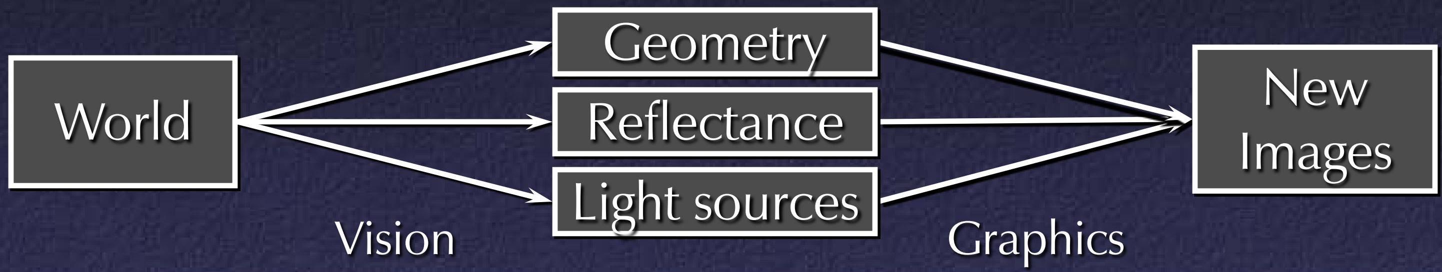
Image-Based Modeling and Rendering

Acknowledgment: Tom Funkhouser

Image-Based Modeling and Rendering

- For many applications, re-rendering is goal
- Traditional vision / graphics pipelines:



- Image-based pipeline:

Image-Based Modeling and Rendering

- Generate new views of a scene directly from existing views
- “Pure” IBR (such as lightfields): no geometric model of scene
- Other IBR techniques try to obtain higher quality with less storage by building a model

Plenoptic Function

- $L(x,y,z,\theta,\phi,t,\lambda)$
- Captures all light flow in a scene
  - to/from any point (x,y,z),
  - in any direction (θ,ϕ),
  - at any time (t),
  - at any frequency (λ)
- Enough information to construct any image of the scene at any time

[Funkhouser]

Plenoptic Function Simplifications

- Represent color as RGB: eliminate λ
- Static scenes: ignore dependence on t
- $7D \rightarrow 3 \times 5D$

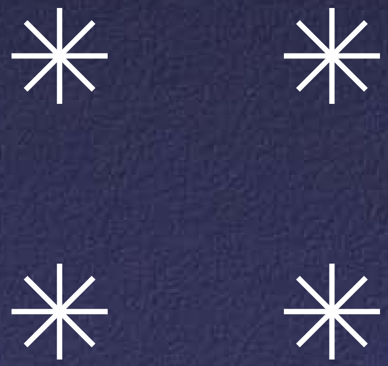
Plenoptic Function – Special Cases

- Sample at one (x,y,z):
  - $L(\theta,\phi)$ is just an (omnidirectional) image
- Full 5D $L(x,y,z,\theta,\phi)$:
  - Omnidirectional image at each point in space
  - Enough information to reconstruct any view

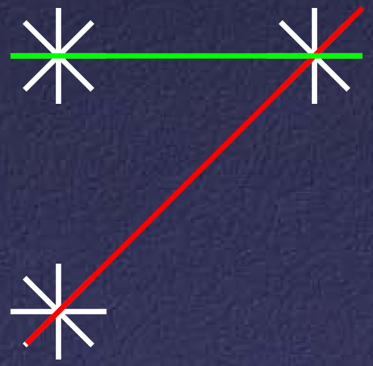
Free Space

- Consider a region of space without occlusion
- Light travels in straight lines → some pixels in different images are the same ray of light

General case



Free space



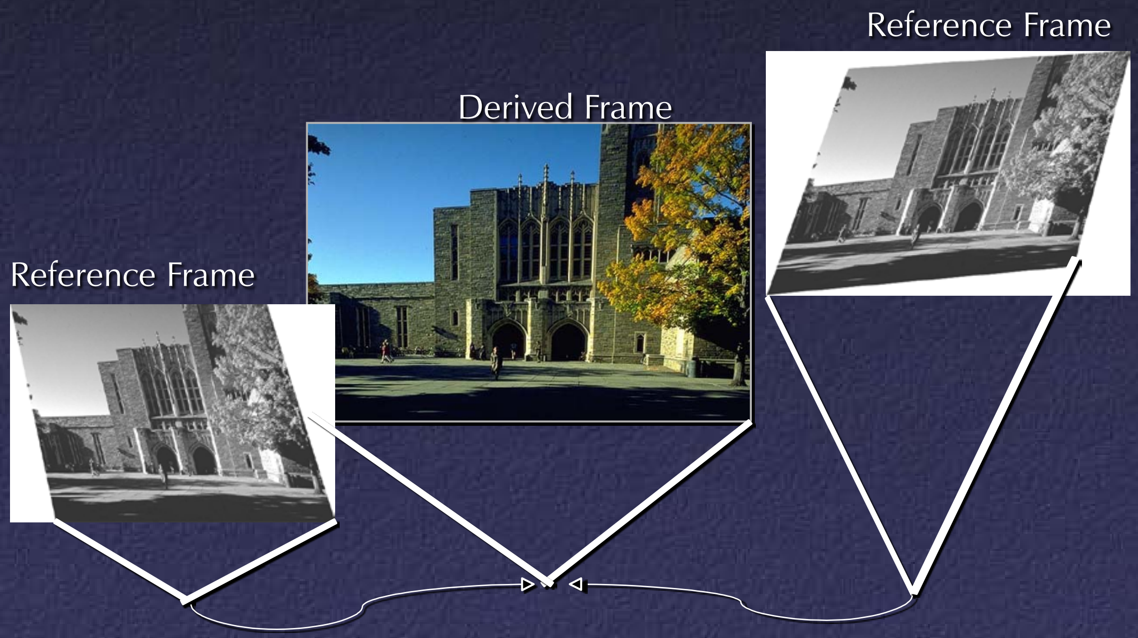
"Rebinning" pixels

Image-Based Representations

7D	Ideal
6D	Consider only 3 frequencies (RGB)
5D	Consider only one time instant (static scene)
4D	Consider only viewpoints inside/outside scene
3D	Consider one dimension fewer directions/positions
2D	Consider viewpoints at finite set points or angles

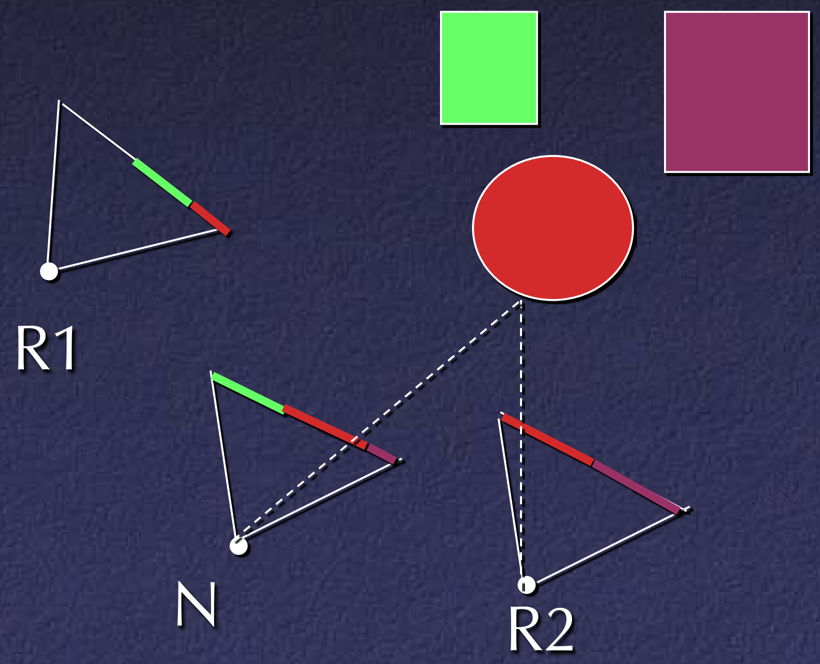
View Interpolation

- Create novel images by resampling photographs
  - Reference images sample 5D plenoptic function



View Interpolation

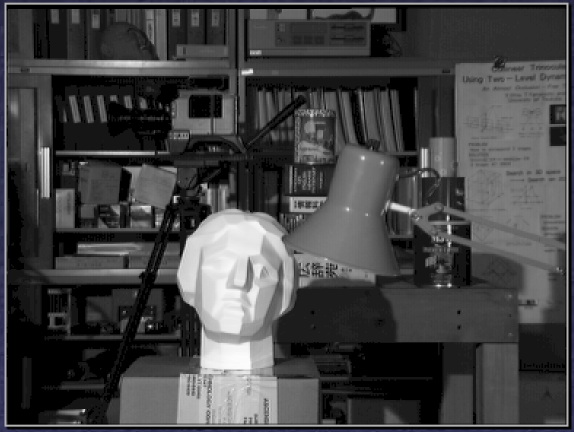
- Method:
  - Warp nearby reference images to novel viewpoint
  - Blend warped images



Morph with warp
defined by
pixel correspondences

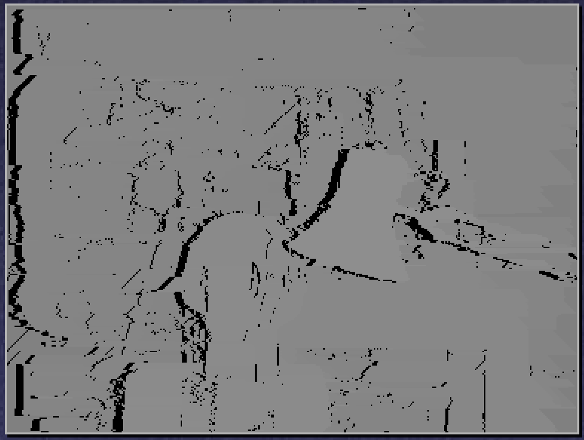
Pixel Correspondences

- Vision (e.g. stereo): disparity
- Feature matching: sparse
- 3D model: possibly coarse



Left

Right

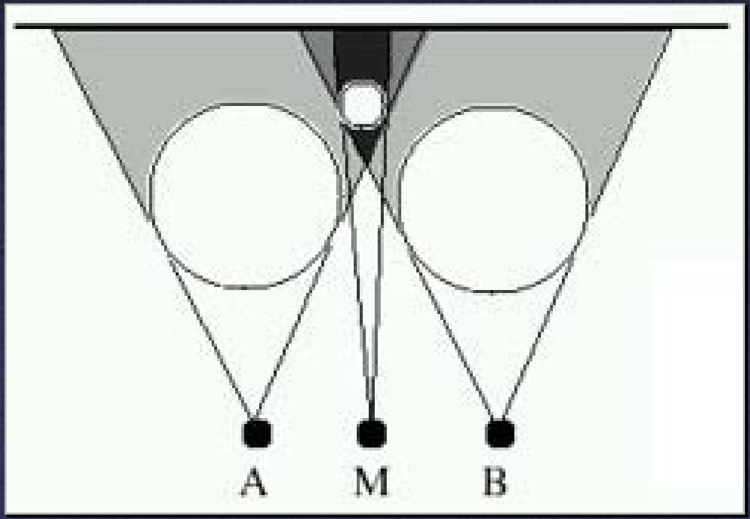


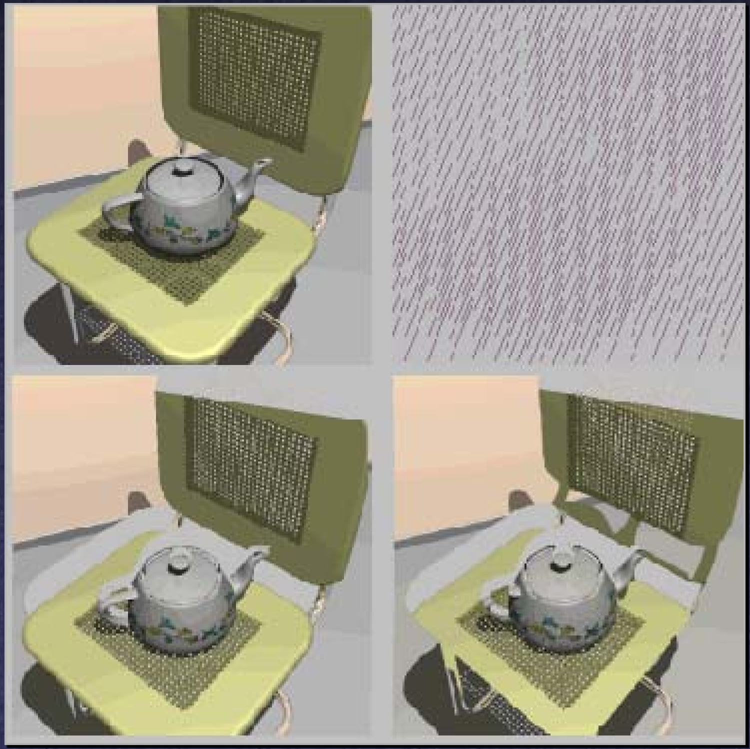
Disparity

[Szeliski]

View Interpolation

- Problem: changes in visibility
  - Disocclusions





[McMillan]

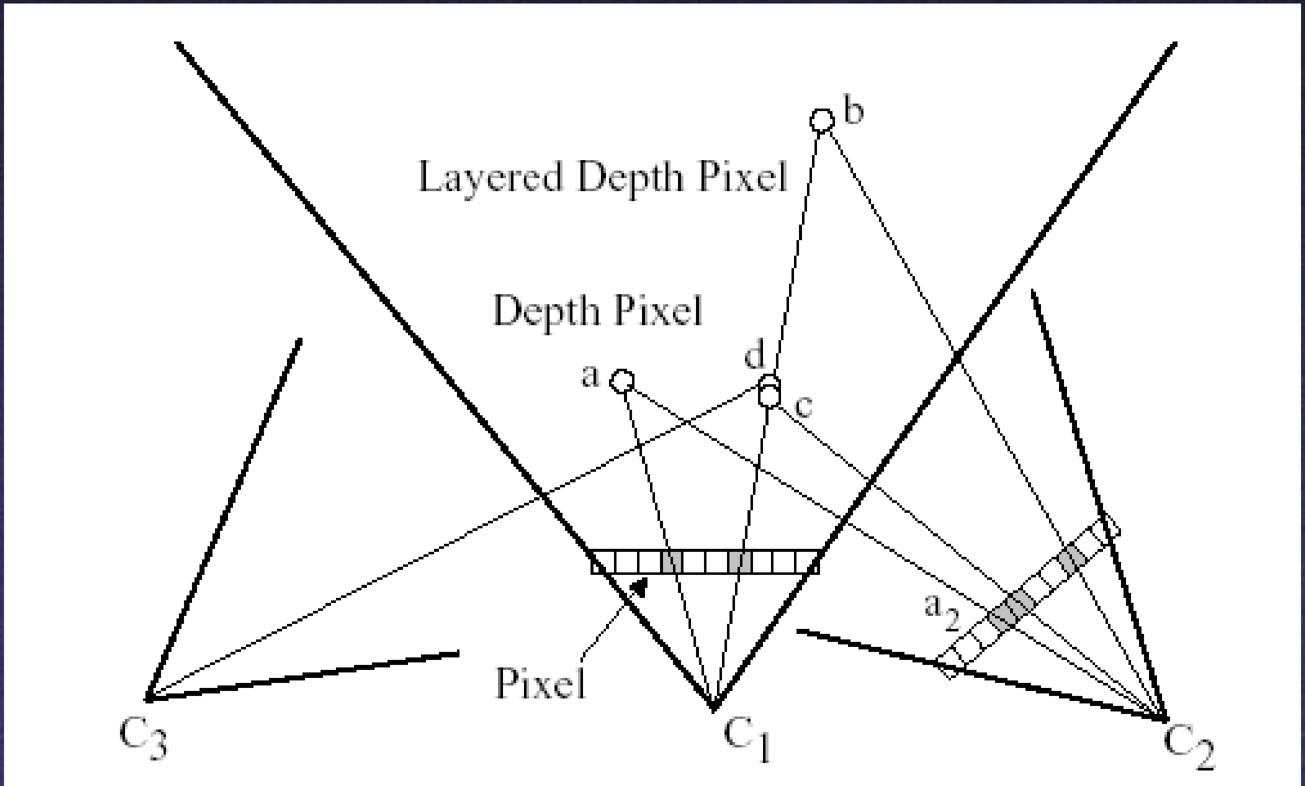
Disocclusions

- Partial solutions:
  - Use more photographs
  - Fill holes by interpolating nearby pixels

[McMillan]

Disocclusions

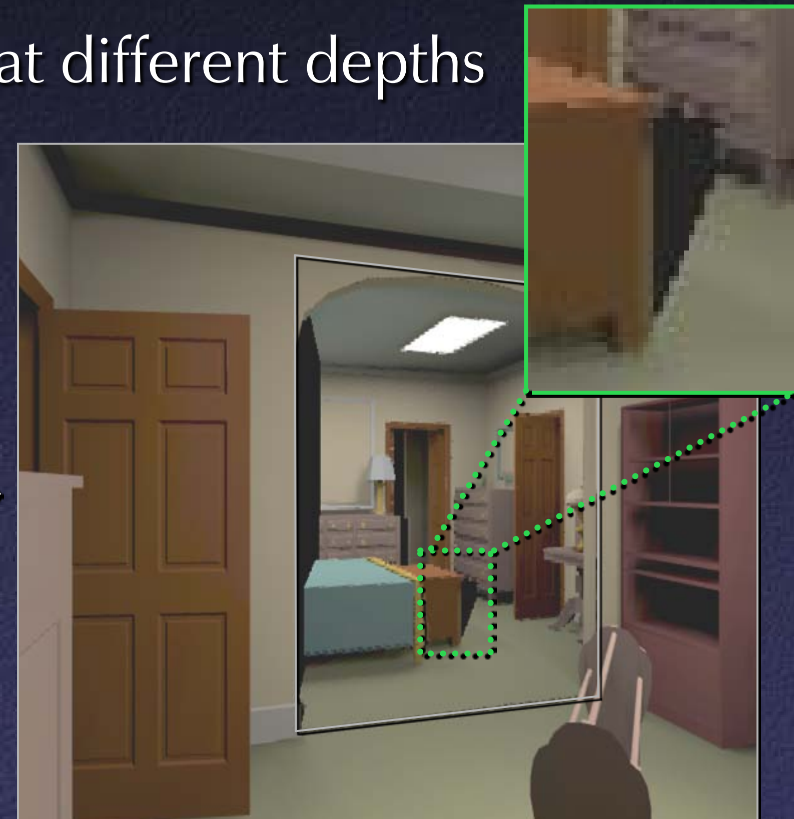
- Better solutions (when possible):
  - Multiple samples per pixel at different depths



Disocclusions

- Better solutions (when possible):
  - Multiple samples per pixel at different depths

Reference Image



Warped Depth Image

[Popescu]

Disocclusions

- Better solutions (when possible):
  - Multiple samples per pixel at different depths

Reference Image

Warped Layered Depth Image

Light Field

- In unoccluded space, can reduce plenoptic function to 4D
- 2D array of 2D images
- Still contains enough information to reconstruct new views

Lightfields

- Advantages:
  - Simpler computation vs. traditional CG
  - Cost independent of scene complexity
  - Cost independent of material properties and other optical effects
  - Avoid hard vision problems
- Disadvantages:
  - Static geometry
  - Fixed lighting
  - High storage cost

Using Lightfields

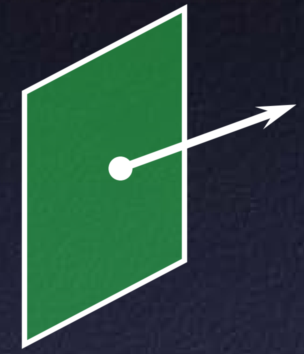
- Obtain 2D slices of 4D data set
- Arbitrary views: take other 2D slices
- Challenges:
  – Capture
  – Parameterization
  – Compression
  – Rendering

Capturing Lightfields

- Need a 2D set of (2D) images
- Choices:
  - Camera motion: human vs. computer
  - Constraints on camera motion
  - Coverage and sampling uniformity
  - Aliasing

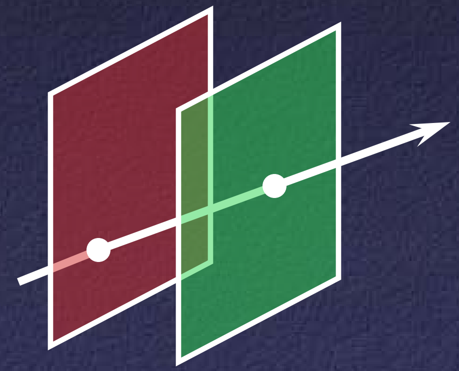
Lightfield Parameterization

- Point / angle



- Two points on a sphere

- Points on two planes



- Original images and camera positions

Compression

- Compress individual images (JPEG, etc.)
- Adapt video compression to 2D arrays
- Decomposition into basis functions
- Vector quantization

Rendering

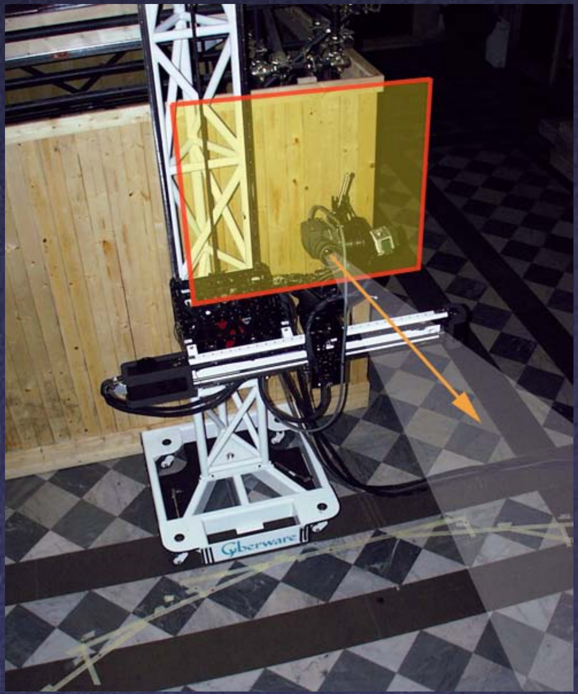
- How to select rays?
- Interpolation
- Taking advantage of hardware
  - Graphics hardware
  - Compression hardware

Implementations

- Lightfields, Levoy and Hanrahan (SIGGRAPH 96)
- Lumigraphs, Gortler et al. (SIGGRAPH 96)
- Unstructured lumigraphs, Buehler et al. (SIGGRAPH 01)

Light Field Rendering

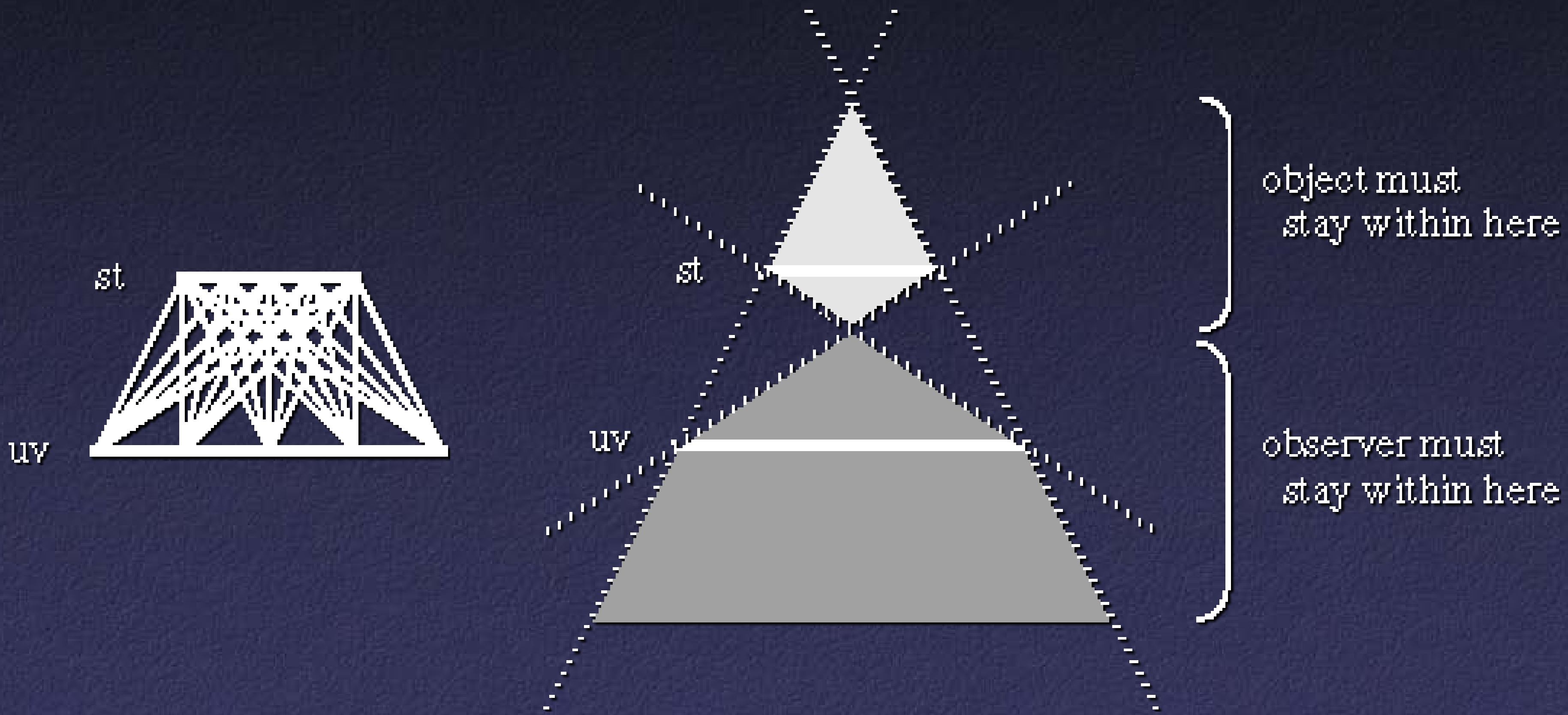
- Capture:
  - Computer-controlled camera rig
  - Move camera to grid of locations on a plane



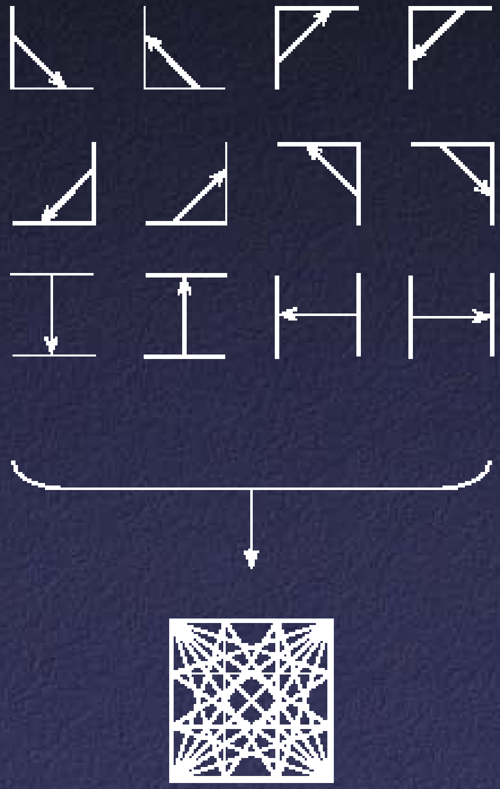
Light Field Two-Plane Parameterization

- Two planes, evenly sampled: “light slab”
- In general, planes in arbitrary orientations
- In practice, one plane = camera locations
  - Minimizes resampling

Light Field Coverage



Multi-Slab Light Fields



Rendering

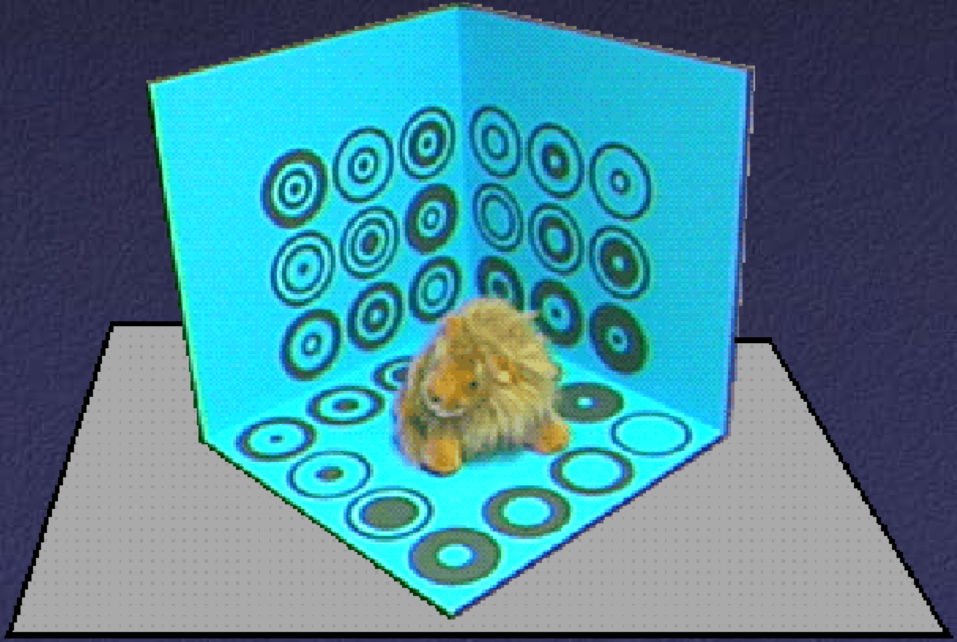
- For each desired ray:
  - Compute intersection with (u,v) and (s,t) planes
  - Take closest ray
- Variants: interpolation
  - Bilinear in (u,v) only
  - Bilinear in (s,t) only
  - Quadrilinear in (u,v,s,t)

Light Field Compression

- Based on vector quantization
- Preprocessing: build a representative codebook of 4D tiles
- Each tile in lightfield represented by index
- Example: 2x2x2x2 tiles, 16 bit index = 24:1 compression

The Lumigraph

- Capture: move camera by hand
- Camera intrinsics assumed calibrated
- Camera pose recovered from markers

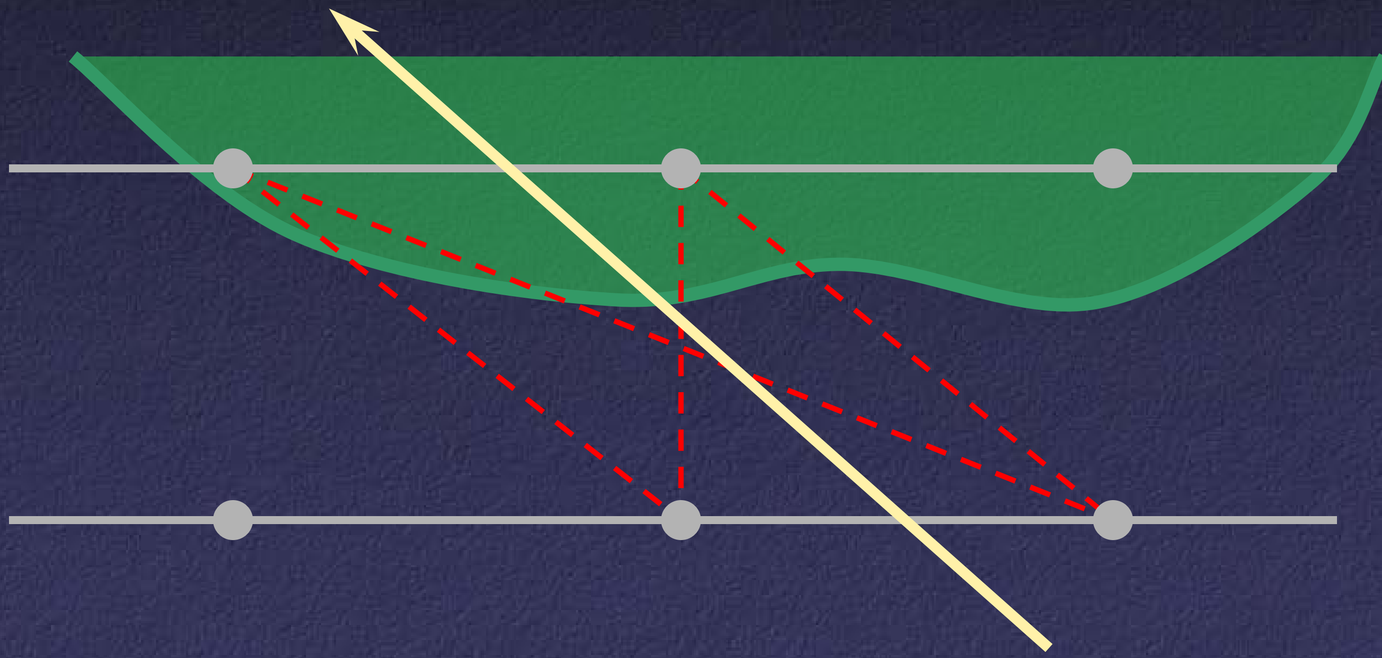


Lumigraph Postprocessing

- Obtain rough geometric model
  - Chroma keying (blue screen) to extract silhouettes
  - Octree-based space carving
- Resample images to two-plane parameterization

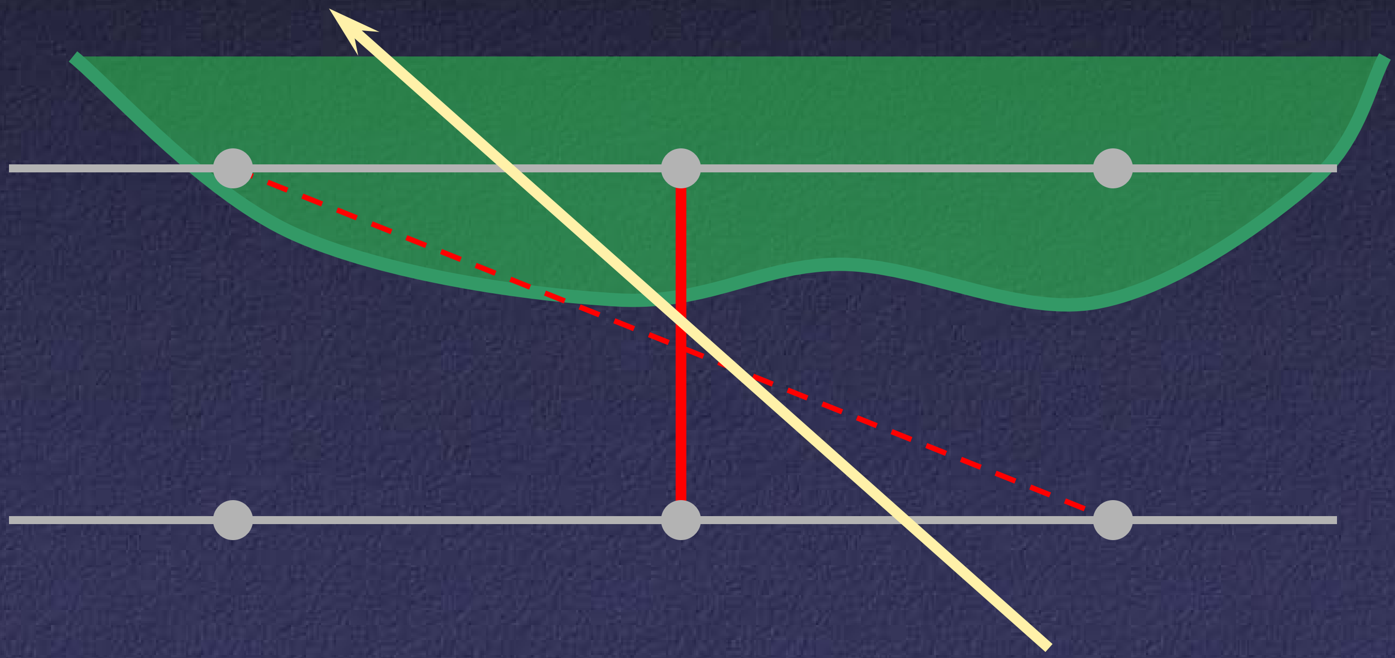
Lumigraph Rendering

- Use rough depth information to improve rendering quality



Lumigraph Rendering

- Use rough depth information to improve rendering quality



Lumigraph Rendering

Without using geometry

Using approximate geometry

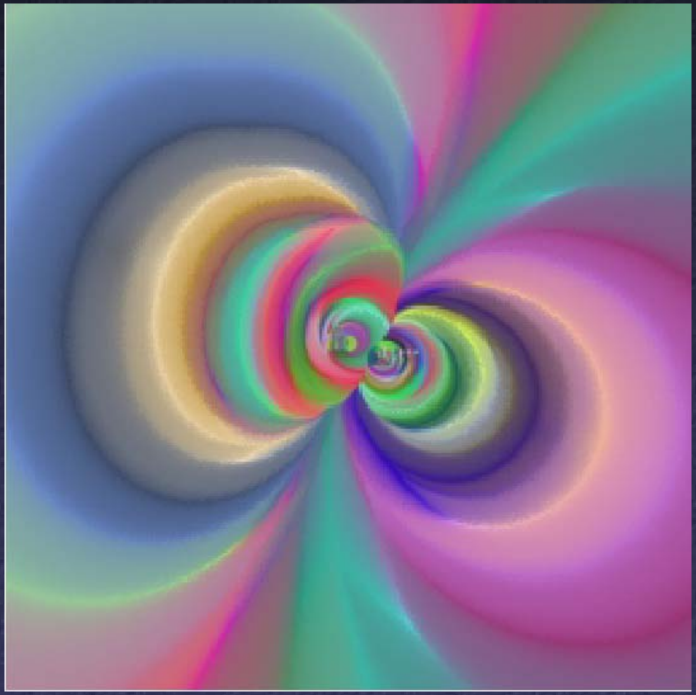
Unstructured Lumigraph Rendering

- Further enhancement of lumigraphs: do not use two-plane parameterization
- Store original pictures: no resampling
- Hand-held camera, moved around an environment

Unstructured Lumigraph Rendering

- To reconstruct views, assign penalty to each original ray
  - Distance to desired ray, using approximate geometry
  - Resolution
  - Feather near edges of image
- Construct "camera blending field"
- Render using texture mapping

Unstructured Lumigraph Rendering



Blending field

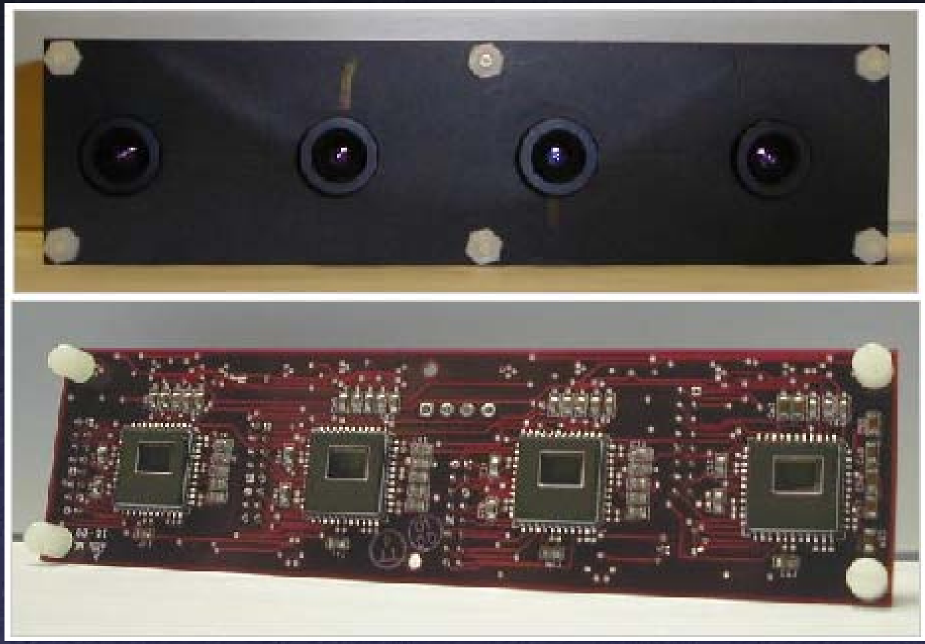
Rendering

Other Lightfield Acquisition Devices

- Spherical motion of camera around an object
- Samples space of directions uniformly
- Second arm to move light source – measure reflectance

Other Lightfield Acquisition Devices

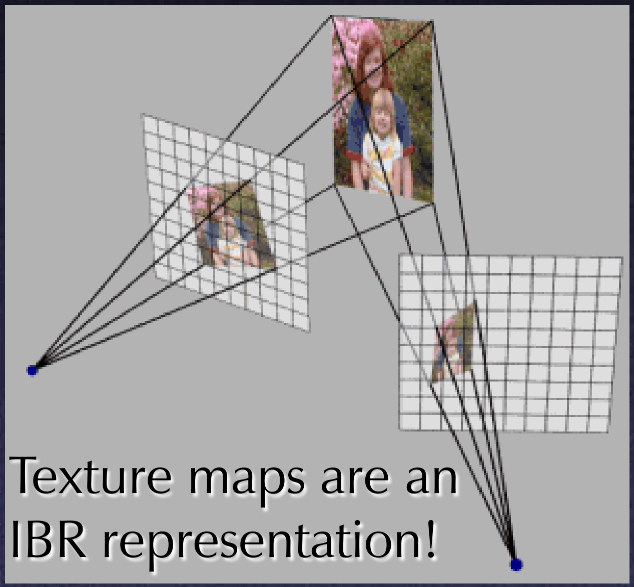
- Acquire an entire light field at once
- Video rates
- Integrated MPEG2 compression for each camera

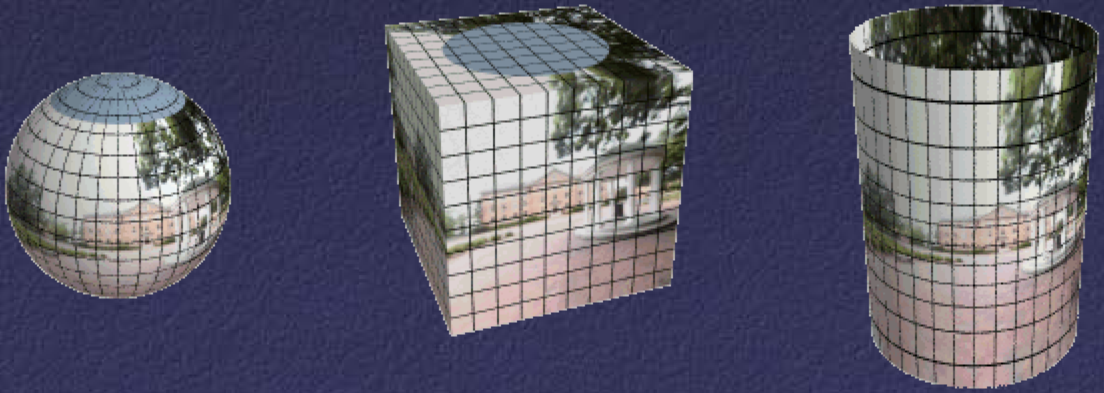


(Bennett Wilburn, Michal Smulski, Mark Horowitz)

Other IBR Representations

- Texture maps
- VDTMs
- Surface light fields
- Concentric mosaics
- Panorama
- Etc.

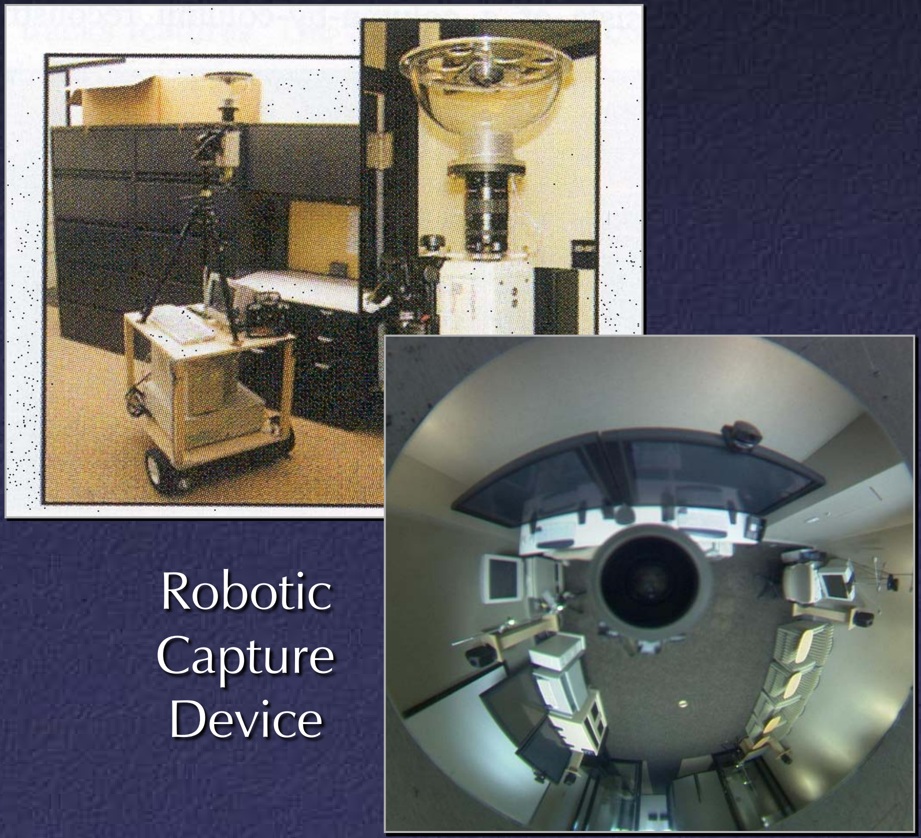




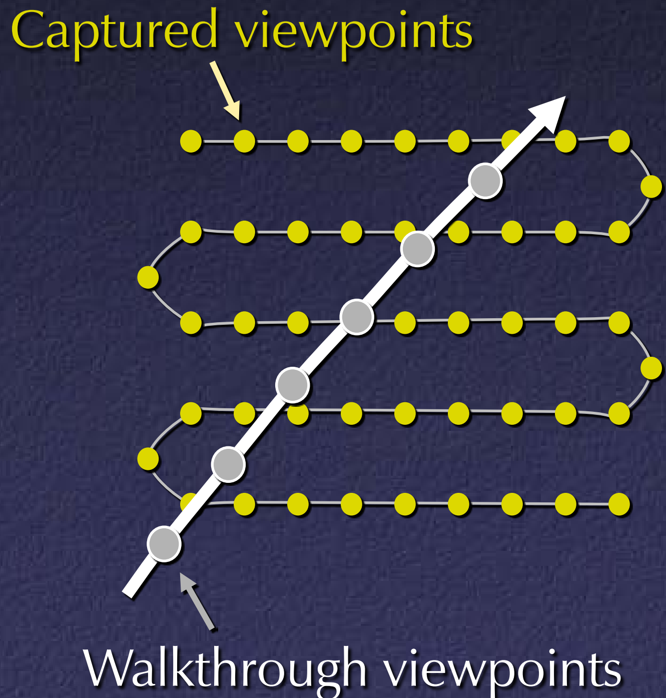
[McMillan]

Sea of Images

- Dense sampling of plenoptic function with hemispherical camera moving on plane



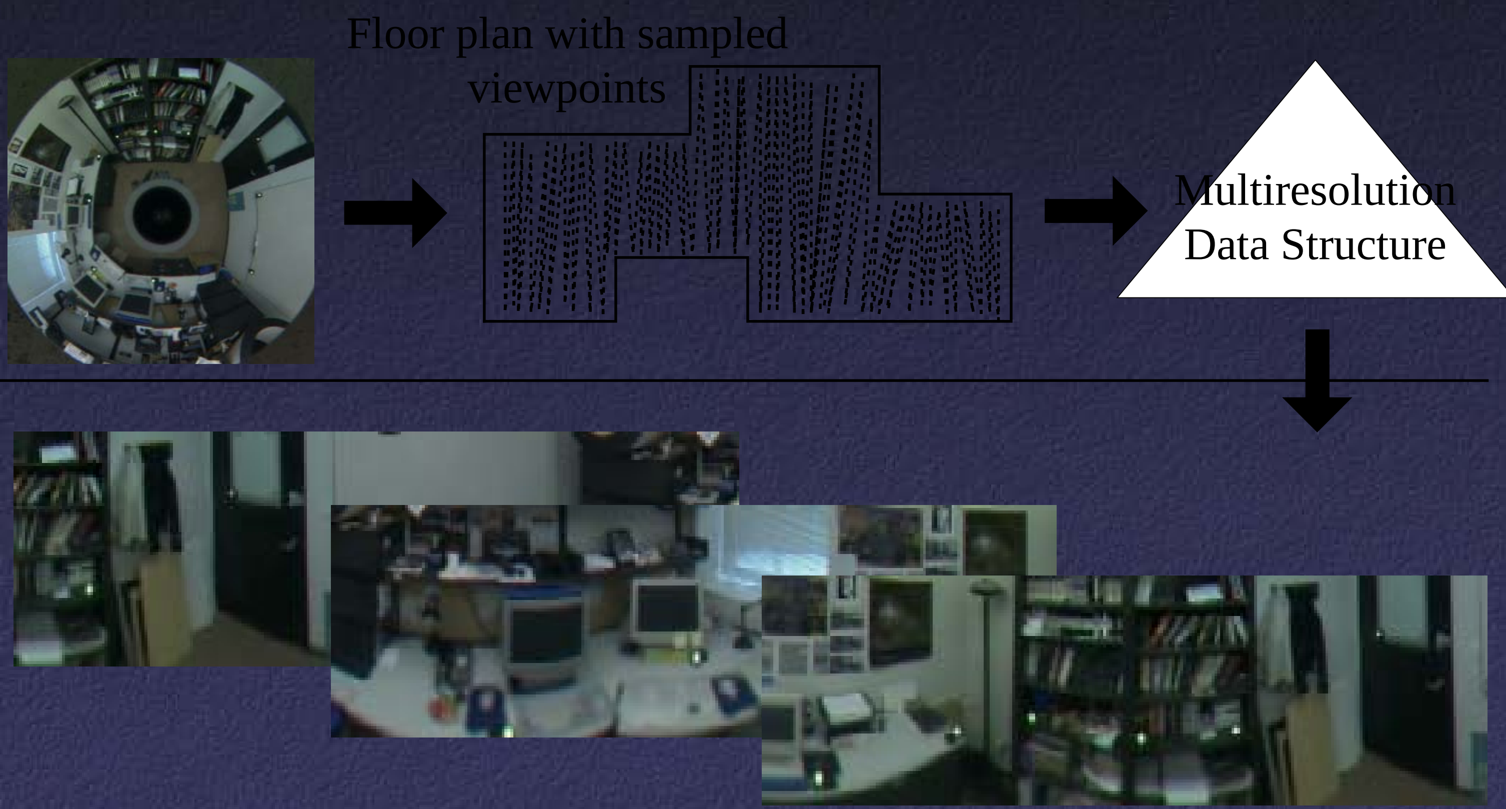
Robotic Capture Device



[Aliaga & Funkhouser]

Sea of Images

- Multiresolution compression for walkthroughs

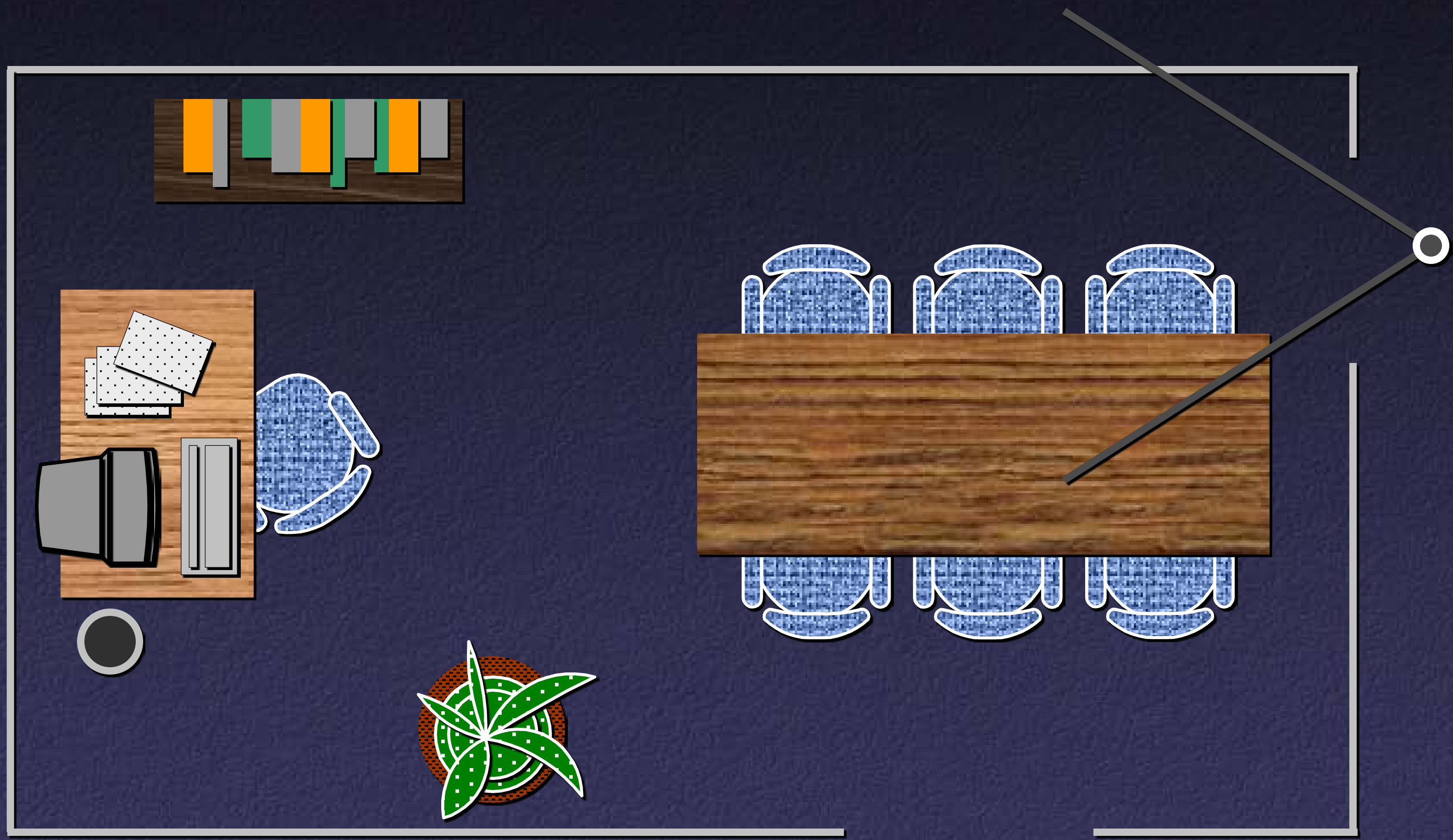


Replacing Geometry with Images

- Algorithm
  - Select subset of model
  - Create image of the subset
  - Cull subset and replace with image
- Why?
  - Image displayed in (approx.) constant time
  - Image reused for several frames

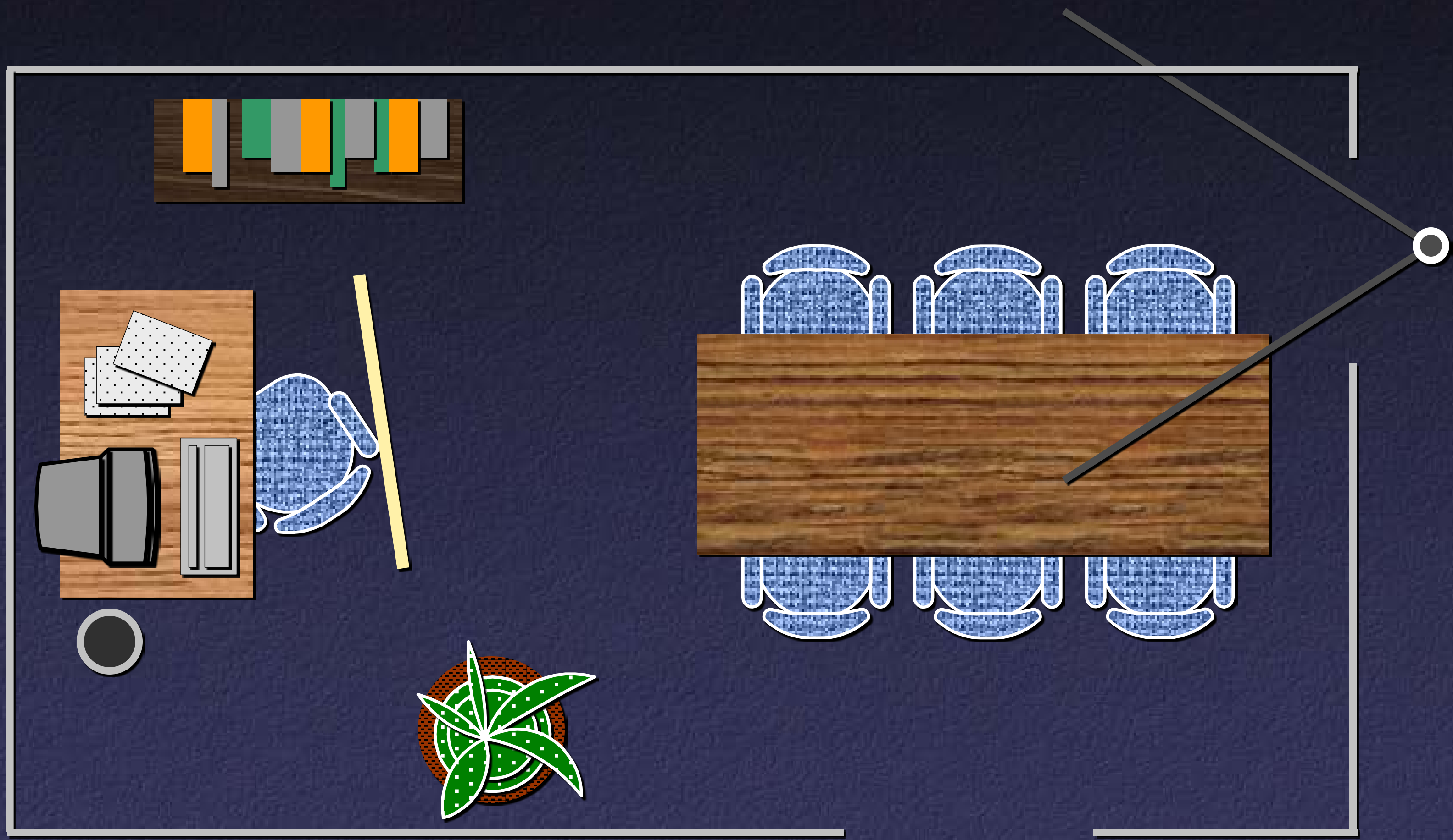
[Aliaga]

Simple Example



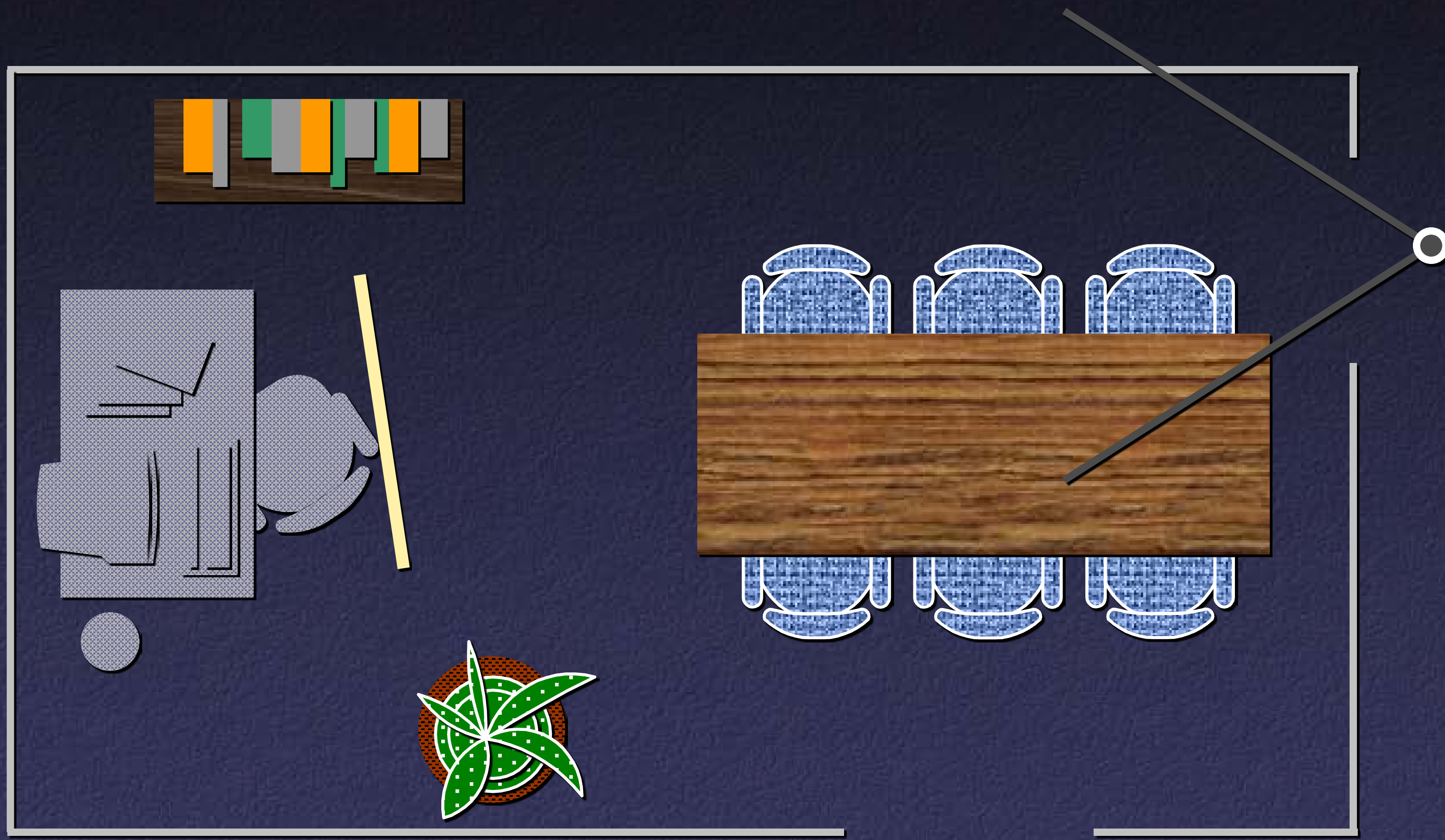
[Aliaga]

Simple Example



[Aliaga]

Simple Example



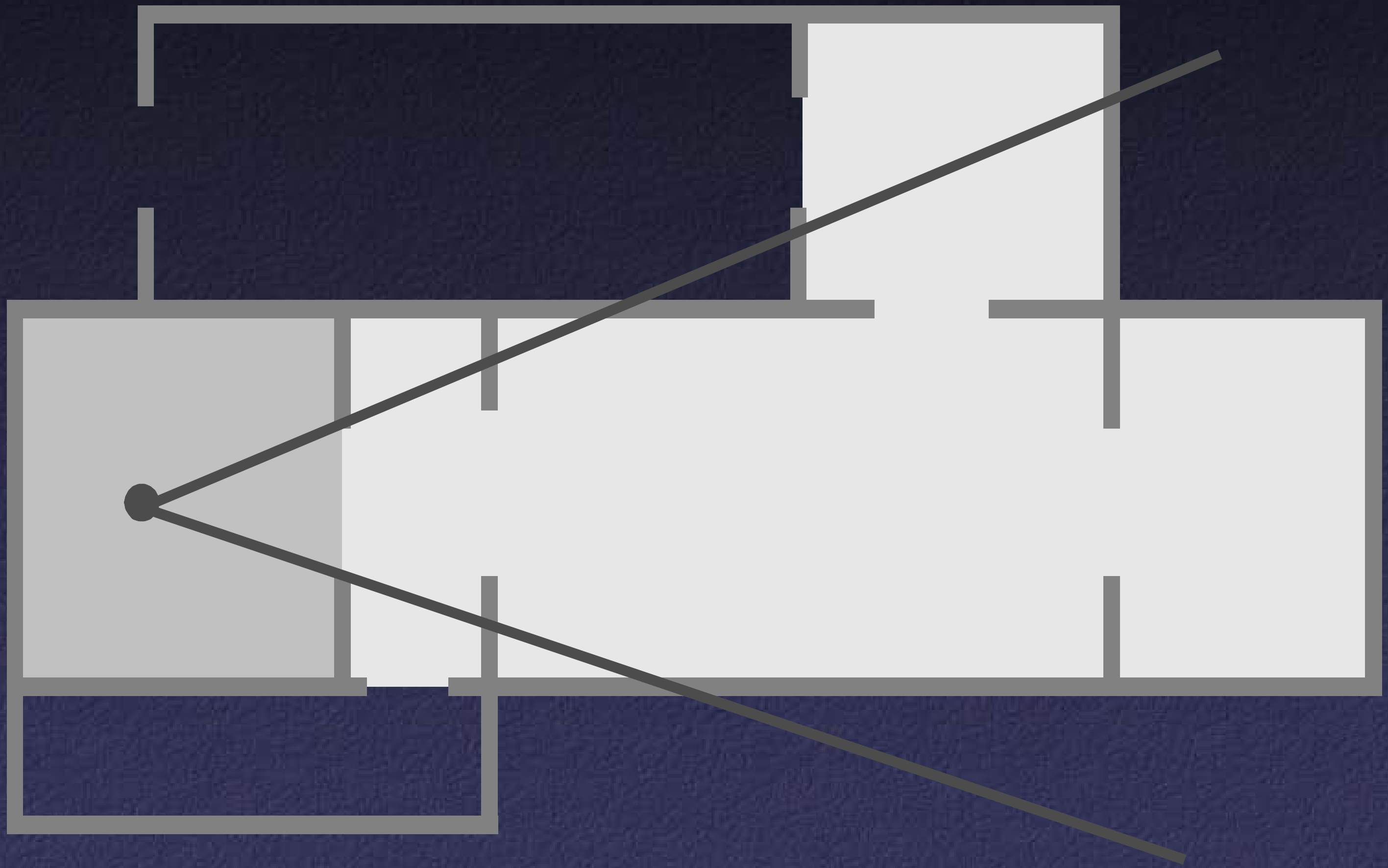
[Aliaga]

Automatic Image-Placement

- Preprocess:
  - Select geometry to replace
- At run time:
  - Display selected geometry as a (depth) image
  - Render remaining geometry normally

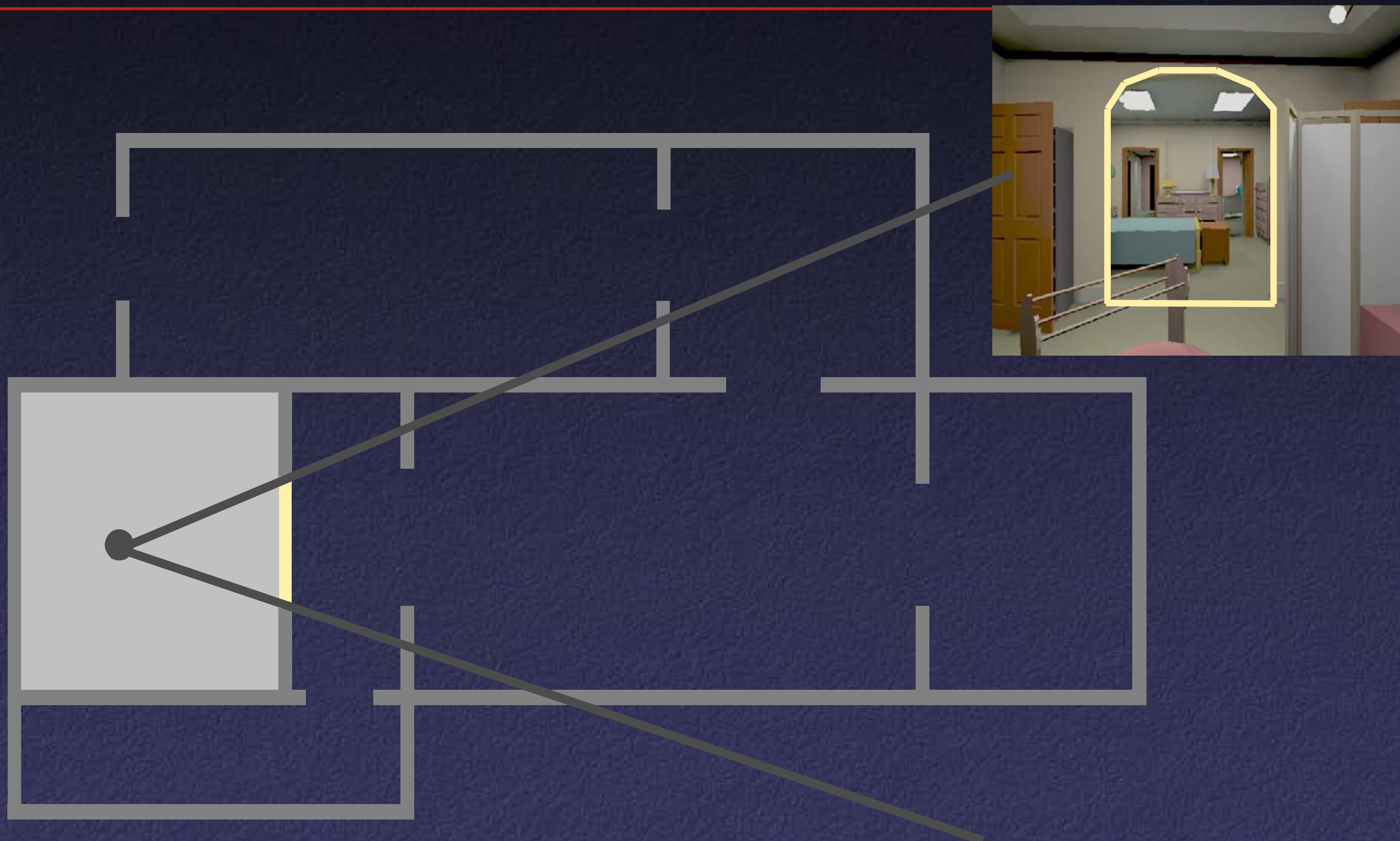
[Aliaga]

Cells and Portals



[Aliaga]

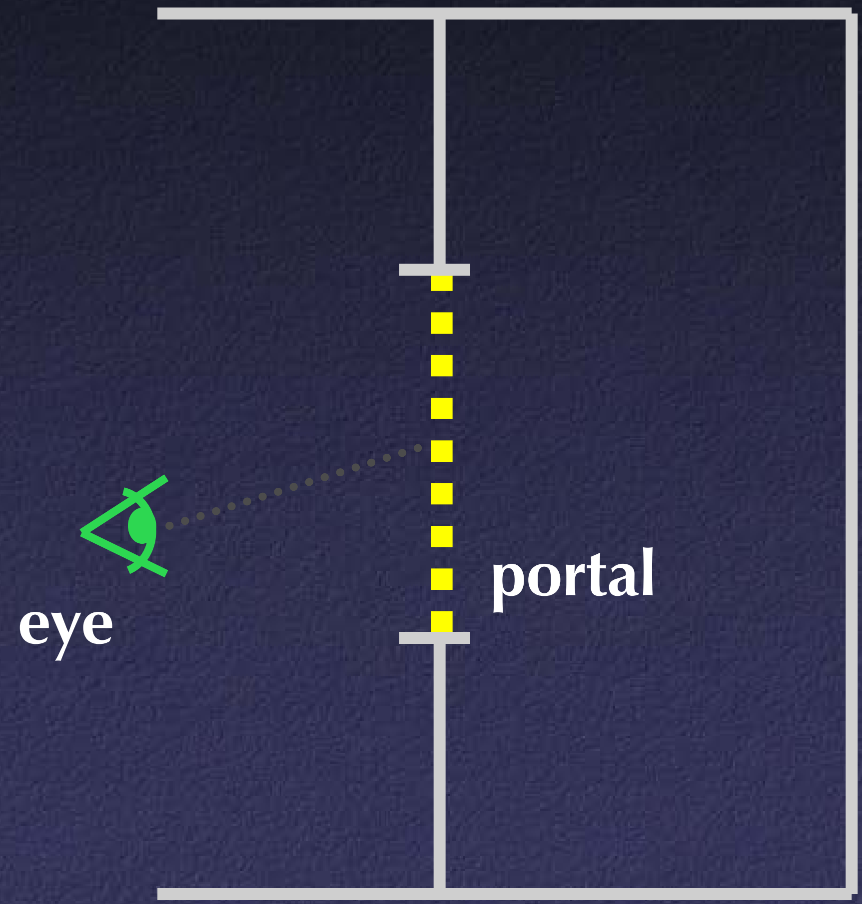
Portal Images



[Aliaga]

Creating Portal Images

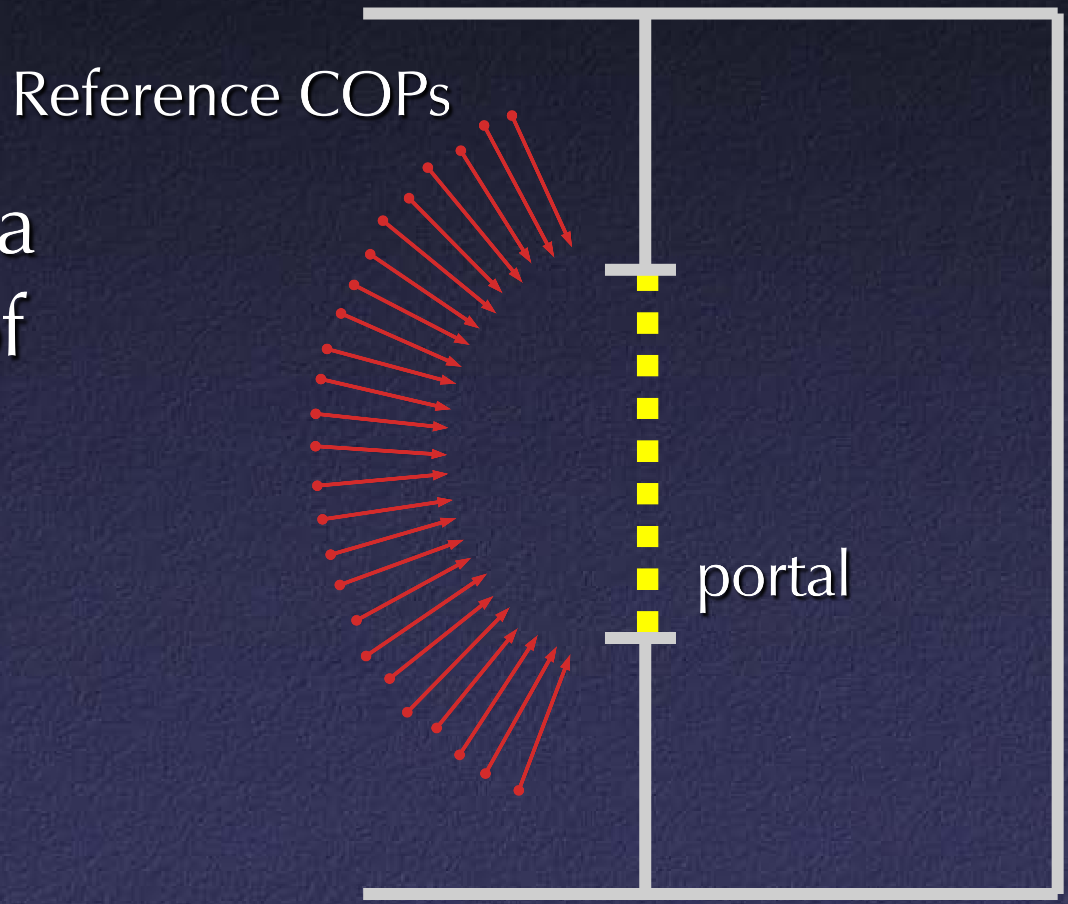
Ideal portal image would be one sampled from the current eye position



[Aliaga]

Creating Portal Images

Display one of a large number of pre-computed images (~120)

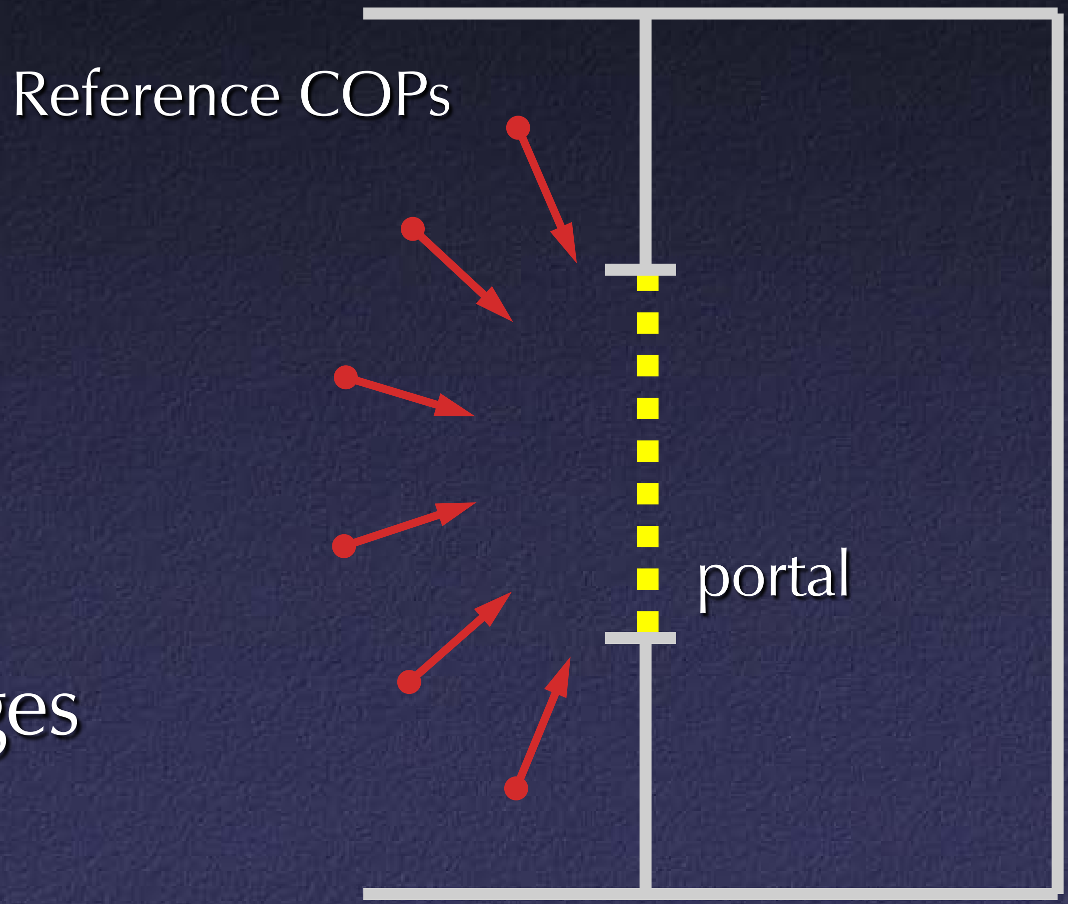


[Aliaga]

Creating Portal Images

or…

Warp one of a much smaller number of reference images



[Aliaga]

IBR Summary

- Advantages
  - Photorealistic - by definition
  - Do not have to create 3D detailed model
  - Do not have to do lighting simulation
  - Performance independent of scene
- Disadvantages
  - Static scenes only
  - Real-world scenes only
  - Difficult for scenes with specularities, etc.
  - Limited range of viewpoints
  - Limited resolution