Virtual Reality Technology and Programming

TNM053: Lecture 10: Interaction Techniques for VR

2D and 3D interaction

- 2D interaction
  - Straightforward since commonly used
  - Pointing device controls interaction point
  - Pick 2D objects as regions of plane
  - Measure 2D interactions
- 3D interaction with 2D pointer
  - All of 2D but in 3D context - object picking
  - Difficulties resolving depth and selected objects
  - Navigation - Control of the camera

VR interaction

- Lots of similarity to 3D
- Difference due to immersion
  - More interaction mechanisms available
- Added ‘natural’ responses
  - due to ‘presence’
  - Mutual dependence on presence

Fundamental forms of interaction

- Movement (of camera)
- Selection:
  - of objects
  - of controls (widgets)
- Manipulation (of objects)
- Scaling of the scene
- Virtual menu and widget interaction

Natural interaction

- Immersive display
  - should give us ‘real world’ interaction
- In reality they don’t:
  - Most displays externalize the world
  - Workbench and wall are just like monitors
  - Caves are more immersive
    - Usually cave surrounded by the scene, not in it!
- Only tracked HMD meets requirements of full immersion

Wall/Workbench display

- Should give us ‘real world’ interaction
- In reality they don’t:
  - Most displays externalize the world
  - Workbench and wall are just like monitors
  - Caves are more immersive
    - Usually cave surrounded by the scene, not in it!
- Only tracked HMD meets requirements of full immersion
Wall/ Workbench display

Cave - more immersive

Only HMD’s are fully immersive but...

- They have limitations in display...
  - Slow updates leading to disorientation
  - Limited field of view
  - Cybersickness
- ...and in tracking...
  - Range limited to a few metres or less
  - Usually no less limited than a cave
- ...and in cables for the displays

Movement (Navigation)

- Essential to control camera point
- Head tracking
  - Used in motion parallax
  - And in limited head motion
- Camera translation – moving the box
  - Use navigation metaphors
  - Direction and speed control required

Movement - direction

- Gaze-directed
- Hand-directed
- Physical controls
- Virtual controls

Gaze-directed motion

- Move in the direction of view
  - Feels quite natural
- Problems with navigation of view
  - Head and eyes are rarely aligned
  - Can’t watch the scene go by
- Rarely used in VR
  - Common in 3D (game) environments
    - Well-defined view direction
Hand-directed motion

- Pointing mode – a driving metaphor
  - User points in desired direction
    - Track hand directly (glove)
  - 5DOF required
- ‘Crosshair’ mode:
  - Use head/hand axis to move
  - Supposed to make it easier for a novice
  - Makes it hard to move and look sideways
  - Can use 3DOF devices

Tracking direction

Physical controls

- Physical (not tracked) devices
  - Buttons, knobs and sliders
    - Once very common (cheap and easy)
  - Not realistic - Lacks a natural mapping
- Realistic – part of the virtual world:
  - Steering wheels
  - Handlebars
  - Joysticks
- Has scope for force feedback
- Current exjobb on this in car simulation

Virtual Controls

- Instead of physical devices, use virtual
  - Virtual steering wheel
  - Virtual ‘flight-stick’
- Place where you want in the scene
- Hard to interact with
  - How do you use a steering wheel if you can’t grip it directly?
  - Harder to incorporate force-feedback

Controlling Speed

- Constant speed
- Constant acceleration
- Controlled speed
- Controlled acceleration

Constant speed

- What speed is the right speed?
  - Close up needs slow movement
  - Examination needs slow movement
  - Navigation typically calls for higher speed
- Not widely used
Constant acceleration

- Start slow and accelerate under...
  - Button control?
  - Hand control?
- Good for allowing user to...
  - Examine small details close up
  - Navigate across long distances
- Tendency to overshoot
  - Needs good depth cueing

Controlled speed or acceleration

- Provide access to a range of speeds
  - What range?
- Controlled how?
  - Hand controlled
  - Physical controls

Hand controlled speed/ acceleration

- E.g. Distance head-hand determines it
  - Used with 'crosshairs' direction control

Hand controlled speed/ acceleration

- Works well
  - Intuitive
  - Natural mapping to the virtual world
- Relies on proprioceptive sensation
  - Not very precise
  - Needs large ‘dead zone’
  - Limited dynamic range

Major problems with hand-related navigation

- Fatigue
  - Large-arm movement for navigation
  - Long periods of use (hours)
- Must be able to turn off navigation
- Long periods of use require other methods
  - Usually physical devices

Object driven navigation

- Build objects into the scene to move:
  - Lifts
  - Moving walkways/stairways
  - 'stepping discs' (teleporters)
- More exotic:
  - Attractors
  - Repellors
- Must know where people want to go
  - Virtual architecture?
### Goal driven navigation

- Drive the view through a set of defined points
  - List or virtual map
- Requires that you know what is ‘interesting’ in the scene
  - Objects
  - Routes (passing interesting objects)

### Goal driven navigation (2)

- User moves by selecting next point of interest:
  - From a list? (as used in VRML)
  - By selecting an object and moving to corresponding point?
- Teleport? – less ‘presence’
- Compute shortest (quickest) route to selected destination and auto-move

### Orbital mode

- Has specialized uses (in HMD or cave)
- Rotate using head orientation:
  - Fix object in front of view
  - move around it by turning and tilting
- Allows full examination of a single scene object (or cluster) without moving or navigating
- Could still zoom

### Orbital mode (2)

### Navigation: Summary

- General applications
  - mostly rely on hand (crosshair) control
    - For both direction and speed
    - most natural(?)
- Real applications usually use a mixture
- Extended use needs physical controls

### Interaction with the world

- Object selection
  - Picking objects
  - Manipulating selected objects
- Control interaction
  - Widgets: defining and using
Object selection in 3D graphics

Object selection in VR

Transform schemes

- Hardest part of using systems like this is the mass of coordinate transforms
- Rendering generates many transforms
- VR generates many more
  - stereo
  - tracking devices

Transformations

- Very complex layered transformation
  - Much difficulty hidden by modern systems
  - Stereo hidden in graphics library
  - Interactor results hidden from programmer
- You only have two frames
  - World
  - ‘CAVE’
**glLoadIdentity**

- Don’t!
- It undoes some of the CAVE initialization for the display function
- Moves the display into the wrong transform

**Selection**

- User probably needs to select objects
- How to specify selected object?
- Close range
  - Within ‘arms-reach’
  - Maybe user can move around the objects
- Long range
  - Outside natural arms reach
  - On the other side of the screen
  - More complex selection scheme required

**Object picking**

- Requires object intersection testing
  - Local probe with object
  - Remote ‘beam’ with object
- Simplest methods:
  - Range checking (from centroid)
  - Object plane intersection testing
  - Bounding box intersection

**Intersection testing**

- Within a coordinate frame
- Use vector mathematics to compute intersections

**Selecting a sphere**

**Facet intersection testing**

- In general we cannot use such a simple scheme
- We must test for intersection by more complex methods
- Must test for intersection using testing on the polygons making up the object
  - or a bounding box?
Facet intersection testing
- 3D space!
  
  Repeat for every triangle in the object!
  Can ignore back-faces with respect to pointer

Bounding box selection
- Can use (3D) bounding box to test
- Box must fit appropriately.

Overlaps
- What if your wand vector goes through many objects?
  - Select them all?
  - Select the nearest?
    - How?
      - Sort objects by distance and test?
      - Test all and sort intersections by distance?
      - Other?

Scene graphs!
- The scene graph knows everything about the objects in the 3D scene at one time
  - Can depth sort them for you
  - Can work out the intersections for you
  - Can use hierarchy to define appropriate bounding boxes for you
  - Can define which objects can and which can not be 'picked'

Selection approaches
- How to interact with the scene
  - Close up
  - Far
  - Sparsely populated scenes
  - Packed scenes

Close-range selection
- Bring hand/cursor to object
- Very natural mode of interaction
- Avoid problem of close-spaced objects
  - Can move easily to the objects
  - Reach into the space
  - Place pointer precisely within/on the object
**Long-range selection**
- Must select at a distance
- Selection requires a pointer
  - Could use gaze-directed selection
- What about multiple close-spaced objects at similar distance?
  - Problems determining object closest to pointer
  - Problems of occlusion of distant objects
- What about selecting multiple objects?

**Long range selection (2)**
- Simple in clear scene
- In complex scene can get problematic
- Other select/manipulate schemes needed like:
  - Fixed length pointer
  - Navigable ‘drone’

**Long range selection (3)**
- Long range selection requires:
  - Highlighting of current object
  - Clear view of objects and probes
- Gravity?
  - Pulls selector to specific objects
  - Allows user to move between objects
- Selection schemes application specific

**Manipulation**
- Having selected objects user might want to manipulate them
  - Rotate
  - Translate
- Centre of rotation
  - About object (centroid)
  - About virtual point of contact
- Again, what about action at a distance?

**Manipulation: Close range**
- Make centres of action at ‘hand’
  - Select object
  - Move ‘hand’ in 6DOF
  - Rotate and translate object accordingly
- Very natural, very intuitive

**Manipulation: Long range**
- What defines the centre of action?
  - Centroid of object?
  - Position of selection point?
- What defines the centre of rotation?
  - Position of selection?
    - Maybe ok for moving an object
    - Doesn’t allow for much rotation
  - Position of hand?
    - Better for rotation
    - Relies on the mouse metaphor
Scaling

- In addition we want to be able to scale
  - Expand around object of interest
- Use hand point and scale around that
  - Natural and easy
- Use selected object and scale about it
  - Use hand point as origin for 3D mouse

Manipulation: Summary

- Quite a few modes of action
- Long range manipulation is a problem
  - Especially in control of rotation
- Most applications use a mixture of these modes

Virtual widgets

- Analogous to widget sets used in 2D interfaces
- No limit on what can be used as a widget
  - Any imaginable object
  - Multi-dimensional menus are possible
  - No real guidance yet
    - Most interfaces look very familiar, like 2D

Floating menus

COVI SE in four-wall cave

Menu dimensionality

- 1D movement of device selects from a circular list
  - Twist
  - Turn
  - Spin

- Could be very natural in a tracked glove environment
  - Gesture driven – Johnny Mnemonic
- Could use other shapes for additional items
  - Sphere? Cylinder? Cube?
  - Hierarchical?
Interface problems

- Occlusion of the display
  - Large displayed widgets block display
    - Resolution problems
  - Good placement vital
  - Context-dependent pop-ups useful
- Bad display damages sense of presence
- Distance problems:
  - Change of focal distance
  - Tiring for the user’s eyes

3D palette

Palm device interfaces

- (Bluetooth?) wireless devices
- Acts as a palette for widgets
- Added interface for interaction
- Tracked
- Tablet PC?
  - With stereo?

Case Study: Air Traffic Management

- Wish to view air traffic flow
- Wish to modify planned routes (waypoints)
- Need to be able to examine
  - Space around airport
  - Space around a specific aircraft
  - Space around a specific waypoint
  - Space around an arbitrary point
Navigational needs

- Select location of points of interest:
  - Airport
  - Aircraft
  - Arbitrary point in the scene
- Navigate around point:
  - Rotate in 2DOF (L-R/U-D)
  - Zoom in and out
  - Scale display (exaggerate altitude)

Navigational modes

Rotation around point

- Camera point transition
  - Made within the scene
  - Long range jumps made by zooming out
  - Instantaneous transition within view
- Camera rotation
  - Use Physical rotation of pointer device
  - Use angling of pointer device

Selection needs

- Selection of navigational points
  - Airport – List
  - Aircraft – Select with pointing device
  - Arbitrary point – use surface crosshair
- Selection of interactive points
  - Select waypoints
  - Insert new ones

Selection of waypoints

- Use wand pointer
- Use ‘gravity’
  - Snap to close point (waypoint/aeroplane)
- Click and hold to move selected point
- Double-click and hold to insert point

Final interface

Select view point:
- Engage wand
- Select new centre of rotation:
  - Scene point
  - Waypoint
  - Aircraft

Select and move waypoint

Navigate

Joystick

Zoom
ATM: Additional feedback

- Aircraft and waypoints have additional information which must be presented
- Achieved through pop-up window
  - Currently at fixed depth in the scene
  - Not very satisfactory - tiring for the eyes

Minority Report

- Stunning VR interface
- Actually 2D but 3D interaction
- Uses hand and head (eye?) tracking
- Uses gestures
- Uses modal virtual widgets
- Also demonstrates the importance of being able to turn the interface off!

In the year 2054 a murder is about to be committed...

MR: Selection Metaphor

MR: Gestures
Summary

- Navigation and interaction using VR features is tricky:
  - Many possibilities...
  - ...But few guidelines
- Lots of potential problems:
  - Fatigue
  - Complex navigation points
- Important to analyze application
  - Navigational requirements
  - Interaction requirements
  - Features in the application display