$\sigma_2 = 0.85$$

$$L = 30$$

$$\mu = \pi$$

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Non-Linear Translation for VE Navigation (1)

- Users lean about the waist to move small to medium distances
 - users can lean and look in different directions
- Users can also lean to translate a floor-based interactive world in miniature (WIM)
 - Step WIM must be active
 - user's gaze must be 25 degrees below horizontal

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Non-Linear Translation for VE Navigation (2)

- Leaning vector \vec{L}_R is the projection of the vector between the waist and the head onto the floor
 - gives direction and raw magnitude components
- Navigation speed is dependent on the user's physical location
 - Leaning sensitivity increases close to a boundary
- Linear function - $L_T = a \cdot D_{\min} + b$
- Mapped velocity - $v = \|\vec{L}_R\| - L_T$

Spring 2017

CAP6121 – 3D User Interfaces for Games and Virtual Reality

©Joseph J. LaViola Jr.

Non-Linear Translation for VE Navigation (3)

- Navigation speed is also dependent on the user's head orientation with respect to the vertical axis
 - especially useful when translating the floor-based WIM

- Mapping is done with a scaled exponential function

$$F = \alpha \cdot e^{-\beta |\vec{H} \cdot \vec{V}_{up}|}$$

- Final leaning velocity is $v_{final} = F \cdot v$

Next Class

- Navigation – Travel
- Readings
 - 3DUI Book – Chapter 5