Multi-user architectures

Peer-to-Peer networking for small numbers of users
one peer acts as host, others are “non-hosts”.
hosts manages the session info – who is connected
need to reset host when current host-peer leaves

Client-Server networking for larger number of users
server “host” manages connect and session information on clients
clients connect and interact via the server
**AR / VR devices**

*HMD w/ tracking, stereo viewing, haptic (glove) input, networking*

Use high level graphics libraries {OpenGL, DirectX, Java 3D}

---

**Net based render loop**

- **Create & connect** (object & action db)
- **Read sensor values**
- **Process time dependent actions**
- **Receive & apply network events**
- **Render world (graphics pipeline)**

**Frame rate**

**network events**

**shared virtual environment server**
Critical Section – Matrix

events “fire” when packets arrive possible race conditions from multiple events could try to update local information (threaded)

Critical Sections and Locks

A section of the code that can be accessed by multiple threads
Locks used for mutually exclusive access of the critical section

Whats in the packets?

Matrix struct
Player state information – player's "object id"

Performance issues –

lag up to 500 ms (half a second)
packet loss – UDP (fire, forget) default, TCP increases lag ...
bandwidth constraints – voip, packet headers, compression prediction && smoothing – reduce traffic, accuracy loss ...
Augmented reality, see through display with "overlay" 3D

Virtual Reality, 3D display with head and body tracking.

Wearable ➔ AR + fashion
more than glass or watches ...

chemistry AR demo

Maelstrom VR/AR Walking Demo

Epson moverio
3D glasses and HMDs

**Nvidia’s 3D Vision** usb IR emitter, shutter glasses

**Real D**
polarized 3D, cinema equipment

**Occulus VR** dev kit

**Google Cardboard**

**Sensics**
high end HMD
Dual Display Stereo Viewing

Scene is rendered from as a view from each eye (often separate buffers). Position, orientation, aspect ratio (X to Y), parallax (inner pupilary distance), convergence, convergence distance (perceived stereo), distance L R eye.

Convergence angle often not implemented.
Eye transforms

Possible physical body translations, convergence orientation = fn(distance) – based on Java3D API (PhysicalBody)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>eyes</td>
<td>-0.033, 0.0, 0.0</td>
<td>0.033, 0.0, 0.0</td>
<td></td>
</tr>
<tr>
<td>ears</td>
<td>-0.080, -0.030, 0.095</td>
<td>0.08, -0.30, -0.095</td>
<td></td>
</tr>
</tbody>
</table>

// Draw for each eye
Render {
  Matrix rightEye, leftEye
  Matrix reTranslation, leTranslation // eye translations
  Matrix reRotation, leRotation       // eye rotations
  set light(s)
  // for each eye
  rightEye = World * reTranslation * reRotation
  set View and Projection
  draw scene
  // repeat for left eye
Smart phone displays both left and right eye images on screen.

Viewer (with lens) focuses separate images for each eye.
Encapsulate input devices. Sensor instances are interchangeable. Recompile w/ new sensor instance in existing world to modify input...

Sensors have a lag, accuracy and frame rate (fps of system)

accuracy is the range of values for an event
± 0.1 units
for an \( n \) unit range

i-o glasses range:
360° yaw
± 60° pitch & roll

Linking sensors to viewpoints for head tracking devices
<< Natural Point's trackir >>
Head tracking with camera (regular || IR) 
tracking targets 
face recognition (eyes, shape of head) 

Search image (for pixel targets) / frame 

3 tracking target 
calibrate, store size and vector lengths of targets: b'.size, bc'.vec 
volume of target \(\leftarrow\) change in distance 
vectors between targets \(\leftarrow\) change in orientation 

\[
\begin{align*}
\text{b.size} &> \text{b'.size} \quad \& \quad \text{c.size} < \text{c'.size} \quad \& \quad \text{bc.vec} < \text{bc'.vec} \quad \& \quad \\
\text{a.size} &< \text{a'.size} \quad \Rightarrow \quad \text{head has turned (Y – yaw) right}
\end{align*}
\]

magnitude of the rotation is proportional to changes 
need to determine expressions and set weights in calibration 

face recognition 
eyes are "targets" (whites w/pupil pattern) w/in field (head)
Gestures

Gestures are motion based commands generated from connected and unconnected sensing devices.

Computer Vision, see next page
Gloves, 5dt.com
Leap controller
tracking references

**Wii controller head tracker**

![Image](image1.png)

**iPad tracker**  **iPad head tracker**

![Image](image2.png)

Open source camera vision (head tracking, etc)  **OpenCV**

[http://opencv.org/](http://opencv.org/)  
several recent books on OpenCV 2

FreeTrack is a free optical motion tracking application for Microsoft Windows, released under the GNU General Public License.


Trackir 4 – an inexpensive IR camera w/ free SDK (Opti-track SDK)