CSE190 3DUI
- Winter 2013

CSE 190: 3D User Interaction

Lecture #7: Selection

Jürgen P. Schulze, Ph.D.

Announcements

- Homework assignment #2 due Friday, February 8th at 1pm in Sequoia lab 142
  - This time grading starts at 12 noon
- Reminder: recommended reading

Augmented Reality Demo

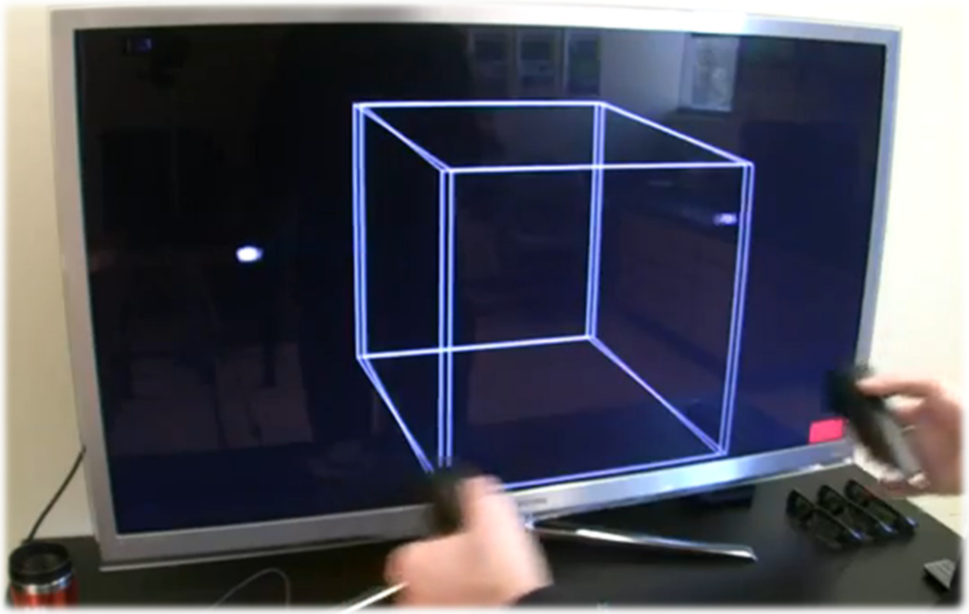
- Web app: http://www.nature.com/naturejobs/science/profiles/kaust-ar

Razer Hydra

- Developed by Sixense Entertainment
- Released June 16, 2011
- Tracks absolute position and orientation (6 DOF)
  - Precision: 1mm and 1 degree
- Uses a weak electro-magnetic field
- Two wired input devices

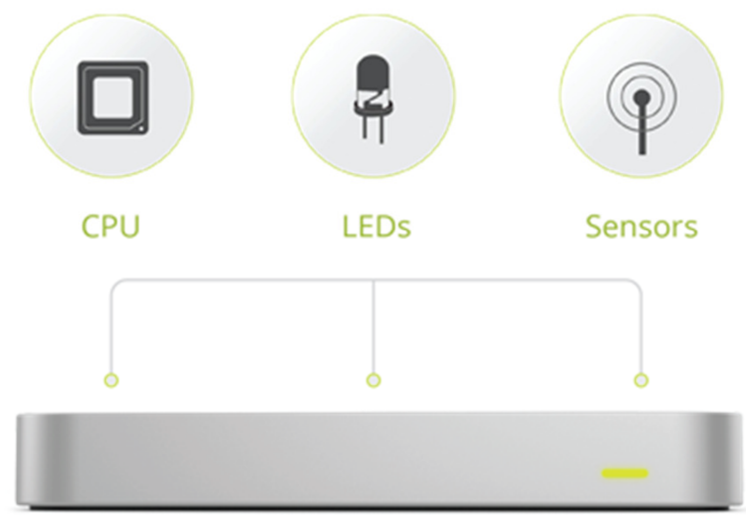
Razer Hydra Video

- Razer Hydra for low-cost 3D displays
  - By Oliver Kreylos, UCD
  - http://www.youtube.com/watch?v=H5bSzVByLjM



Leap Motion

- Short range finger tracking
  - To date no access to depth map
- Inexpensive ($70 on pre-order)
- Not yet available (promised for early 2014)
- SDK available today
- Developer units being shipped
- More and more demo videos available



Leap Video

- https://www.leapmotion.com/

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

Selection & Manipulation

- Selection: specifying one or more objects from a set
- Manipulation: modifying object properties (<u>position</u>, <u>orientation</u>, 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
  - Presence of occluding objects

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

Input Device Parameters

- Number of control dimensions
- Control integration: how many DOF are controlled simultaneously
- Force vs. position control
- Form factor: impact on accuracy

Sensor attached to hand

Sensor rolled with fingers

Technique Classification by Metaphor

- Manipulation techniques
  - Egocentric metaphor
    - Virtual pointer metaphor
      - Ray-casting
      - Two-handed pointing
      - Flashlight
      - Image plane
    - Direct manipulation
      - “Classical” virtual hand
      - Go-Go
    - Hybrid techniques
      - HOMER
  - Exocentric metaphor
    - World-in-miniature
    - Scaled-world grab
  - Hybrid techniques
    - Voodoo Dolls

Isomorphic vs. Non-Isomorphic View

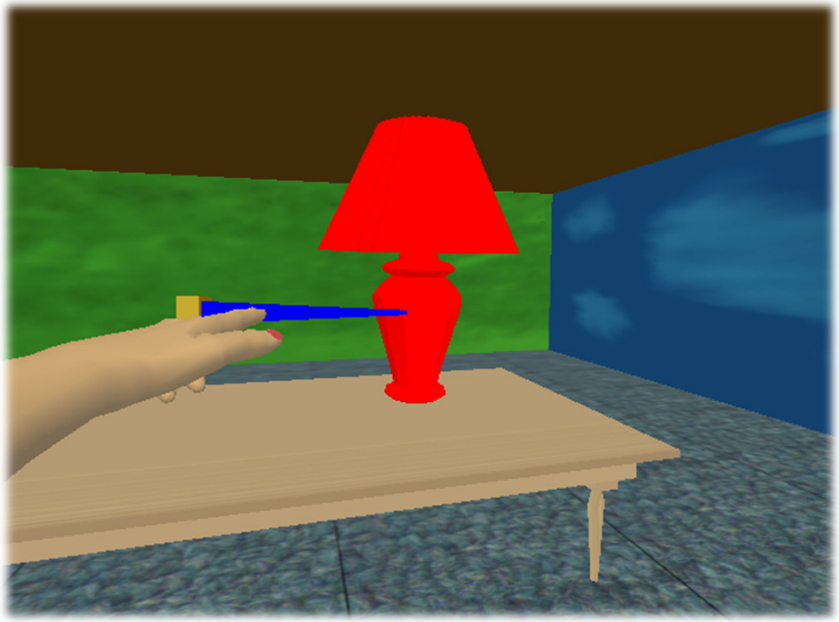
- Isomorphic
  - Geometrical on-to-one correspondence between hand motions in physical and virtual worlds
  - Natural interactions
- Non-Isomorphic
  - “Magic” virtual tools (laser beams, rubber arms, etc.)

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

Ray-Casting

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



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

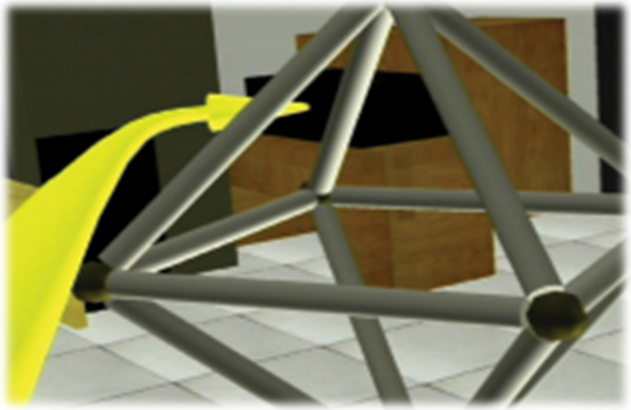
where $0 < \alpha < \infty$, determined by first object intersection

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

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

Two-Handed Pointing

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



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

where $0 < \alpha < \infty$, fixed parameter

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

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

Flashlight

- Soft selection technique
  - 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

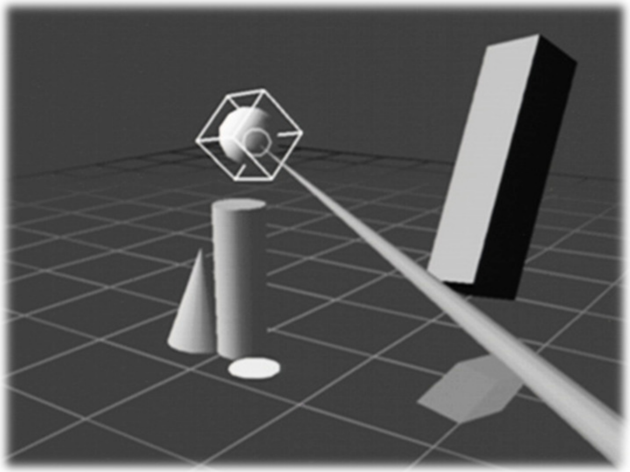
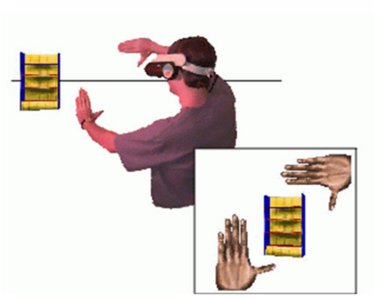


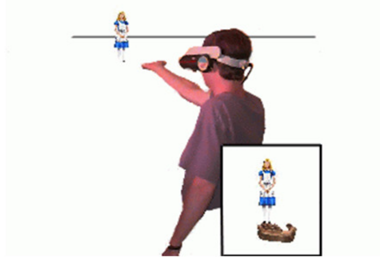
Image Plane Techniques

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

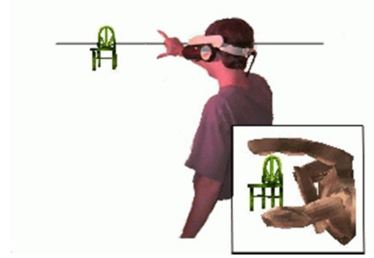
Framing



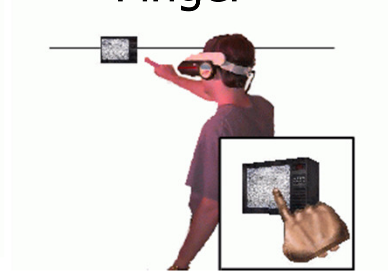
Lifting Palms



Head-Crusher

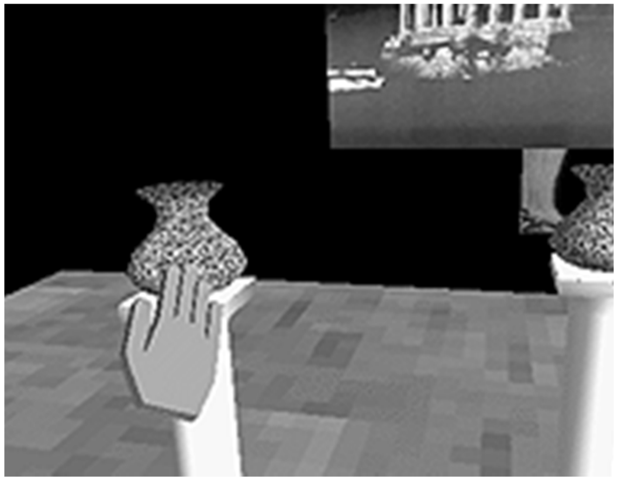


Sticky Finger



Virtual Hand

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



$$\mathbf{p}_\mathbf{v} = \alpha \cdot \mathbf{p}_\mathbf{r}, \mathbf{R}_\mathbf{v} = \mathbf{R}_\mathbf{r}$$

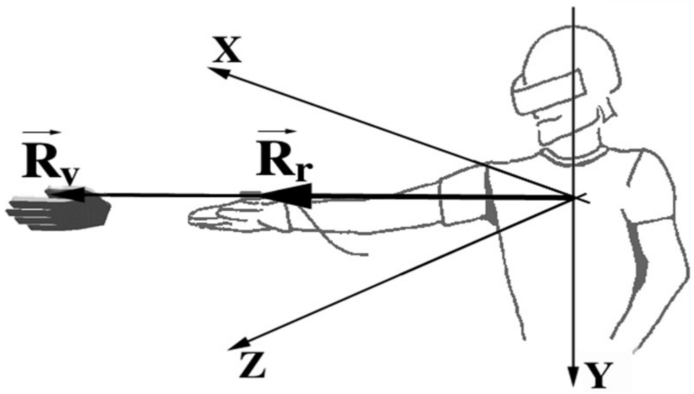
$\mathbf{p}_\mathbf{r}, \mathbf{R}_\mathbf{r}$ = position and orientation of real hand

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

α = fixed scaling factor

Go-Go

- By 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

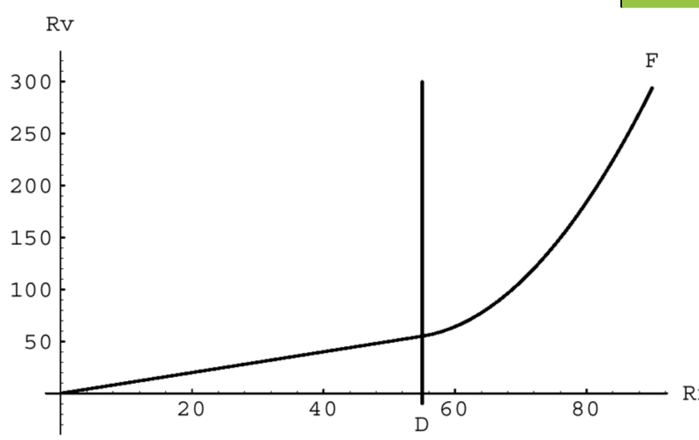


$$r_v = F(r_r) = \begin{cases} r_r & \text{if } r_r \leq D \\ r_r + \alpha(r_r - D)^2 & \text{otherwise} \end{cases}$$

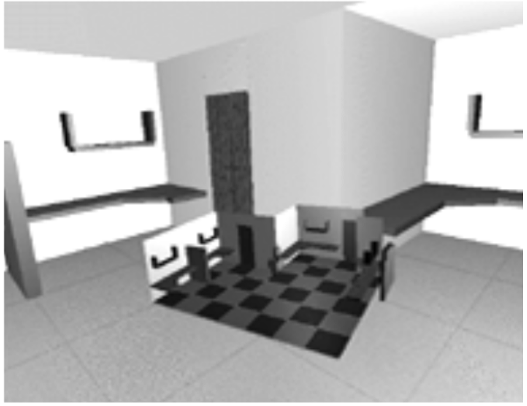
where r_r = length of $\vec{\mathbf{R}}_\mathbf{r}$

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

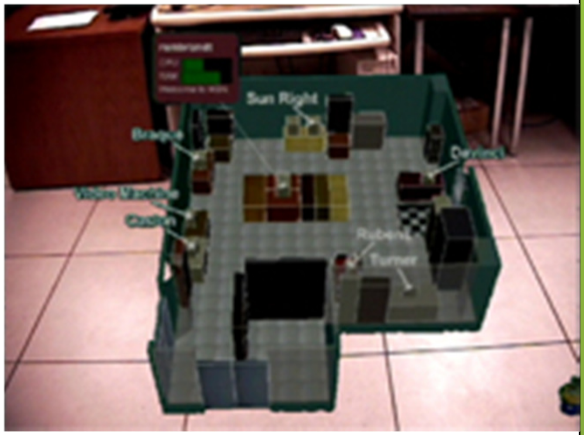
D, α are constants



World-in-Miniature (WIM)



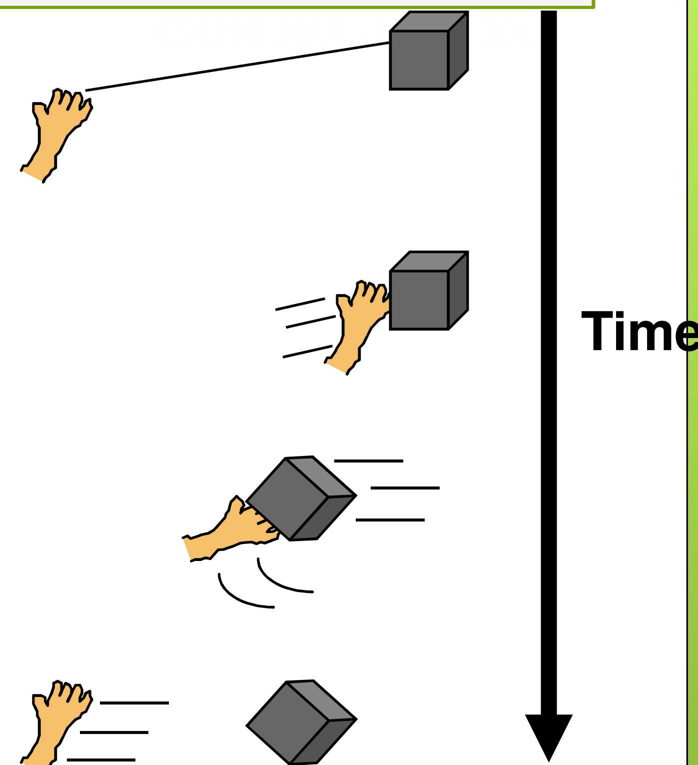
- By Stoakley, 1995
- "Dollhouse" world held in user's hand
- Miniature objects can be manipulated directly
- Moving miniature objects affects full-scale objects
- Can also be used for navigation

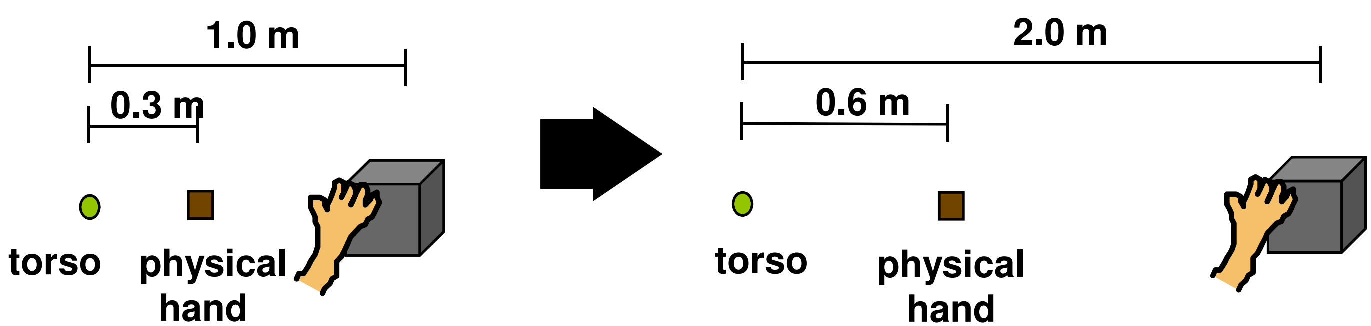


HOMER

Hand-Centered
Object
Manipulation
Extending
Ray-Casting

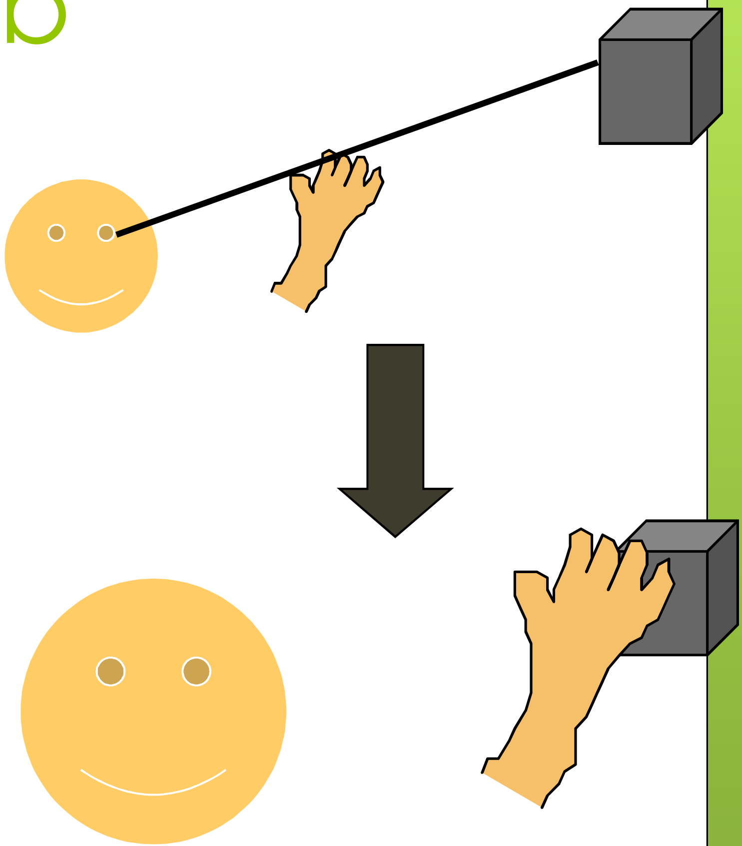
- By Bowman/Hodges, 1997
- Select: ray-casting
- Manipulate: hand-centered





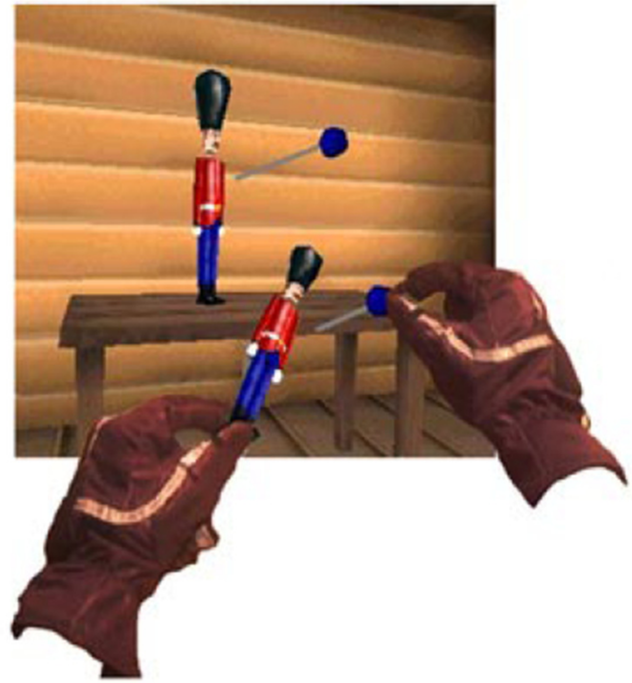
Scaled-World Grab

- 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



Voodoo Dolls

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



Rotational Mappings

- Most listed techniques deal only with selection and translation
- Many do not work well for rotations
- Rotation options:
  - Direct mapping of object rotation to rotation of device
    - Can cause clutching: repeated grabbing and releasing of object to rotate further than wrist allows
    - Tracking jitter can make small rotations difficult
  - Rotation amplification or slow-down

Rotation Calculations

- Simplest way to calculate rotations: Euler angles
- Euler angles define rotation by 3 rotations about coordinate axes
- Typical problem with Euler angles: Gimbal Lock, occurs in certain object orientations
  - Video (play until 1:12)
    - http://www.youtube.com/watch?v=zc8b2Jo7mno
- Better than Euler angles: 4x4 rotation matrices
  - Problem: 16 numbers required to specify rotation
- Quaternions: greatly improve rotation calculations

Quaternions

- OSG defines mathematical operators for quaternions to add, subtract, multiply, etc.
- In OSG, quaternions can be specified by rotation angle and axis:
  - `osg::Quat(value_type angle, const Vec3d &axis)`
- Or mathematically:
  - `osg::Quat(value_type x, value_type y, value_type z, value_type w)`

Quaternion Definition

- [w, x, y, z]
  - w = cos(a/2)
  - x = sin(a/2) * nx
  - y = sin(a/2) * ny
  - z = sin(a/2) * nz
- a: angle of rotation
- {nx,ny,nz}: normalized axis of rotation

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

- Quaternions in Ogre3D:
  - http://www.ogre3d.org/tikiwiki/Quaternion+and+Rotation+Primer
- Quaternions in OSG:
  - http://www.openscenegraph.org/projects/osg/wiki/Support/Maths/QuaternionMaths