CSE 165: 3D User Interaction

Lecture #2: Overview

Announcements

- Homework Assignment #1
  - Due next Friday at 2:00pm
  - To be presented in CSE lab 220
- Paper presentations
  - Title/date due by **entering into wiki table** on Ted by Sunday, January 17th
  - Date and paper selection is first come first serve
- Everyone should have been added to Piazza
- Siggraph student volunteers: https://sis.siggraph.org

Course Topics

- Introduction to 3D Interaction
- Application Domains
- Output Devices
- Input Devices
- Selection and Manipulation
- Navigation (Travel, Wayfinding)
- System Control
- Symbolic Input
- 3D user Interface Design
- Evaluation

What are 3D UIs?

- 3D interaction: human-computer interaction in which the user's tasks are carried out in a 3D spatial context
  - 3D input devices
  - 2D input devices with direct mappings to 3D
- 3D user interface (3D UI): A UI that involves 3D interaction
- 3D interaction technique: A method (hardware and software) allowing a user to accomplish a task in a 3D UI

Why 3D Interfaces?

- 3D applications should be useful
  - immersion
  - natural skills
  - immediacy of visualization
- But, many real world applications have low complexity of interaction
- More complex applications have serious usability problems
- Technology alone is not the solution

Interaction Goals

- Performance
  - efficiency
  - accuracy
  - productivity
- Usability
  - ease of use
  - ease of learning
  - user comfort
- Usefulness
  - interaction helps meet system goals
  - interface relatively transparent so users can focus on tasks

What makes 3D interaction difficult?

- Spatial input
- Lack of constraints
- Lack of standards
- Lack of tools
- Lack of precision
- Fatigue
- Layout more complex
- Perception

Universal 3D Interaction Tasks

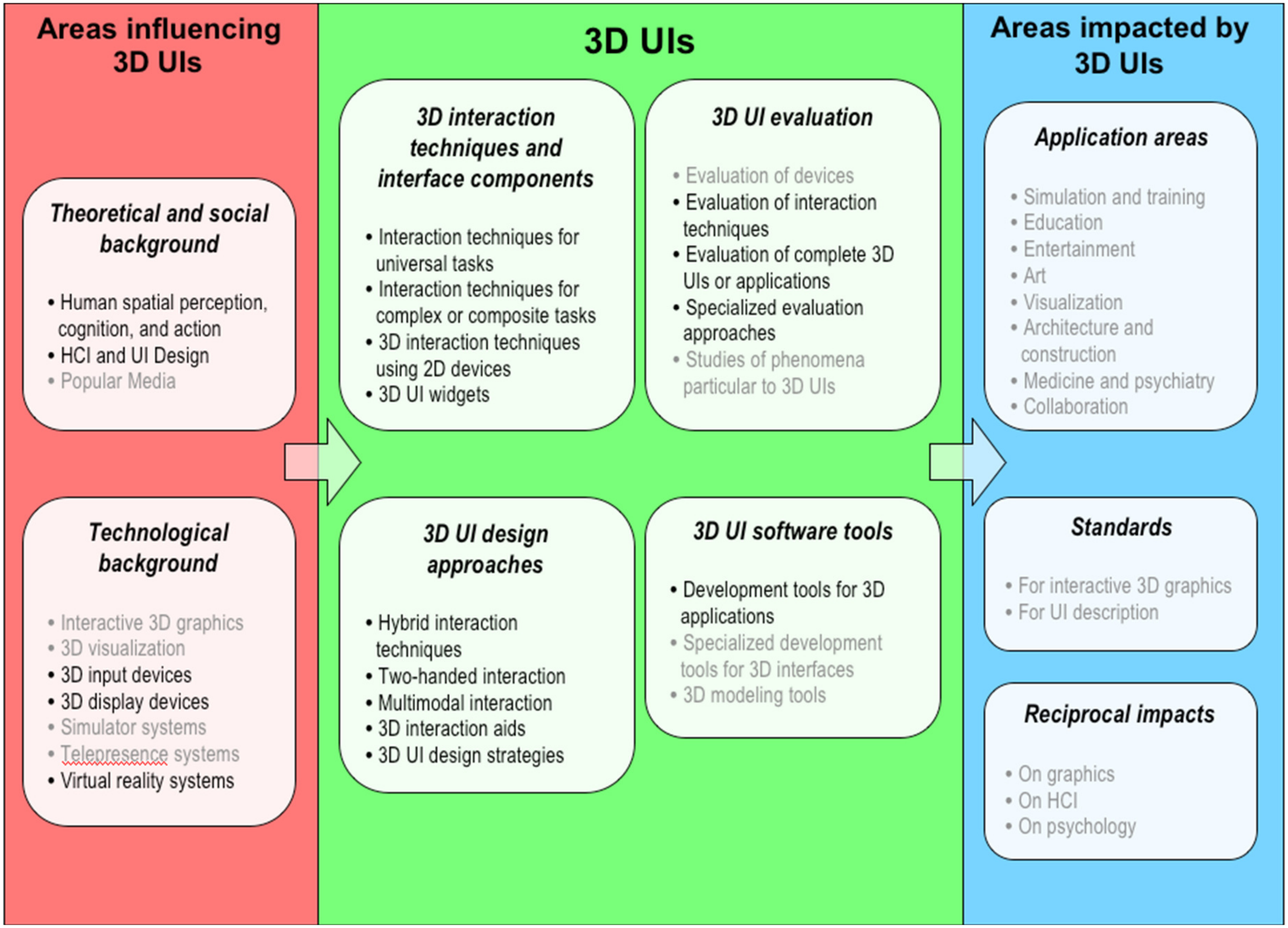
- Navigation
  - travel: motor component
  - wayfinding: cognitive component
- Selection/Picking
- Manipulation
  - specification of object position & orientation
  - specification of scale, shape, other attributes
- System Control
  - changing the system state or interaction mode (e.g., menus)
  - may be composed of other tasks
- Symbolic Input (text, numbers)

3D UI Design Philosophies

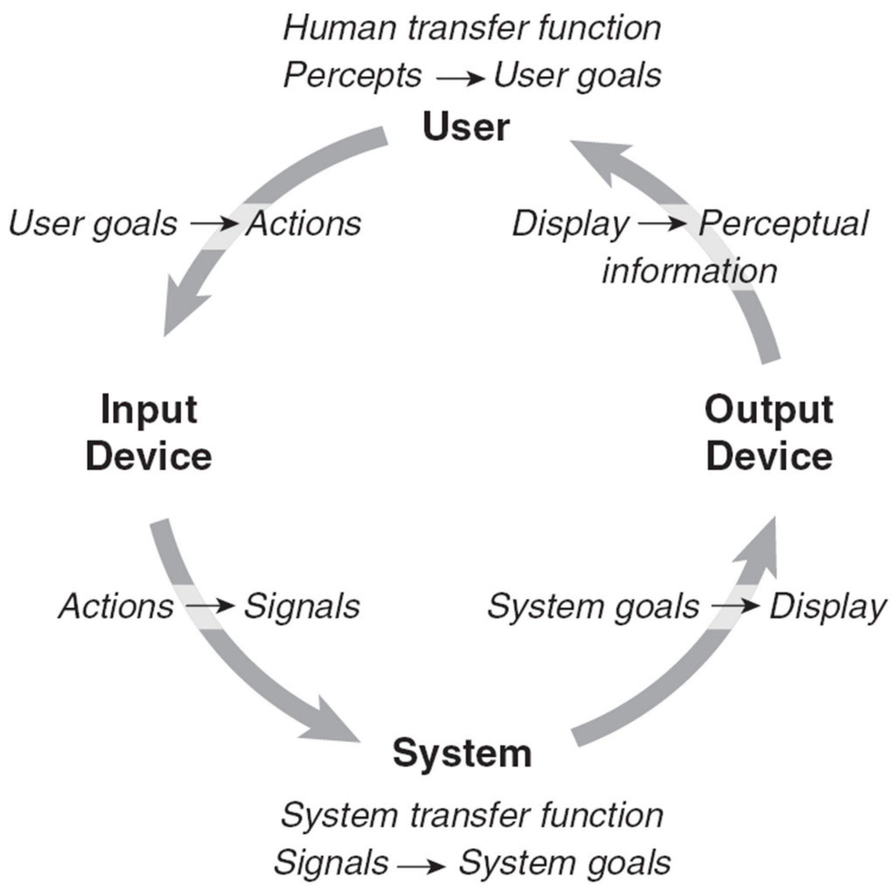
- Artistic approach: Base design decisions on
  - intuition about users, tasks, and environments
  - heuristics, metaphors, common Sense
  - aesthetics
  - adaptation/inversion of existing interfaces
- Scientific approach: Base design decisions on
  - formal characterization of users, tasks, and environments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

Applications

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


Areas influencing 3D UIs
Theoretical and social background
• Human spatial perception, cognition, and action
• HCI and UI Design
• Popular Media
Technological background
• Interactive 3D graphics
• 3D visualization
• 3D input devices
• 3D display devices
• Simulator systems
• Telepresence systems
• Virtual reality systems
3D UIs
3D interaction techniques and interface components
• Interaction techniques for universal tasks
• Interaction techniques for complex or composite tasks
• 3D interaction techniques using 2D devices
• 3D UI widgets
3D UI evaluation
• Evaluation of devices
• Evaluation of interaction techniques
• Evaluation of complete 3D UIs or applications
• Specialized evaluation approaches
• Studies of phenomena particular to 3D UIs
3D UI design approaches
• Hybrid interaction techniques
• Two-handed interaction
• Multimodal interaction
• 3D interaction aids
• 3D UI design strategies
3D UI software tools
• Development tools for 3D applications
• Specialized development tools for 3D interfaces
• 3D modeling tools
Areas impacted by 3D UIs
Application areas
• Simulation and training
• Education
• Entertainment
• Art
• Visualization
• Architecture and construction
• Medicine and psychiatry
• Collaboration
Standards
• For interactive 3D graphics
• For UI description
Reciprocal impacts
• On graphics
• On HCI
• On psychology

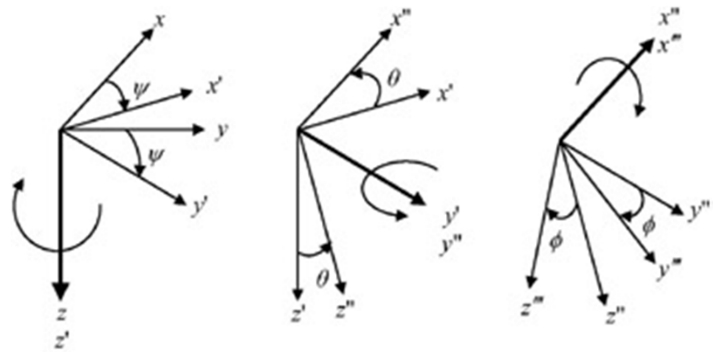
Interaction Workflow



Quaternions

Rotation Calculations

- Intuitive approach: Euler Angles:
  - Simplest way to calculate rotations
  - Defines rotation by 3 sequential rotations about coordinate axes
  - Example Z-Y-X:



1st, the body frame yaws right an angle ψ about the z-axis

2nd, the body frame pitches up an angle θ about the y'-axis

3rd, the body frame rolls CW an angle φ about the x''-axis

http://www.globalspec.com/reference/49379/203279/3-3-euler-angles

Problems With Euler Angles

- Problems with Euler angles:
  - No standard for order of rotations
  - Gimbal Lock, occurs in certain object orientations
    - Video (0:20-1:12)
      - http://www.youtube.com/watch?v=zc8b2Jo7mno
- Better: rotation about arbitrary axis (no Gimbal lock)
  - Can be done with 4x4 matrix
  - But: smoothly interpolating between two orientations is difficult

➔ Quaternions

Quaternion Definition

- Given angle and axis of rotation:
  - a: rotation angle
  - {nx,ny,nz}: normalized rotation axis
- Calculation of quaternion coefficients w, x, y, z:
  - w = cos(a/2)
  - x = sin(a/2) * nx
  - y = sin(a/2) * ny
  - z = sin(a/2) * nz

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 in Bullet Physics

- Quaternions can be specified by rotation angle and axis:
  - `btQuaternion(const btVector3 &_axis, const btScalar &_angle)`
- Or by an x, y, z, w tuple:
  - `btQuaternion(const btScalar &_x, const btScalar &_y, const btScalar &_z, const btScalar &_w)`
- Bullet defines mathematical and other operators:
  - `+, -, *, /, inverse, getAngle, getAxis, slerp`, etc.

Quaternions: Further Reading

- Gamasutra:
  Rotating Objects Using Quaternions
  - http://www.gamasutra.com/view/feature/131686/rotating_objects_using_quaternions.php
- 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

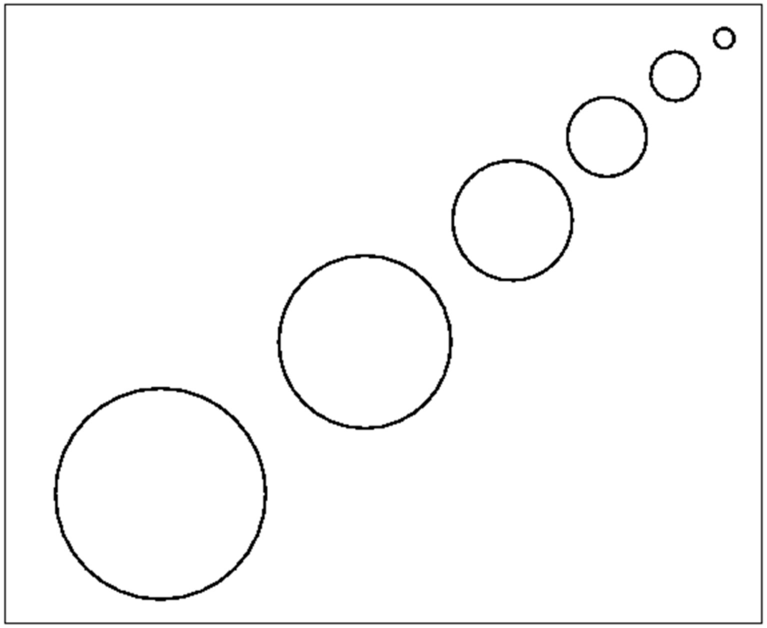
3D Vision

Depth Cues – How Do We See 3D?

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

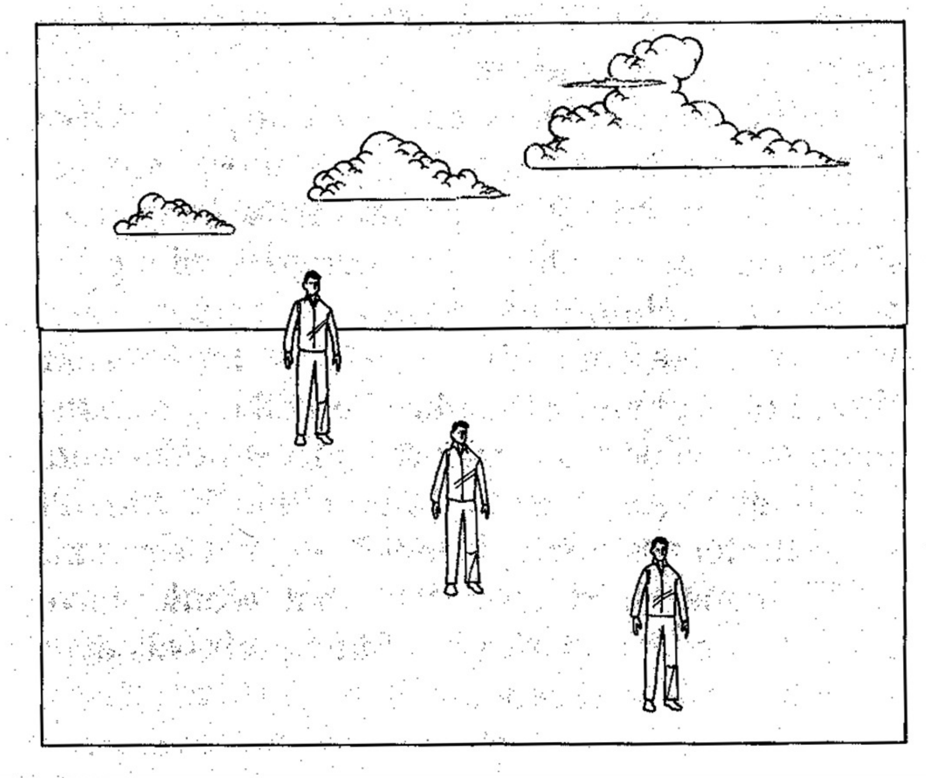
Monocular/Static Cues

- Relative Size



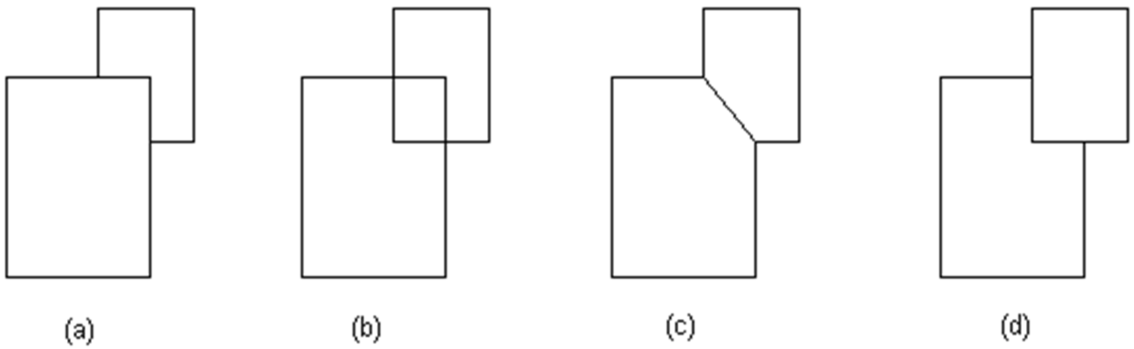
Monocular/Static Cues

- Height relative to horizon



Monocular/Static Cues

- Occlusion

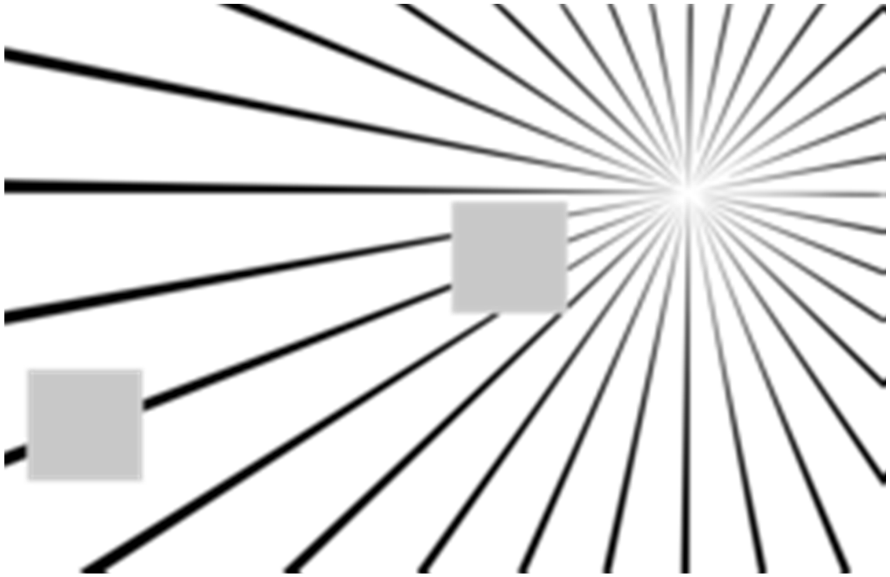


Depth perception based on overlapping. The object with more continuous border line is felt to lie closer. In figure (a) it is the larger rectangle and in figure (d) it is the smaller. In figures (b) and (c) no depth information can be obtained.

http://www.hitl.washington.edu/projects/knowledge_base/virtual-worlds/EVE/III.A.1.c.DepthCues.html

Monocular/Static Cues

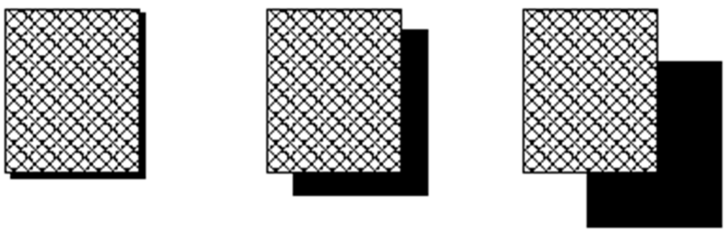
- Linear Perspective



http://anthonysaba.wikispaces.com/Depth+Perception

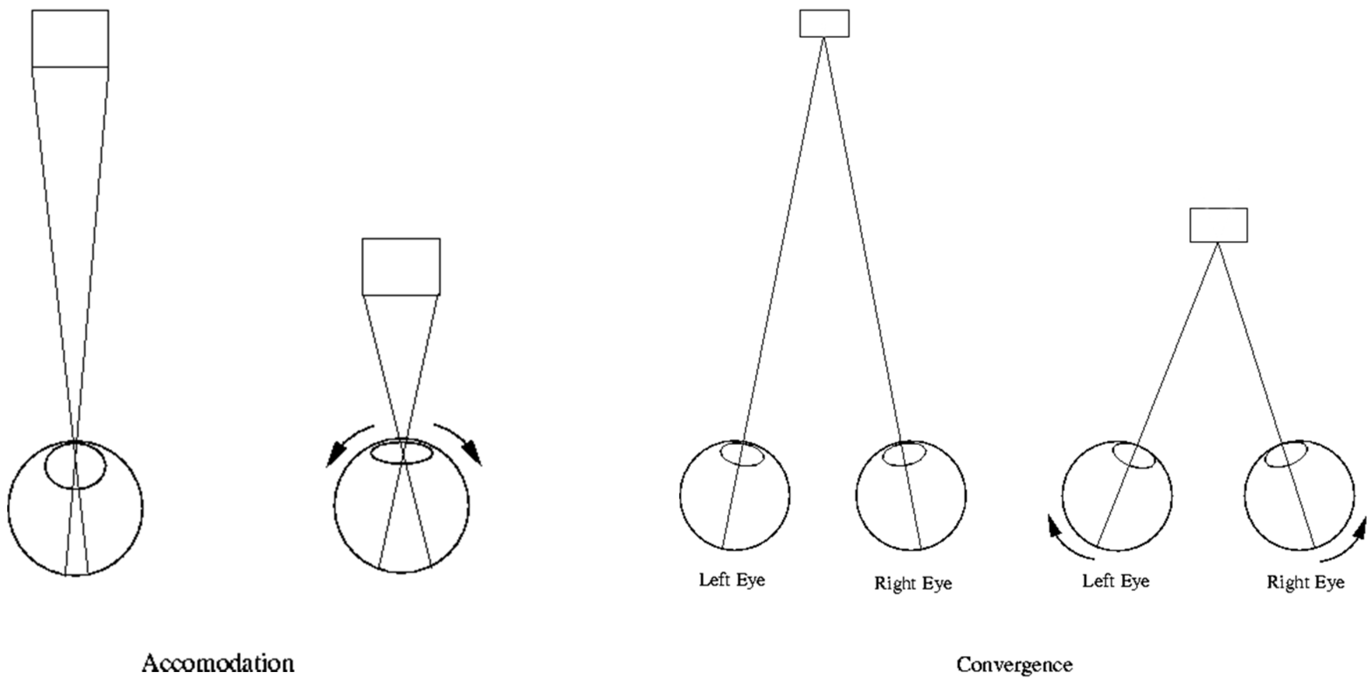
Monocular/Static Cues

- Shading, Lighting, and Texture



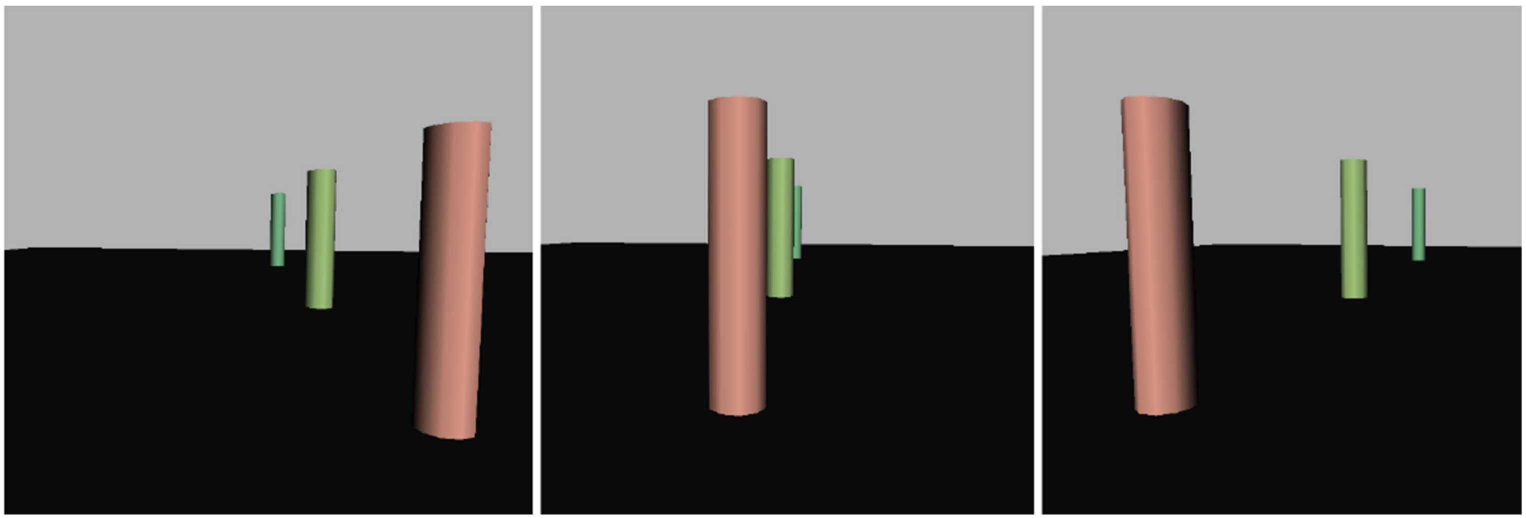
Oculomotor Cues

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



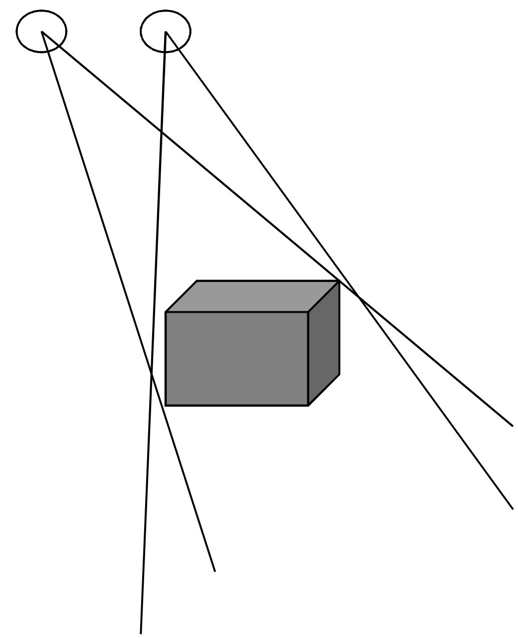
Motion Parallax

- Stationary viewer vs. moving viewer



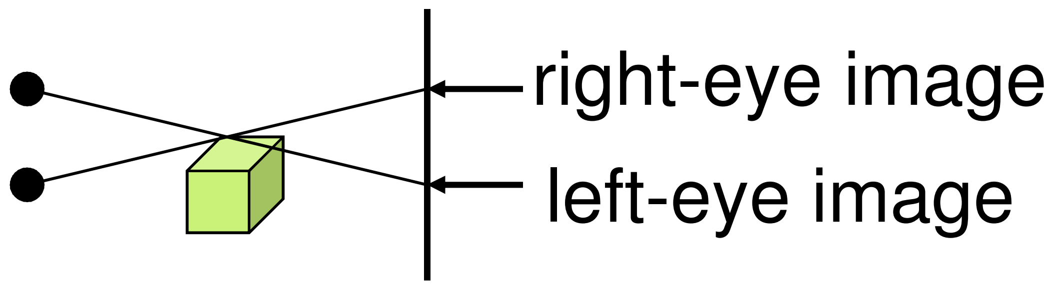
Binocular Disparity and Stereopsis

- 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