CSE 165 - Winter 2014

CSE 165: 3D User Interaction

Lecture #3: Stereo

Jürgen Schulze

Announcements

- Homework Assignment #1
  - Due date: January 24th at 1:30pm
  - To be presented in CSE lab 260
- Reminder: paper presentations
  - Email title + date to instructor by Friday 1/17
- Pickup Kinect in office hour after class

Overview continued

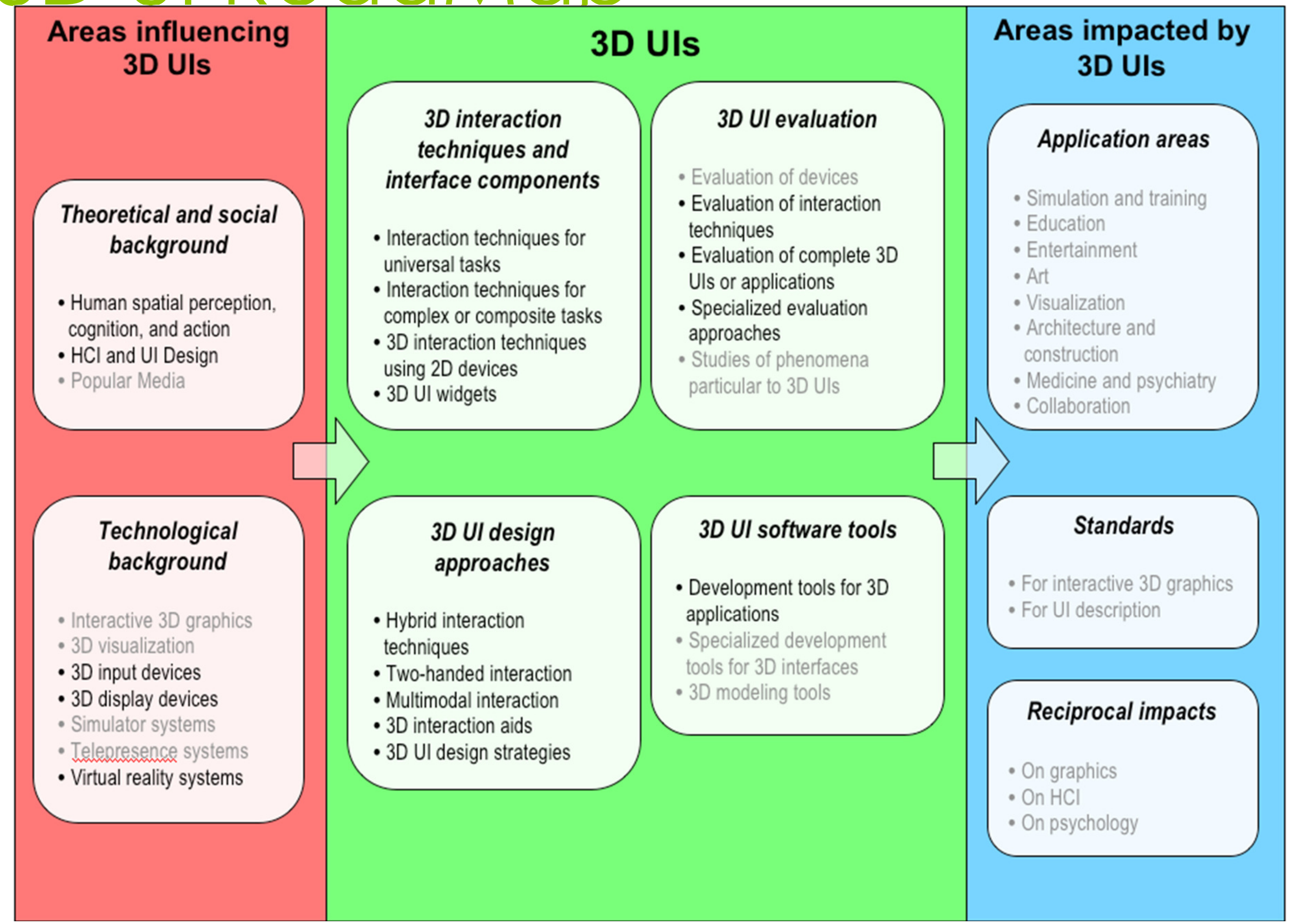
3D UI Design Philosophies

- Artistic approach: Base design decisions on
  - intuition about users, tasks, and environments
  - heuristics, metaphors, common Sense
  - aesthetics
  - adaptation/inversion of existing interfaces
- Scientific approach: Base design decisions on
  - formal characterization of users, tasks, and environments
  - quantitative evaluation results
  - performance requirements
  - examples: taxonomies, formal experimentation

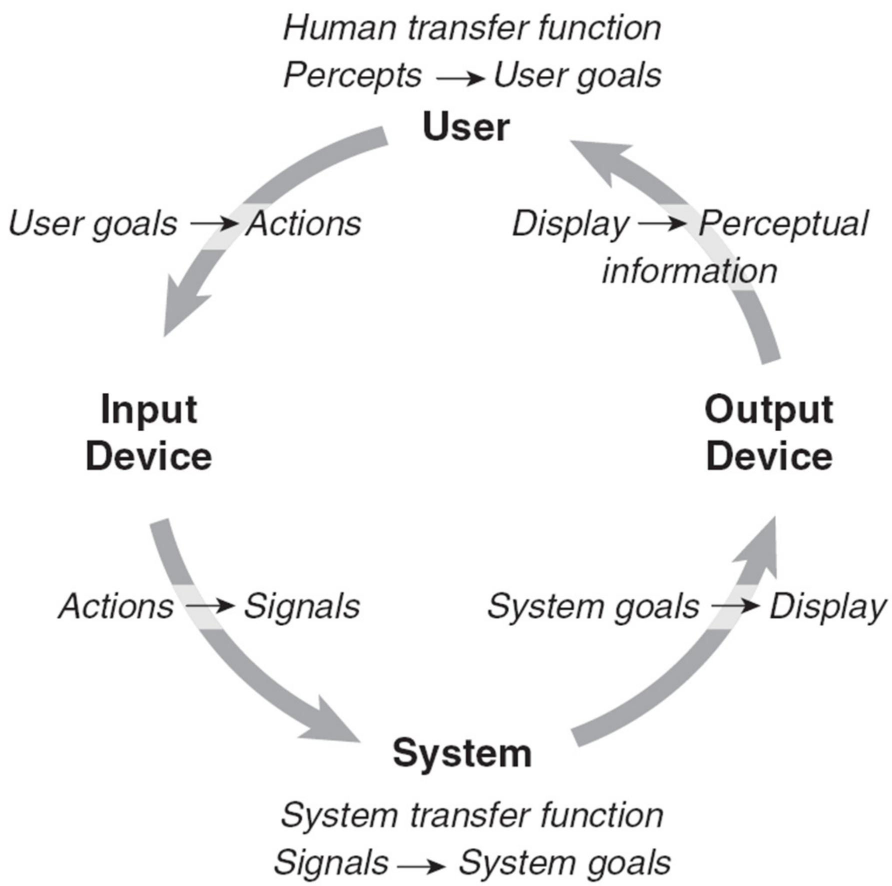
Applications

- Architecture / CAD
- Education
- Manufacturing
- Medicine
- Simulation / Training
- Entertainment – *Games!*
- Design / Prototyping
- Information / Scientific Visualization
- Collaboration / Communication

3D UI RoadMap



Interaction Workflow



Quaternions

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

3D Displays

Introduction To Displays

- *Display*: device which presents perceptual information
- Often 'display' used to mean 'visual display'
- Goal: display devices which accurately represent perceptions in simulated world

Lecture Outline

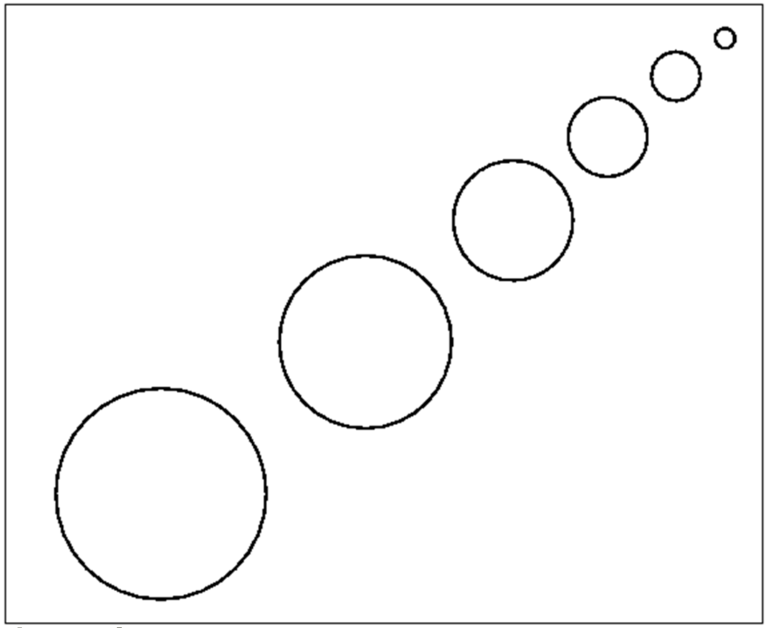
- Visual System
- Depth Cues
- Visual Display Characteristics
- Visual Display Examples
  - monitors
  - surround screen displays
  - workbenches
  - head mounted displays
  - arm-mounted displays
  - virtual retinal displays
  - autostereoscopic displays

Depth Cues – How Do We See 3D?

- Monocular/static cues
- Occulomotor cues (Accommodation/Convergence)
- Motion Parallax
- Binocular Disparity and Stereopsis

Monocular/Static Cues

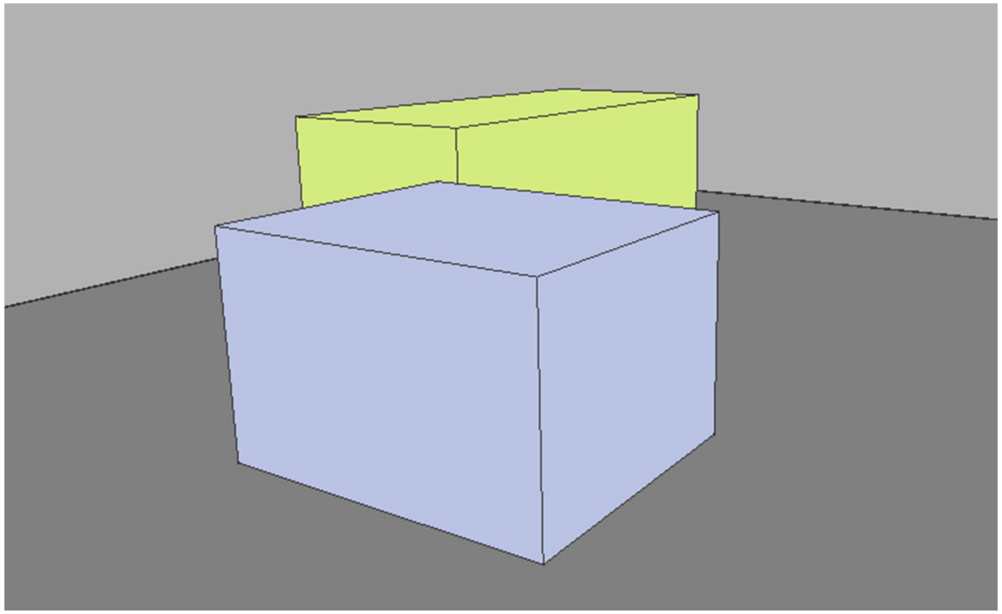
- Relative Size



- Height relative to horizon

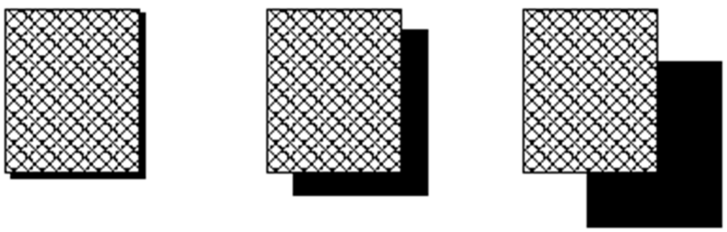
Monocular/Static Cues

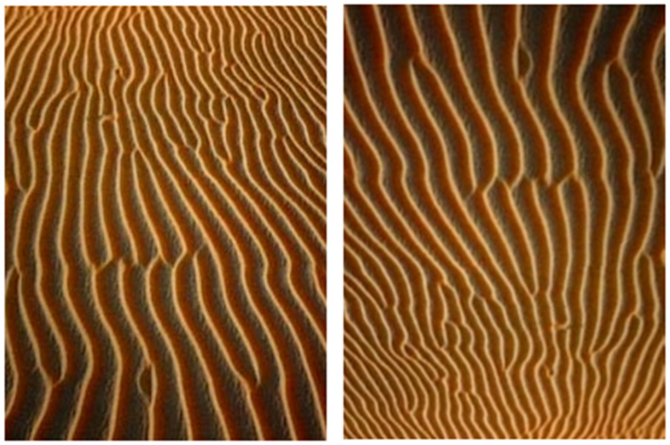
- Occlusion and Linear Perspective



Monocular/Static Cues

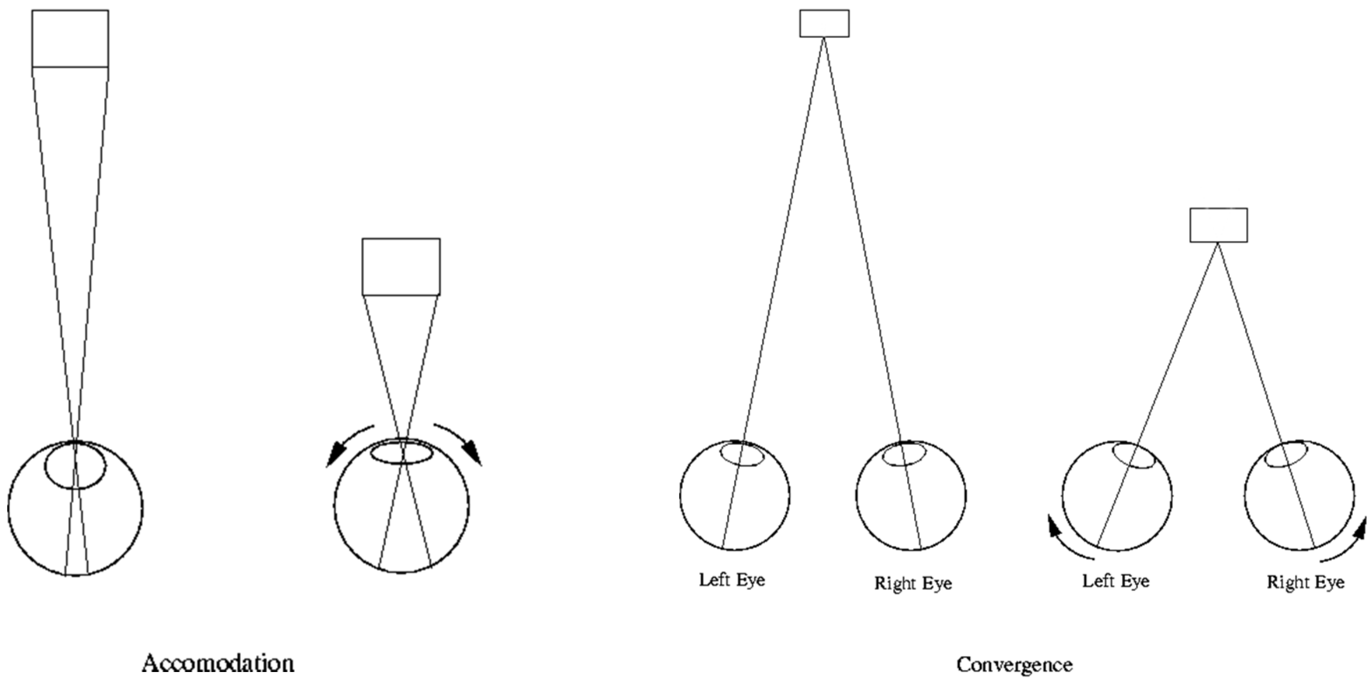
- Shading, Lighting, and Texture





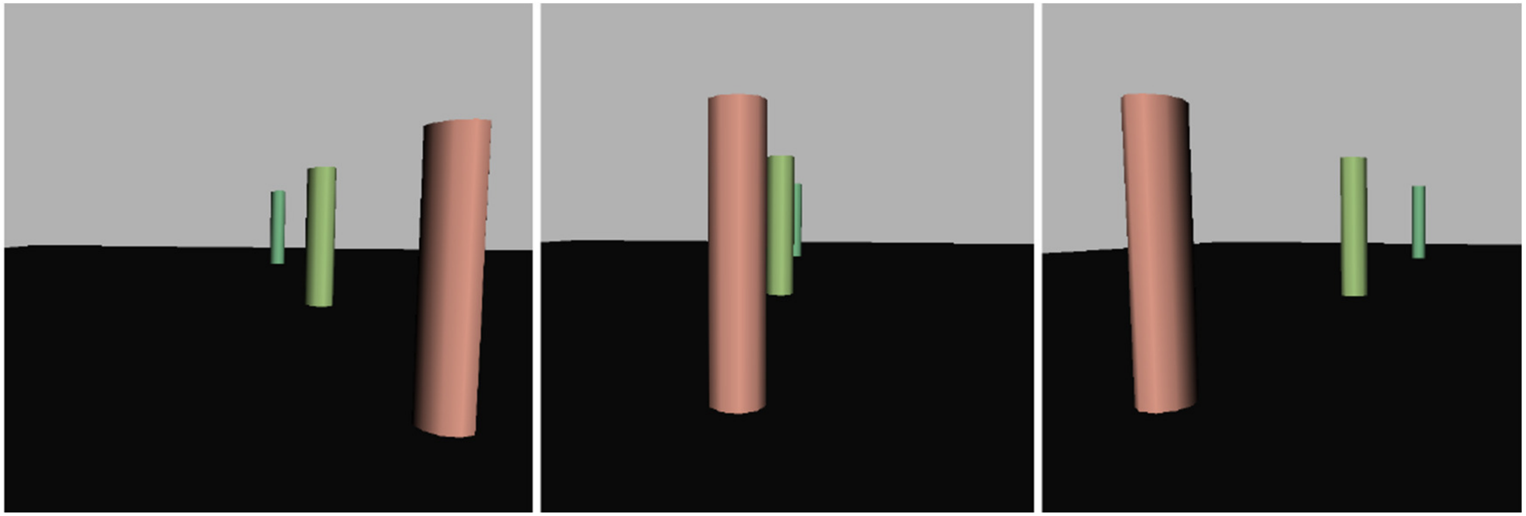
Oculomotor Cues

- Accommodation – physical stretching and relaxing of eye lens
- Convergence – rotation of viewer's eyes so images can be fused together at varying distances



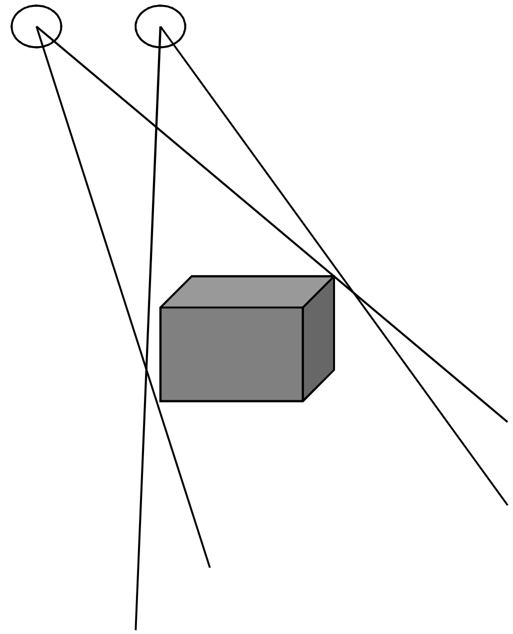
Motion Parallax

- Stationary viewer vs. moving viewer



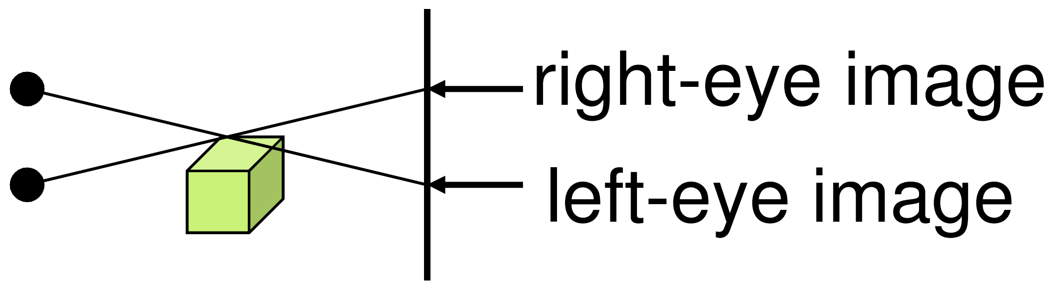
Binocular Disparity and Stereopsis

- Each eye gets a slightly different image
- Only effective within a few feet of viewer
- Many implementation schemes



Accommodation-Convergence Mismatch

- Standard stereo displays confuse the brain based on oculomotor cues



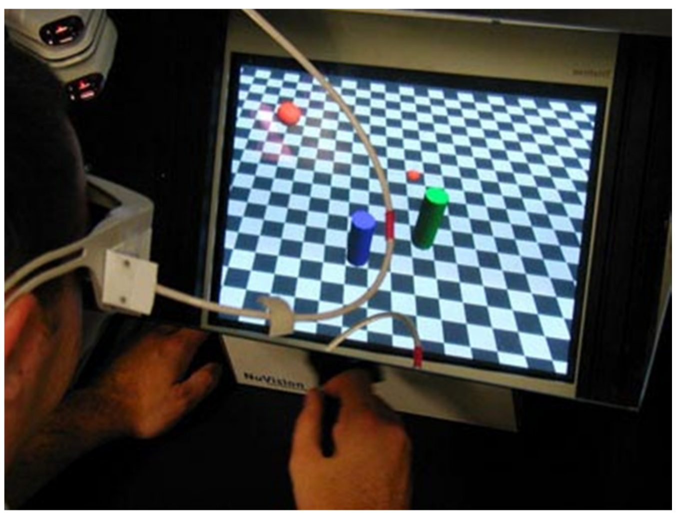
- Only "true 3D" displays can provide these correctly

Visual Display Characteristics

- Field of View (FOV) and Field of Regard (FOR)
  - FOR – amount of physical space surrounding viewer in which visual images appear
  - FOV – maximum visual angle seen instantaneously
- Spatial Resolution
  - number of pixels and screen size
- Screen Geometry
  - rectangular, hemispherical, etc…
- Light Transfer Mechanism
  - front projection, rear projection, laser light, etc…
- Refresh Rate
  - not the same as frame rate
- Ergonomics

Stereo Monitor

- Ordinary workstation equipped with emitter and shutter glasses



Stereo Monitor – Advantages

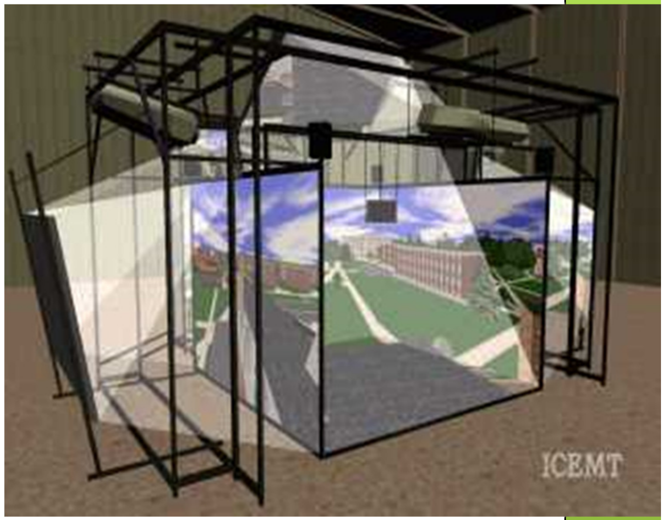
- Least expensive in terms of additional hardware over other output devices
- Allows usage of virtually any input device
- Good resolution
- User can take advantage of keyboard and mouse

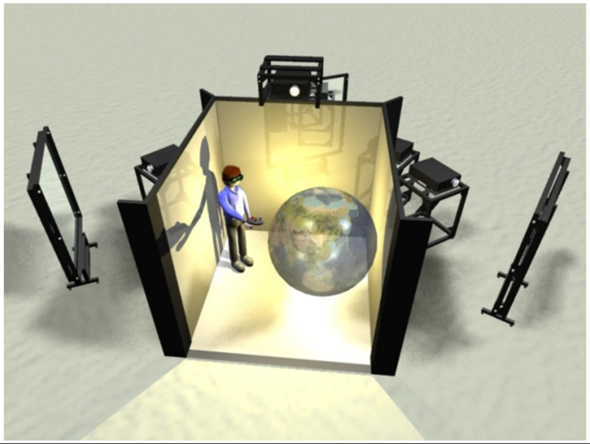
Stereo Monitor – Disadvantages

- Not very immersive
- User really cannot move around
- Does not take advantage of peripheral vision
- Stereo can be problematic
- Occlusion from physical objects can be problematic

Surround Screen VE

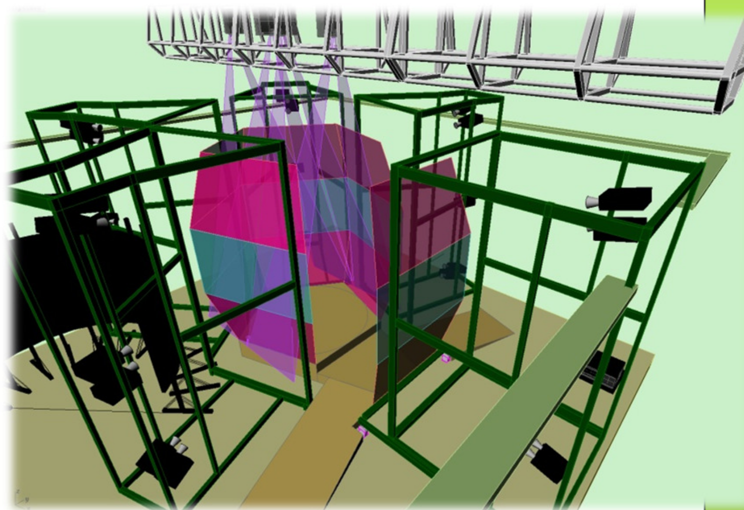
- Has 3 to 6 large screens
- Puts user in a room for visual immersion
- Usually driven by a single or group of powerful graphics engines

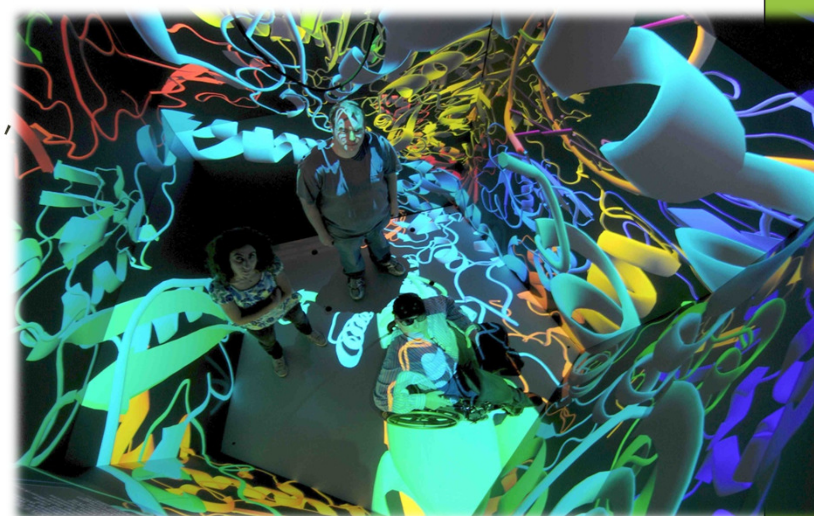




The StarCAVE

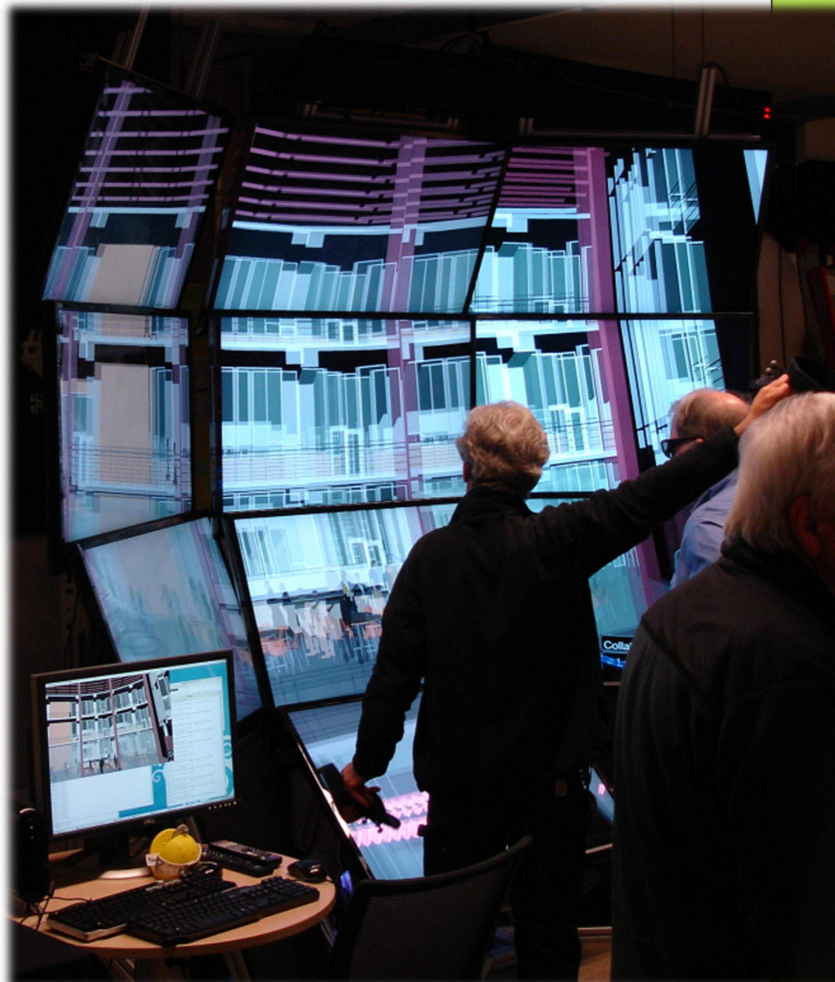
- 18 Dell XPS 710 PCs
- Dual Nvidia GeForce 285 graphics cards
- CentOS Linux
- 34 JVC HD2k projectors (1920x1080 pixels):
  ~34 megapixels per eye
- 360 degrees immersion
- Passive stereo, circular polarization
- 15 screens on 5 walls, ~8 x 4 foot each, plus floor projection
- 4-camera optical tracking system

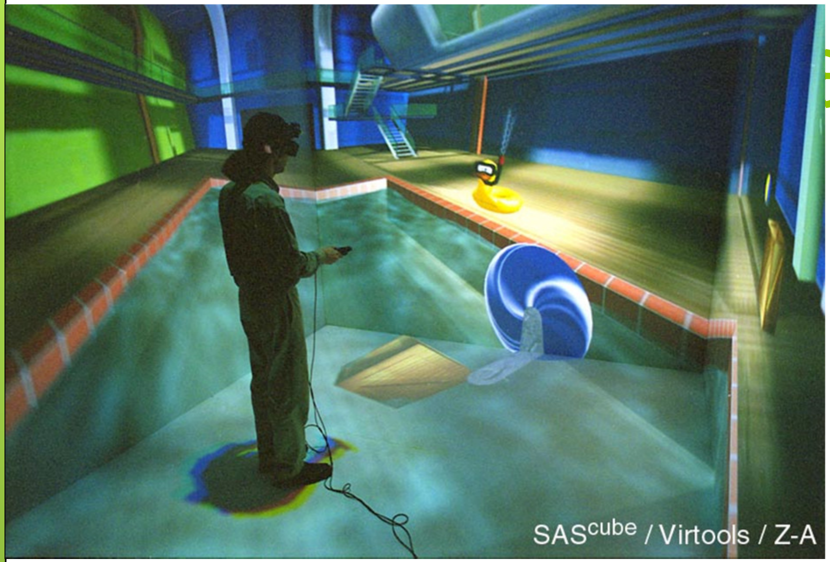




NexCAVE

- 14 42" JVC Xpol displays:
  LCD panels with polarizing filters,
  1920x1080 pixels
- 8 rendering PCs
- Nvidia GeForce 480 GPUs
- 2-camera ART TrackPack optical tracking system




SAS^cube / Virtools / Z-A