3D User Interface Travel Techniques

Lecture #9: Navigation I – Travel

Spring 2009

Joseph J. LaViola Jr.

Universal 3D Interaction Tasks

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

Travel

- The motor component of navigation
- Movement between 2 locations, setting the position (and orientation) of the user's viewpoint
- The most basic and common VE interaction technique, used in almost any large-scale VE

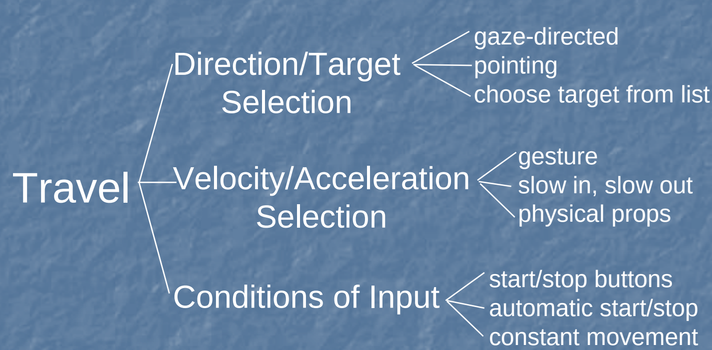
Travel Tasks

- Exploration
  - travel which has no specific target
  - build knowledge of environment
- Search
  - naïve: travel to find a target whose position is not known
  - primed: travel to a target whose position is known
  - build layout knowledge; move to task location
- Maneuvering
  - travel to position viewpoint for task
  - short, precise movements

Travel Characteristics

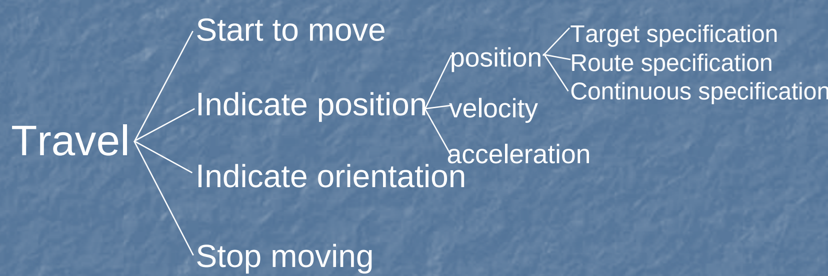
- Travel distance
- Amount of curvature/number of turns in path
- Target visibility
- DOF required
- Accuracy required
- Other tasks during travel
- Active vs. passive
- Physical vs. virtual

A Technique Classification – Component Decomposition



Spring 2009 CAP6938 – 3D User Interfaces for Games and Virtual Reality ©Joseph J. LaViola Jr.

Alternate Technique Classification – User Control Level



Travel Techniques

- Physical locomotion ("natural" metaphors)
- Steering techniques
- Route planning
- Target-based techniques
- Manual manipulation
- Viewpoint orientation techniques

Physical Locomotion Techniques

- Walking techniques
  - large-scale tracking
  - Walking in place (GAITER)
- Treadmills
  - single-direction with steering
  - omni-directional
- Bicycles
- Other physical motion techniques
  - VMC / Magic carpet
  - Disney's river raft ride

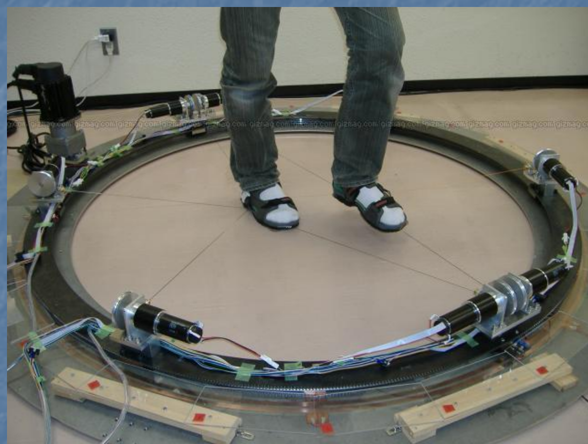
Physical Locomotion Devices (I)

Omni-Directional Treadmill

GaitMaster II

Large Scale Tracking

Physical Locomotion Devices (II)



String Walker

Steering Techniques

- continuous specification of direction of motion
  - gaze-directed
  - pointing
  - torso-directed
  - camera-in-hand
  - semi-automated
  - physical device (steering wheel, flight stick)

Steering – Gaze-Directed

- Move viewpoint in direction of "gaze"
- Gaze direction determined from head tracker
- Cognitively simple
- Doesn't allow user to look to the side while traveling

Steering – Gaze-Directed Implementation

- Each frame while moving:
  - Get head tracker information
  - Transform vector *[0,0,-1]* in head CS to *v=[x,y,z]* in world CS
  - Normalize v: $\hat{v} = \frac{v}{\|v\|}$
  - Translate viewpoint by $(\hat{v}_x, \hat{v}_y, \hat{v}_z) \times current\_velocity$

Pointing Technique

- Also a steering technique
- Use hand tracker instead of head tracker
- Slightly more complex, cognitively
- Allows travel and gaze in different directions – good for relative motion

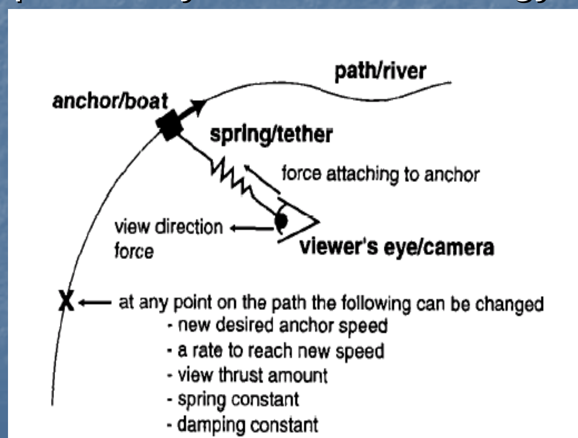
Pointing Implementation

- Each frame while moving:
  - Get hand tracker information
  - Transform vector *[0,0,-1]* in hand CS to *v=[x,y,z]* in world CS
  - Normalize v: $\hat{v} = \frac{v}{\|v\|}$
  - Translate viewpoint by $(\hat{v}_x, \hat{v}_y, \hat{v}_z) \times current\_velocity$

Spring 2009 CAP6938 – 3D User Interfaces for Games and Virtual Reality
©Joseph J. LaViola Jr.

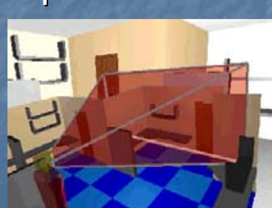
Semi-Automated Travel

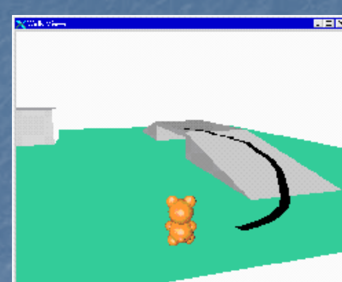
- Example – Galyean's river analogy (1995)

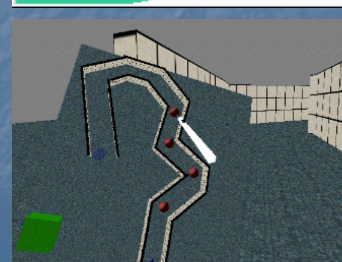


Route-Planning

- one-time specification of path
  - draw path
  - points along path
  - manipulating user representation





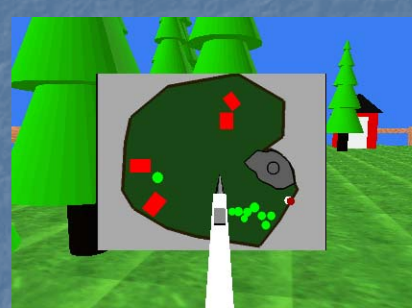


Target-Based Techniques

- discrete specification of goal
  - point at object
  - choose from list
  - enter coordinates
- Map/WIM-based target specification

Map-Based Travel Technique

- User represented by icon on 2D map
- Drag icon with stylus to new location on map
- When released, viewpoint animated smoothly to new location



Map-based Travel Implementation

- Must know
  - map scale relative to world: s
  - location of world origin in map CS: $o=(x_o, y_o, z_o)$
- On button press:
  - if stylus intersects user icon, then each frame:
    - get stylus position in map CS: (x, y, z)
    - move icon to $(x, 0, z)$ in map CS

Map-Based Travel Implementation (cont.)

- On button release:
  - Get stylus position in map CS: (x, y, z)
  - Move icon to $(x, 0, z)$ in map CS
  - Desired viewpoint: $p_v = (x_v, y_v, z_v)$ where
    - $x_v = (x - x_o)/s$
    - $z_v = (z - z_o)/s$
    - y_v = *desired height at* (x_v, y_v)
  - Move vector: $m = (x_v\text{-}x_{curr}, y_v\text{-}y_{curr}, z_v\text{-}z_{curr})$ * *(velocity/distance)*
  - Each frame for *(distance/velocity)* frames: translate viewpoint by m

Spring 2009 CAP6938 – 3D User Interfaces for Games and Virtual Reality ©Joseph J. LaViola Jr.

Manual Manipulation – Grabbing the Air Technique

- Use hand gestures to move yourself through the world
- Metaphor of pulling a rope
- Often a 2-handed technique
- May be implemented using Pinch Gloves™

Grabbing The Air Implementation (one-handed)

- On pinch:
  - Obtain initial hand position in world CS: (x_h, y_h, z_h)
- Each frame until release:
  - Obtain current hand position in world CS: (x'_h, y'_h, z'_h)
  - Hand motion vector: $m = ((x'_h, y'_h, z'_h) - (x_h, y_h, z_h))$
  - Translate world by m (or viewpoint by $-m$)
  - $(x_h, y_h, z_h) = (x'_h, y'_h, z'_h)$
- Cannot simply attach objects to hand – do not want to match hand rotations

Viewpoint Orientation Techniques

- Head tracking
- Orbital viewing
- Non-isomorphic rotation
- Virtual sphere

Next Class

- Navigation – Wayfinding
- Readings
  - 3DUI Book – Chapter 6

