3D User Interface Techniques for Selection and Manipulation

Lecture #8: Selection and Manipulation
Spring 2011
Joseph J. LaViola Jr.

Interaction Workflow
Universal 3D Interaction Tasks

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

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
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

Selection & Manipulation

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

3D Interaction Techniques and the Input Device

- Number of control dimensions
- Control Integration
- Force vs. Position control
- Device placement
- Form Factor
Technique Classification by Metaphor

VE manipulation techniques

- Exocentric metaphor
  - World-In-Miniature
  - Scaled-world grab

- Egocentric metaphor
  - Virtual Hand metaphor
    - "Classical" virtual hand
    - Go-Go
    - Indirect, stretch Go-Go
  - Virtual Pointer metaphor
    - Ray-casting
    - Aperture
    - Flashlight
    - Image plane

Technique Classification by Components

Manipulation

- Object Attachment
  - attach to hand
  - attach to gaze
  - hand moves to object
  - object moves to hand
  - user/object scaling
  - no control

- Object Position
  - 1-to-N hand to object motion
  - maintain body-hand relation
  - other hand mappings
  - indirect control

- Object Orientation
  - no control
  - 1-to-N hand to object rotation
  - other hand mappings
  - indirect control

Feedback

- graphical
- force/tactile
- audio
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

Pointing – Ray-Casting

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

\[ p(\alpha) = h + \alpha \cdot \hat{p} \]

where \( 0 < \alpha < \infty \)

\( h \) = 3D position of virtual hand
\( \hat{p} \) = ray attached to \( h \)
Pointing – Two-Handed Pointing

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

\[ p(\alpha) = h_l + \alpha \cdot (h_r - h_l) \]
where \( 0 < \alpha < \infty \)

\[ h_l = \text{3D position of left hand} \]
\[ h_r = \text{3D position of right hand} \]

Pointing – Flashlight and Aperture

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

\[ p(\alpha) = e + \alpha \cdot (h - e) \]
where \( 0 < \alpha < \infty \)

\[ h = \text{3D position of hand} \]
\[ e = \text{3D coordinates of viewport} \]
Pointing – Image Plane Family

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

Direct Manipulation – Virtual Hand

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

\[ p_v = \alpha \cdot p_r, R_v = R_r \]

- \( p_r, R_r \) = position and orientation of real hand
- \( p_v, R_v \) = position and orientation of hand in VE
- \( \alpha \) = a scaling factor
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

\[ r_i = F(r_i) = \begin{cases} r_i & \text{if } r_i \leq D \\ r_i + \alpha(r_i - D) & \text{otherwise} \end{cases} \]

where \( r_i \) = length of \( \mathbf{R}_r \)
\( r_c \) = length of \( \mathbf{R}_c \)
\( D, \alpha \) are constants

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
Hybrids - HOMER

- Hand-Centered
- Object
- Manipulation
- Extending
- Ray-Casting
  - Select: ray-casting
  - Manipulate: hand

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 $d_o = \text{dist}(o, t)$
HOMER Implementation (cont.)

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

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
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

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
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

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?
Rotational Space

- Rotations in 3D space are a little tricky
  - do not follow laws of Euclidean 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

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 \(\theta\) as
  \[
  q = (\sin(\frac{\theta}{2}) \hat{u}, \cos(\frac{\theta}{2})) = e^{\frac{\theta}{2} \hat{u}}
  \]
Linear 0th Order 3D Rotation

- Let $q_c$ be the orientation of the input device and $q_d$ be the displayed orientation then
  
  \begin{align*}
  (1) \quad q_c &= (\sin(\frac{\theta}{2}) \hat{u}_c),\cos(\frac{\theta}{2})) = e^{i\frac{\theta}{2}}
  
  (2) \quad q_d &= (\sin(\frac{k\theta}{2}) \hat{u}_c),\cos(\frac{k\theta}{2})) = e^{k\frac{\theta}{2}} = q_c^k
  
  \end{align*}

- Final equations w.r.t. identity or reference orientation $q_o$ are
  
  \begin{align*}
  (3) \quad q_q &= q_c^k \\
  (4) \quad q_d &= (q, q_o^{-1})^k q_o, \quad k = \text{CD gain coefficient}
  
  \end{align*}

Non-Linear 0th Order 3D Rotation

- Consider
  
  \begin{align*}
  (3) \quad q_d &= q_c^k \\
  (4) \quad q_d &= (q, q_o^{-1})^k q_o
  
  \end{align*}

- Let $k$ be a non-linear function as in
  
  $\omega = 2\arccos(q_c \cdot q_o)$ or $\omega = 2\arccos(w)$

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

  where $c$ is a coefficient and $\omega_o$ is the threshold angle
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

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
Amplified Non-Linear Rotation for VE Navigation (2)

- Apply a non-linear mapping function to the user’s waist orientation $\theta$ 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 $\theta_{new} = \theta (1 - \phi)$

Amplified Non-Linear Rotation for VE Navigation (3)

$$ \sigma_1 = 0.57 $$
$$ \sigma_2 = 0.85 $$
$$ L = 30 $$
$$ \mu = \pi $$
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

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 $L_T = a \cdot D_{\text{min}} + b$
- Mapped velocity $v = \| \vec{L}_R \| - L_T$
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