Image-Based Rendering

COS 426, Spring 2016

Slides from: Dan Aliaga, Thomas Funkhouser,
Marc Levoy, and Szymon Rusinkiewicz

Image-Based Rendering

Make new views of scene from existing views

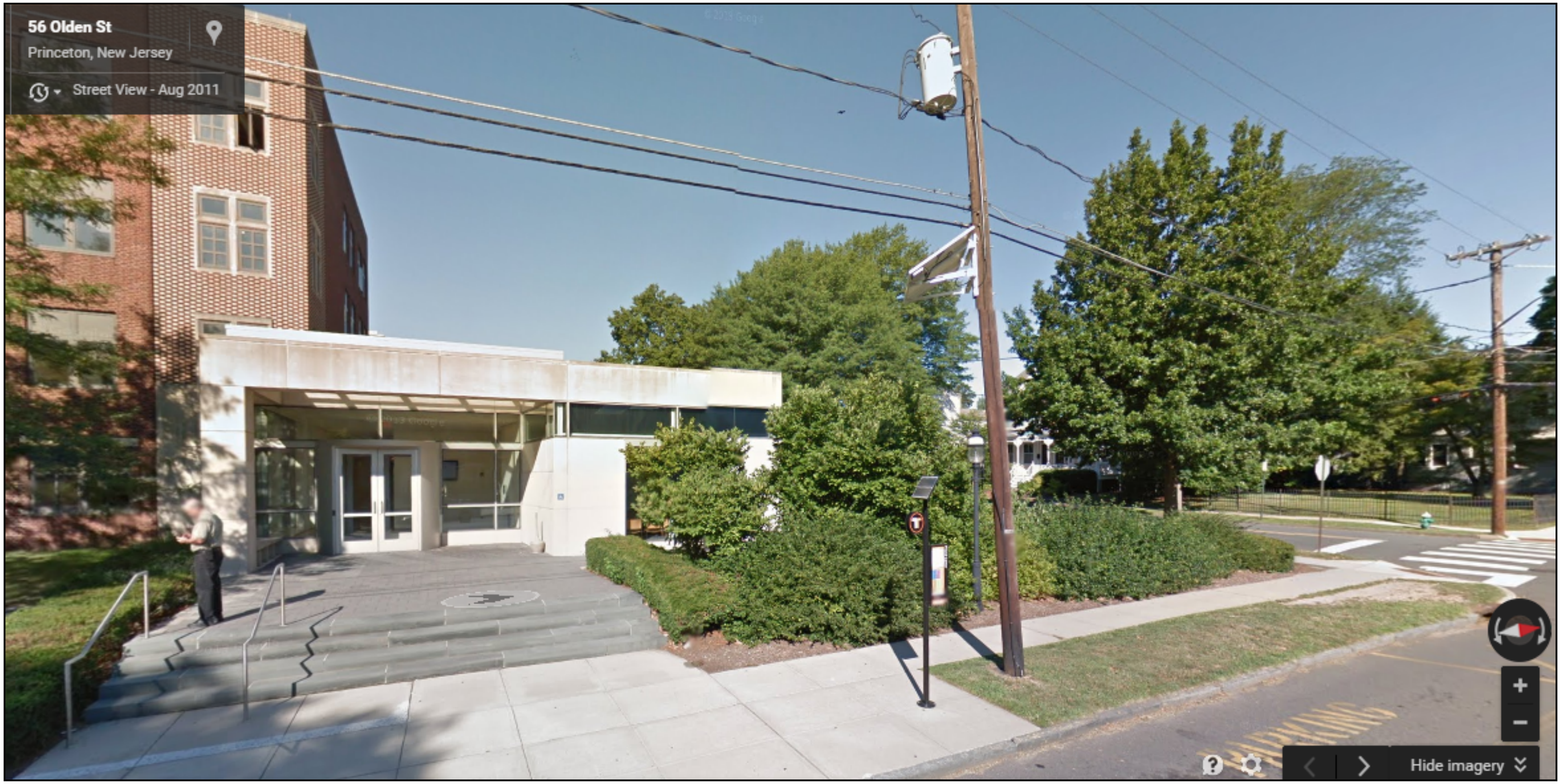
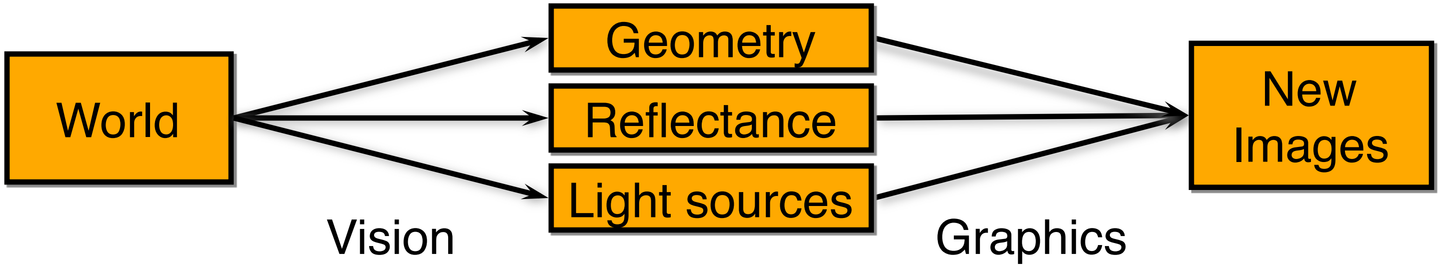
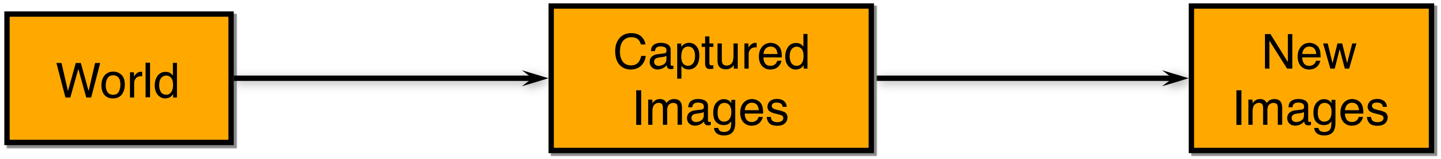


Image-Based Rendering

- Traditional vision / graphics rendering:



- Image-based rendering:

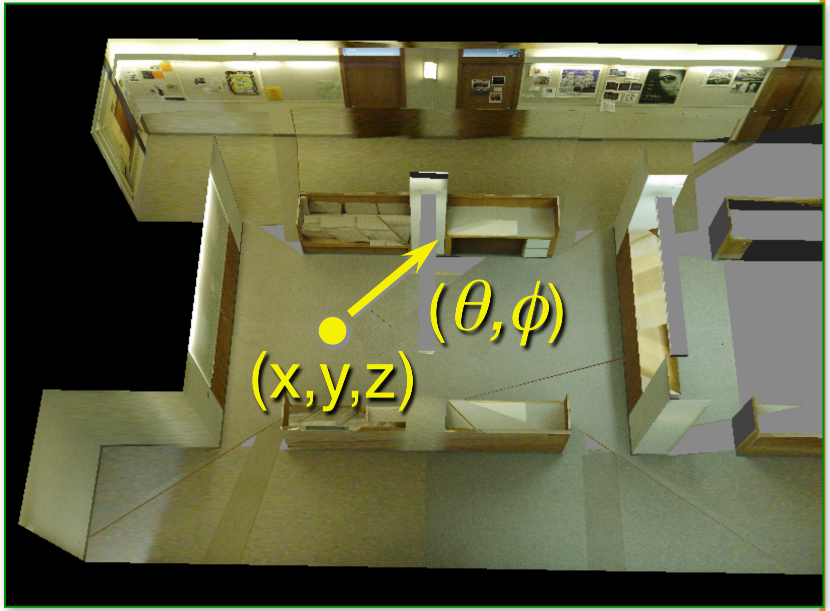


Plenoptic Function

- $L(x,y,z,\theta,\phi,t,\lambda)$

- 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

- Simplification from 7D to 3 × 5D
  - Represent color as RGB: eliminate λ
  - Static scenes: eliminate t

- Other simplfications?

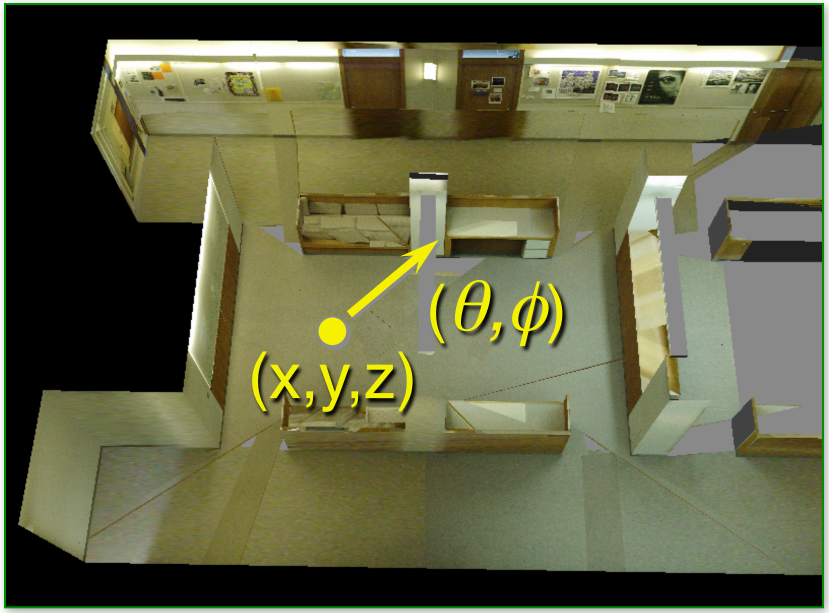
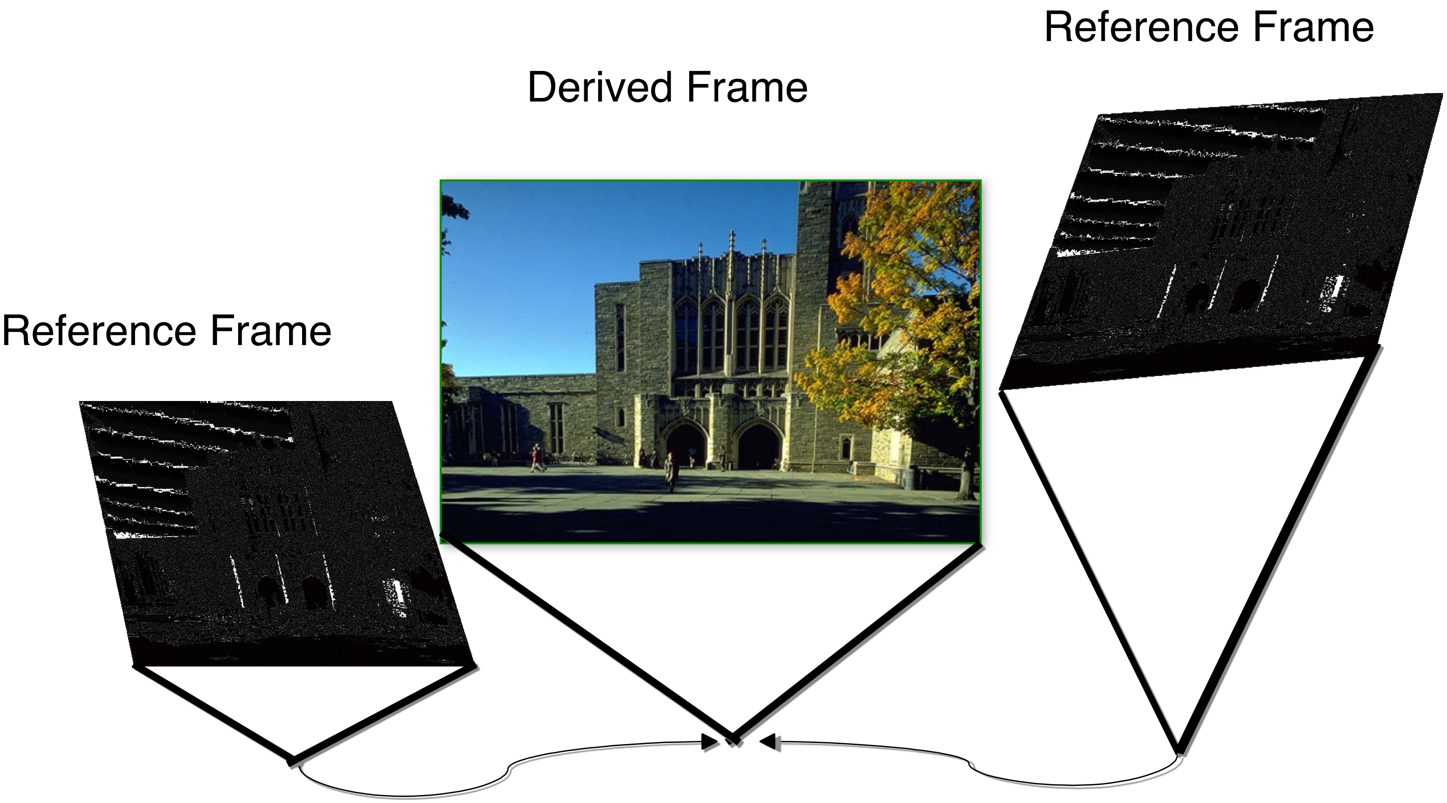


Image-Based Representations

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

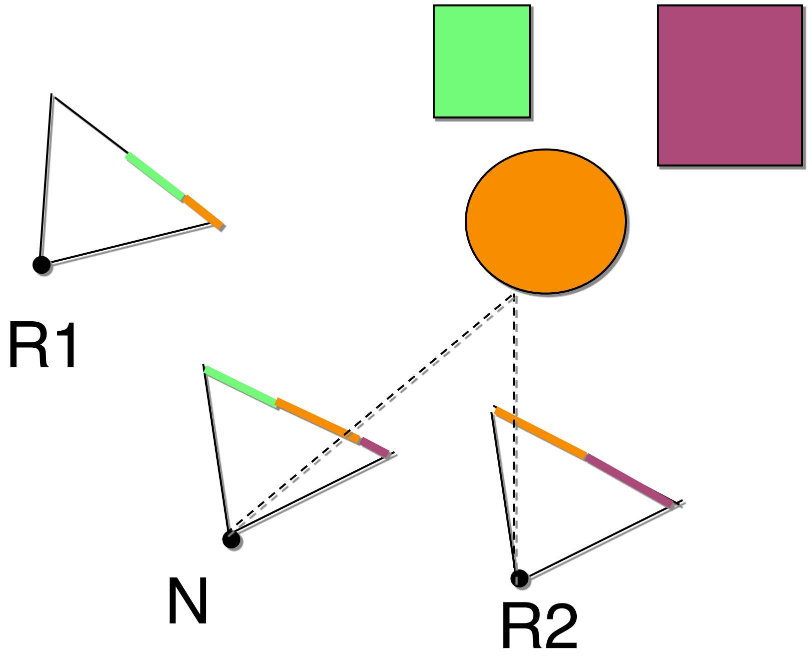
View Interpolation

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



View Interpolation

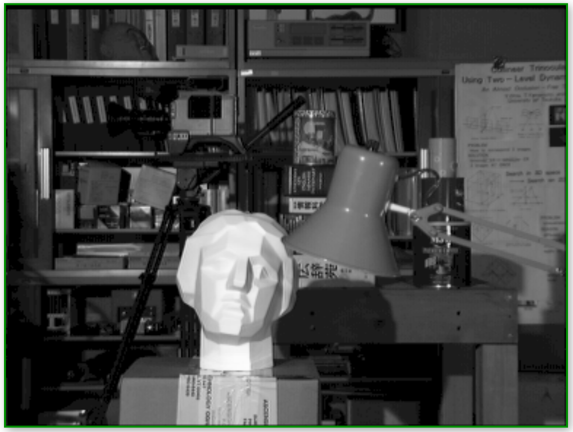
- Method:
  - Warp nearby reference images to novel viewpoint
  - Blend warped images



Morph with warp
defined by
pixel correspondences

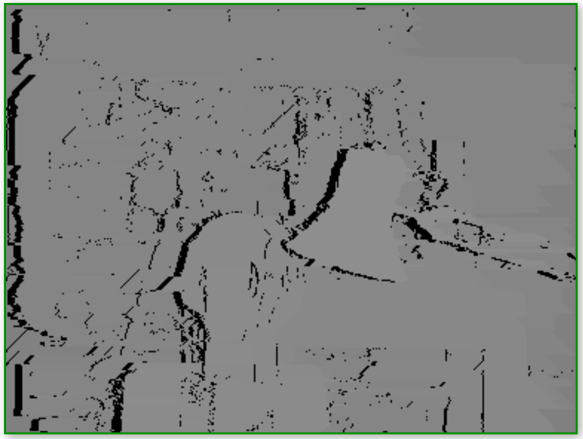
Pixel Correspondences

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



Left

Right

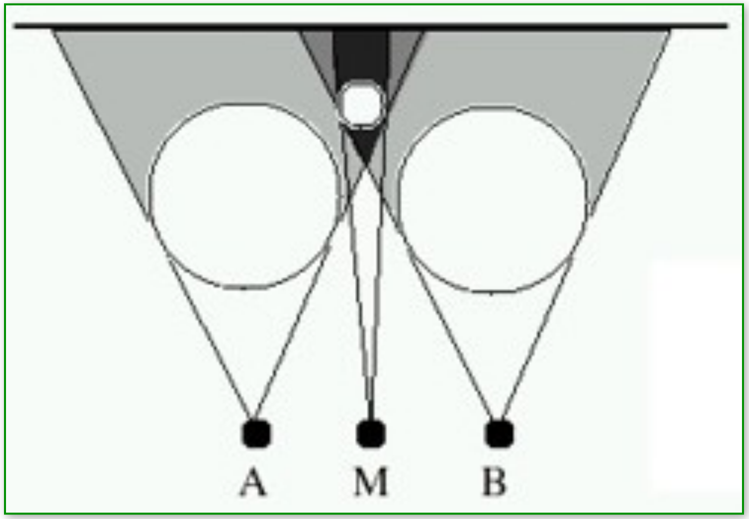


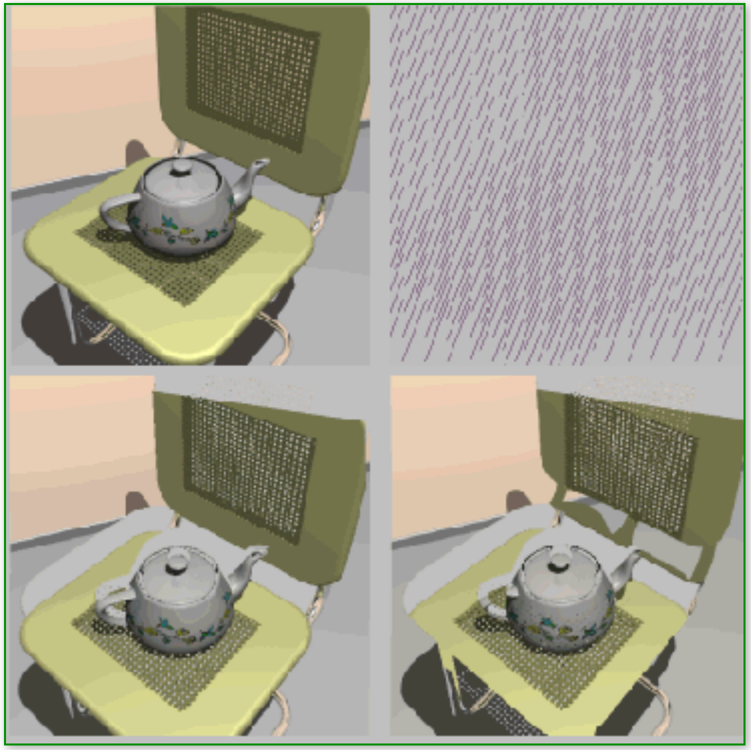
Disparity

[Szeliski]

View Interpolation

- Problem: changes in visibility
  - Disocclusions

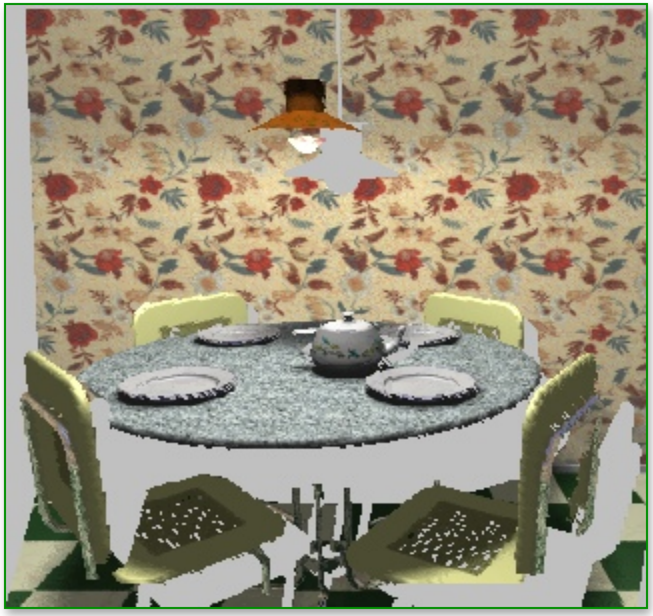


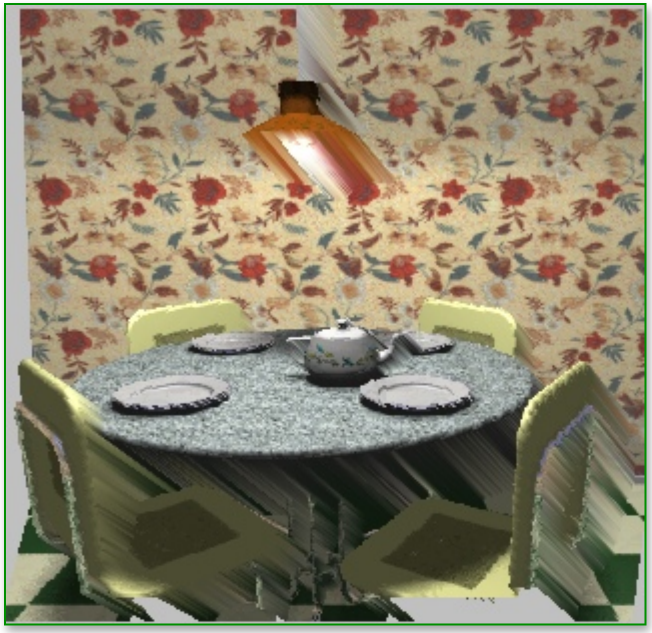


[McMillan]

Disocclusions

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

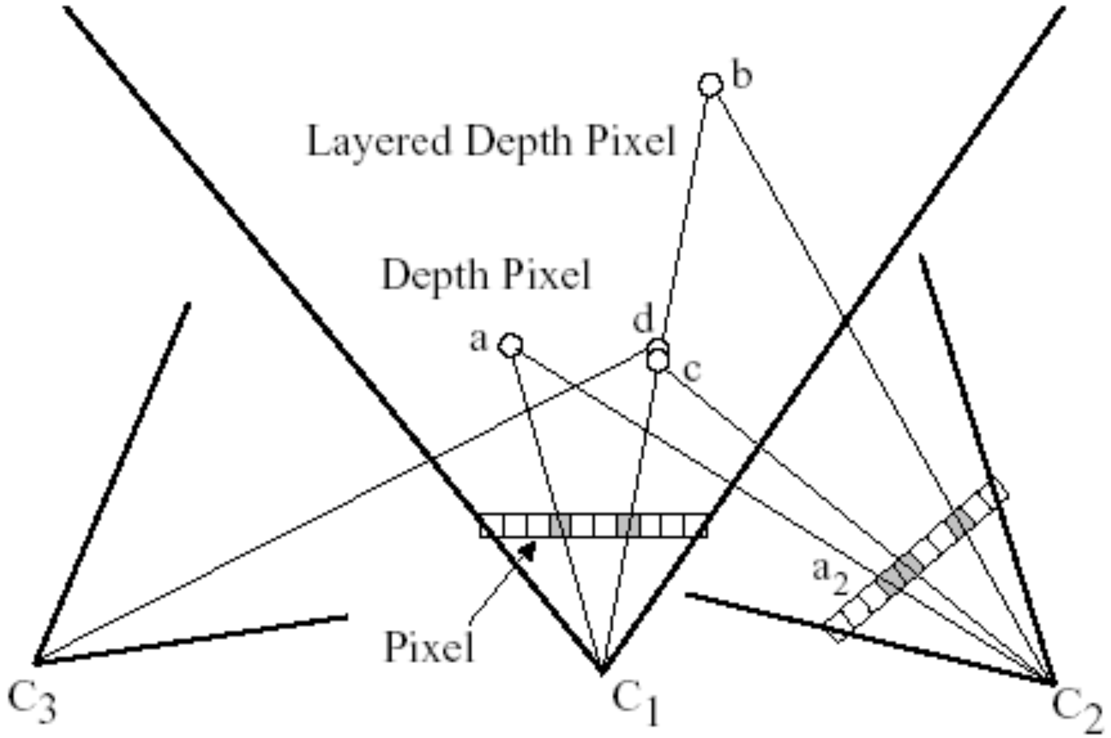




[McMillan]

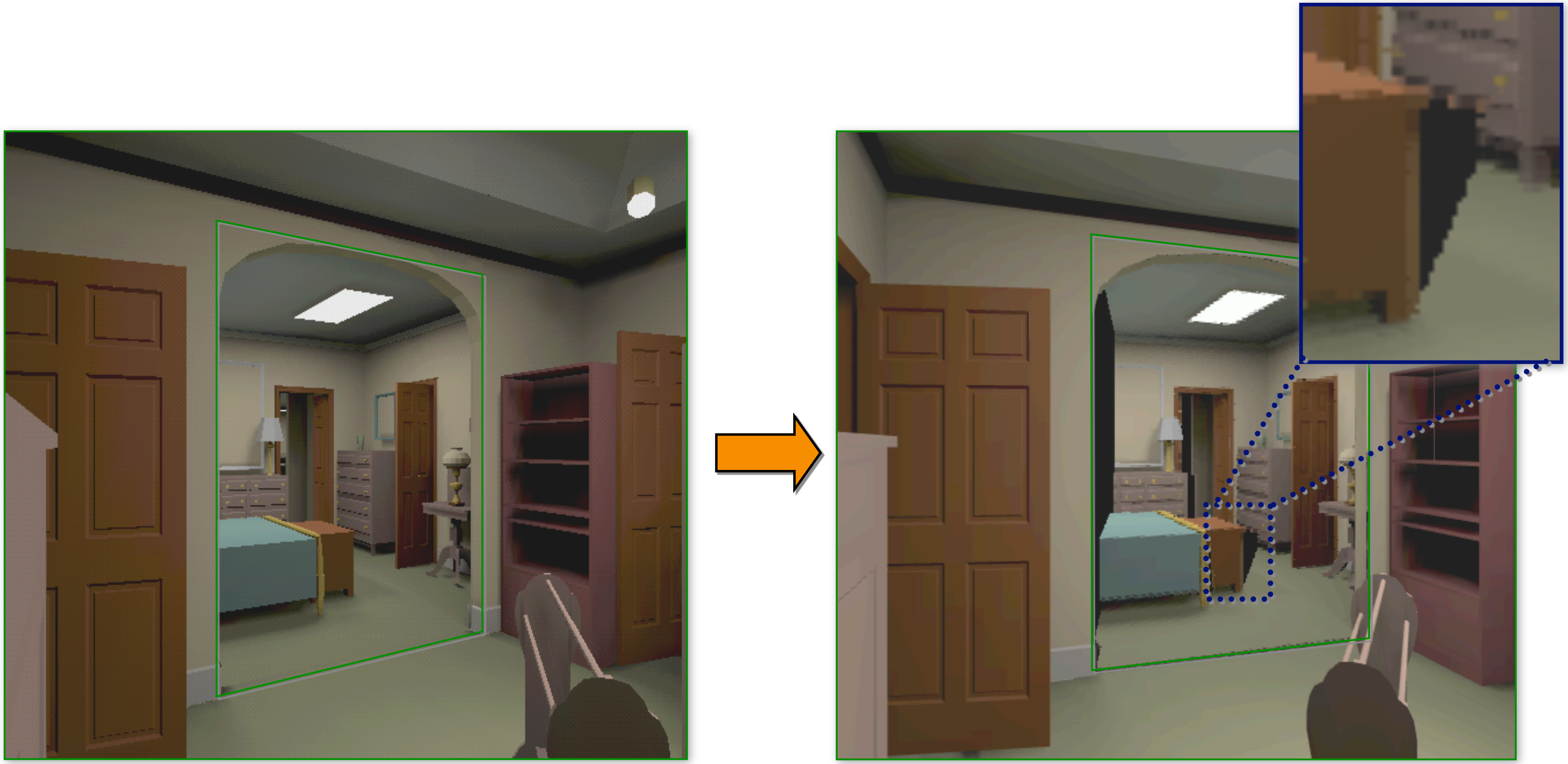
Disocclusions

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



Disocclusions

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



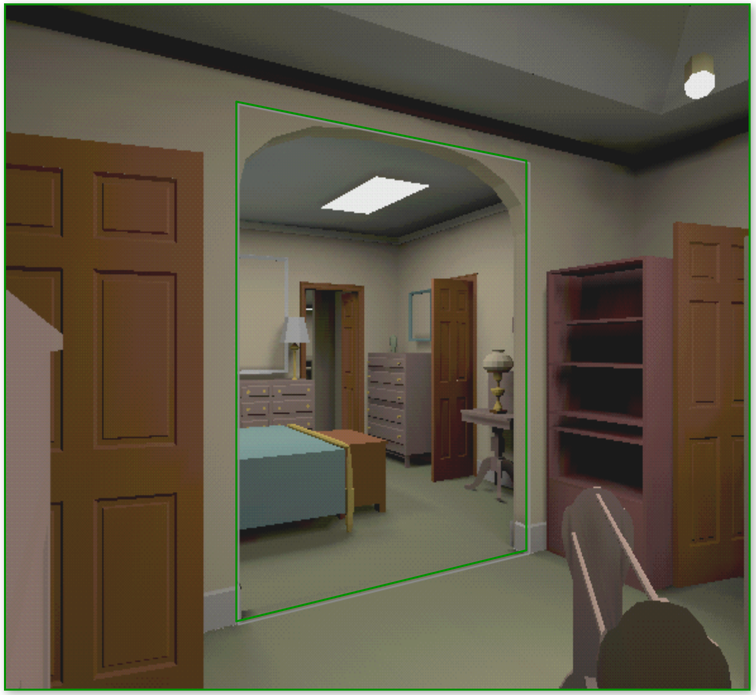
Reference Image

Warped Depth Image

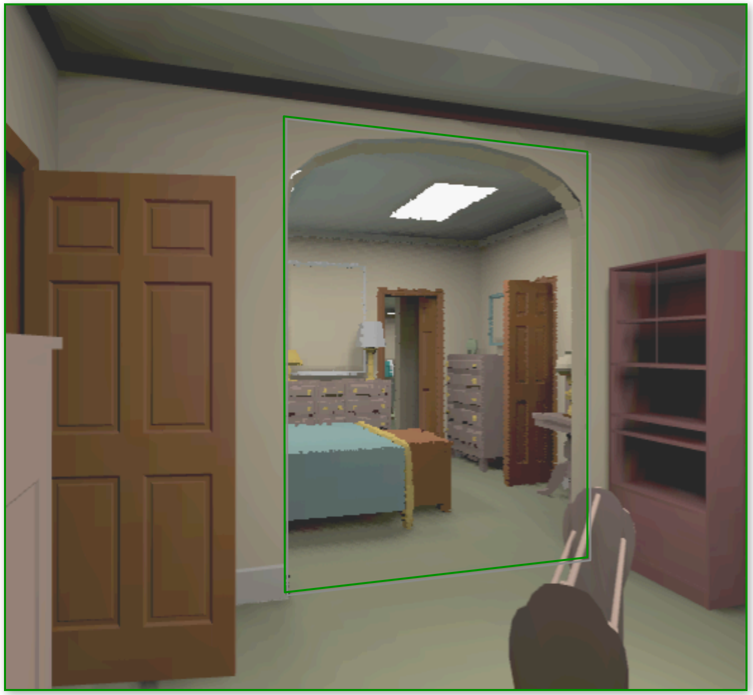
[Popescu]

Disocclusions

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



Reference Image



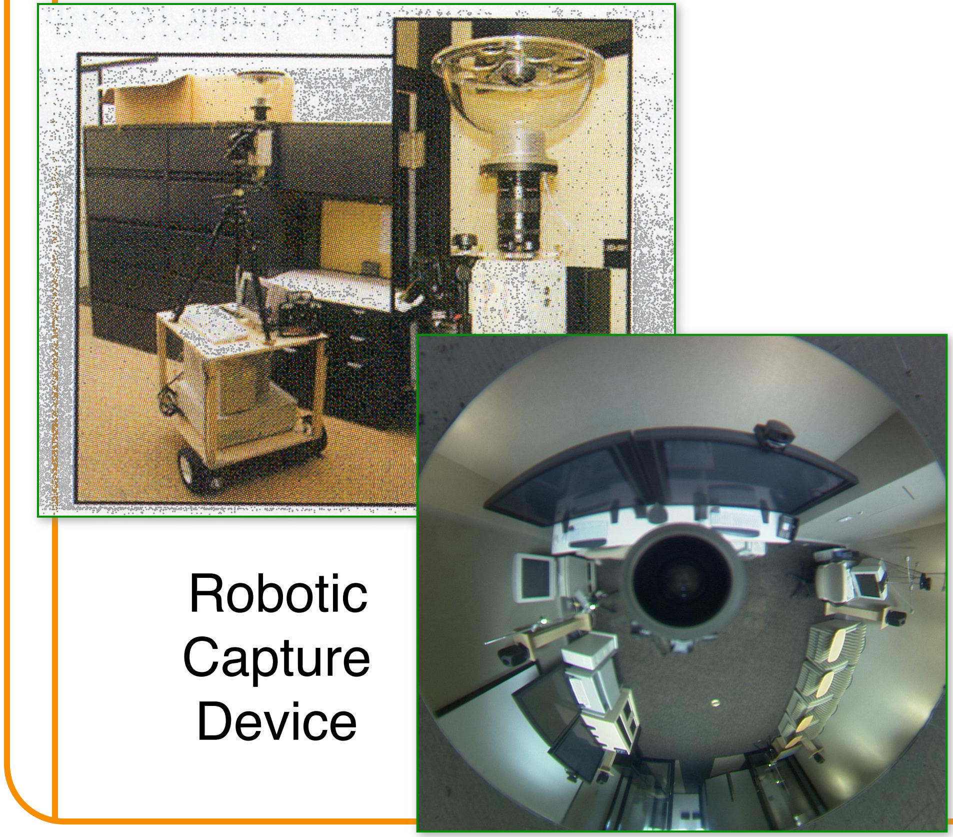
Warped Layered Depth Image

View Interpolation Challenges

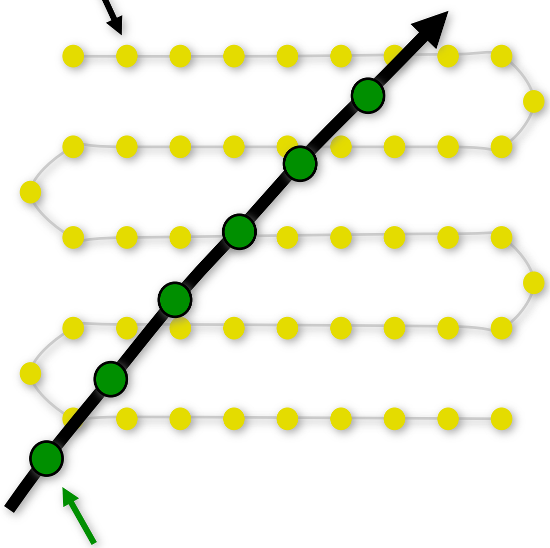
- Capture
  - How do we obtain a dense set of calibrated images over a large area in a practical manner?
- Data Management
  - How do we store and access the large amount of data?
- Rendering
  - How do we create novel views from a dense sampling of images in real-time?

Sea of Images

Dense sampling of plenoptic function with hemispherical camera moving on plane



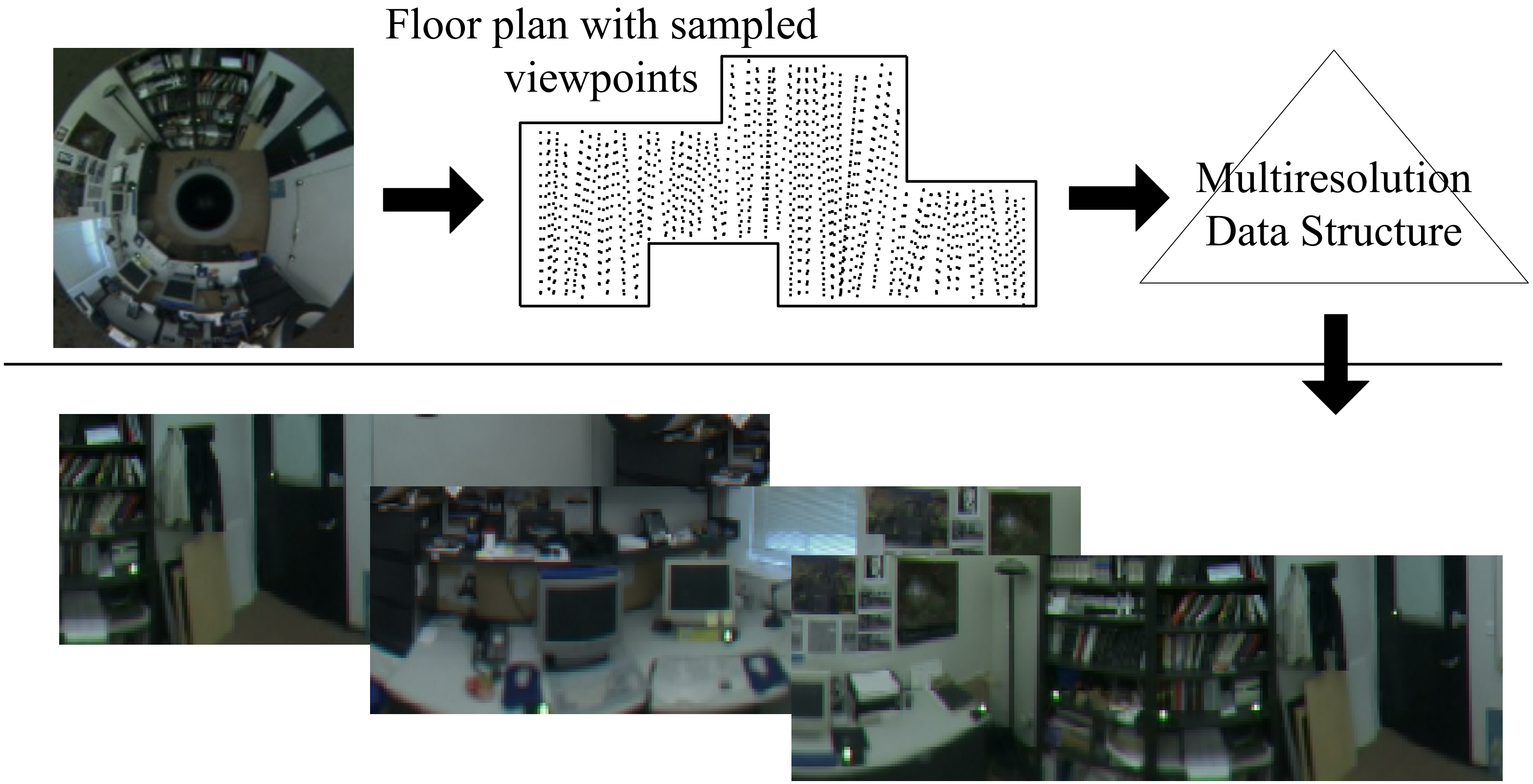
Robotic Capture Device



[Aliaga & Funkhouser]

Sea of Images

- Multiresolution compression for walkthroughs

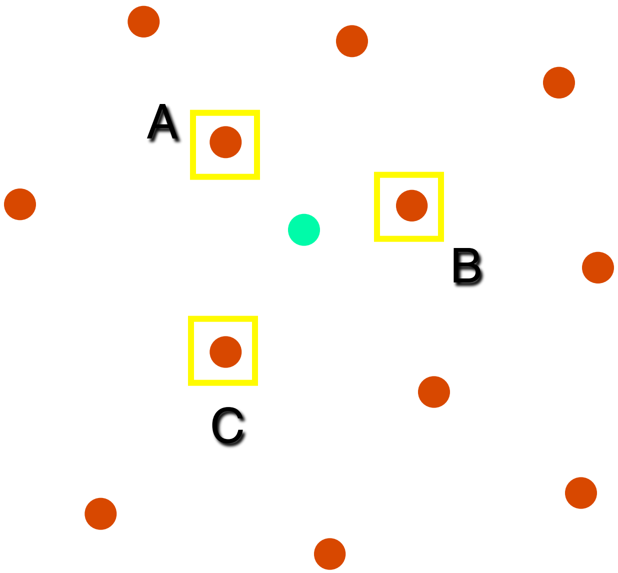


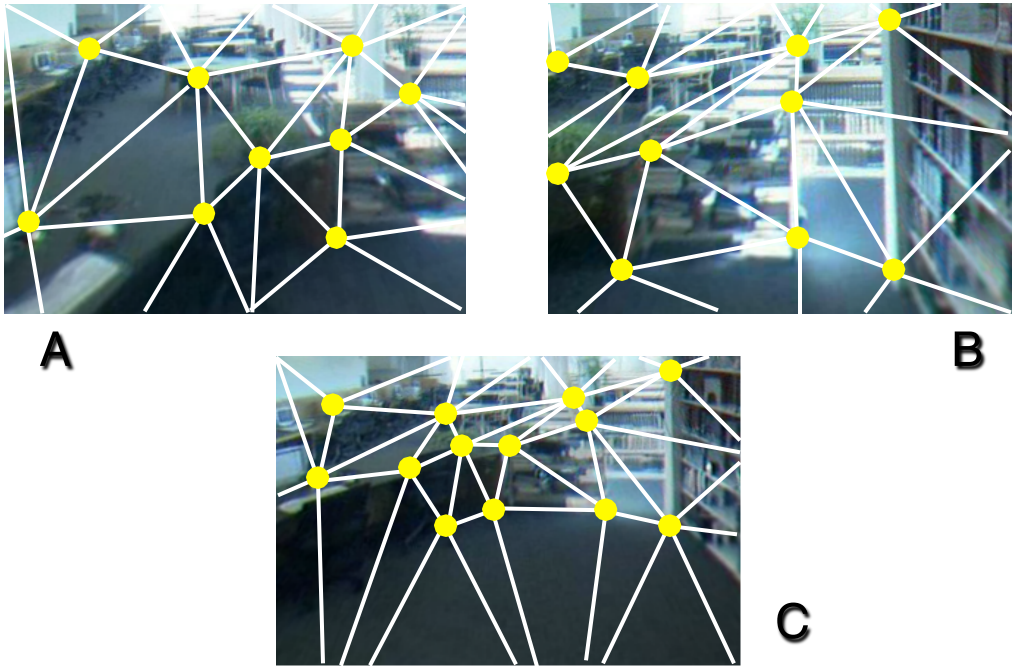
Floor plan with sampled viewpoints

Multiresolution Data Structure

Sea of Images Rendering

Interpolate three nearest views using detected feature correspondences



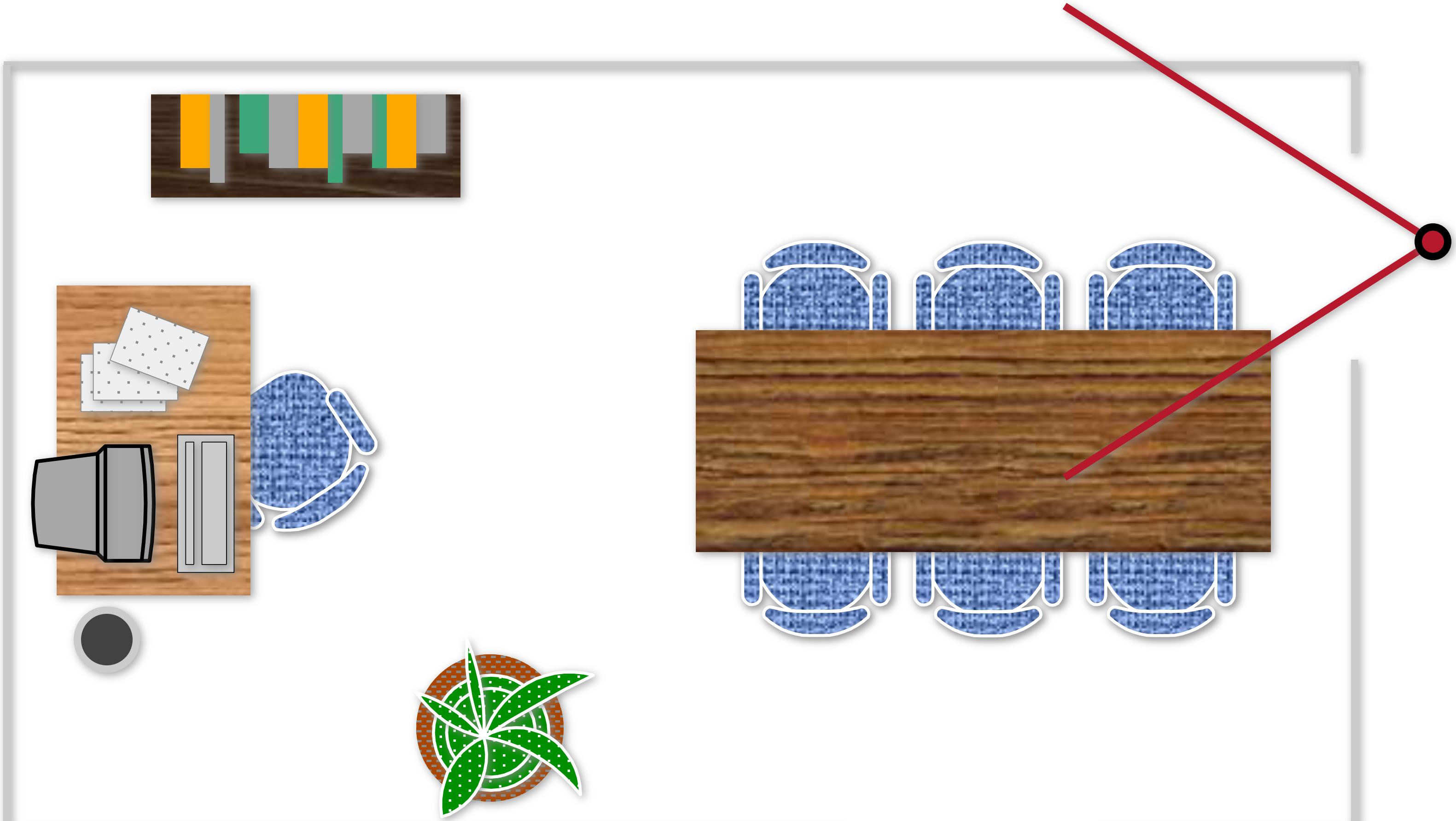


Replacing Geometry with Images

- 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

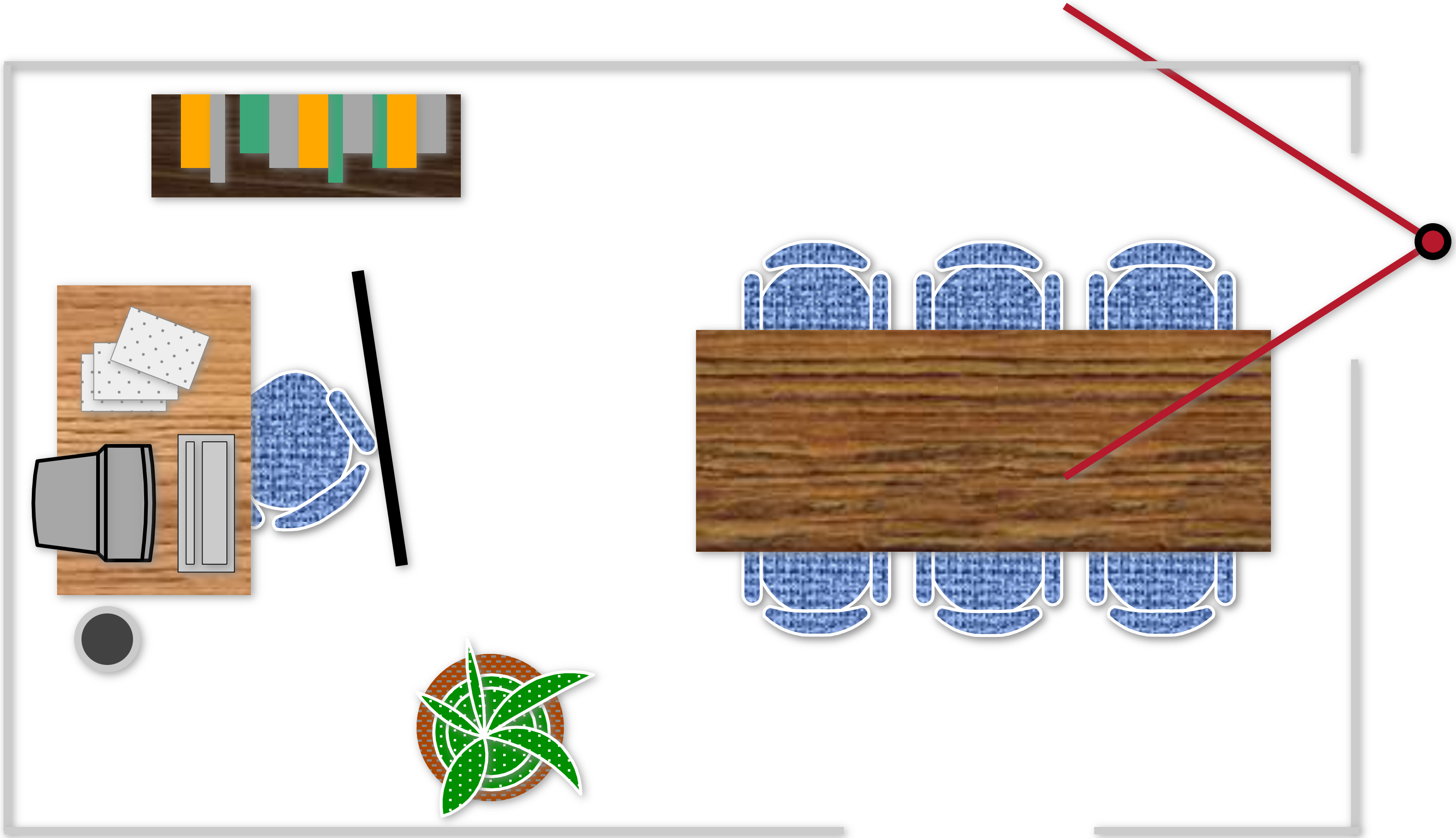
[Aliaga]

Simple Example



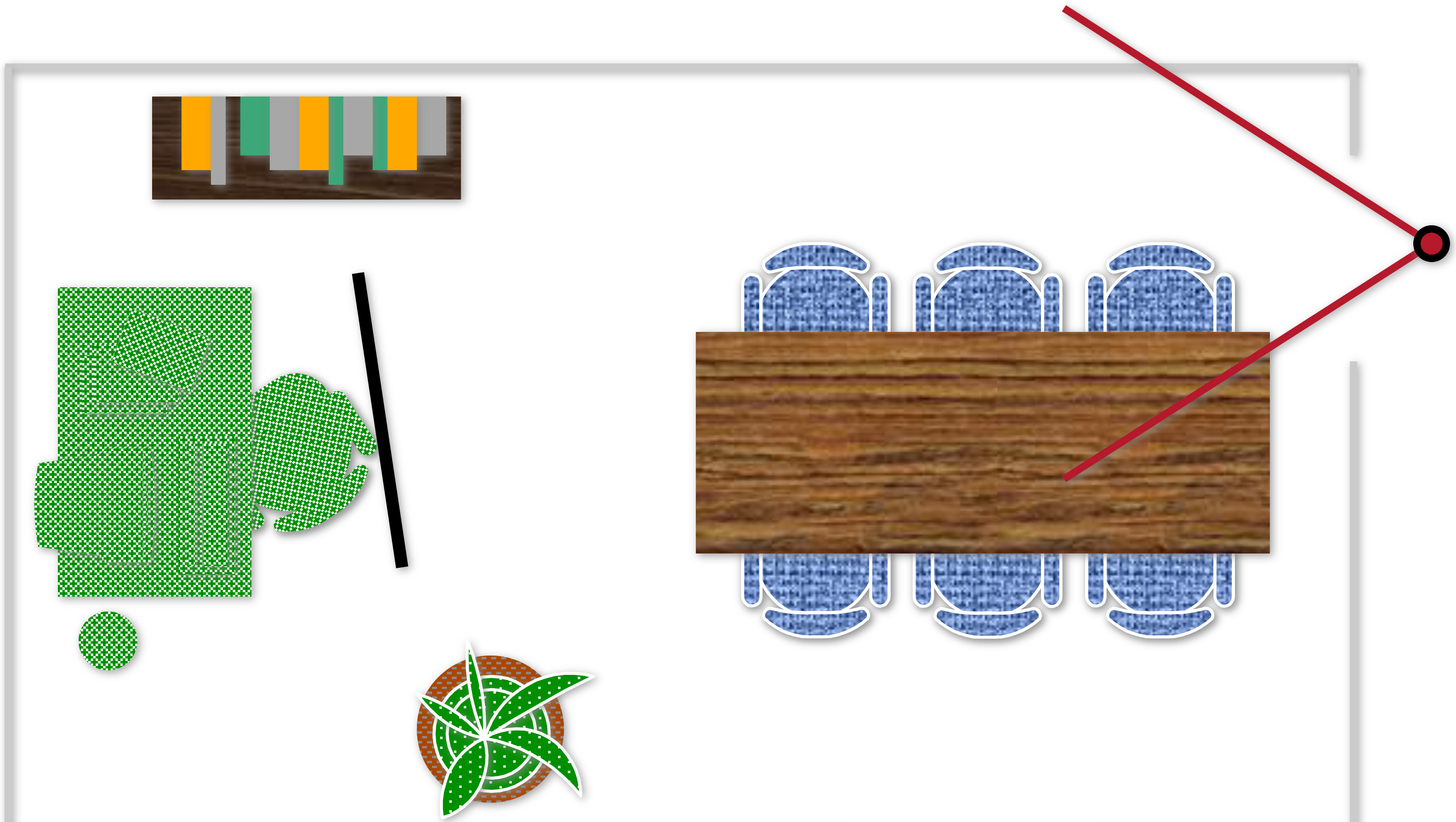
[Aliaga]

Simple Example



[Aliaga]

Simple Example



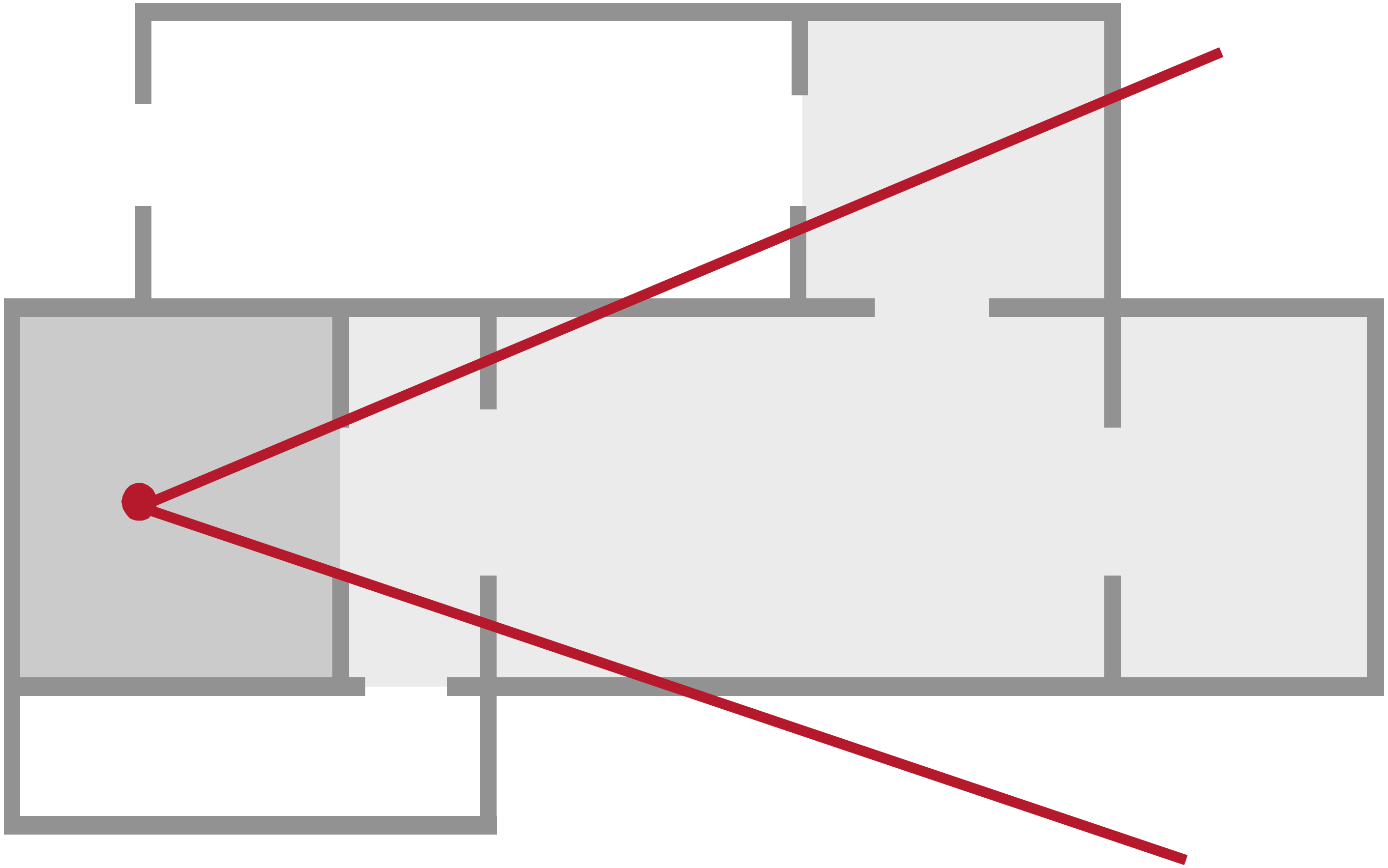
[Aliaga]

Automatic Image-Placement

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

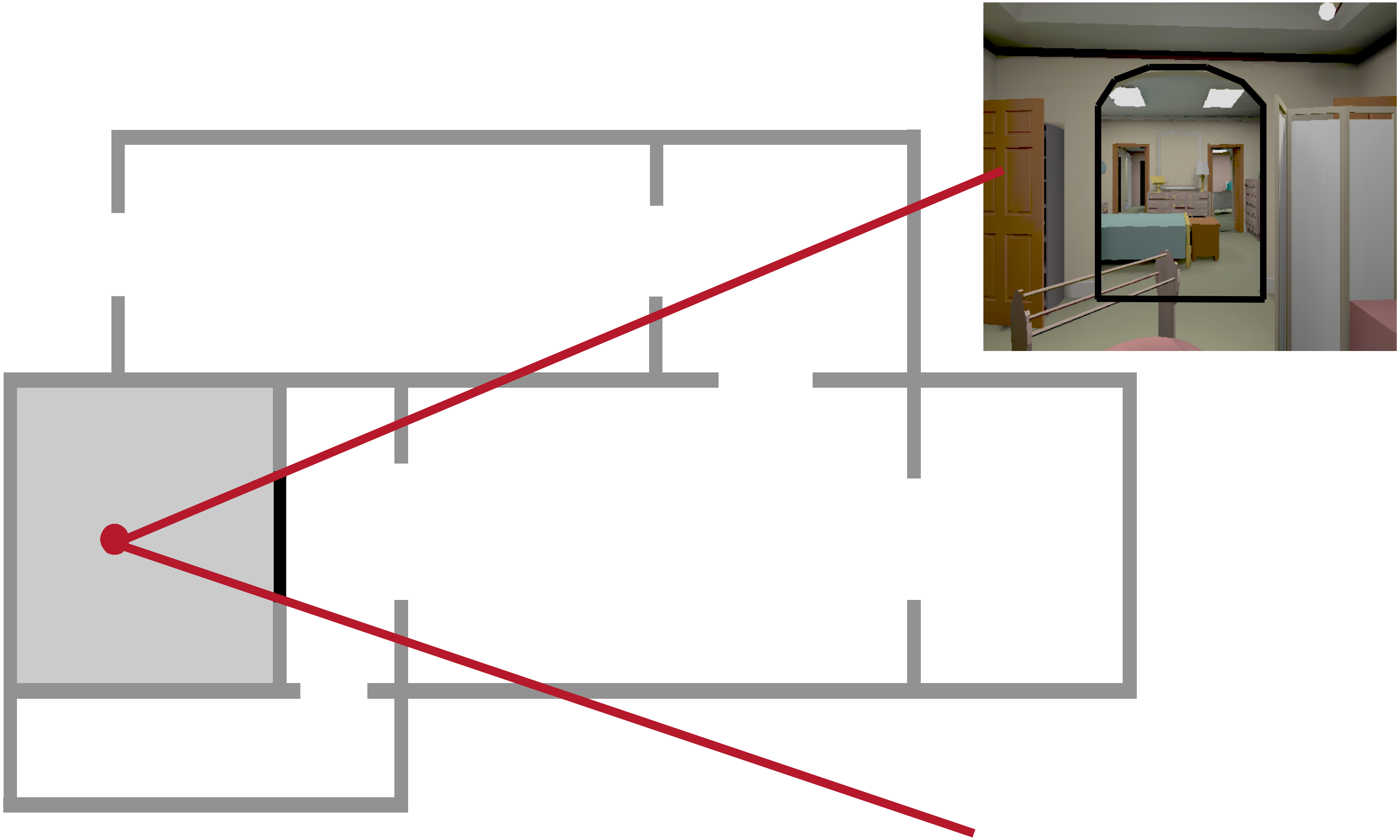
[Aliaga]

Cells and Portals



[Aliaga]

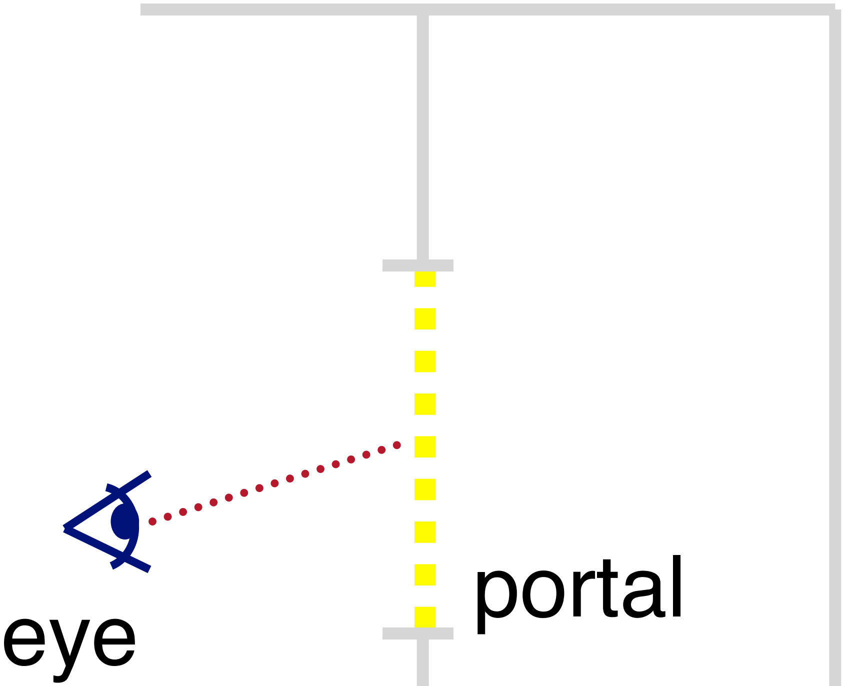
Portal Images



[Aliaga]

Creating Portal Images

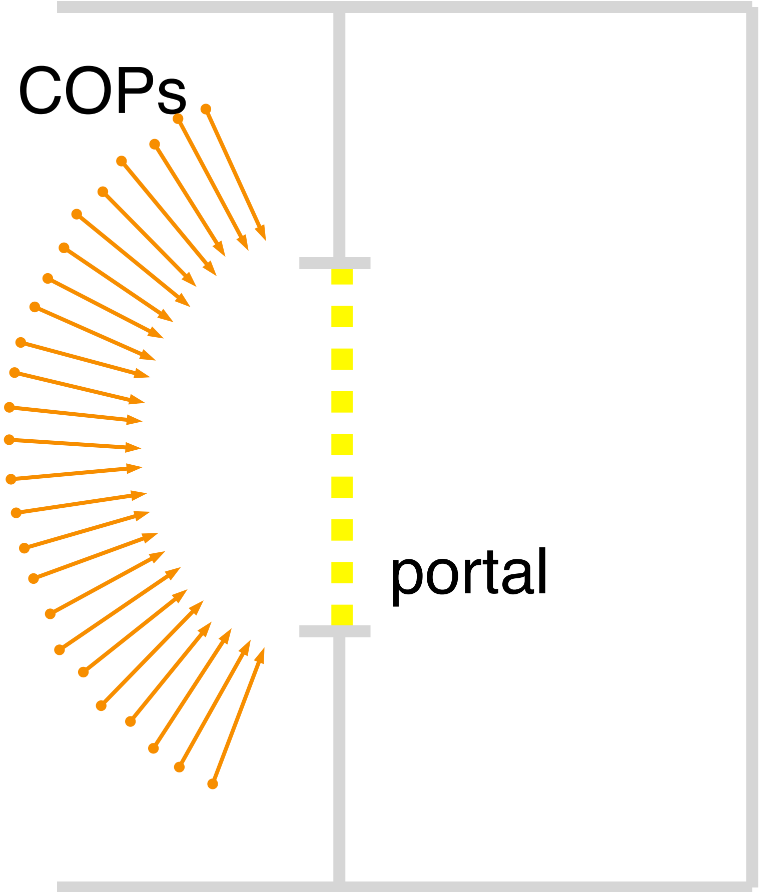
Ideal portal image would be one sampled from the current eye position



[Aliaga]

Creating Portal Images

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

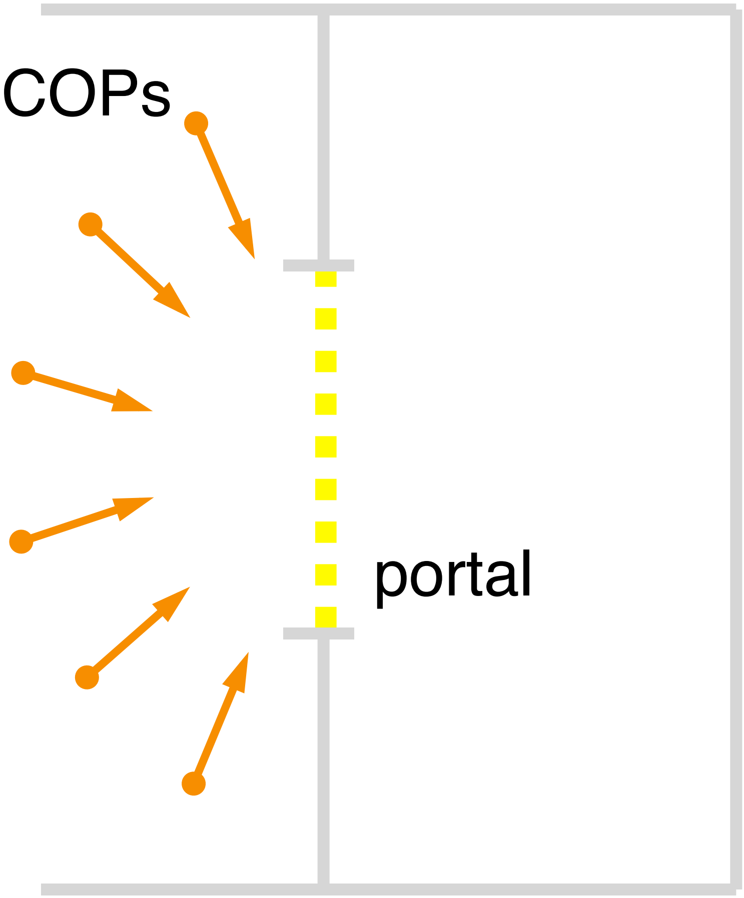


[Aliaga]

Creating Portal Images

or…

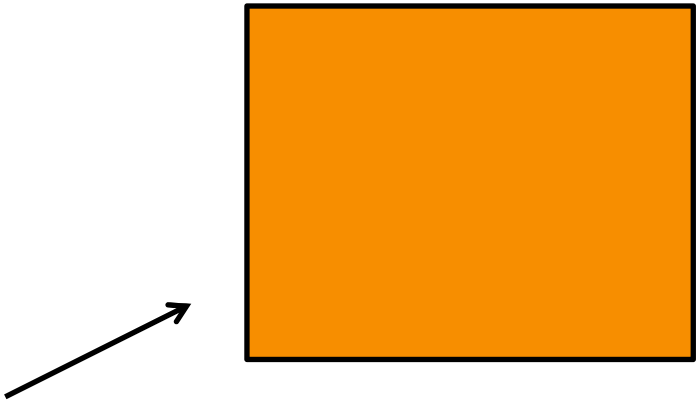
Warp one of a much smaller number of reference images



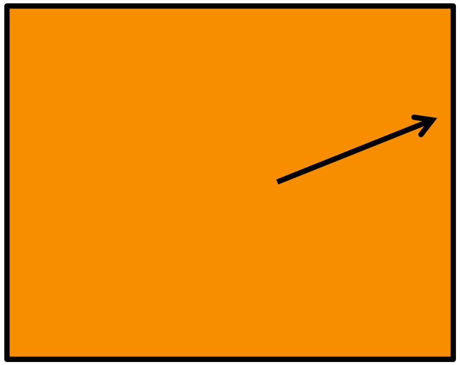
[Aliaga]

Lightfields

In unoccluded space reduce plenoptic function to 4D



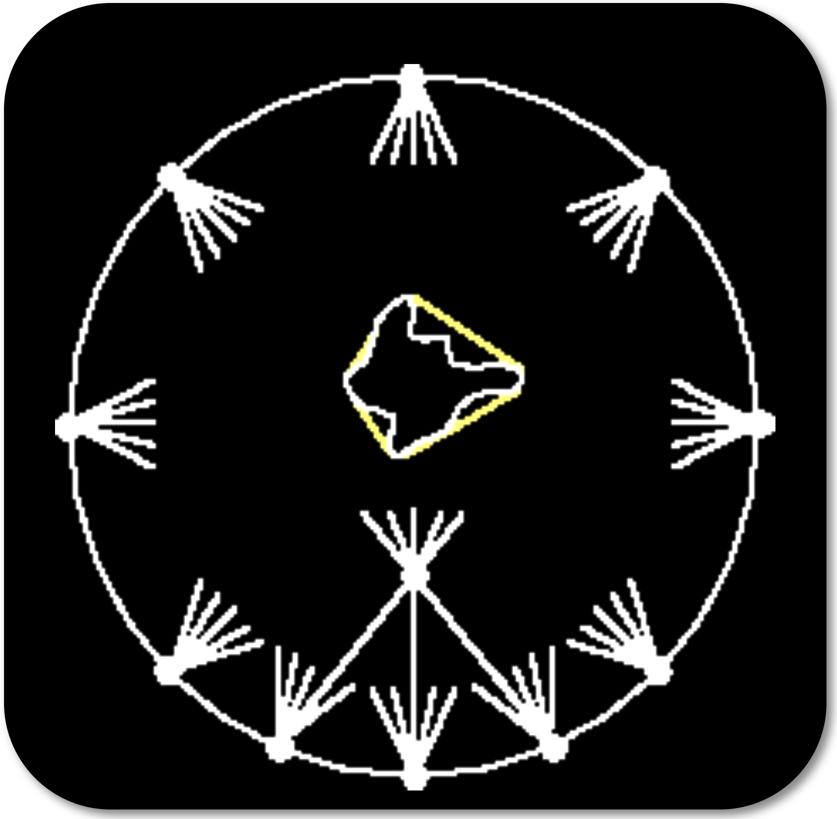
Outside looking in



Inside looking out

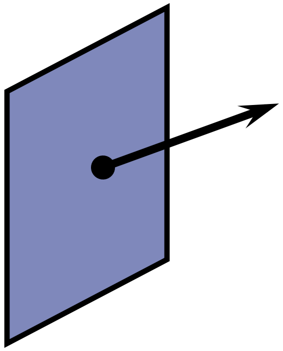
Using Lightfields

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

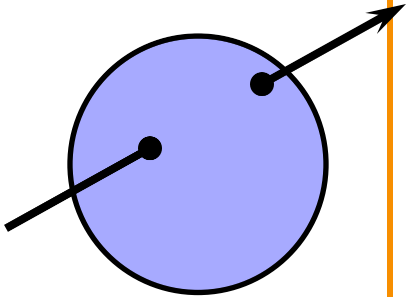


Lightfield Parameterization

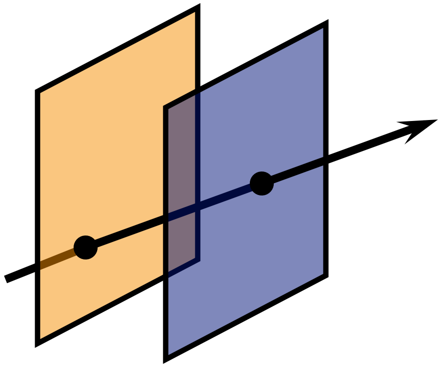
- Point / angle



- Two points on a sphere



- Points on two planes

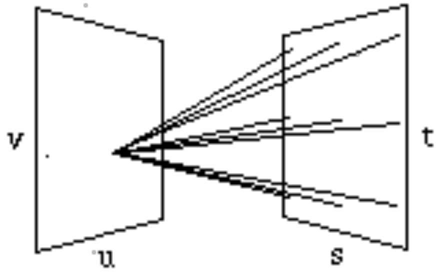


- Original images and camera positions

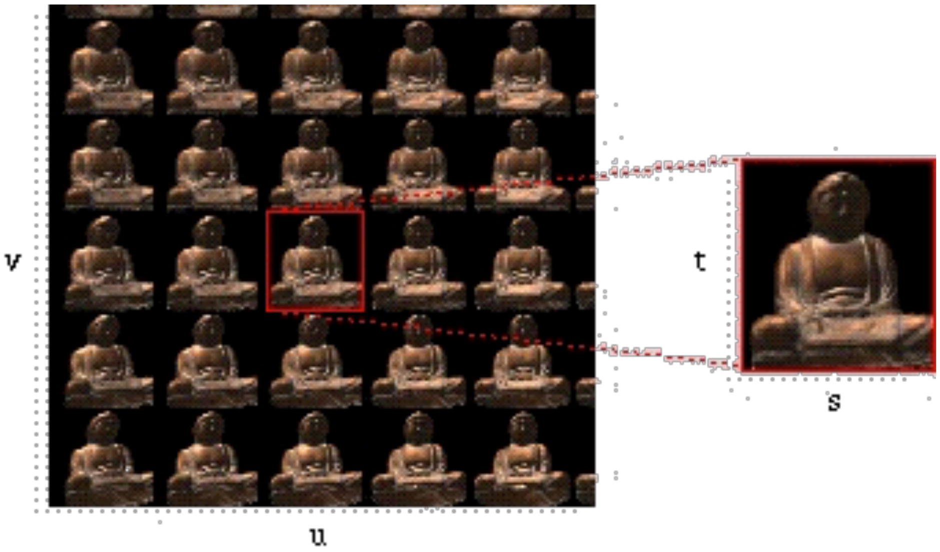
Two-Plane Parameterization

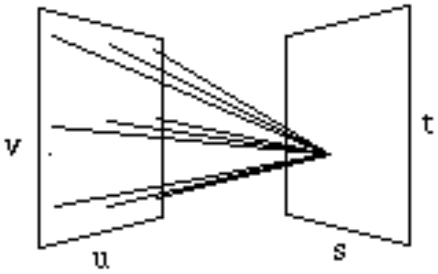
- Two planes, evenly sampled: “light slab”
- In general, planes in arbitrary orientations
- In practice, one plane = camera locations
  - Minimizes resampling

Two-Plane Parameterization

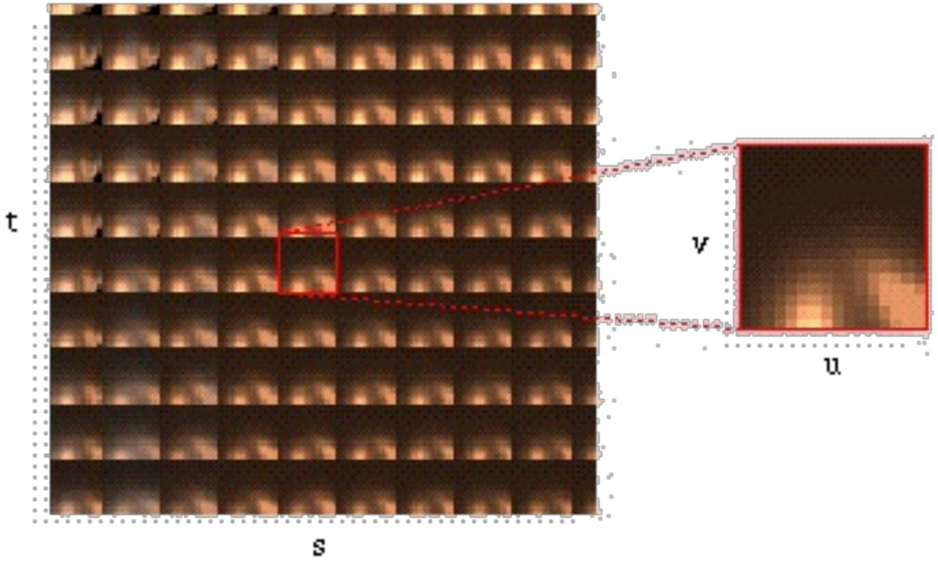


(a)

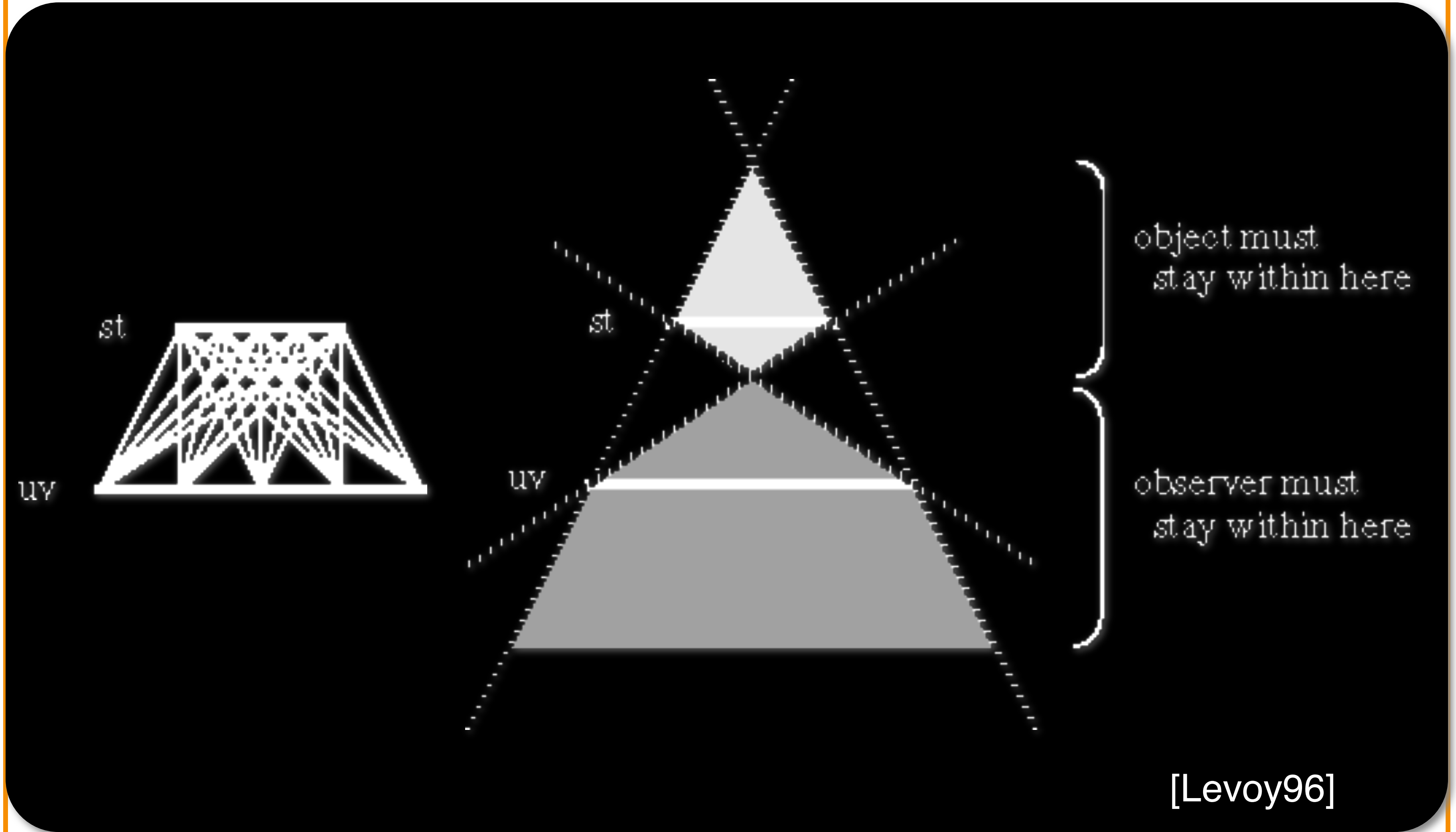




(b)

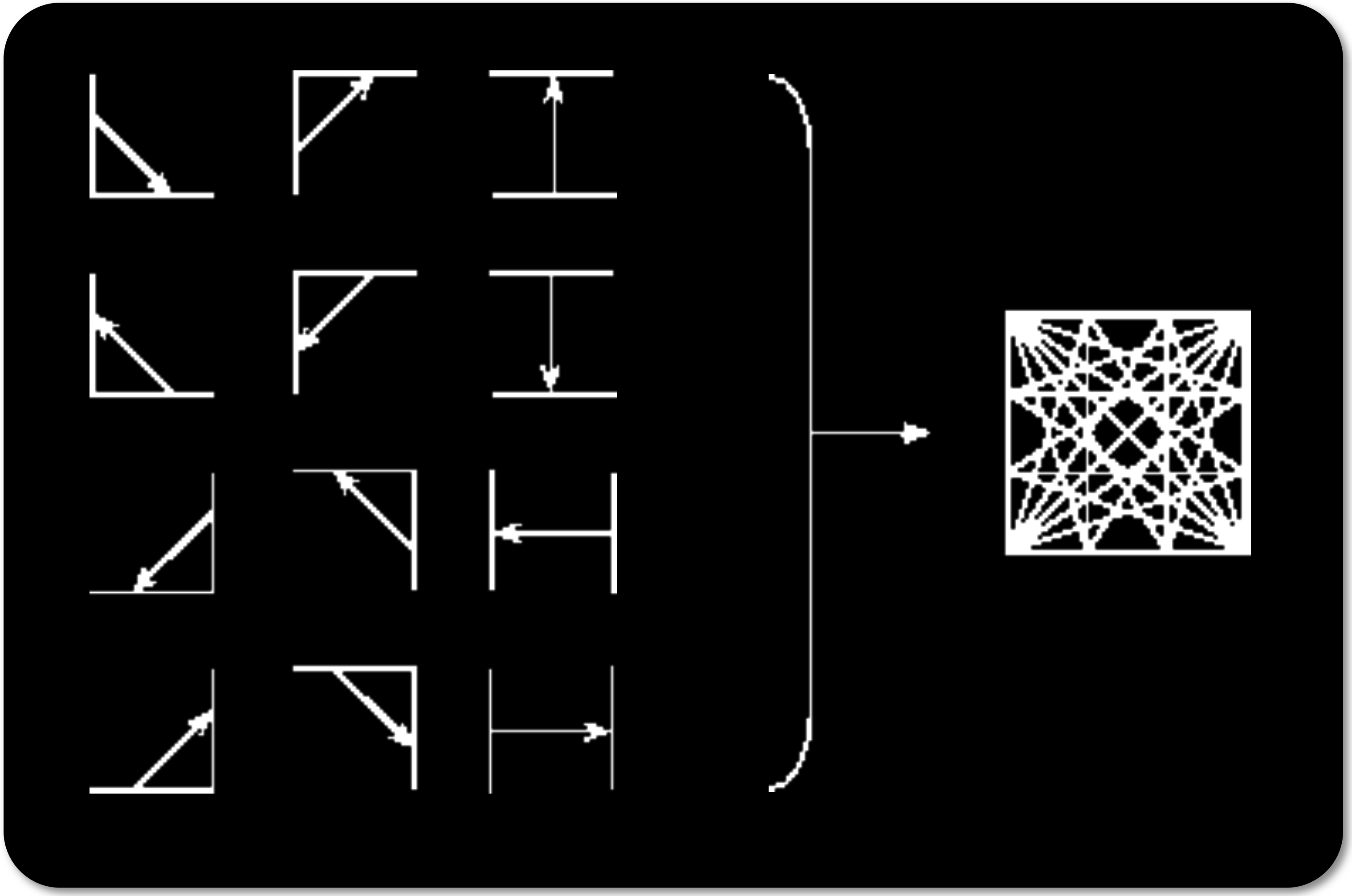


Light Field Coverage



[Levoy96]

Multi-Slab Light Fields



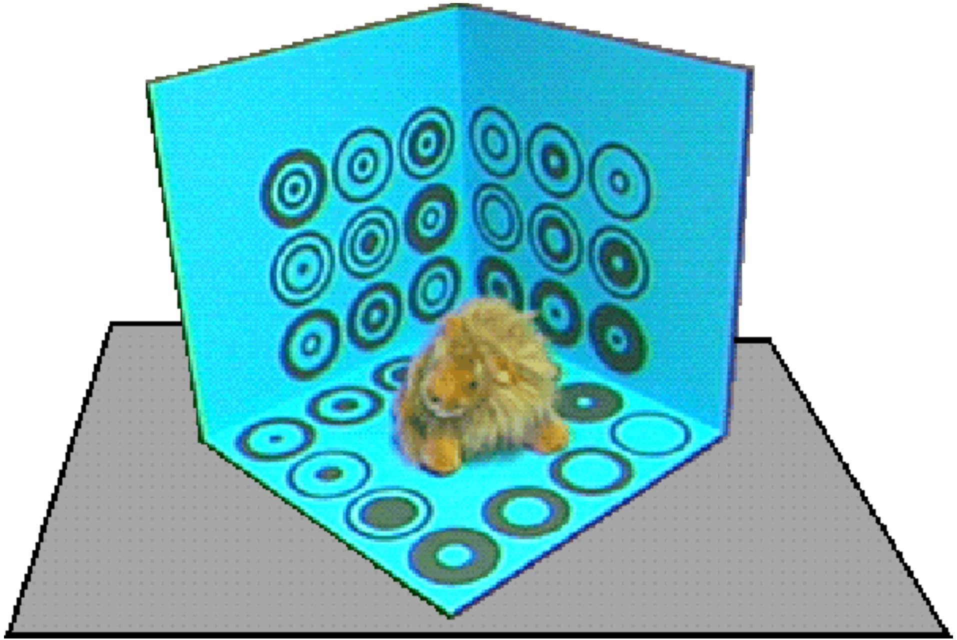
[Levoy96]

Lightfield Capture

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

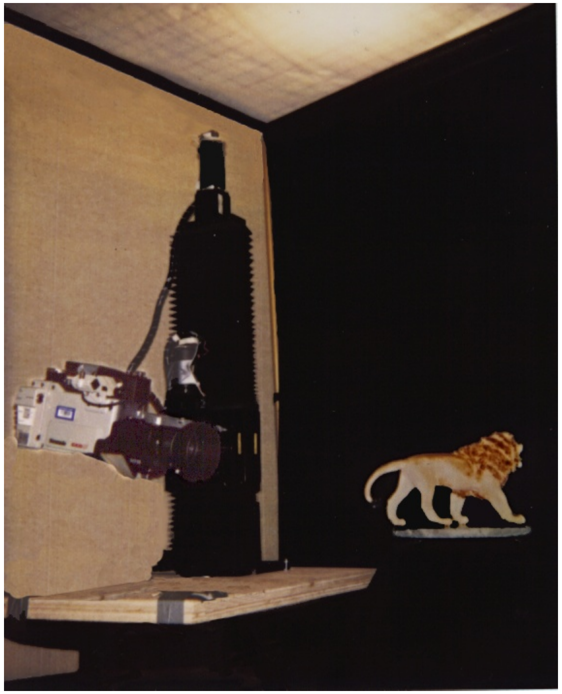
Lightfield Capture

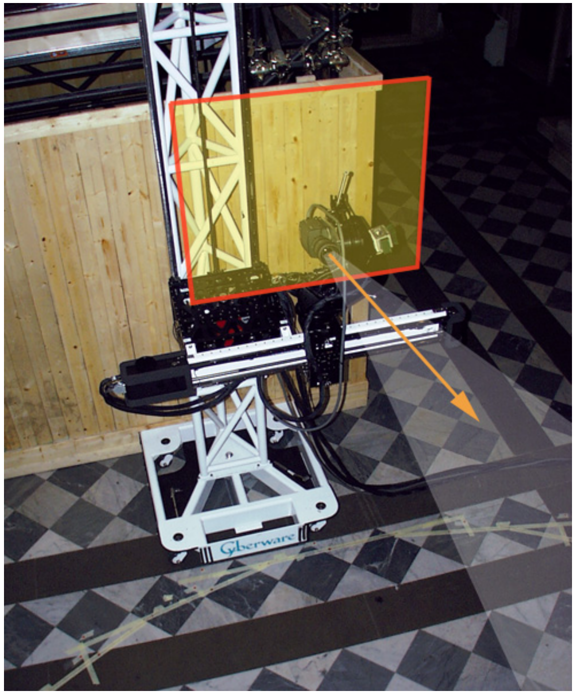
- Capture: move camera by hand
- Camera intrinsics assumed calibrated
- Camera pose recovered from markers



Lightfield Capture

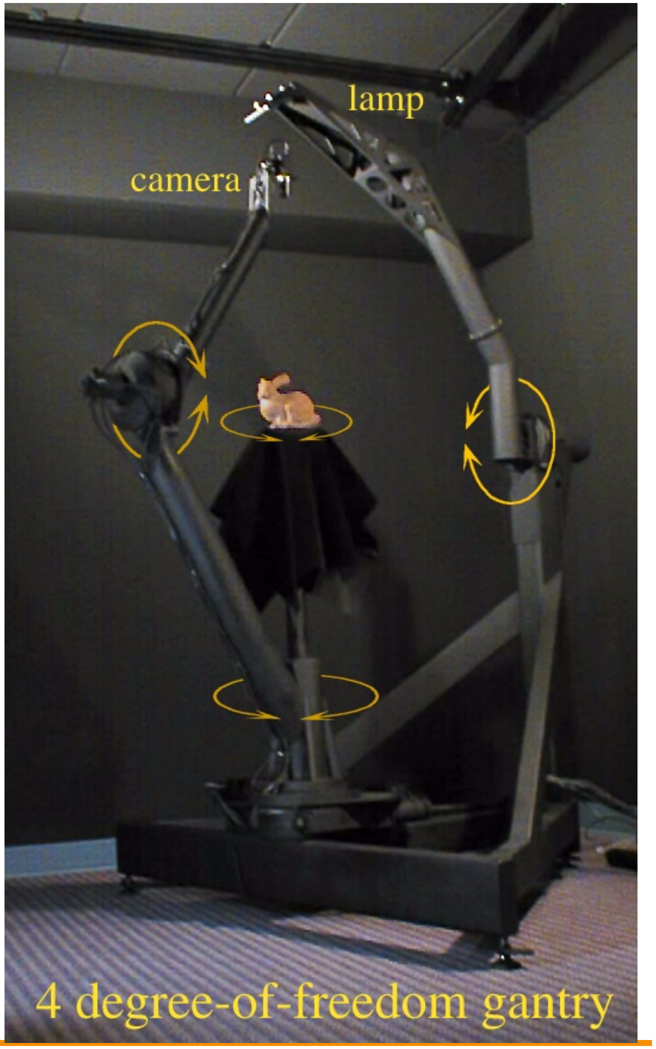
- Computer-controlled camera rig
  - Move camera to grid of locations on a plane





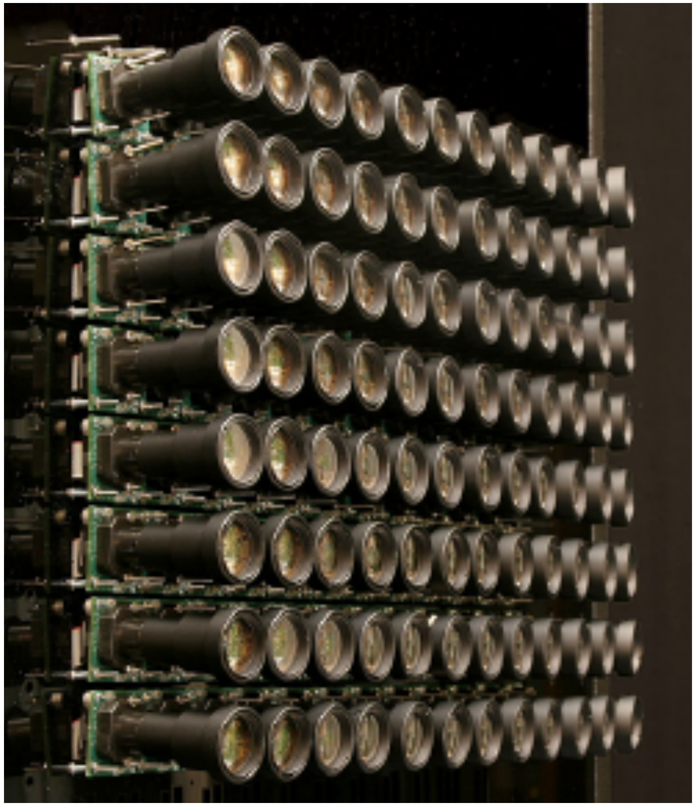
Lightfield Capture

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



Lightfield Capture

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



(Bennett Wilburn, Michal Smulski, Mark Horowitz)

Lightfield Capture

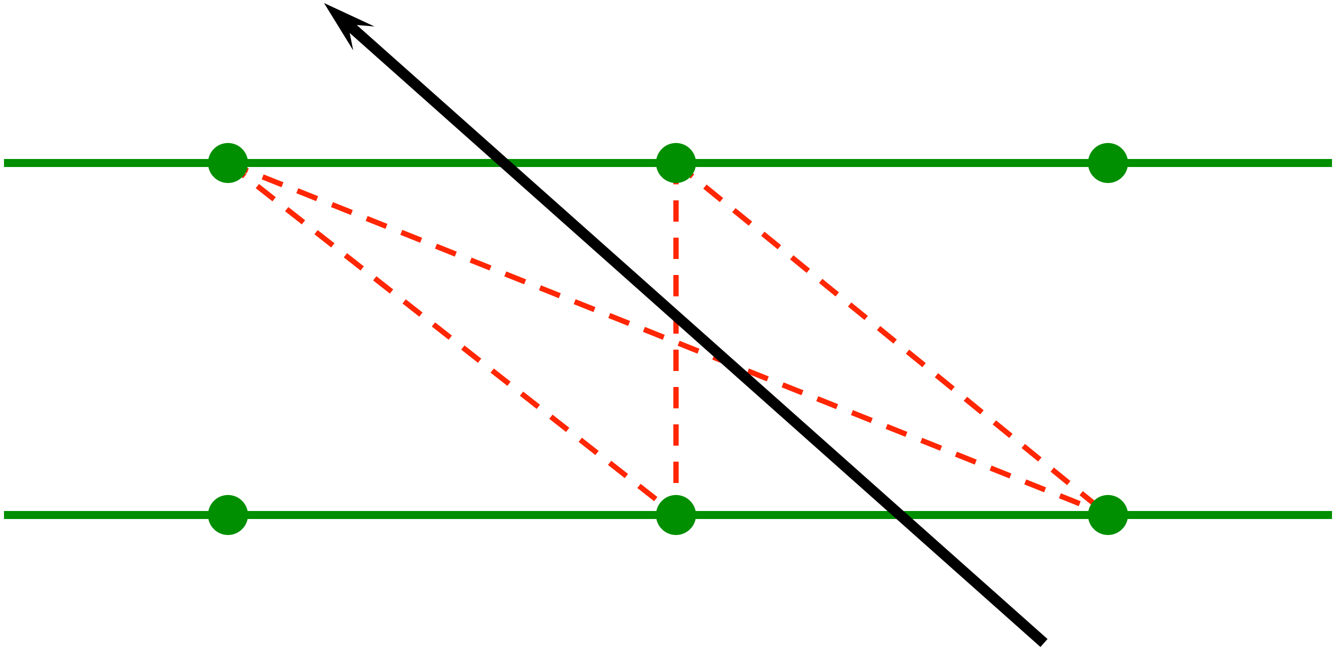
Lytro

Lightfield Compression

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

Lightfield Rendering

- How to select rays?
- How to interpolate

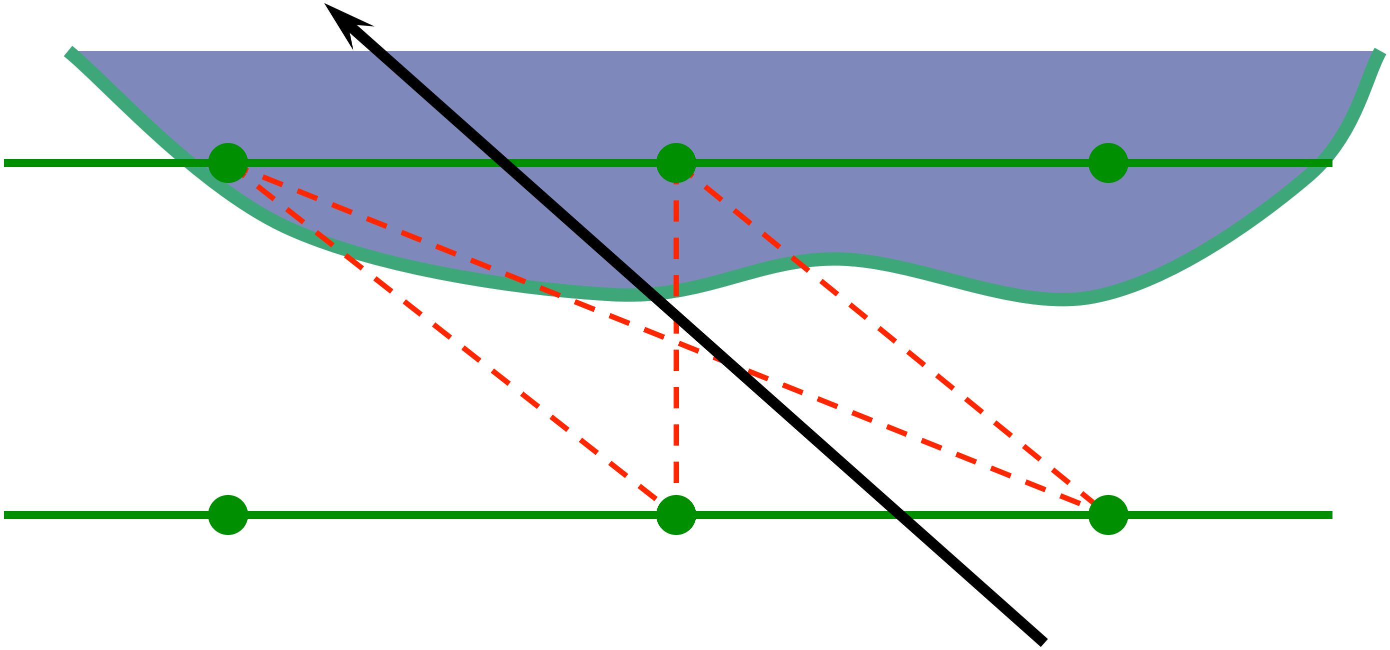


Lightfield Rendering

- 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)

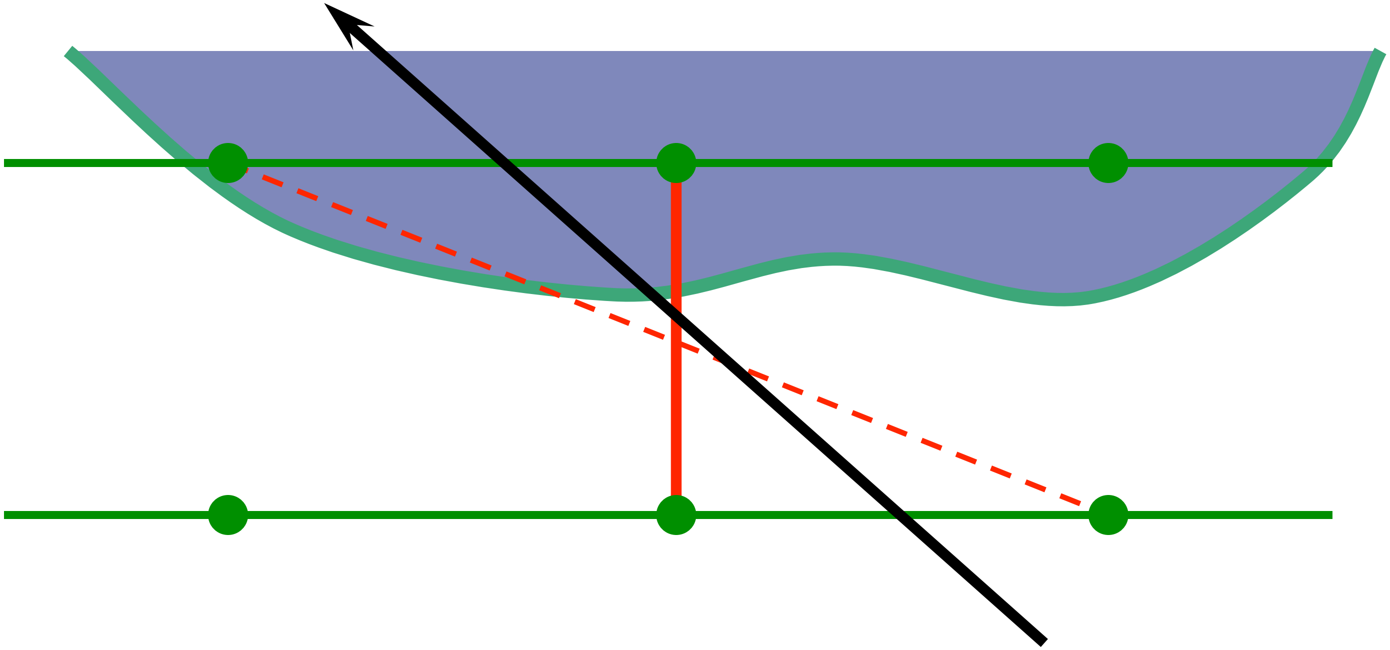
Lightfield Rendering

- Use rough depth information to improve rendering quality

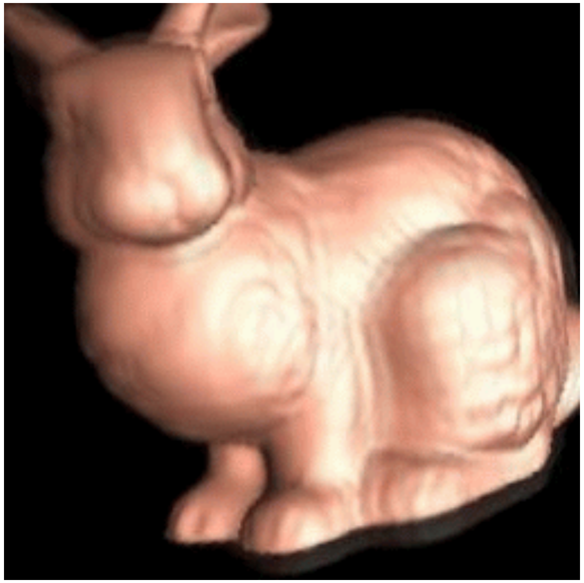


Lightfield Rendering

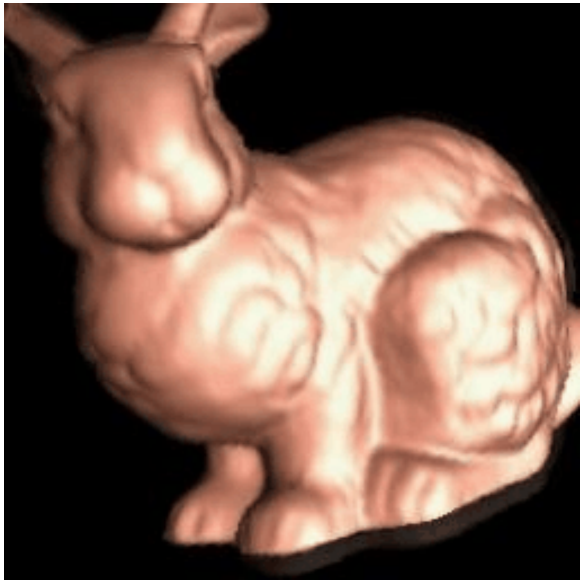
- Use rough depth information to improve rendering quality



Lightfield Rendering

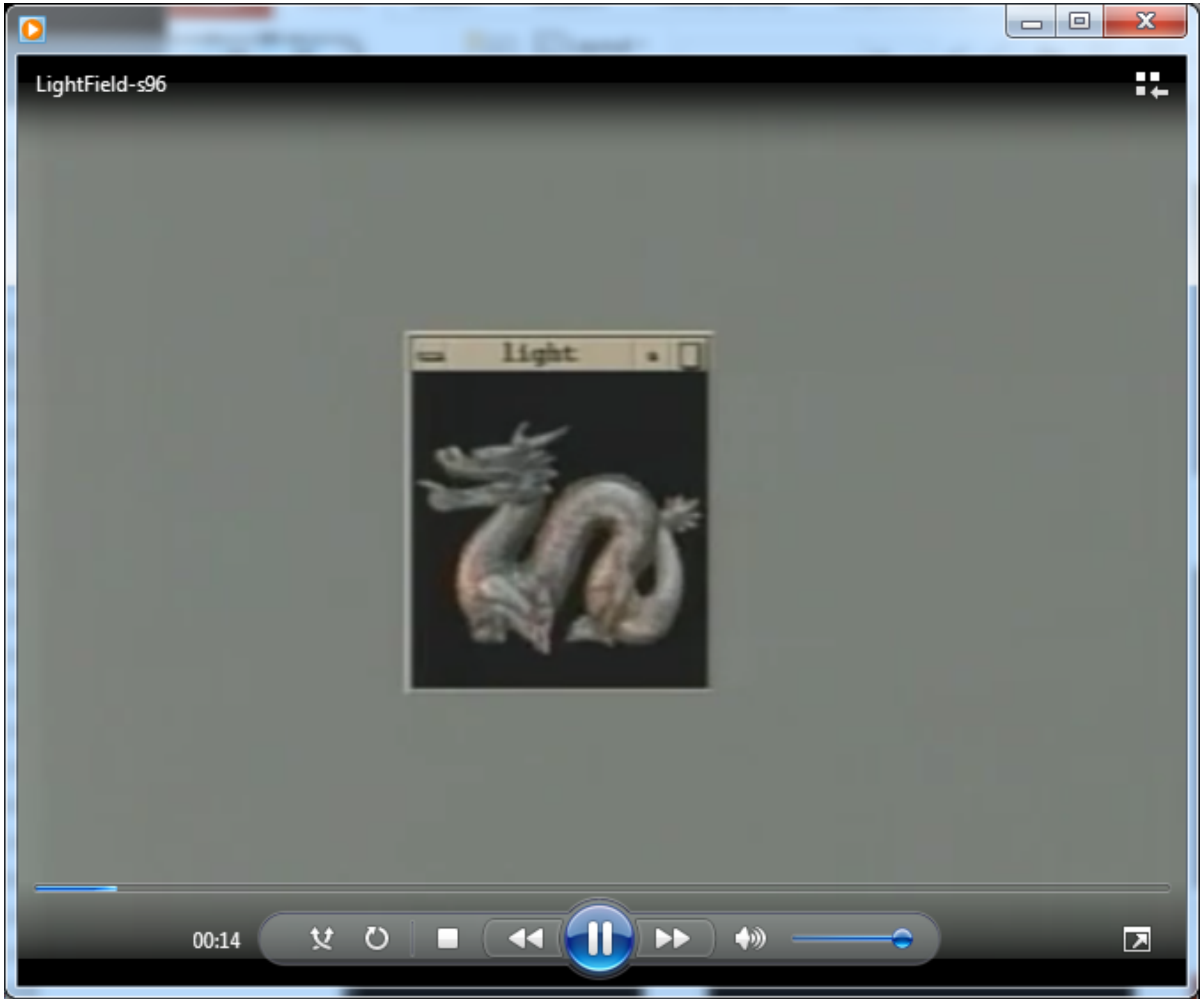


Without using geometry



Using approximate geometry

Lightfield Video

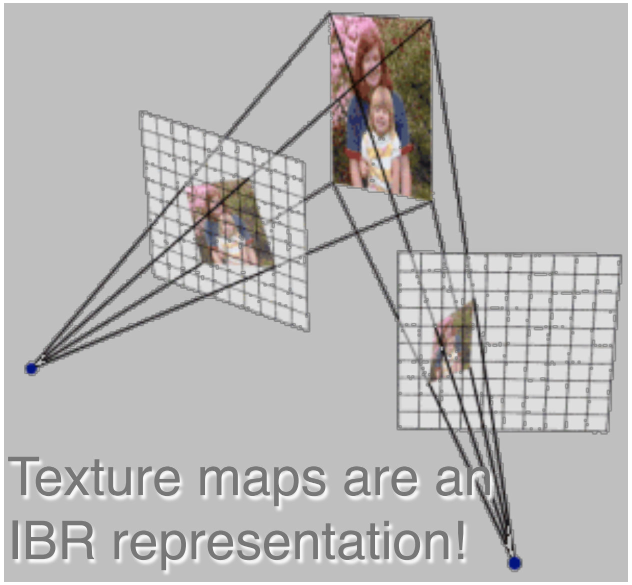


Lightfields

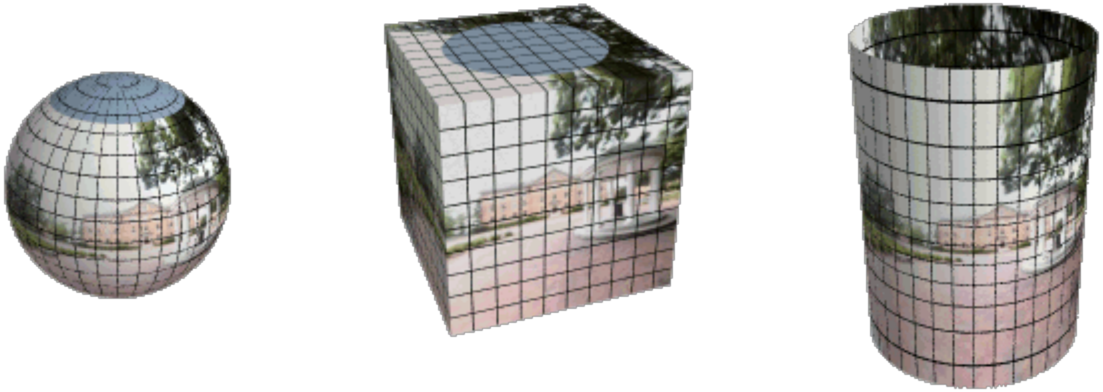
- Advantages:
  - Simpler computation vs. traditional CG
  - Cost independent of scene complexity
  - Cost independent of material properties and other optical effects
  - Avoid hard vision problems
- Disadvantages:
  - Static geometry
  - Fixed lighting
  - High storage cost

Other IBR Representations

- Texture maps
- VDTMs
- Surface lightfields
- Unstructured lightfields
- Concentric mosaics
- Panorama
- Etc.



Texture maps are an IBR representation!



[McMillan]

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