# Computer Graphics I

## - Image based Rendering -

# Motivation

## Photography

- Easy acquisition
- Fast display
- Natural impression

## Computer Graphics

- Time-consuming scene modeling
- Computation-intensive rendering
- Artifical appearance

# Motivation II

- All we usually care about in rendering is generating images from new viewpoints

# Motivation II

- All we usually care about in rendering is generating images from new viewpoints
- In geometry-based methods, we *compute* these new images
  - Projection
  - Lighting
  - Z-buffering

# Motivation II

- All we usually care about in rendering is generating images from new viewpoints
- In geometry-based methods, we *compute* these new images
  - Projection
  - Lighting
  - Z-buffering
- Why not just *look-up* this information?

# Overview

- Theoretical Basis
- “Pure” IBR Algorithms
- Geometry-assisted IBR Techniques

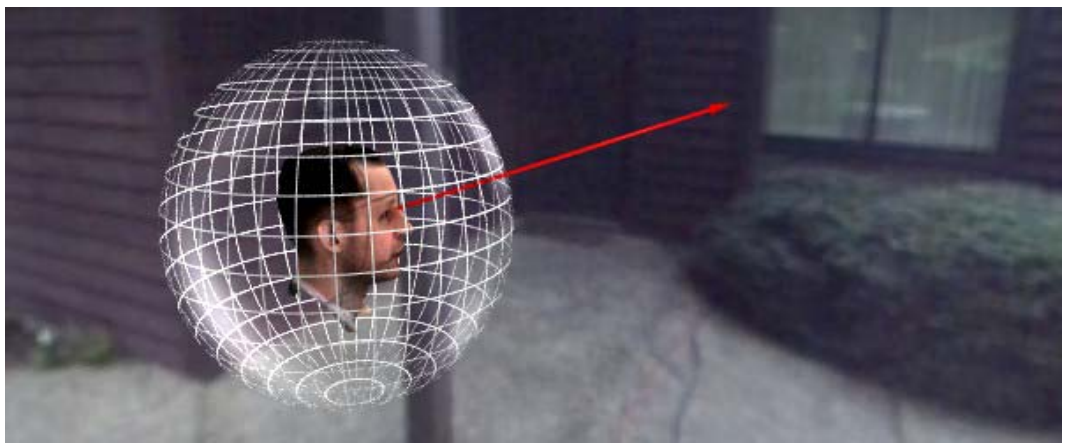
# Overview

- **Plenoptic function**
- **Panoramas**
- **Concentric Mosaics**
- **Light Field Rendering**
- **The Lumigraph**
- **Layered Depth Images**
- **View-dependent Texture Mapping**
- **Surface Light Fields**
- **View Morphing**

# The Plenoptic Function

- Observable light properties (wavelength, 1D) at every point in space (+3D) in all directions (+2D) at every time (+1D): 7D function

$$p = P(\lambda, V_x, V_y, V_z, \theta, \phi, t)$$



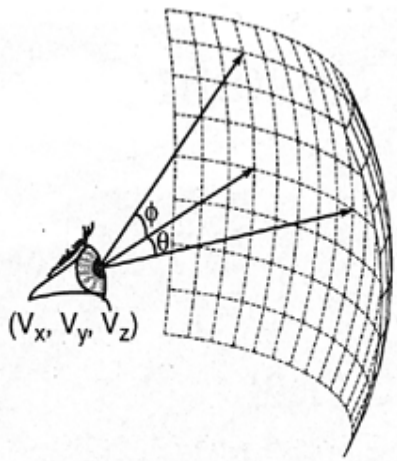



FIGURE 1. The plenoptic function describes all of the image information visible from a particular viewing position.

# The Plenoptic Function II

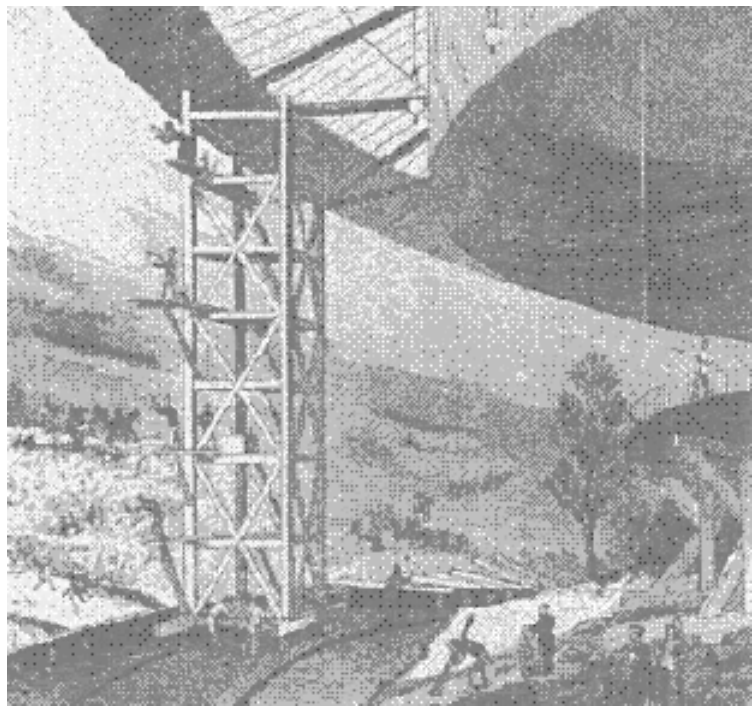
- Acquisition
  - Continuous function $\Rightarrow$ appropriate discretization
  - High-dimensional $\Rightarrow$ reduction in storage requirements
- Rendering
  - Continuous function $\Rightarrow$ look up function value
  - Discretized data $\Rightarrow$ re-sample and interpolate

# Plenoptic Rendering Taxonomy

- Reduced Plenoptic Function
  - 5D: time and wavelength omitted
    $\Rightarrow$ static scene, RGB values
- Light Field Rendering
  - 4D: transparent space, viewpoint outside bounding box
- Concentric Mosaics
  - 3D: viewpoint constrained to lie within a circle
- Panoramas
  - 2D: fixed viewpoint

# Panoramas - History

- Robert Barker‘s Panorama (1792)
  - Up to 17 meters high, 130 meters circumference
- Raoul Brimoin-Sansons Cineorama (1897)
  - 10 synchronized movie projectors, 100 meter circumference
- Disneys CircleVision
  - 9 35mm cameras
- Modern Cinemas
  - IMAX
  - OMNIMAX



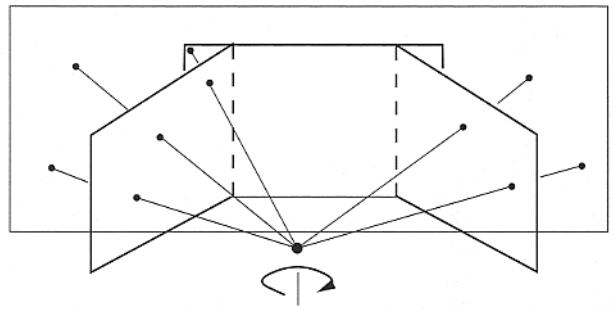
# Panoramas

Fixed viewpoint, arbitrary viewing direction

- Acquisition
  - multiple conventional images
  - Special panorama cameras
- Mosaicing
  - Image registration
  - Stitching
  - Warping
- Rendering
  - Resampling in real-time

# Panoramic Mosaicing

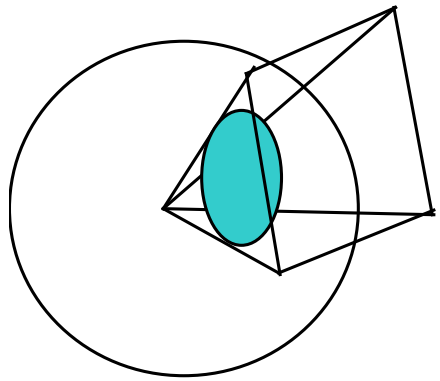
- Projection onto one common plane



$\Rightarrow$ Bow-tie shape

# Panorama Parameterization

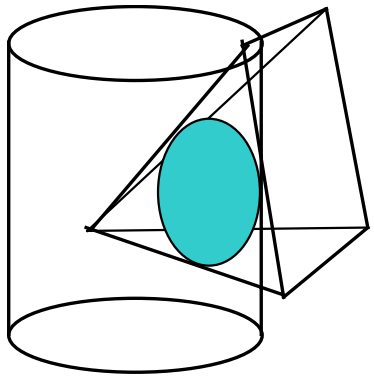
- Spherical projecting surface



  - Advantage
    - Area-constant representation
  - Disadvantage
    - Irregular resampling area

# Panorama Parameterization II

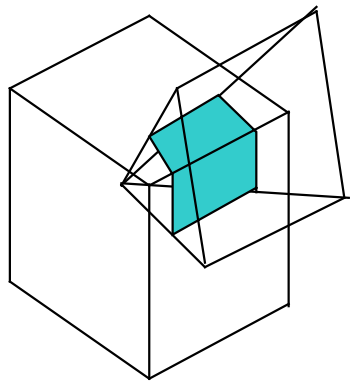
- Cylindrical projecting surface



  - Advantages
    - Simple querying
    - one data structure for all directions
  - Disadvantage
    - Vertical field of view is limited

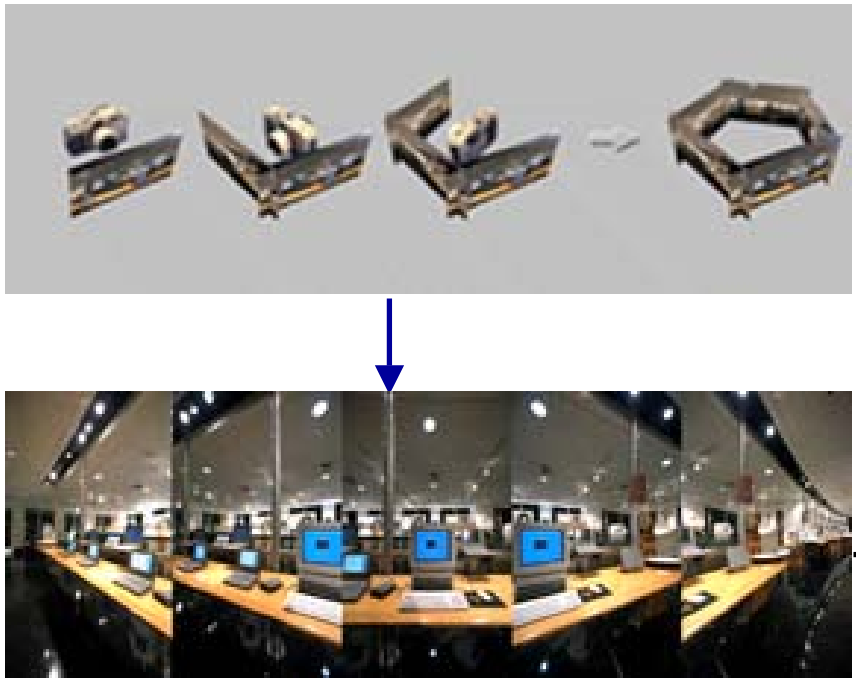
# Panorama Parameterization

- Cubic projecting surface



  - Advantages
    - Simple data representation
    - All viewing directions
  - Disadvantages
    - 6 separate data slabs
    - Distortion towards edges

# Cylindrical Panoramas



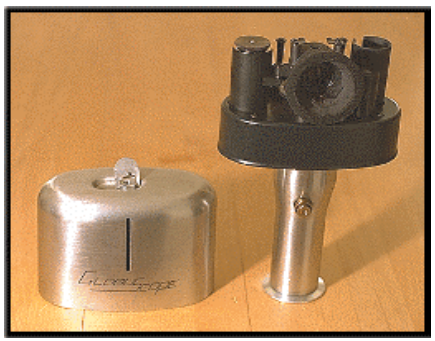
# Panorama Mosaicing

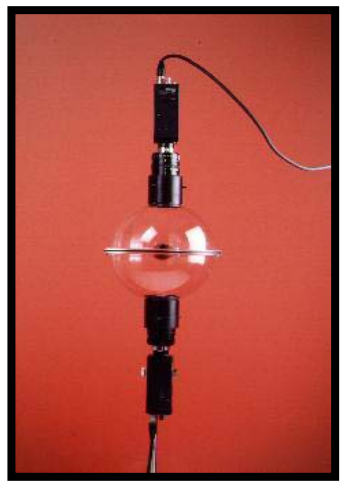
© James Davis

- Prewarping
  - Lens correction, radiometric correction, cylindrical projection
- Image Registration
  - Feature alignment
  - Minimizing  pixel differences
- Compositing
  - Eliminate moving objects
- Resampling
  - Filling holes
  - Blending
  - Filtering

# Panorama Cameras

- Rotating Cameras
  - Kodak Cirkut
  - Globuscope
- Stationary Cameras
  - Be Here
  - OmniCam
  - ...

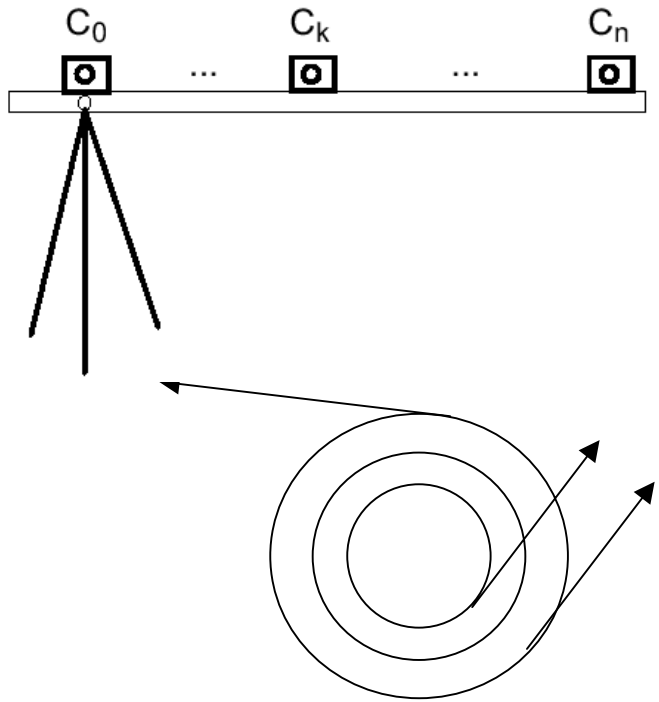




# Concentric Mosaics

H.-Y. Shum and L.-W. He, “Rendering with Concentric Mosaics”, Siggraph’99

www-scf.usc.edu/~csci576/lectures/ concentric-mosaics.pdf

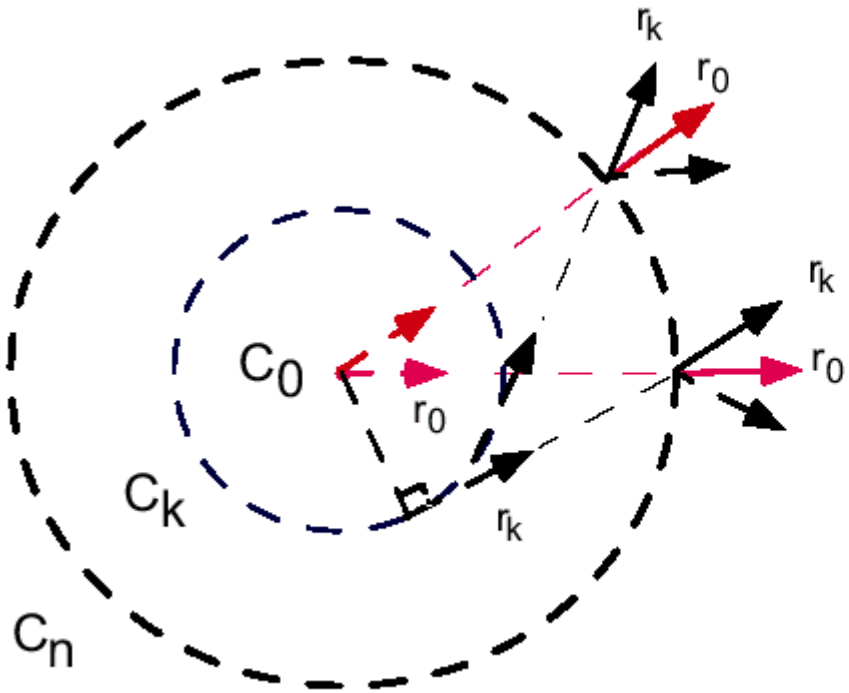


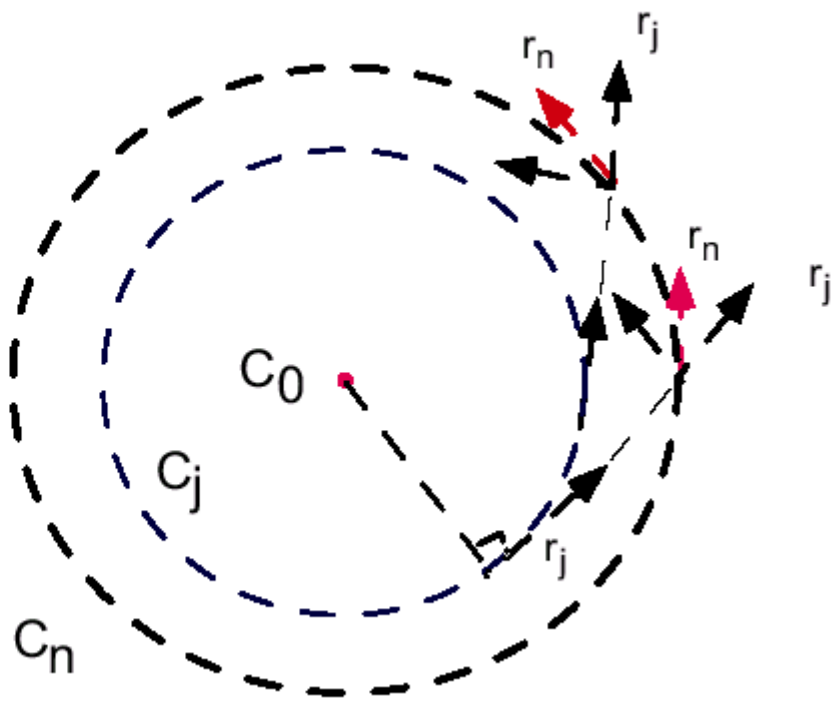

Viewpoint on confined plane

- Acquisition
  - Off-axis rotating camera
  - Different radii
- Optical axis alignment
  - Tangential or radial
- Stored as tangent slit images (vertical lines)

$\Rightarrow$ Conveys horizontal parallax

# Concentric Mosaics – Optical Axis Orientation



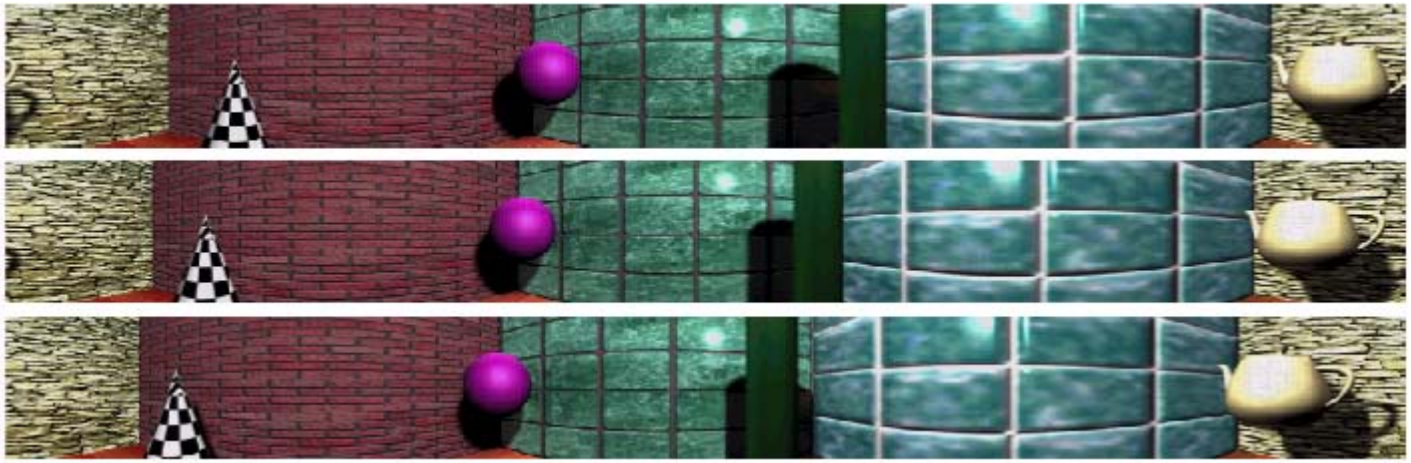




Radial axis alignment

Tangential axis alignment

- Parameterize each vertical line as tangent to circle

# Concentric Mosaics – Data Representation

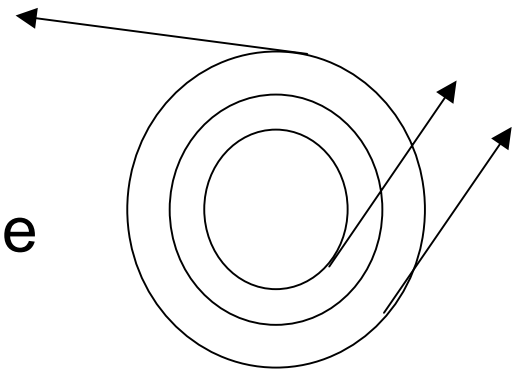


- Along one circle

  Tangential parameterization

  – Multiple centers-of-projection image

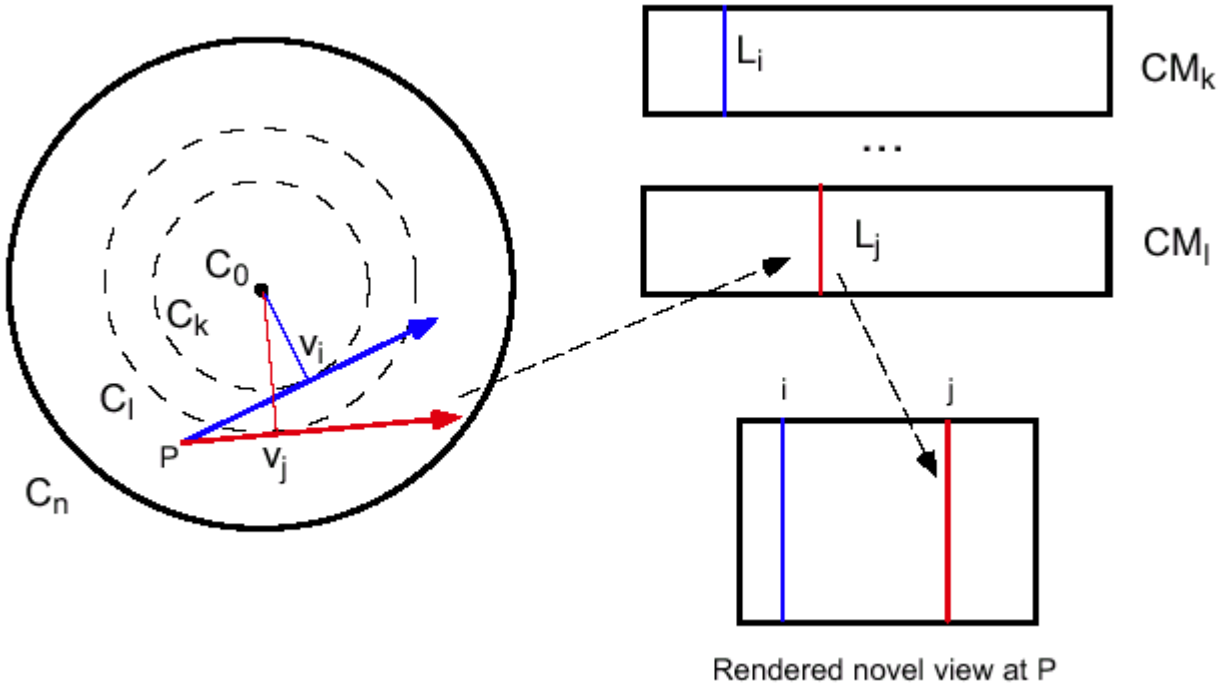
  – Pushbroom camera



- Between circles of different radii

  – Horizontal parallax for different scene depths

# Concentric Mosaics - Rendering




**For each vertical line:**

- Find tangent line to circle
- Select nearest circle
- Select closest recording position

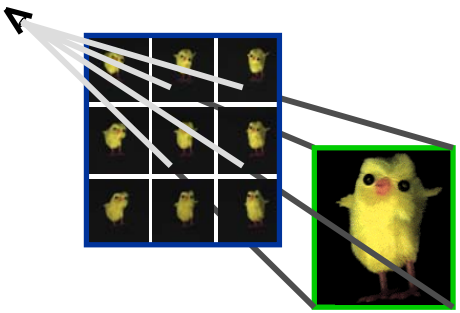
# Concentric Mosaic – Example

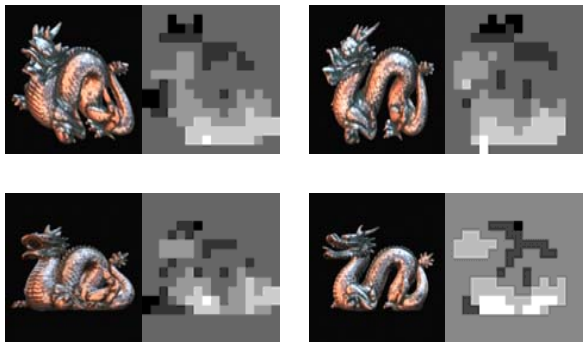
- Horizontal parallax
- Reflection effects
- Dense sampling to avoid aliasing

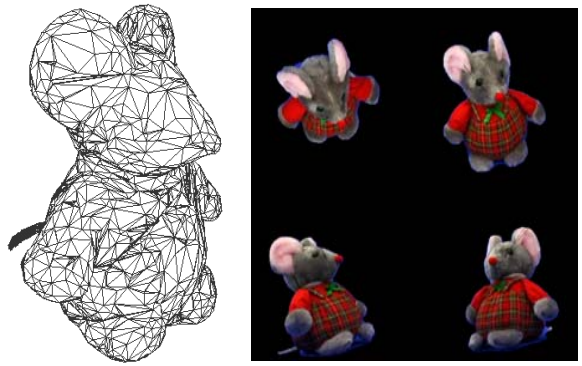
# IBR Taxonomy

image-based

geometry-based







## Light Field Rendering

- images only
+ no scene restrictions
– lots of images

## Lumigraph, Layered Depth Images

- images & per-pixel depth
+ improved rendering quality
– increased rendering complexity

## View-dep. Text. Map., Surface Light Fields

- images & 3-D model
+ fast rendering (hardware)
– geometry acquisition/ reconstruction

# Light Field Rendering

Levoy and Hanrahan, “Light Field Rendering”, Siggraph’96

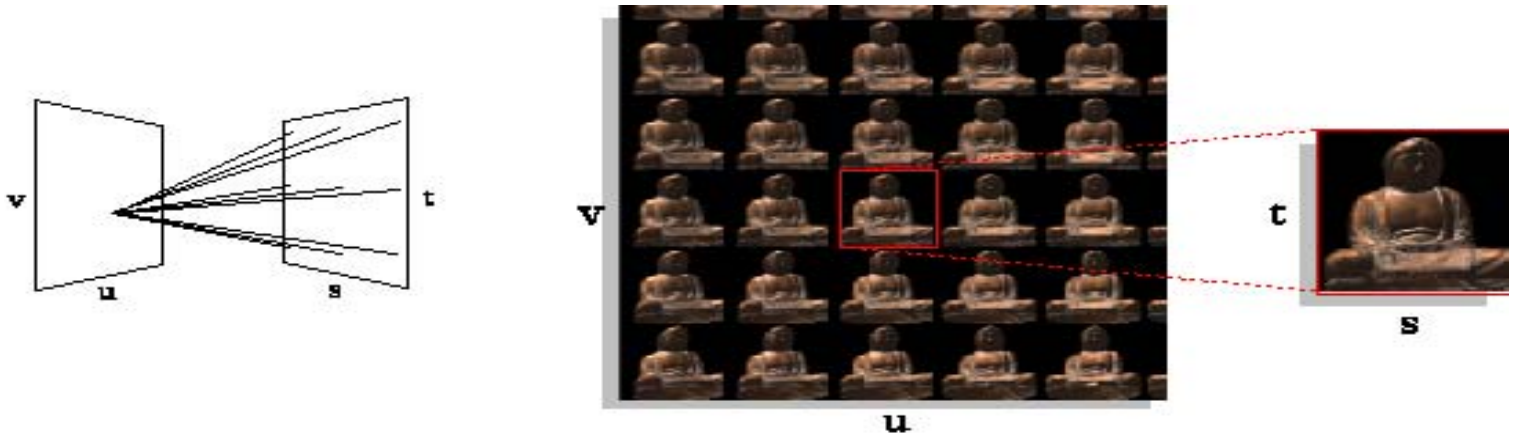
graphics.stanford.edu/projects/lightfield/

- Viewpoint outside bounding visual hull
- Assumption: light properties don’t change along ray
- 2D matrix of 2D images: 4D structure

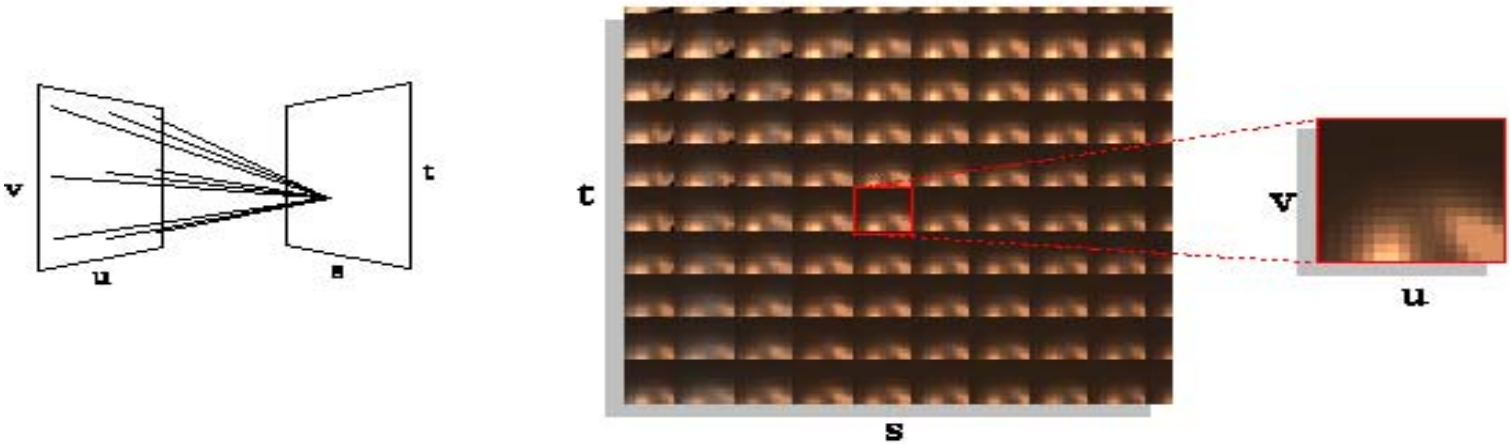
$\Rightarrow$ Conveys full parallax

$\Rightarrow$ captures complex BRDFs

# Two-Plane Parameterization

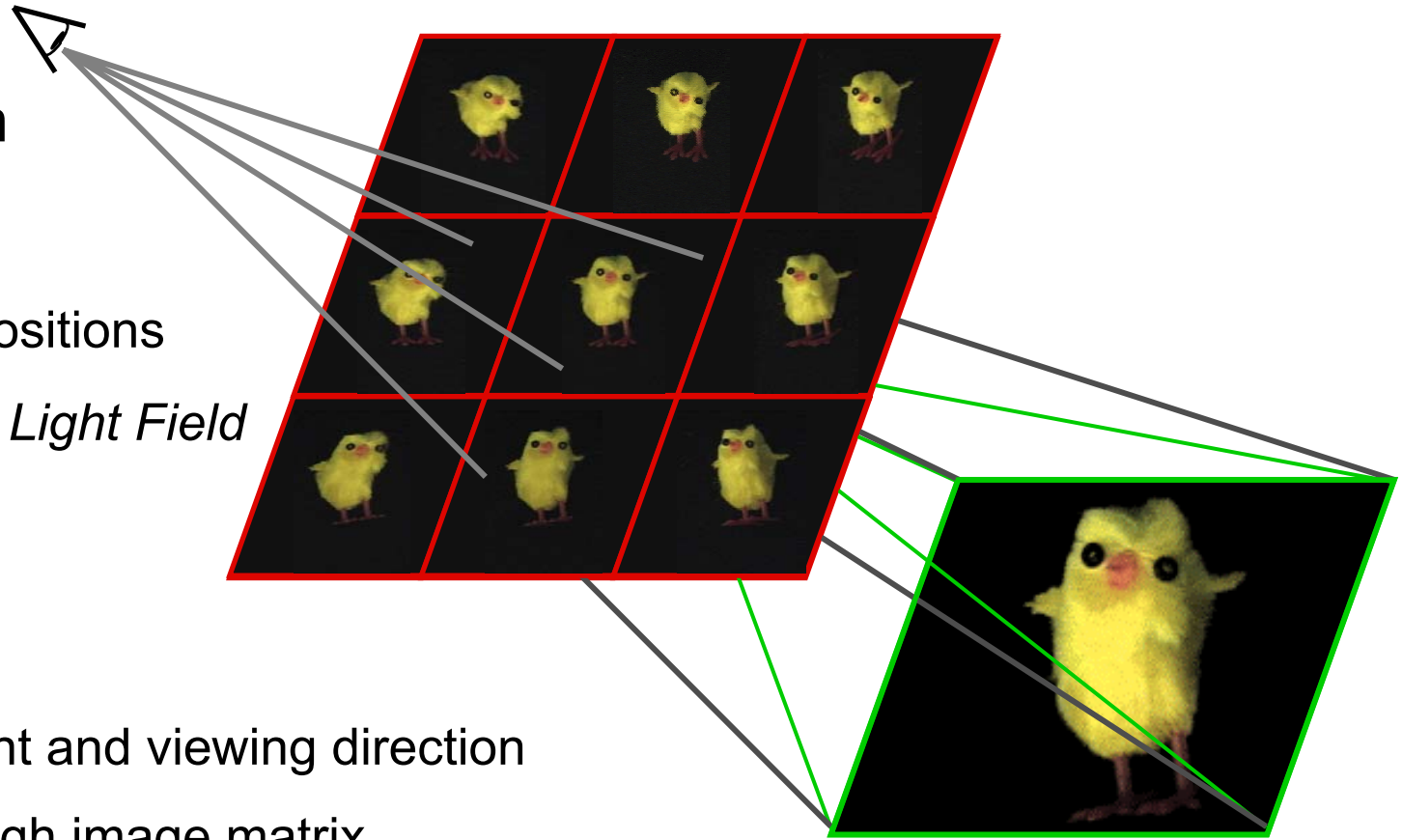



uv array of st images




st array of uv images

# Light Field Rendering



## Acquisition

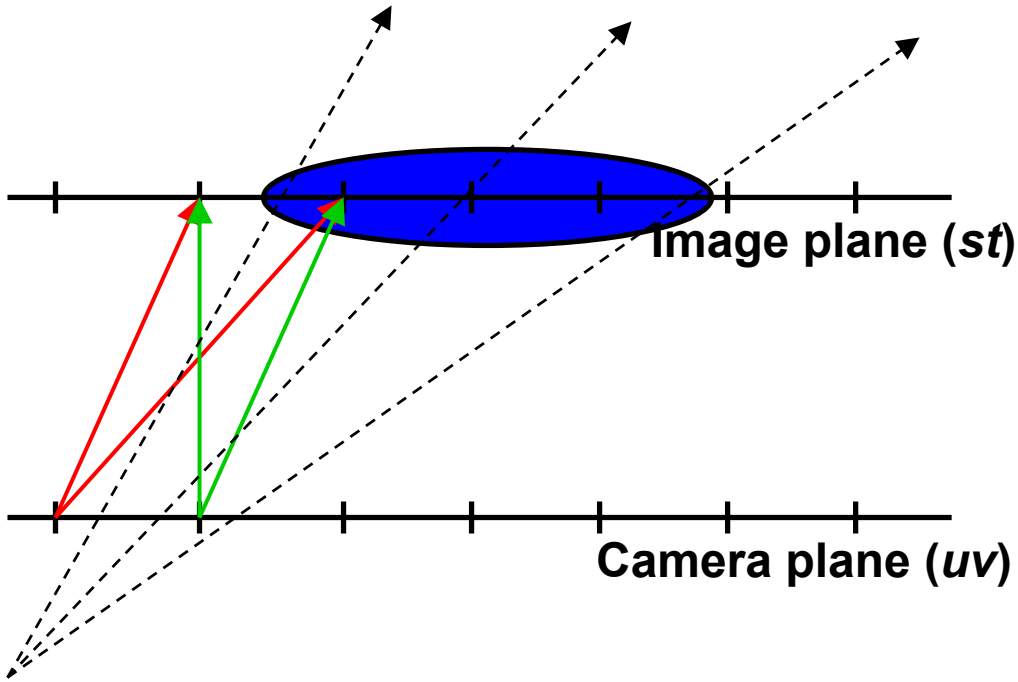
- Static 3D scene
- Known camera positions
- 2D image matrix: *Light Field*

## Rendering

- Arbitrary viewpoint and viewing direction
- Ray-tracing through image matrix
- Pixel color determined by intersection with camera plane/image plane

# Light Fields – Quadralinear Interpolation

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



# Light Field Rendering - Example

2-plane parameterization | closest image | quadralinear interpolation

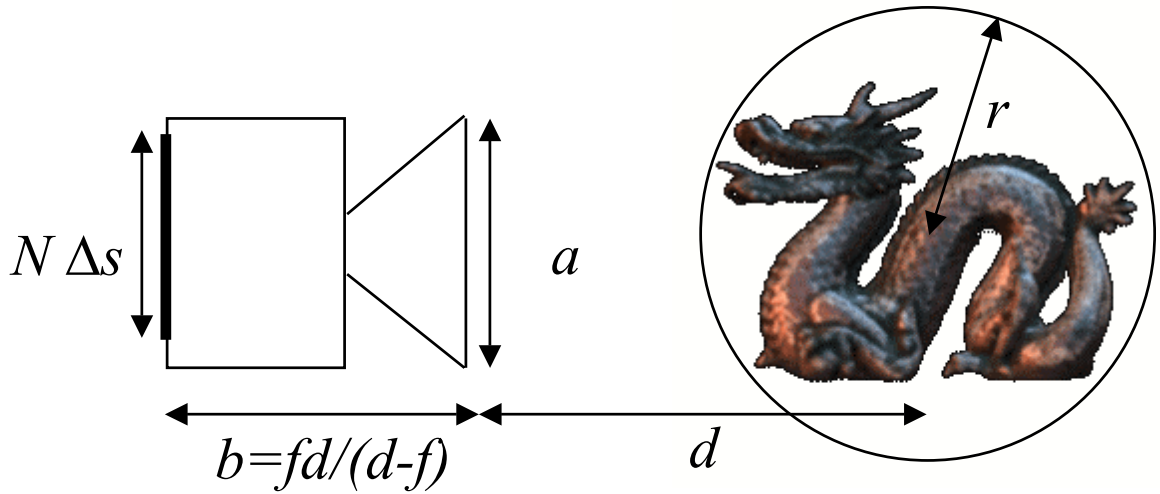
Aliasing-free rendering: number of images $\propto$ image resolution

**Photorealistic rendering results: lots of images necessary !**

# Light Field Sampling

known

- Focal length $f$
- Object radius $r$
- Pixel size $\Delta s$
- Pixel number $N$




- Optimal recording distance $d$
  - Object fills camera format

$$d = f\left(1 + \frac{2r}{N\Delta s}\right)$$

- Maximum lens diameter $a_{max}$
  - Depth-of-field < pixel size $\Delta$s

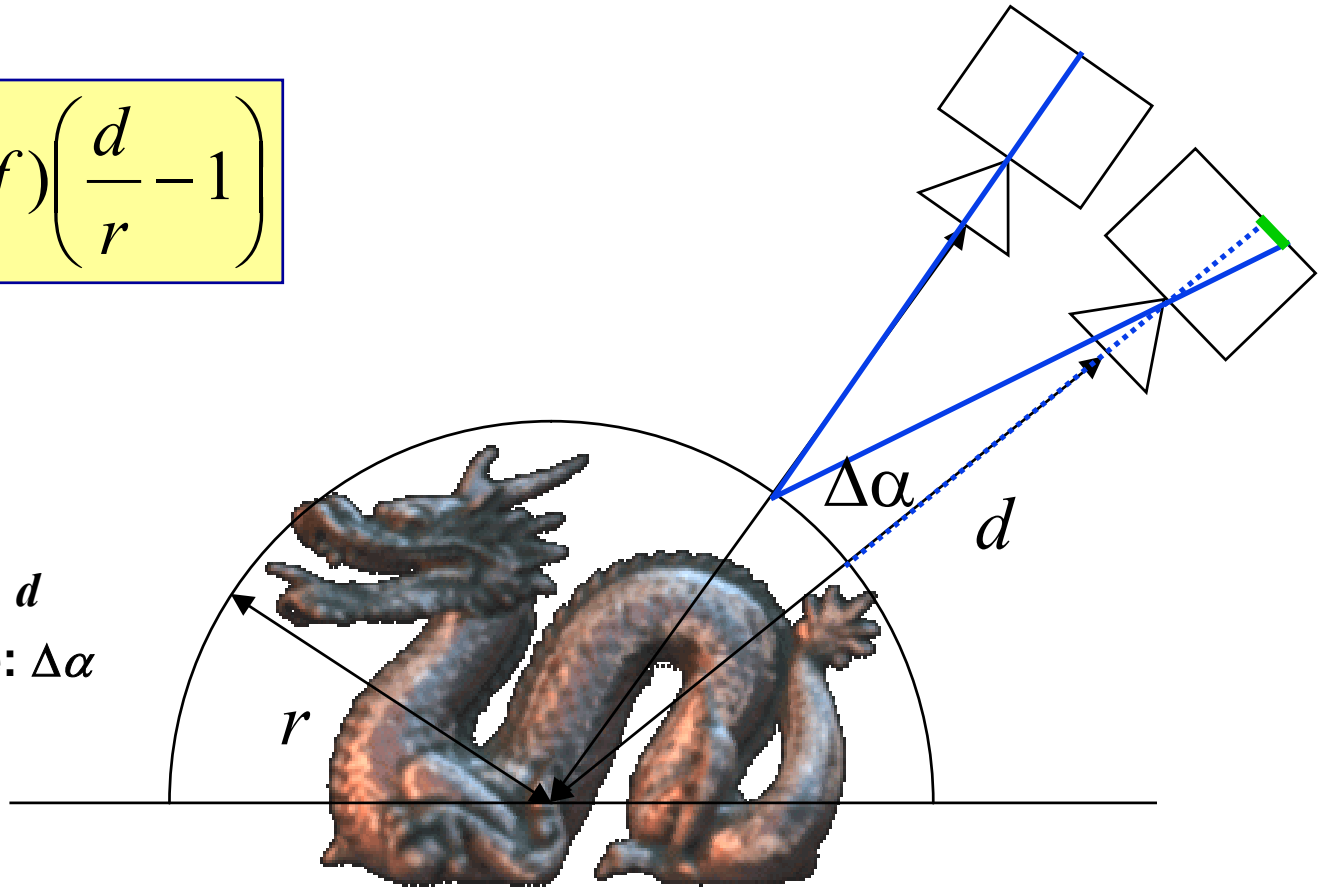
$$a < \frac{\Delta s}{f}(d - f)\left(\frac{d}{r} - 1\right)$$

# Light Field Sampling II

- **Maximum camera movement**
  - Disparity < 1 pixel

$$\Delta\alpha = \frac{\Delta s}{df}(d - f)\left(\frac{d}{r} - 1\right)$$

- **pixel size:** $\Delta s$
- **object radius:** $r$
- **focal length:** $f$
- **recording distance:** $d$
- **displacement angle:** $\Delta\alpha$



# Light Field Sampling III

- Maximum lens diameter $$a = \frac{\Delta s}{f}(d-f)\left(\frac{d}{r}-1\right)$$

- **Maximum camera movement** $$\Delta\alpha = \frac{\Delta s}{df}(d-f)\left(\frac{d}{r}-1\right)$$

- **Angular extend of lens** $$\Delta\alpha \approx \frac{a}{d}$$

$\Rightarrow$ **Same sampling criterion**

$\Rightarrow$ **Camera lens serves as low-pass filter**

$\Rightarrow$ **number of images over sphere**

$$Q \approx \frac{4\pi}{\Delta\alpha^2}$$

# Light Field Sampling IV

- $N\,\Delta s$ = 2.56 mm
- $f$ = 12 mm
- $r$ = 100 mm
- 24 bits/pixel

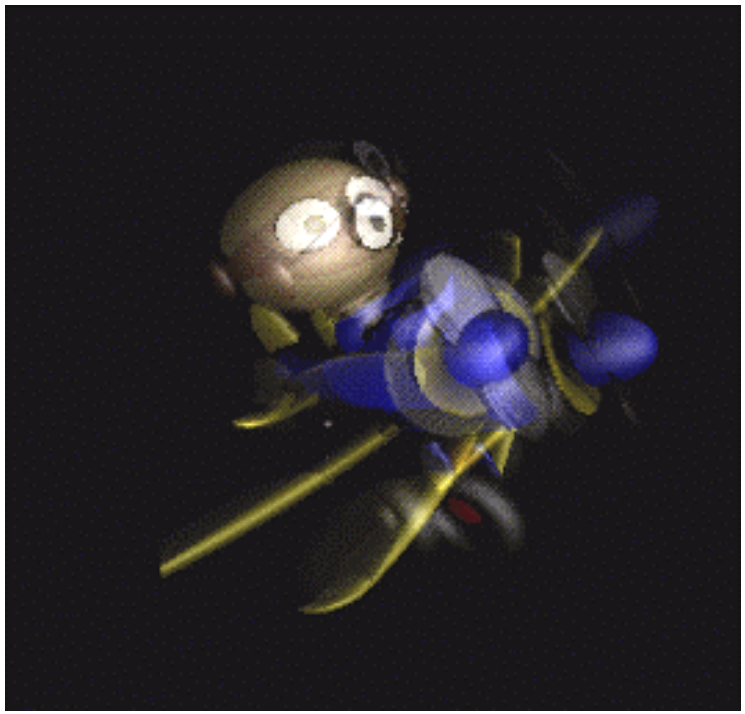
$\Rightarrow$ $d=950$ mm

| pixel number $N^2$ | max lens dia. $d\Delta\alpha$ (mm) | image number $Q$ | total pixel number $N^2Q$ | memory (MB) |
|---|---|---|---|---|
| $256^2$ | 6.64 | 257213 | $1.7*10^{10}$ | 48600 |
| $128^2$ | 13.27 | 64303 | $1.05*10^{9}$ | 3000 |
| $64^2$ | 26.5 | 16076 | $7.68*10^{7}$ | 220 |
| $32^2$ | 53.1 | 4019 | $4.12*10^{6}$ | 11.8 |
| $16^2$ | 106 | 1005 | 257280 | 0.75 |

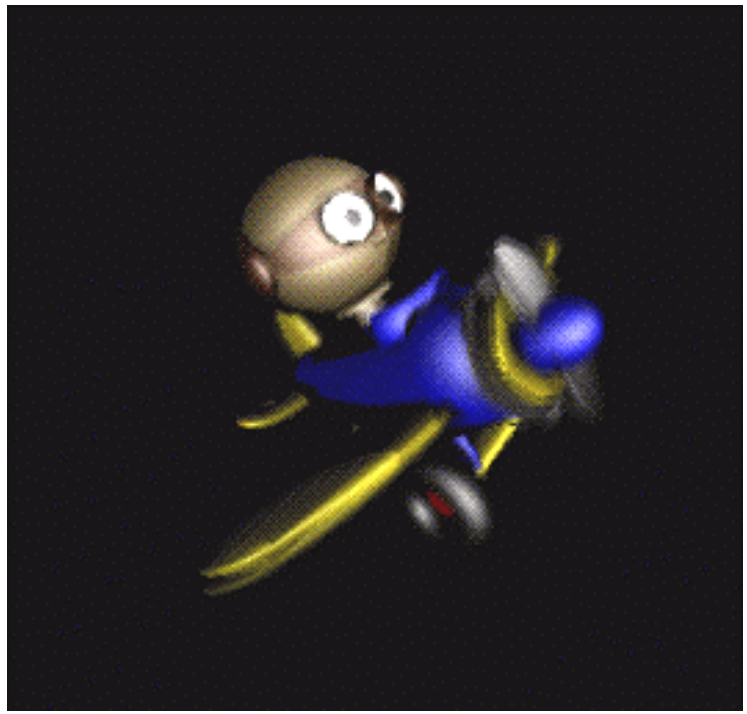
Huge amounts of image data necessary

$\Rightarrow$ IBR from subsampled light-field representations only

# Light Field Rendering – Aliasing Artifacts



rendered from heavily subsampled representation



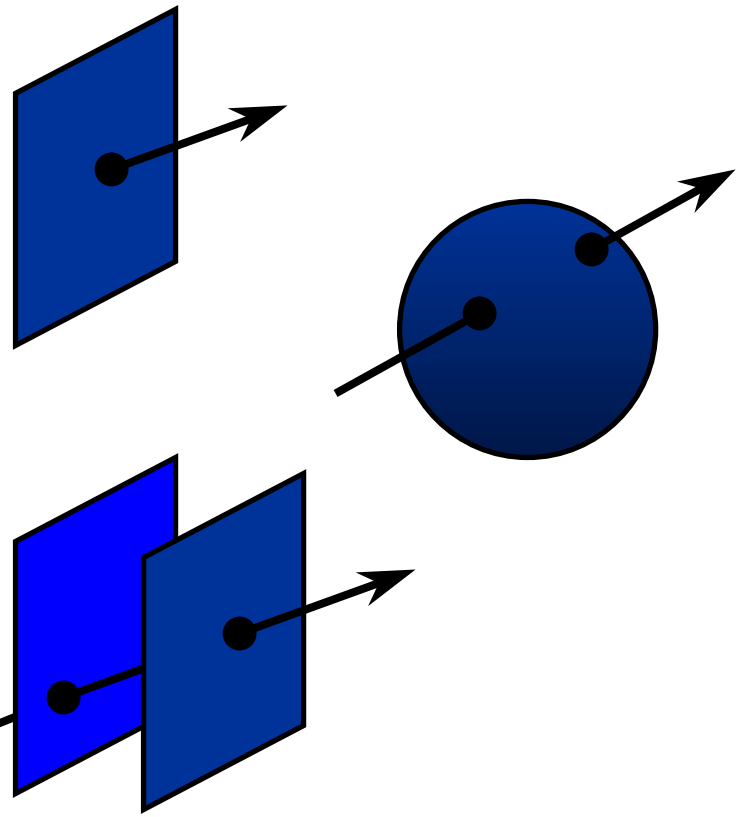
rendered from moderately subsampled representation

Aliasing / blurring artifacts

⇒ Apply scene geometry to estimate missing light-field information

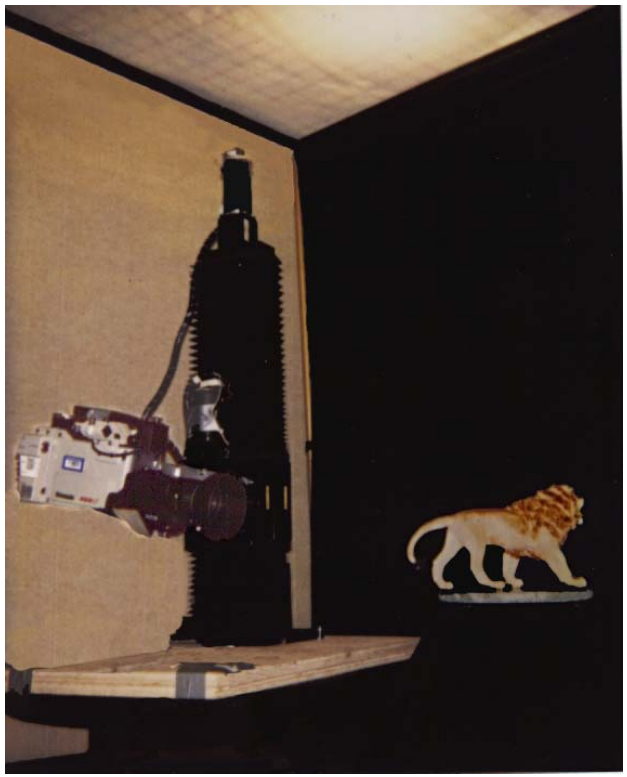
# Light Field Parameterization

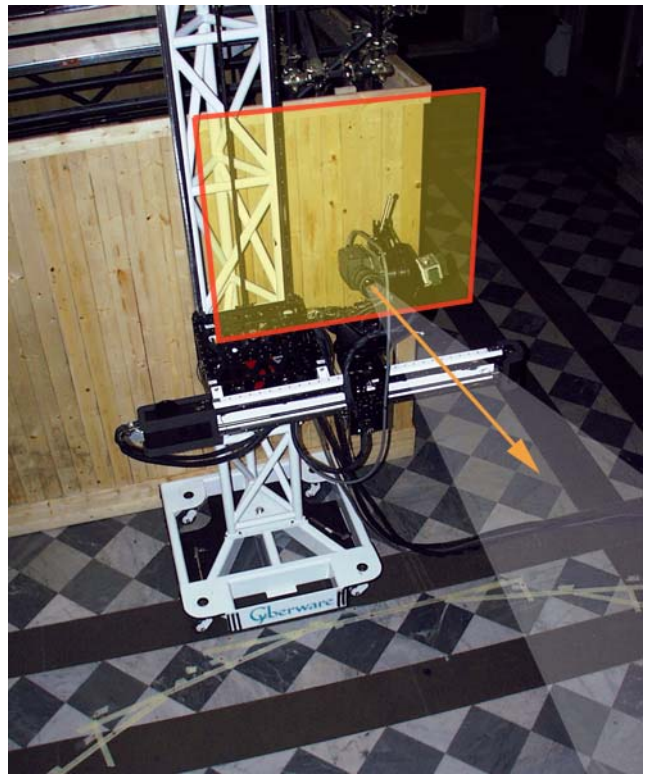
- Point / angle
- Two points on a sphere
- Points on two planes
- Original images and camera positions



# Light Field Acquisition

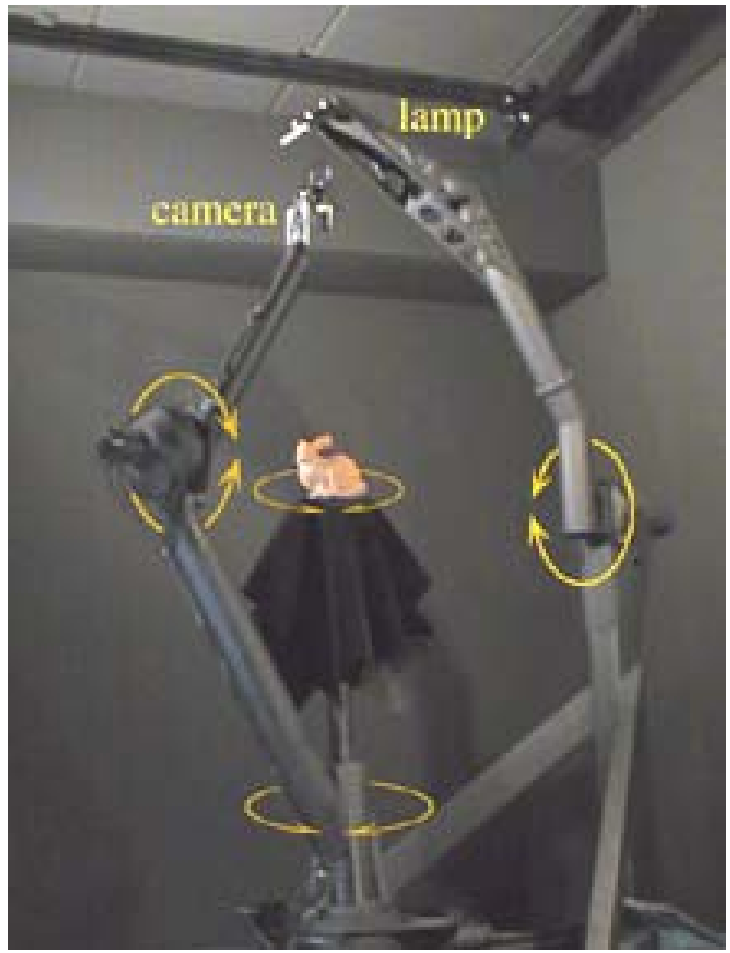
- Calibrated light field capture
  - Computer-controlled camera rig
  - Moves camera to grid of locations on a plane





# Light Field Acquisition II

- Spherical motion of camera around object
- Samples space of directions uniformly
- Single Camera
  - Static scene – Sequential recording
  - Calibrated motion – Mechanical gantry
  - Photometric calibration easy



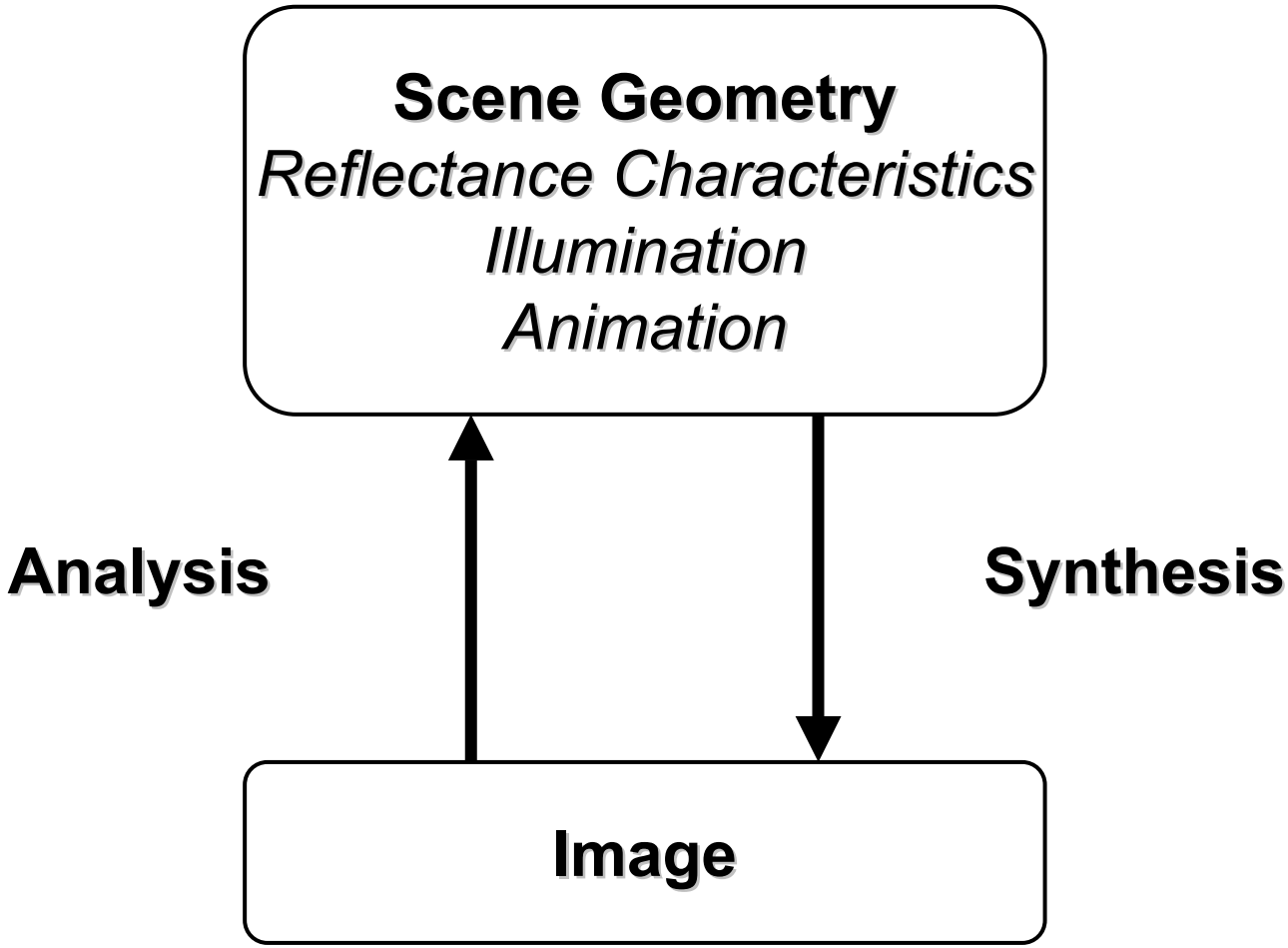
# Light Fields - Summary

- Advantages
  - Simpler computation vs. traditional CG
  - Cost independent of scene complexity
  - Cost independent of material properties and other optical effects
- Disadvantages
  - Static geometry
  - Fixed lighting
  - High storage cost / aliasing

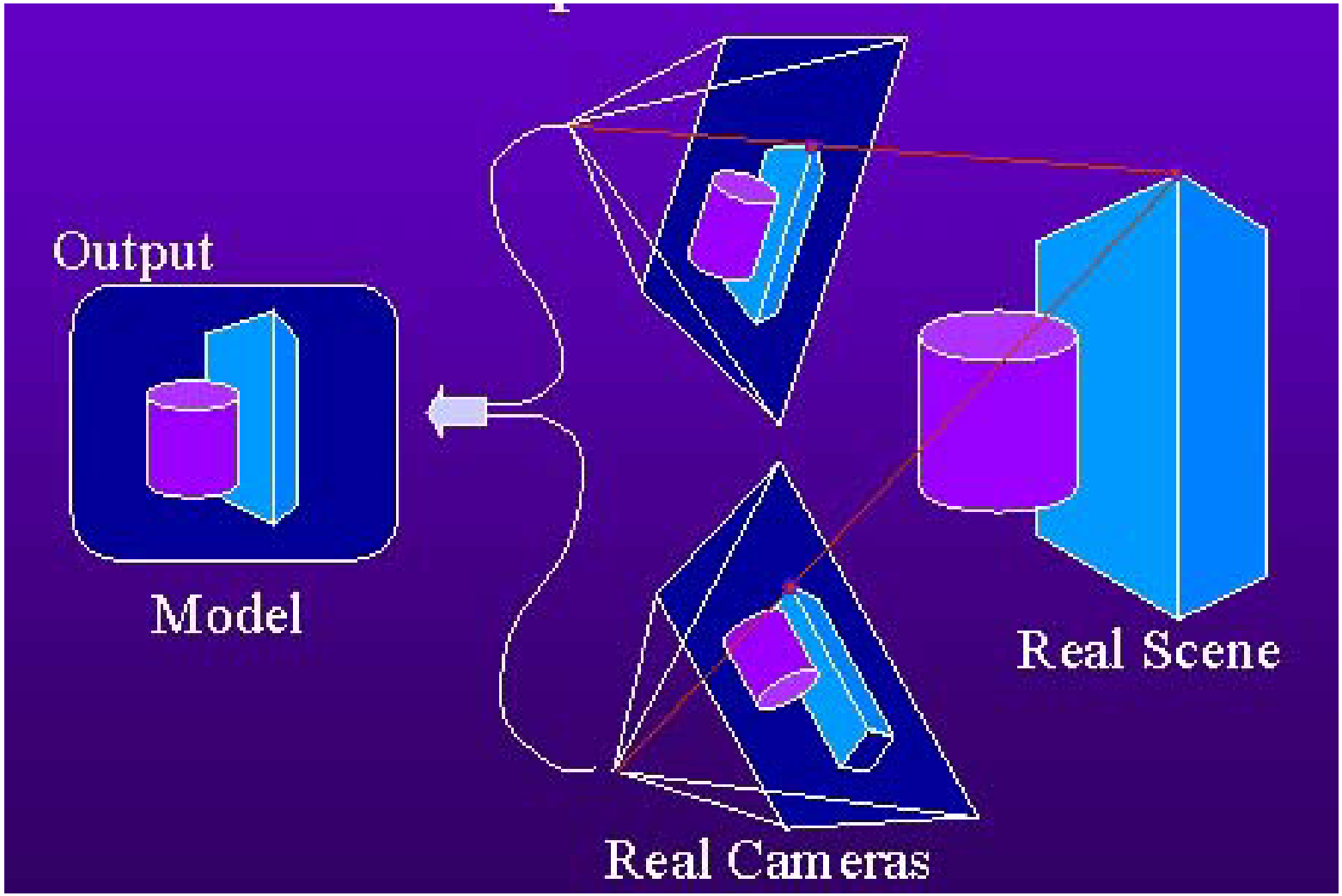
# Geometry-assisted IBR Methods

- Fundamental idea of IBR:
  Generate new views of a scene directly from recorded views
- “Pure” IBR $\Rightarrow$ Light Field Rendering
- Enormous amount of images necessary
- Highly redundant data
- Other IBR techniques try to obtain higher quality with less storage by exploiting scene geometry information

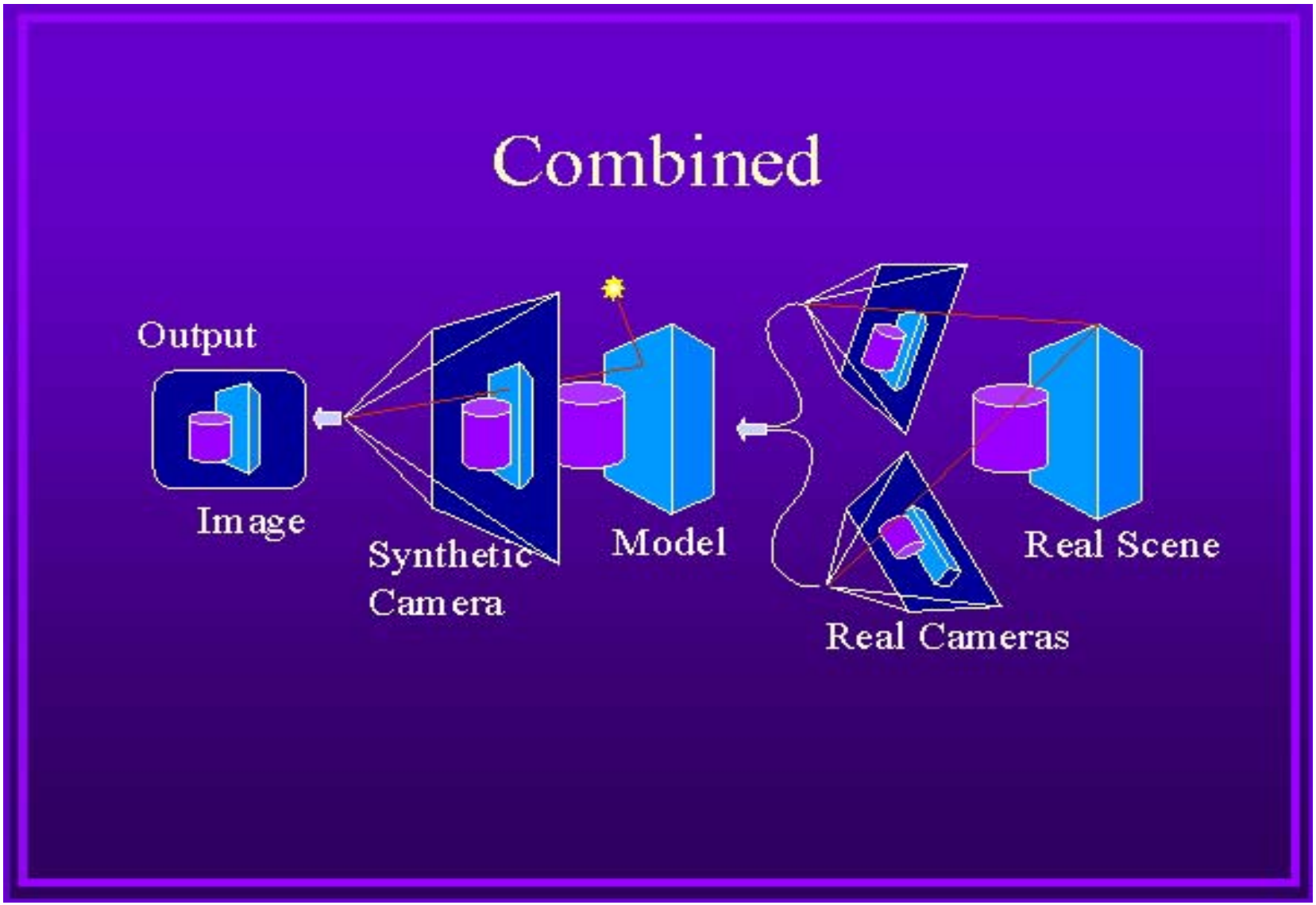
# Computer Graphics – Computer Vision



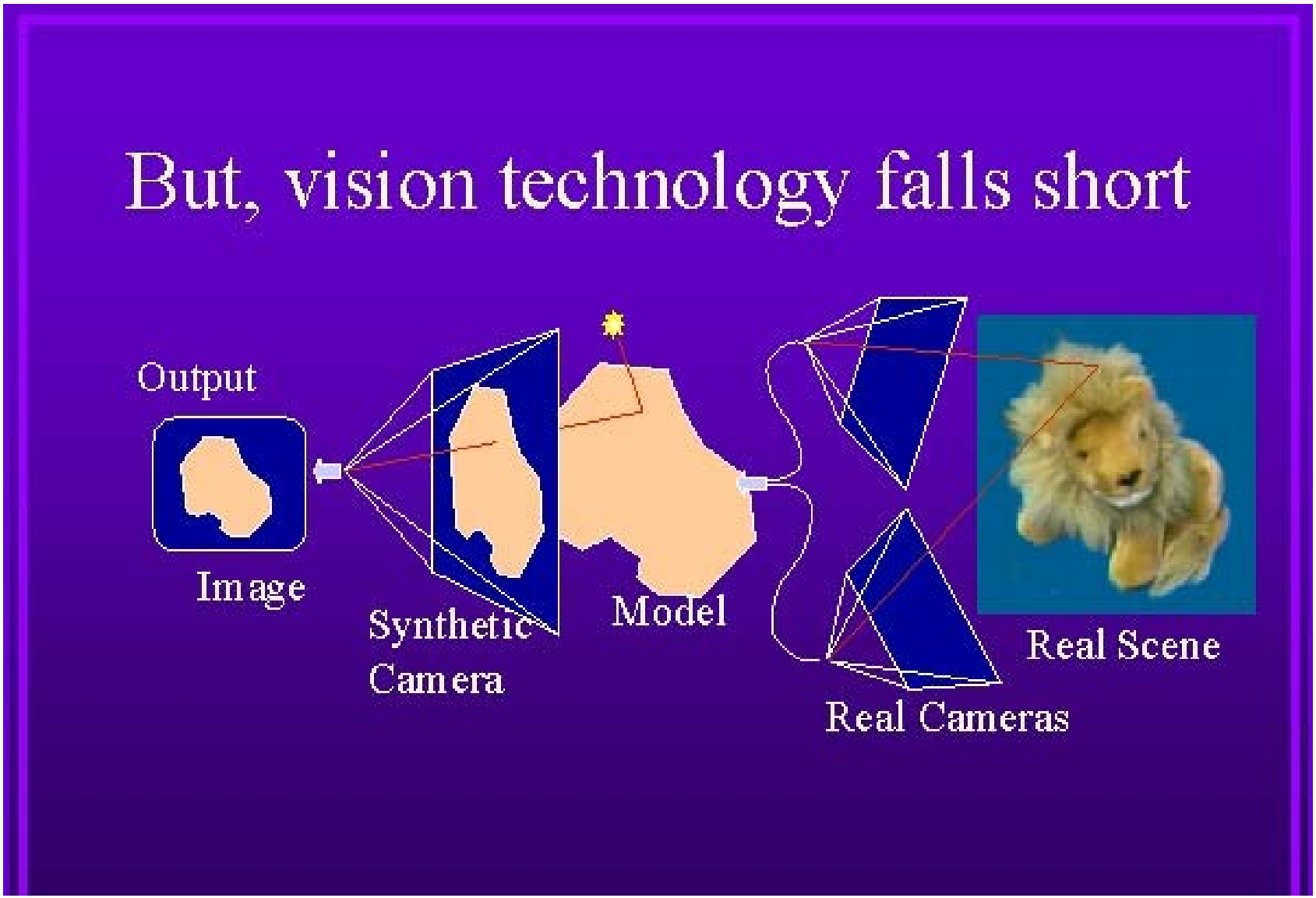
# Geometry Reconstruction



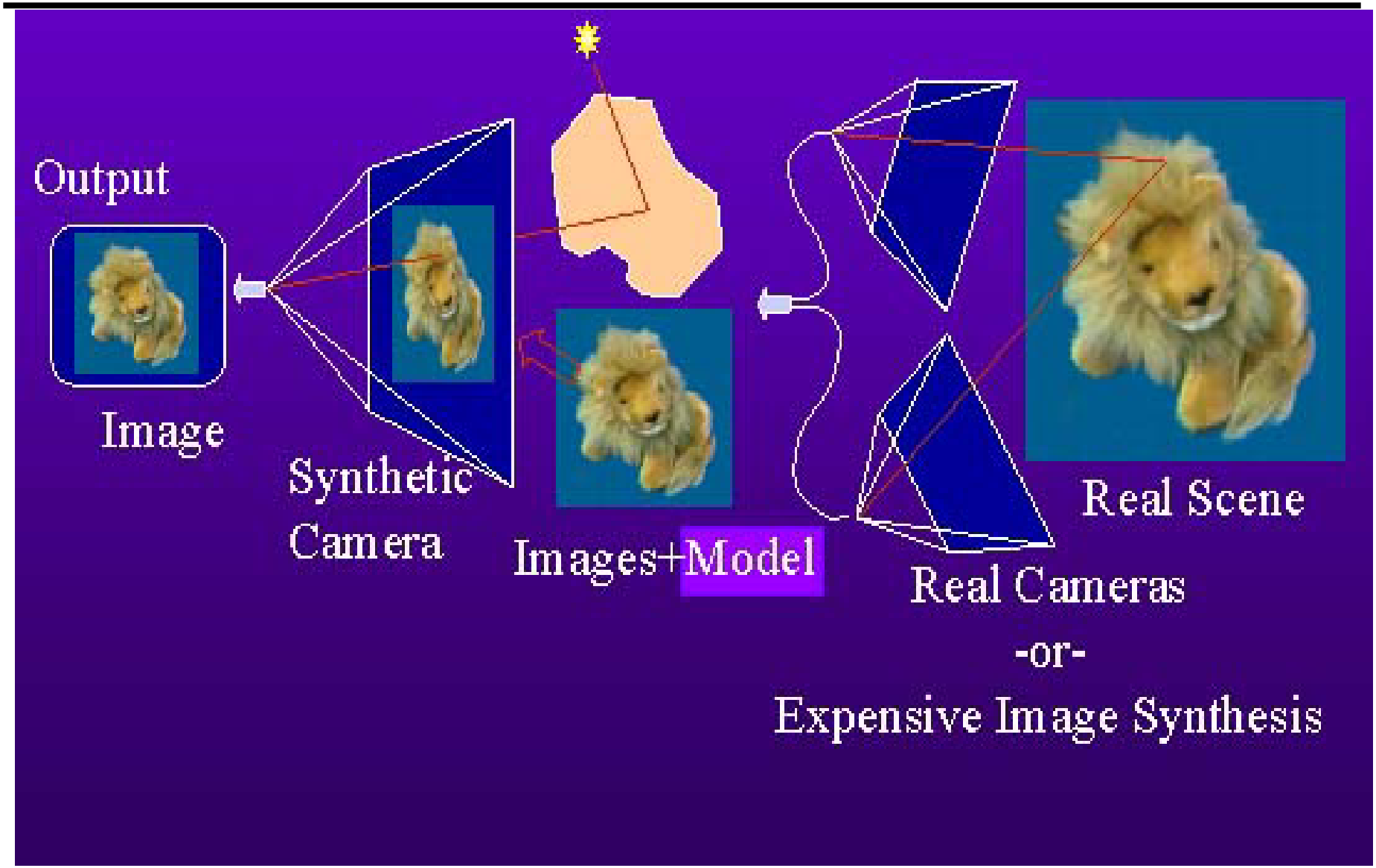
# Vision – Geometry Pipeline



# Approximate Geometry

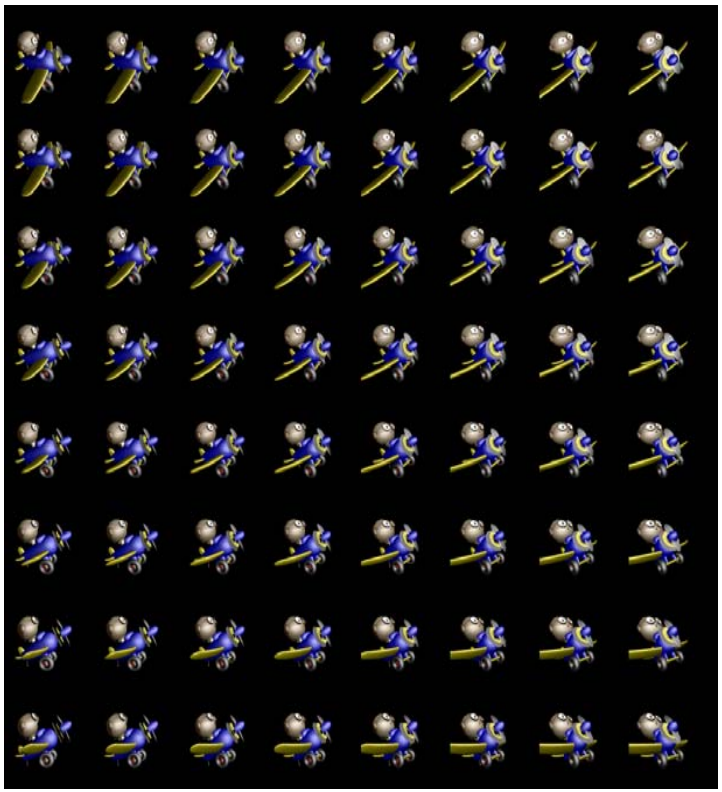


# Geometry-assisted IBR



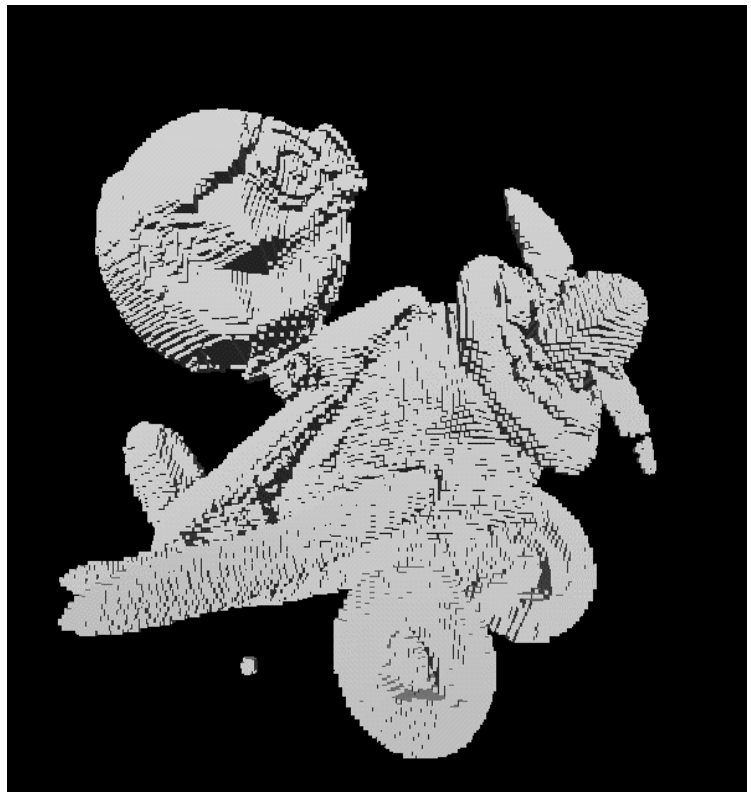
# Geometry-assisted IBR: Example

*Airplane* Light Field



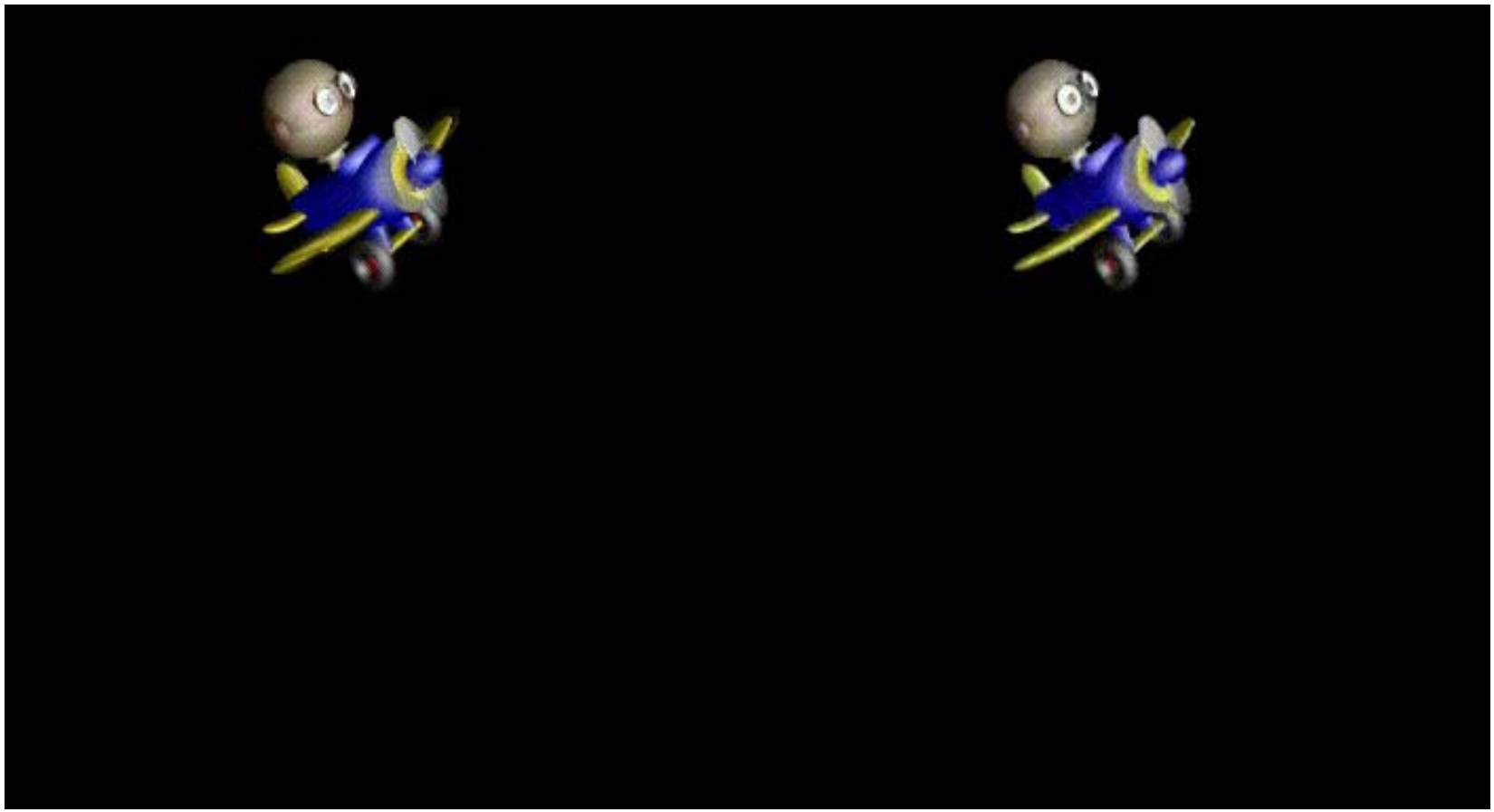
- **8 × 8 images, 256 × 256 pixels**

Reconstructed voxel model



- **250 × 260 × 200 voxels**
- **object surface: 450,000 voxels**

# Geometry-assisted LF Rendering



original *Airplane* light field
8x8 RGB images, 256x256 pixel
12 MBytes

Model-aided Coder
LF augmented to 29x29 images
32.2 dB PSNR, 0.095 bpp (48.6 KBytes)

# The Lumigraph

Gortler et al., “The Lumigraph”, Siggraph’96, pp. 43-52
research.microsoft.com/siggraph96/96/lumigraph.htm

- Input: multiple images
- Resample into 2-plane parameterization

$\Rightarrow$ Equivalent to Light Field Rendering

- Reconstruct approximate per-pixel depth from images

$\Rightarrow$ Disparity-corrected rendering

# Lumigraph – Depth-corrected Rendering

What color has ray (s,u) ?

- Closest recorded ray

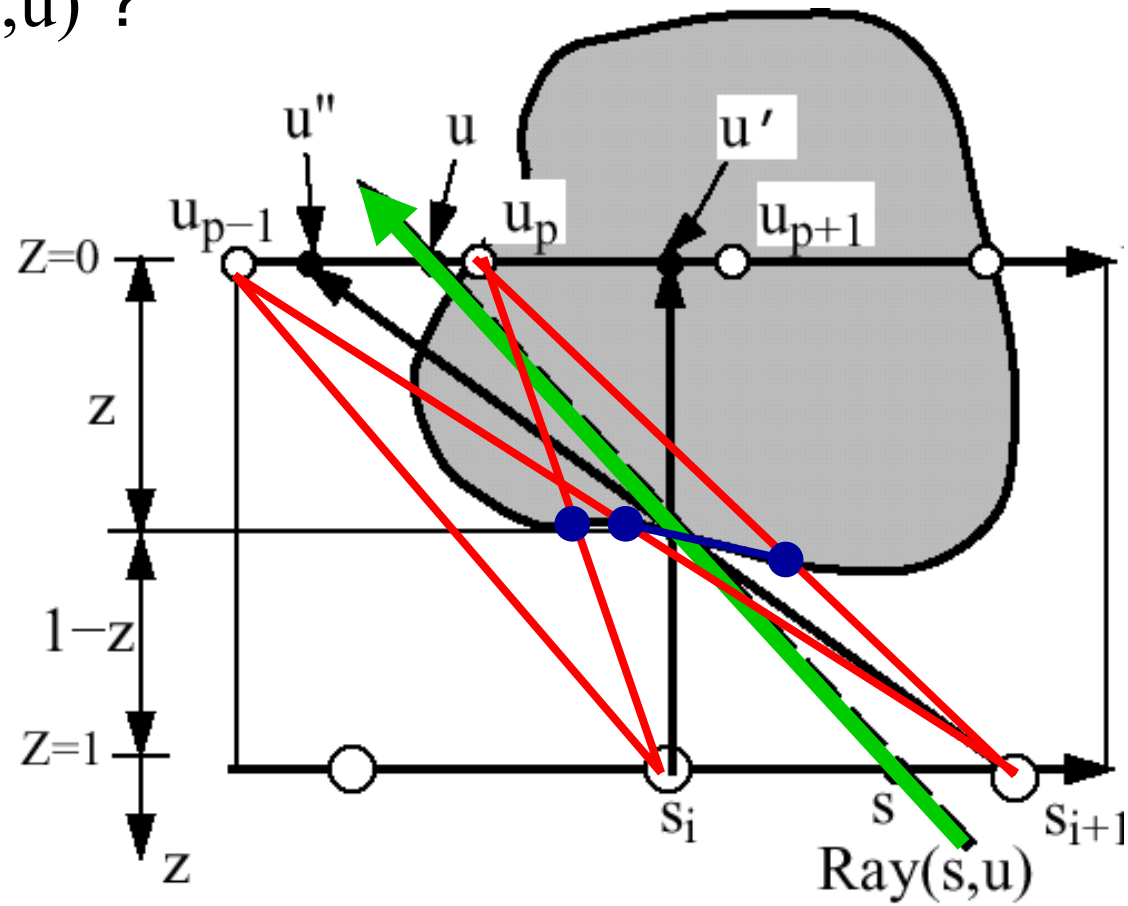
⇒ Wrong surface point

- Neighboring rays

⇒ find surface point closest to ray (s,u)

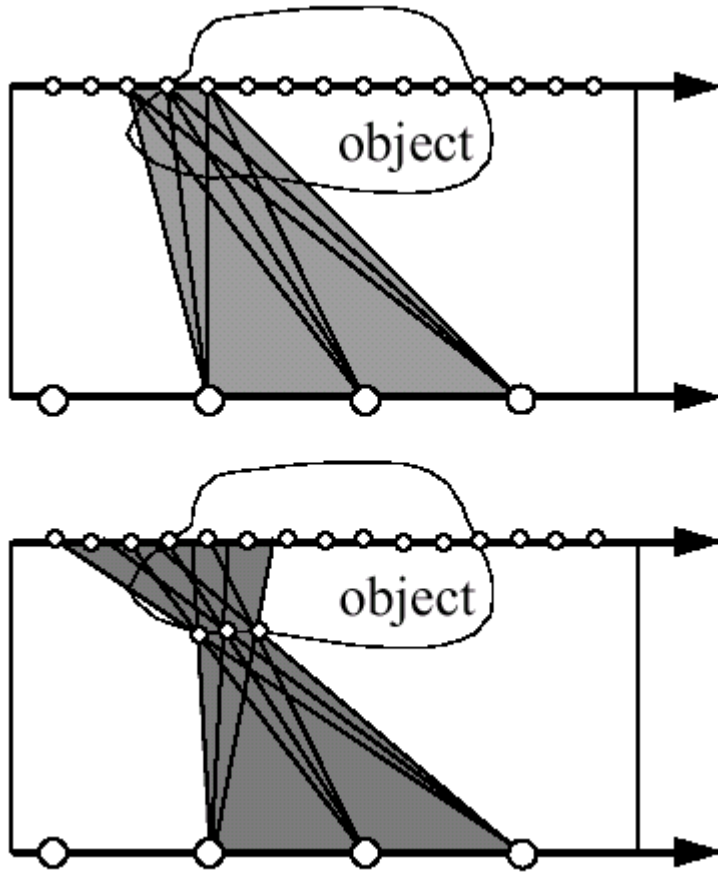
- Fit planar surface

⇒ interpolate between closest rays



# Lumigraph Rendering

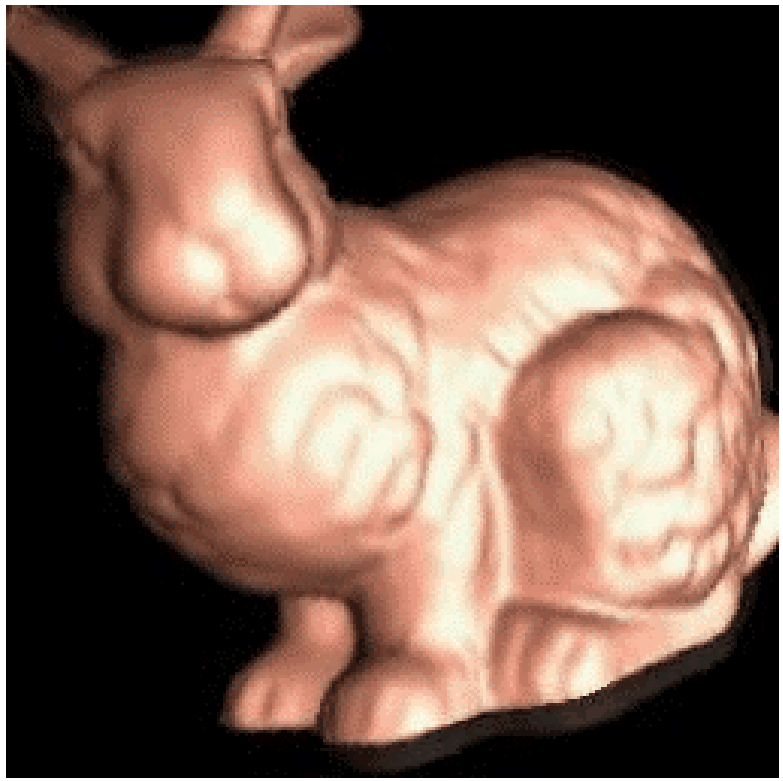
- **Approximate depth correction**
  - Backward problem
- **Parallax included**
- **Reduced aliasing**
- **Occluded regions still a problem**



# The Lumigraph – Rendering Results

Without using geometry

⇒ Light Field Rendering



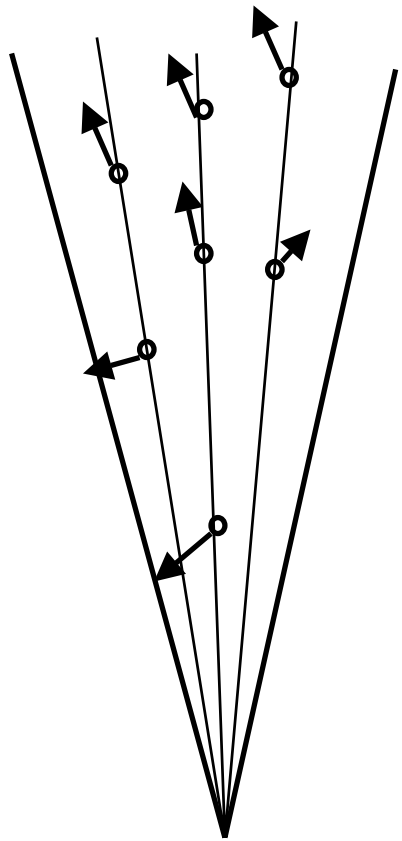
Using approximate geometry

# Layered Depth Images

J. Shade et al., “Layered Depth Images”, Siggraph’98

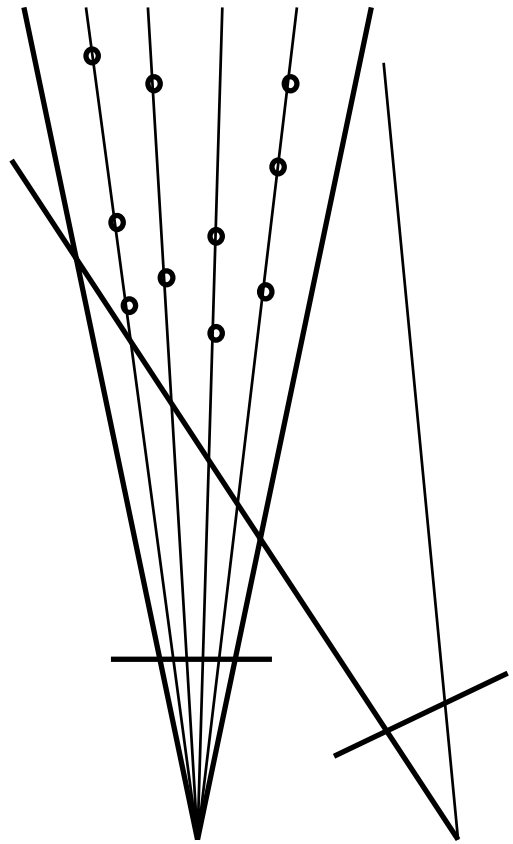
grail.cs.washington.edu/projects/ldi/

- **Idea:**
  - Handle disocclusion
  - Store invisible geometry in depth images
- **Data structure:**
  - Per pixel list of depth samples
  - Per depth sample:
    - RGBA
    - Z
    - Encoded: Normal direction, distance



# Layered Depth Images II

- **Computation:**
  - Incremental warping computation
  - Implicit ordering information
  - Splat size computation
    - Table lookup
    - Fixed splat templates
  - Clipping of LDIs



# Layered Depth Images III

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

# View Morphing

S. Seitz and C. Dyer, “View Morphing”, Siggraph’96

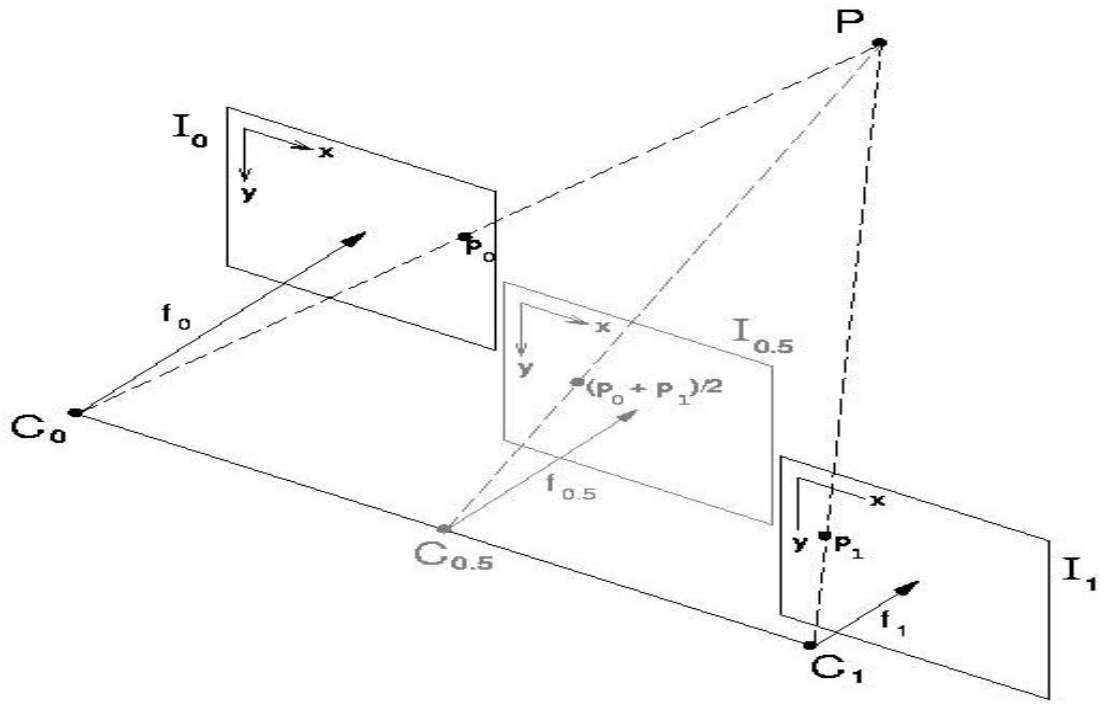
www.cs.washington.edu/homes/seitz/vmorph/vmorph.htm

- Warping between 2 (or more) images
- Cameras’ **F** matrices known
- Image correspondences known for all pixels

$\Rightarrow$ Continuously warp one image into the other giving a physically plausible impression

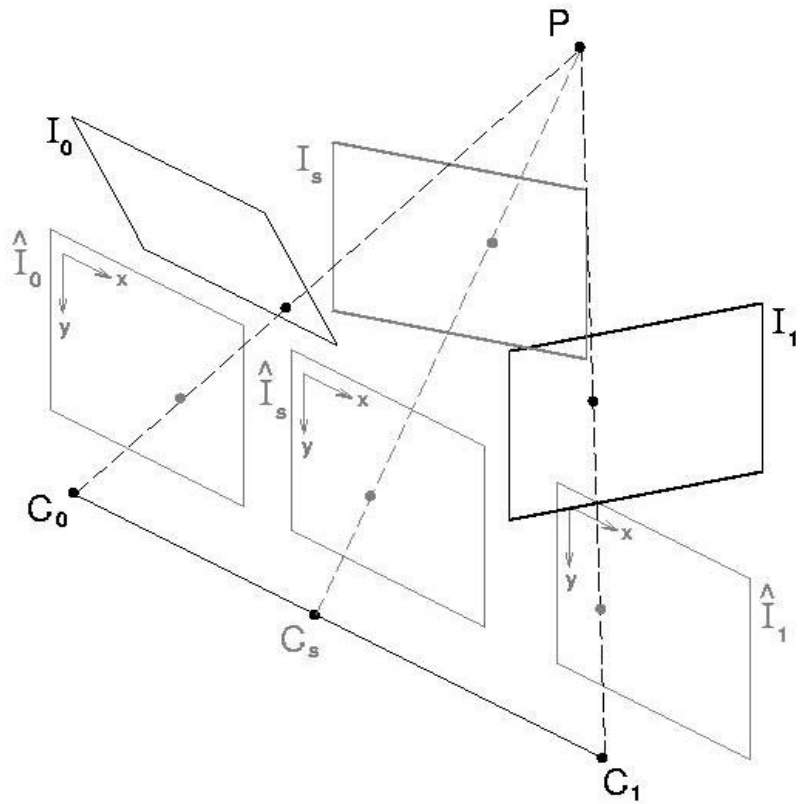
# View Morphing II

- Morphing between parallel views
  - epipolar lines are parallel
  - simple depth - disparity correspondence
  - linear interpolation of pixels of both images

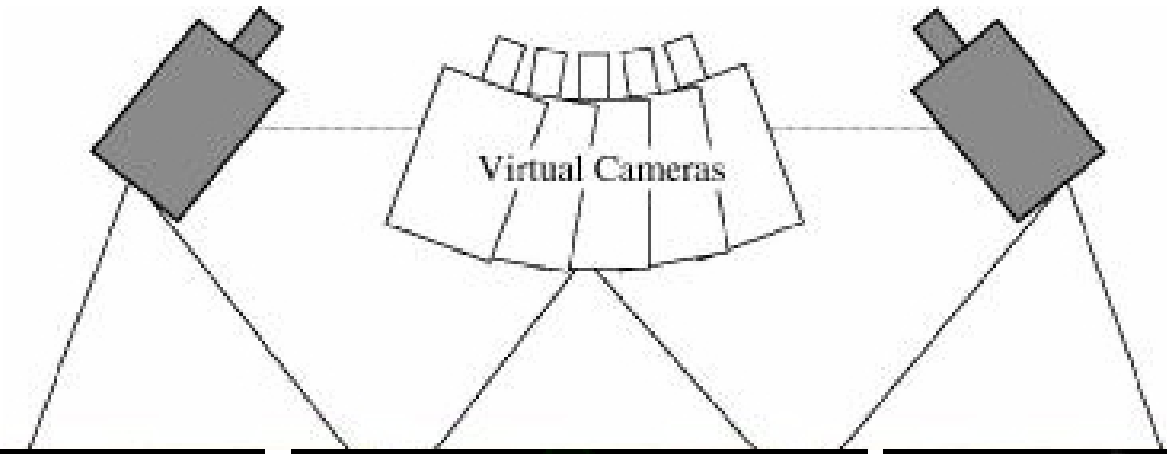


# View Morphing III

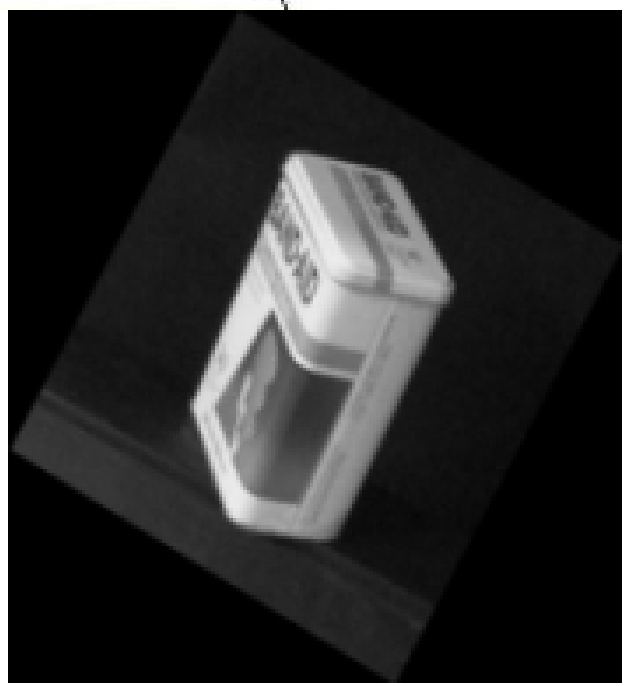
- Morphing between Non-parallel views
  - Prewarp to common plane (homography)
  - Morph
  - Postwarp



# View Morphing - Results





# View-dependent Texture Mapping

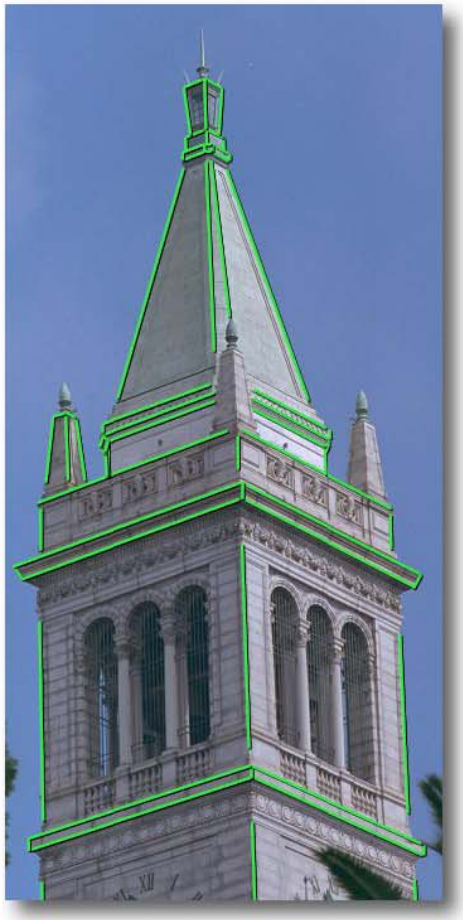
P. Debevec et al.,
"Efficient View-Dependent Image-based Rendering with Projective Texture-Mapping",
Eurographics Rendering Workshop'98
www.debevec.org/Research/VDTM/

- Complete 3D scene geometry model
- Multiple photographs of scene
- Fully calibrated images

$\Rightarrow$ Map photos as texture onto geometry

$\Rightarrow$ Use image closest to viewing direction for texturing

# View-dependent Texture Mapping



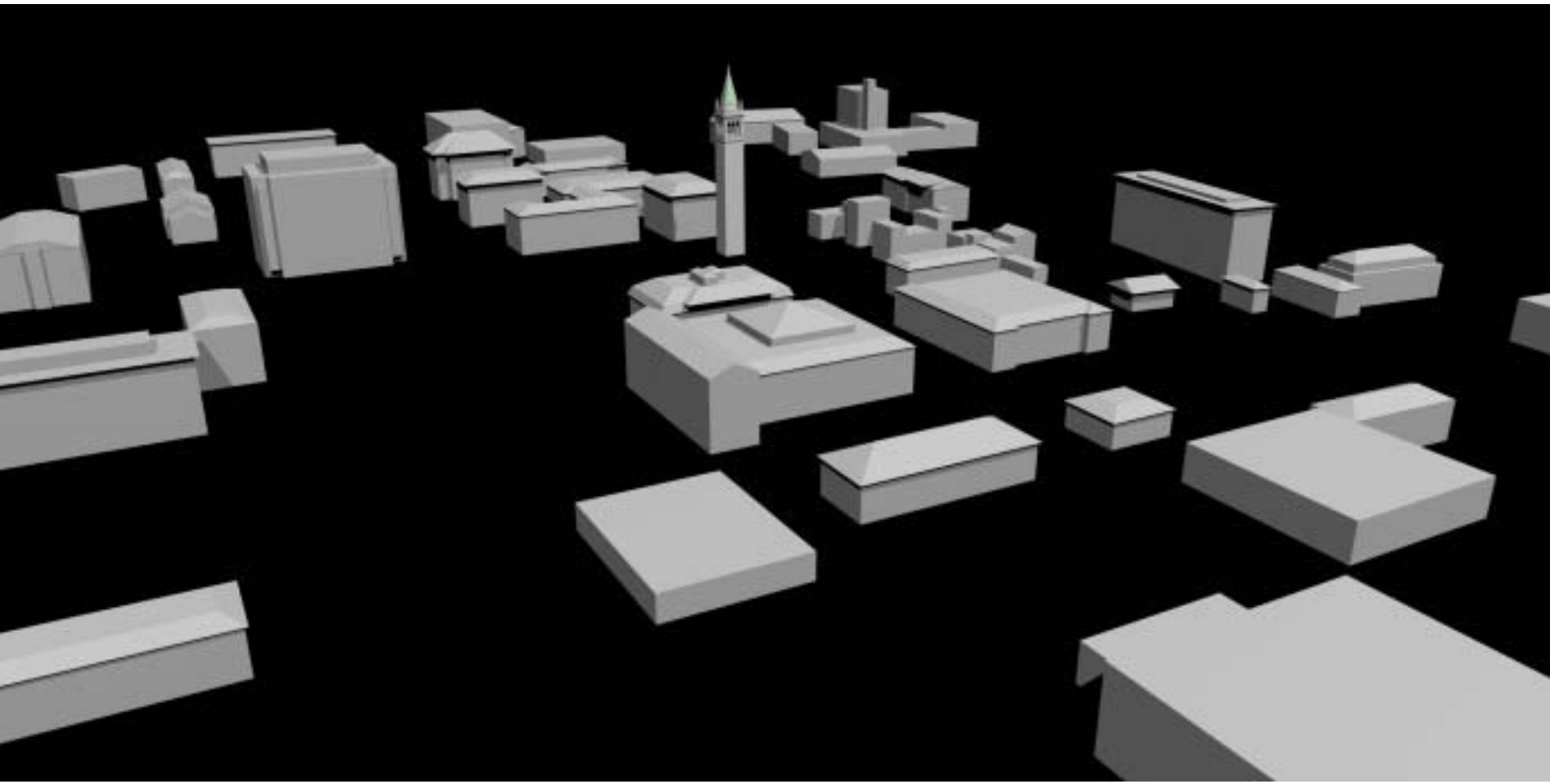
Original photograph with marked edges

Recovered model

Model edges projected onto photograph

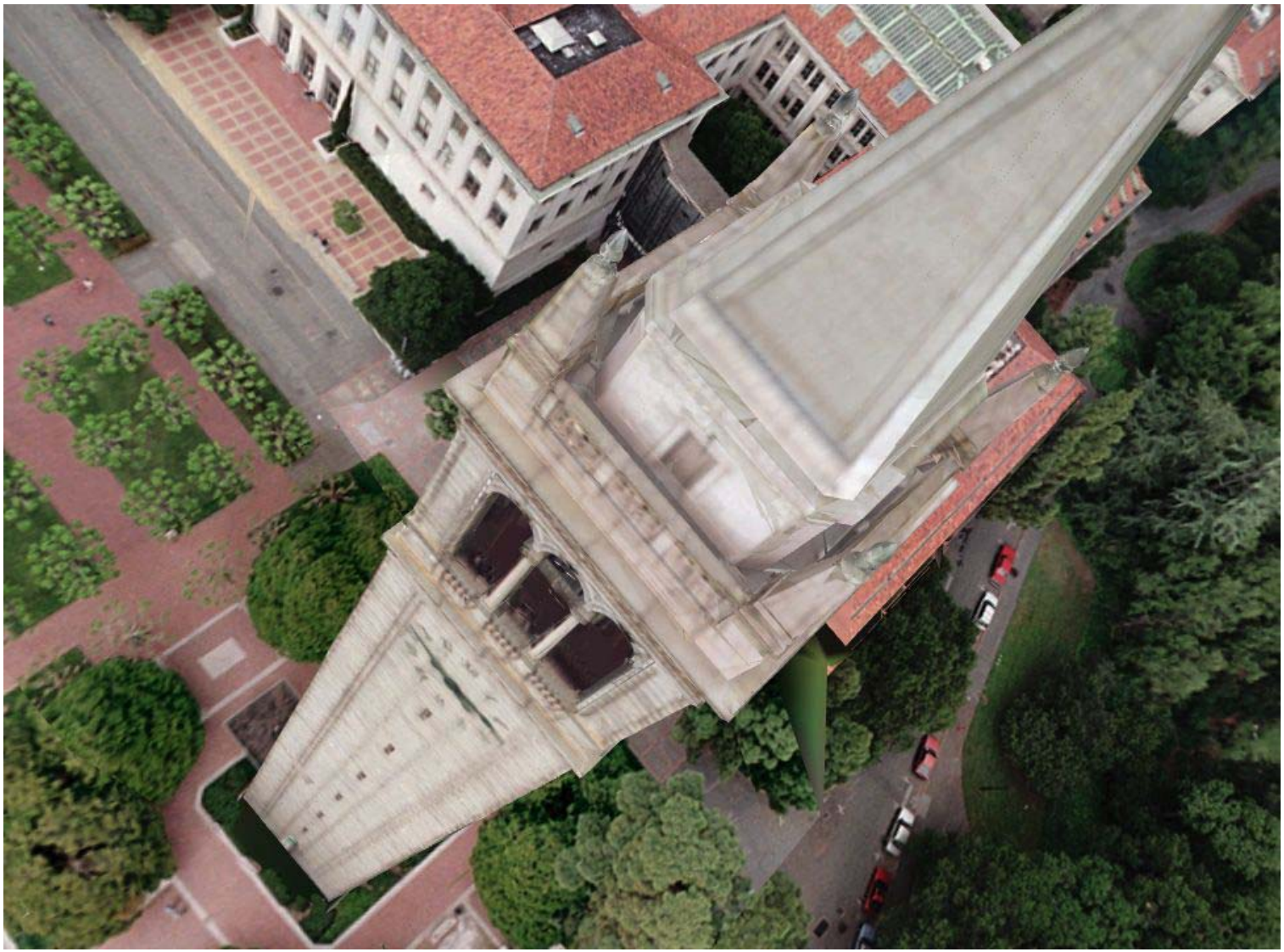
Synthetic rendering

# View-dependent Texture Mapping



# View-dependent Texture Mapping

# View-dependent Texture Mapping



QuickTime Movie

QuickTime Movie

QuickTime Movie

www.debevec.org

# Surface Light Fields

D. Wood et al., “Surface Light Fields for 3D Photography”, Siggraph’00

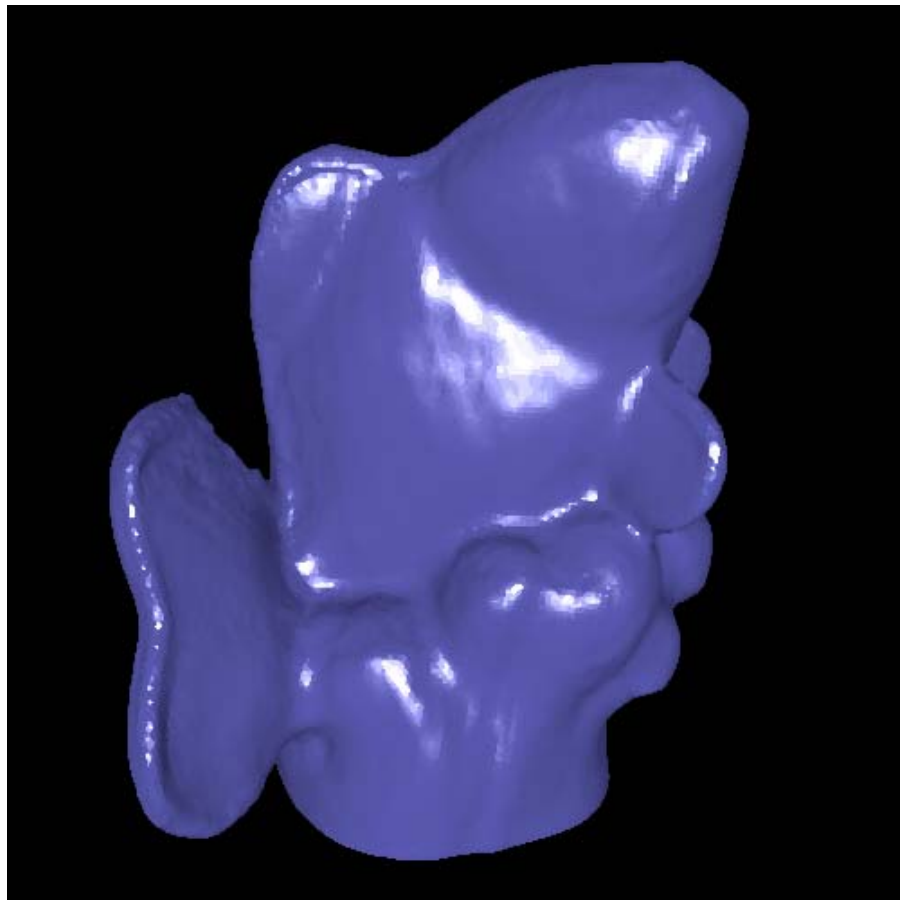
grail.cs.washington.edu/projects/slf/

- Complete 3D scene geometry model
- Multiple photographs
- Fully calibrated camera

$\Rightarrow$ Parameterize Light Field over object surface

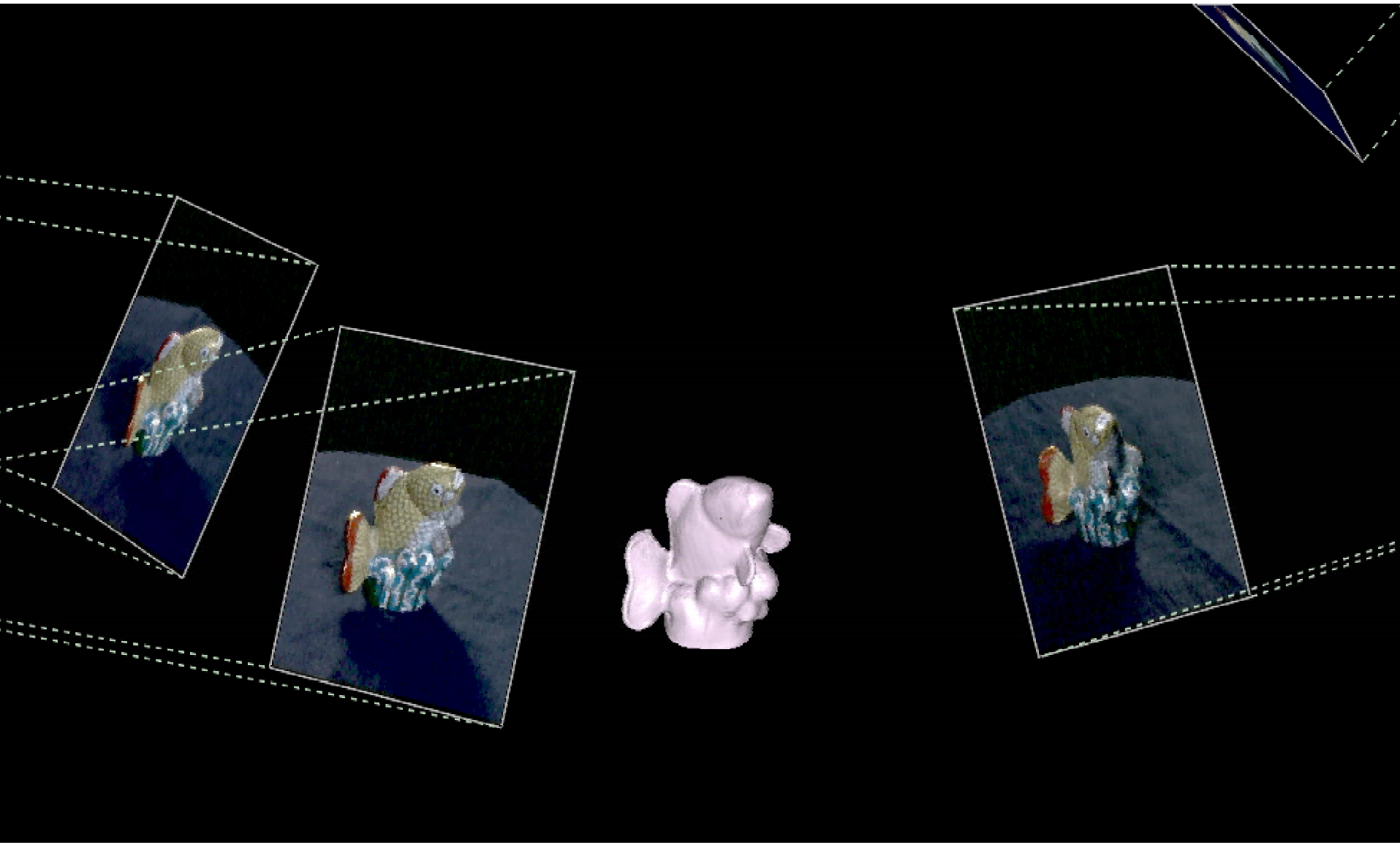
# SLF: Geometry Model Acquisition

Range scans
(only a few shown . . .)

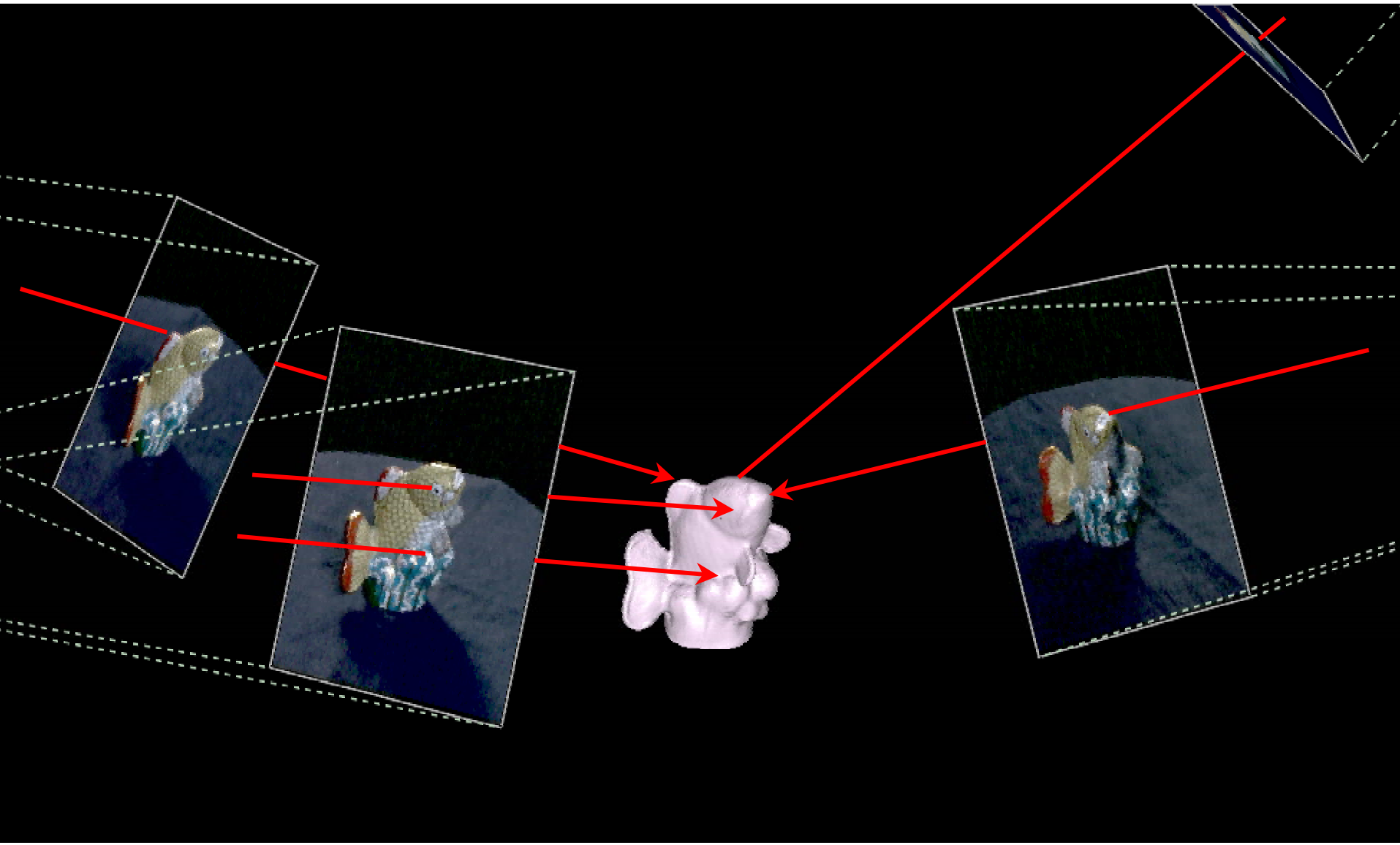


Merged geometry model

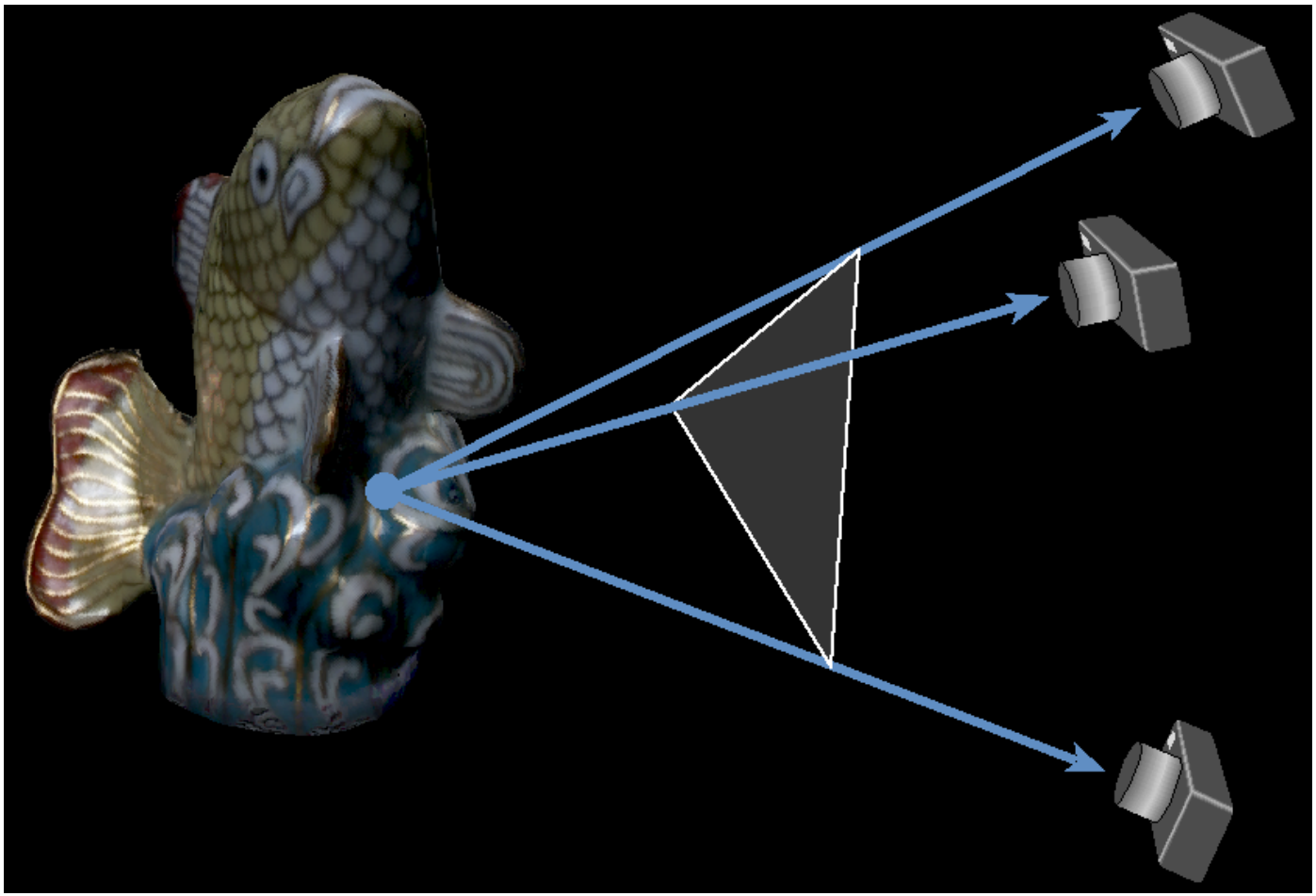
# SLF: Register Images to Geometry



# SLF: Register Images to Geometry

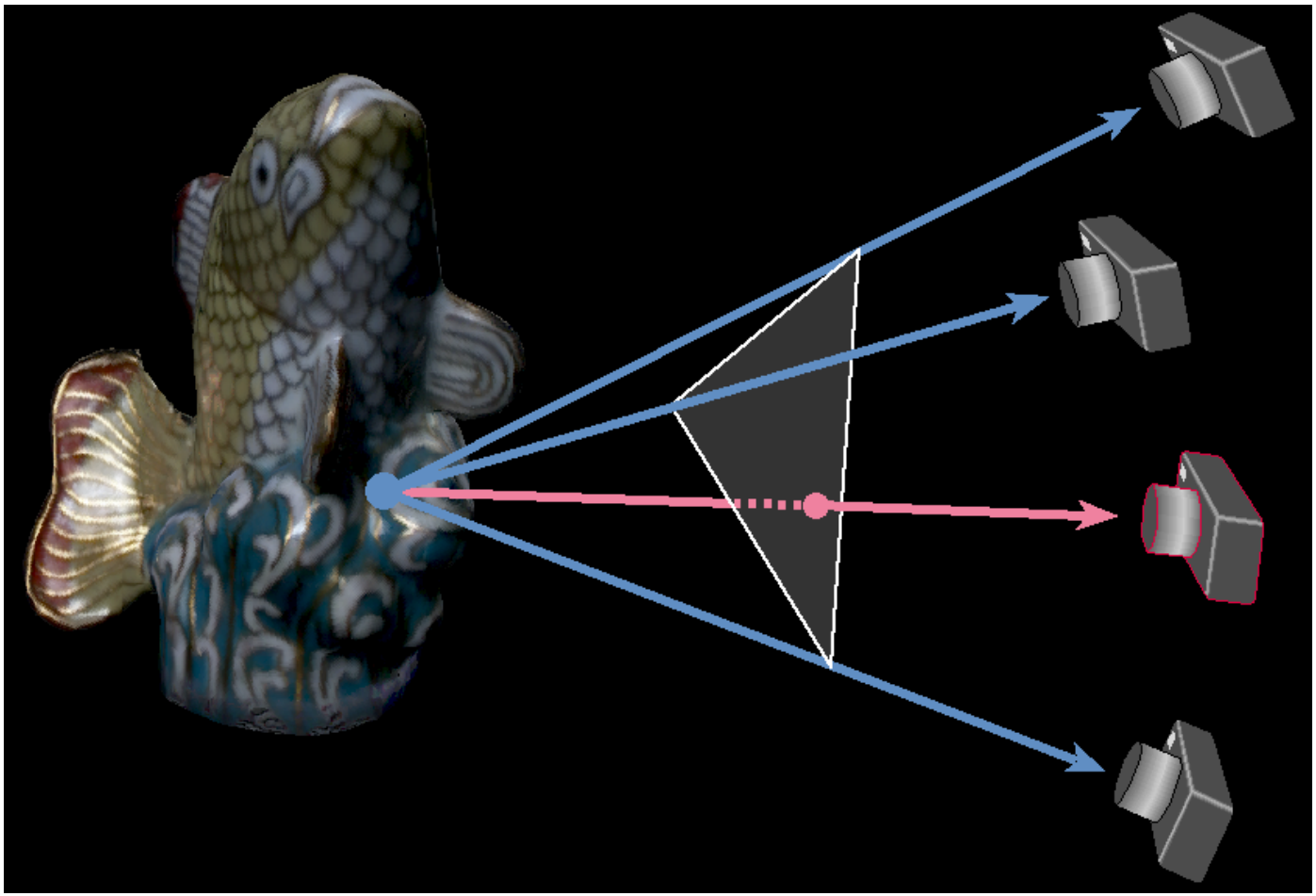


# SLF vs. View-dependent Texture Mapping



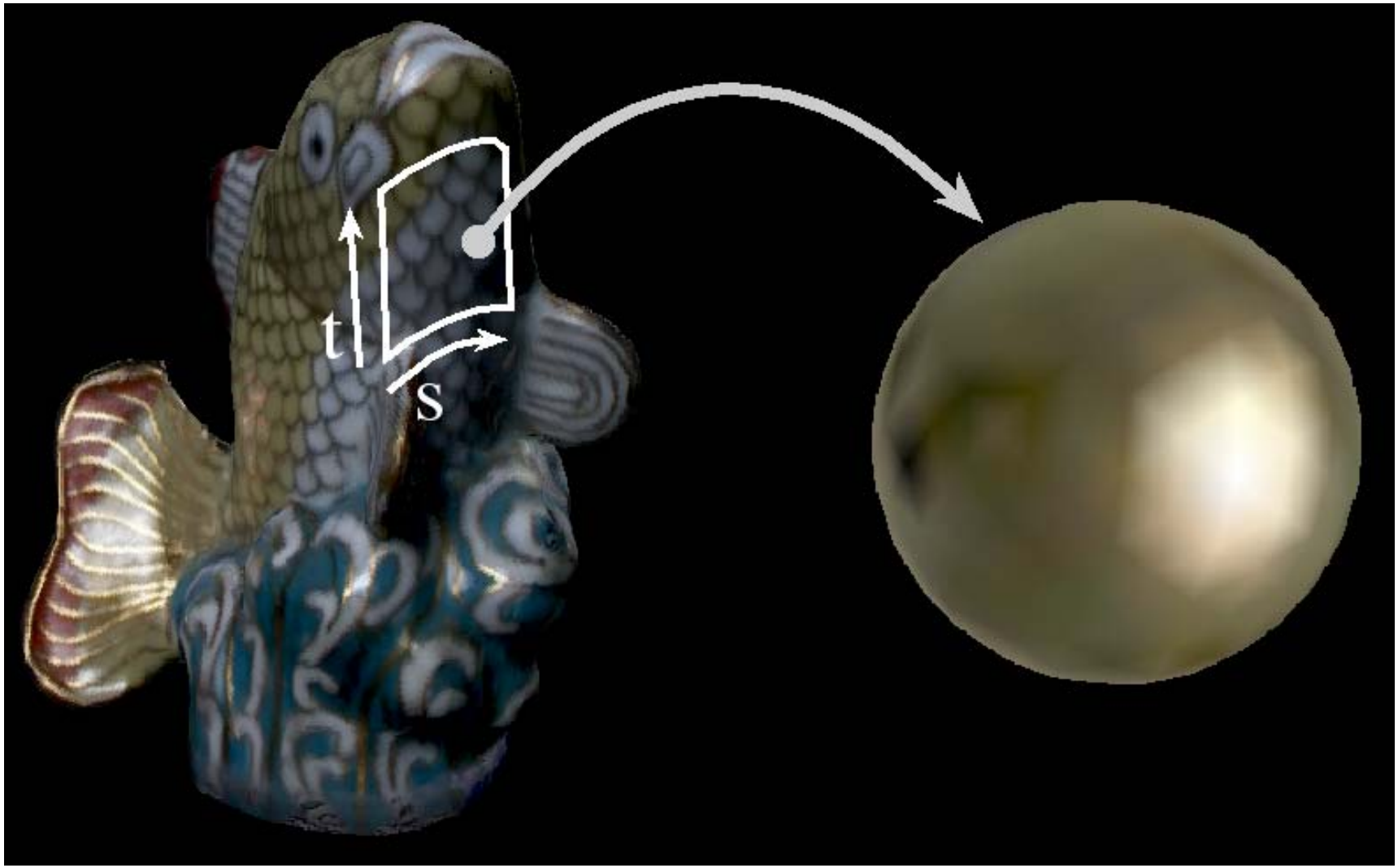
Debevec *et al*. 1996, 1998
Pulli *et al*. 1997

# SLF vs. View-dependent Texture Mapping

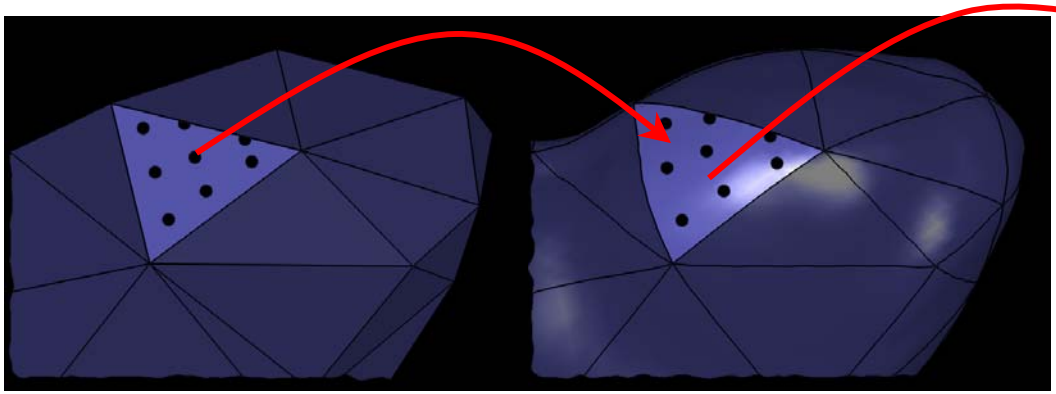


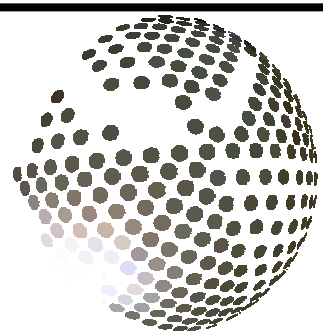
Debevec *et al*. 1996, 1998
Pulli *et al*. 1997

# SLF: Lumispheres

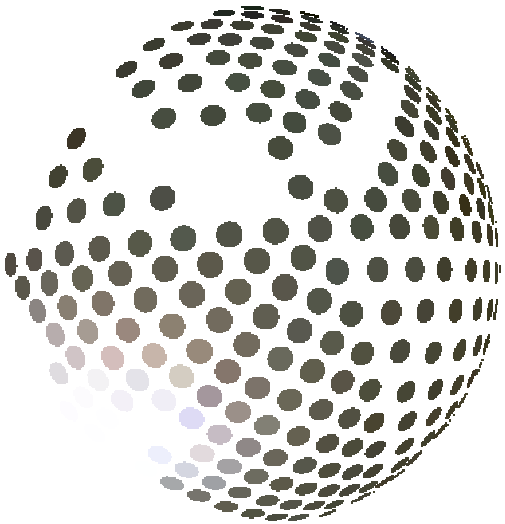


# SLF: Lumisphere Fairing





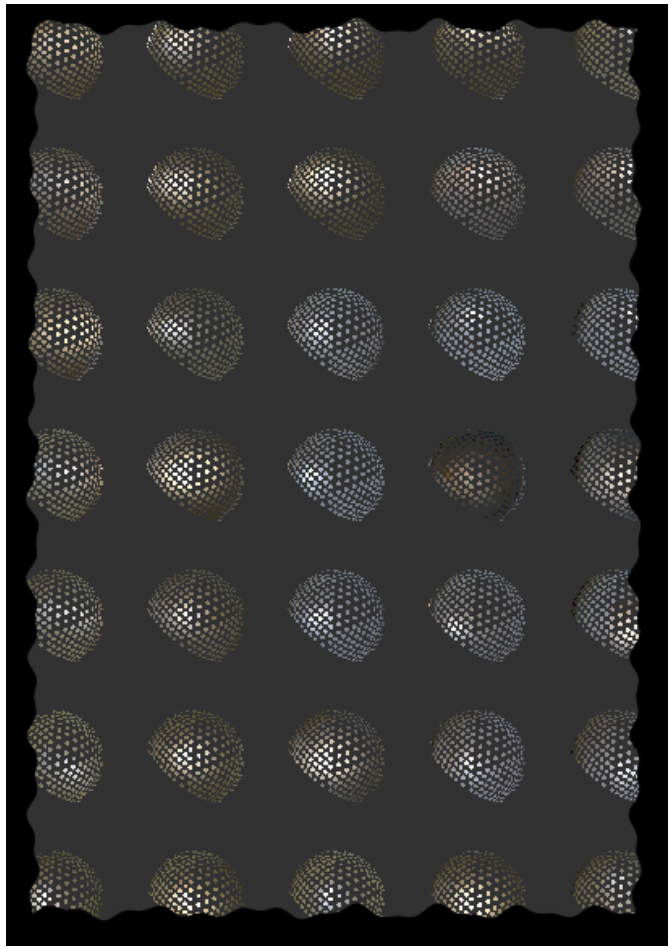
Data lumisphere



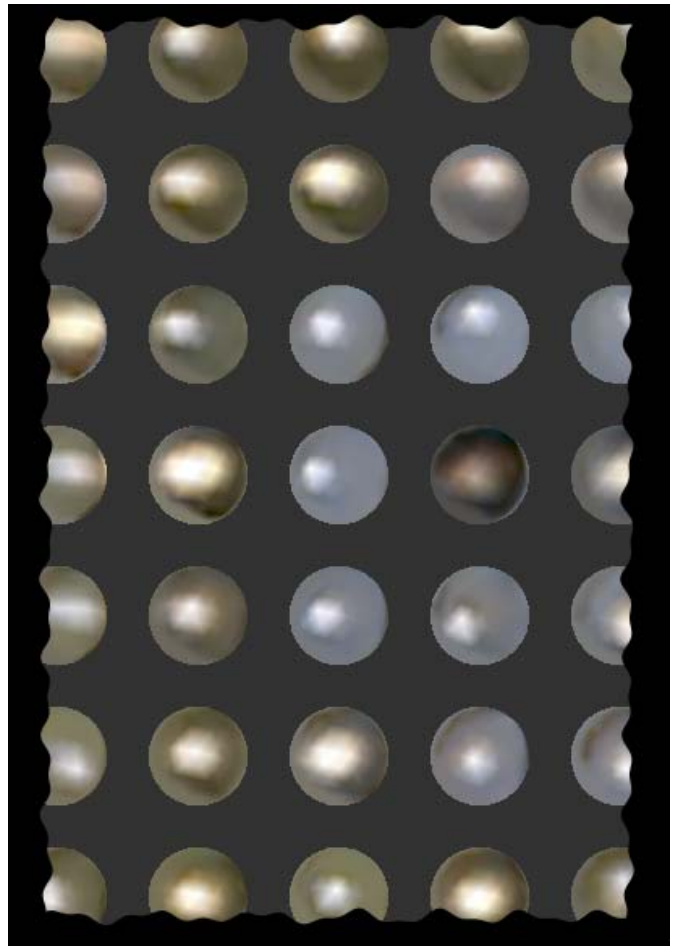
Data lumisphere

Faired lumisphere

# SLF: Lumisphere Matrix



Many input data lumispheres



Many faired lumispheres

# SLF - Results

# Wrap-Up

**Theoretical Background**

- Plenoptic Function

**“Pure” IBR**

- Panoramas
- Concentric Mosaics
- Light Field Rendering

**Geometry-assisted IBR**

- The Lumigraph
- Layered Depth Images
- View Morphing
- View-dependent Texture Mapping
- Surface Light Fields