# Image-Based Rendering

**Computer Vision**

CSE576, Spring 2005

Richard Szeliski

# Today’s lecture

Image–Based Rendering

- Light Fields and Lumigraphs
- Panoramas and Concentric Mosaics
- Environment Matting
- Image-Based models

# Today's lecture

Video-Based Rendering

- Facial animation
- Video matting and shadow matting
- Video Textures and Animating Stills
- Video-based tours

# Readings

- S. J. Gortler , R. Grzeszczuk , R. Szeliski and M. F. Cohen, The Lumigraph, SIGGRAPH'96.
- M. Levoy and P. Hanrahan, Light field rendering, SIGGRAPH'96.
- H.-Y. Shum and L.-W. He. Rendering with concentric mosaics, SIGGRAPH’99.

# Readings

- D. E. Zongker *et al.* Environment matting and compositing, SIGGRAPH'99.
- Y.-Y. Chuang et al. Environment matting extensions: Towards higher accuracy and real-time capture. SIGGRAPH'2000, pp.121-130, 2000.
- P. E. Debevec , C. J. Taylor and J. Malik, Modeling and rendering architecture from photographs:…, SIGGRAPH'96.

# Readings

- Y.-Y. Chuang *et al*. Video matting of complex scenes. ACM Trans. on Graphics, 21(3):243-248, July 2002
- Y.-Y. Chuang *et al*. Shadow matting. ACM Transactions on Graphics, 22(3):494-500, July 2003.
- A. Schödl *et al.,* Video textures. SIGGRAPH'2000, pp. 489-498, 2000.
- M. Uyttendaele *et al*. Image-based interactive exploration of real-world environments. IEEE Comp. Graphics and Applications, 24(3), May/June 2004.

# Lightfields and Lumigraphs

(with lots of slides from Michael Cohen)

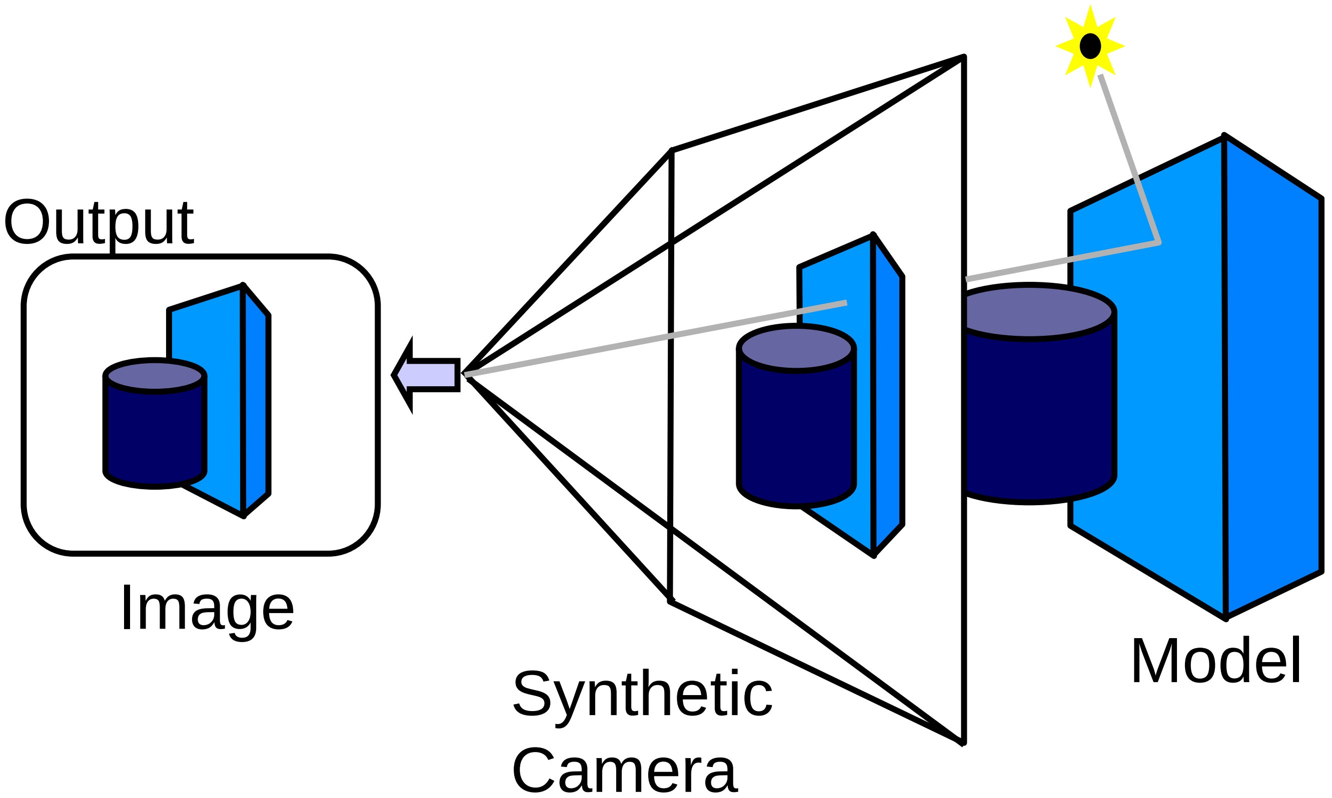
# Modeling light

How do we generate new scenes and animations from existing ones?

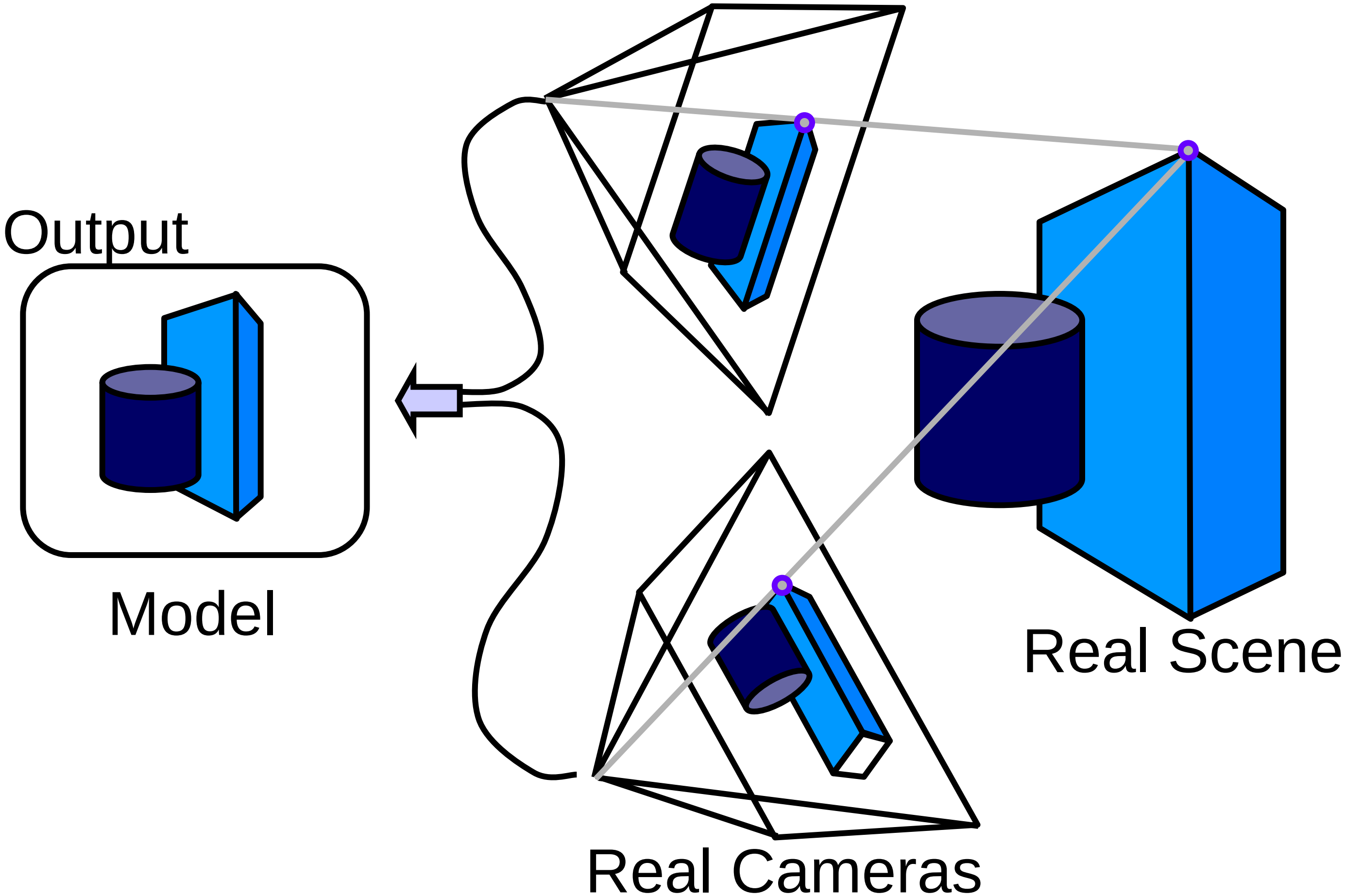
Classic “3D Vision + Graphics”:

- take (lots of) pictures
- recover camera pose
- build 3D model
- extract texture maps / BRDFs
- synthesize new views

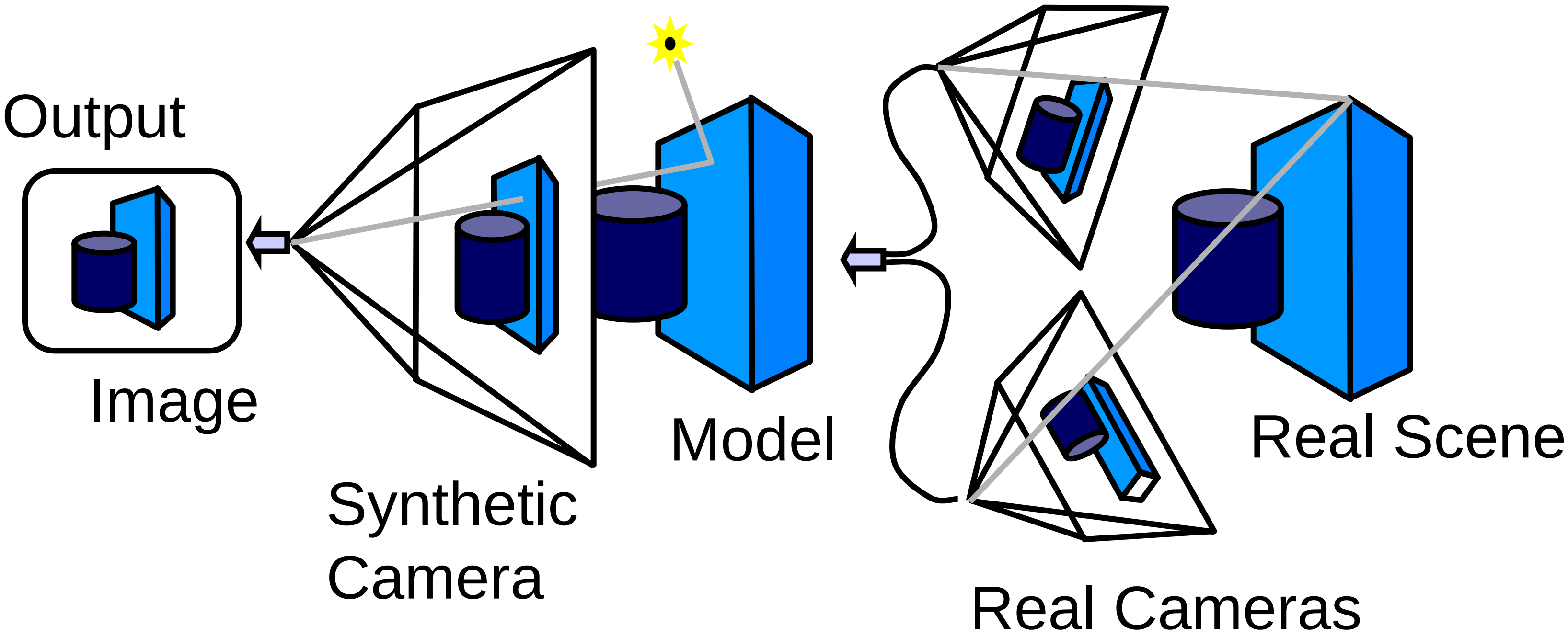
# Computer Graphics



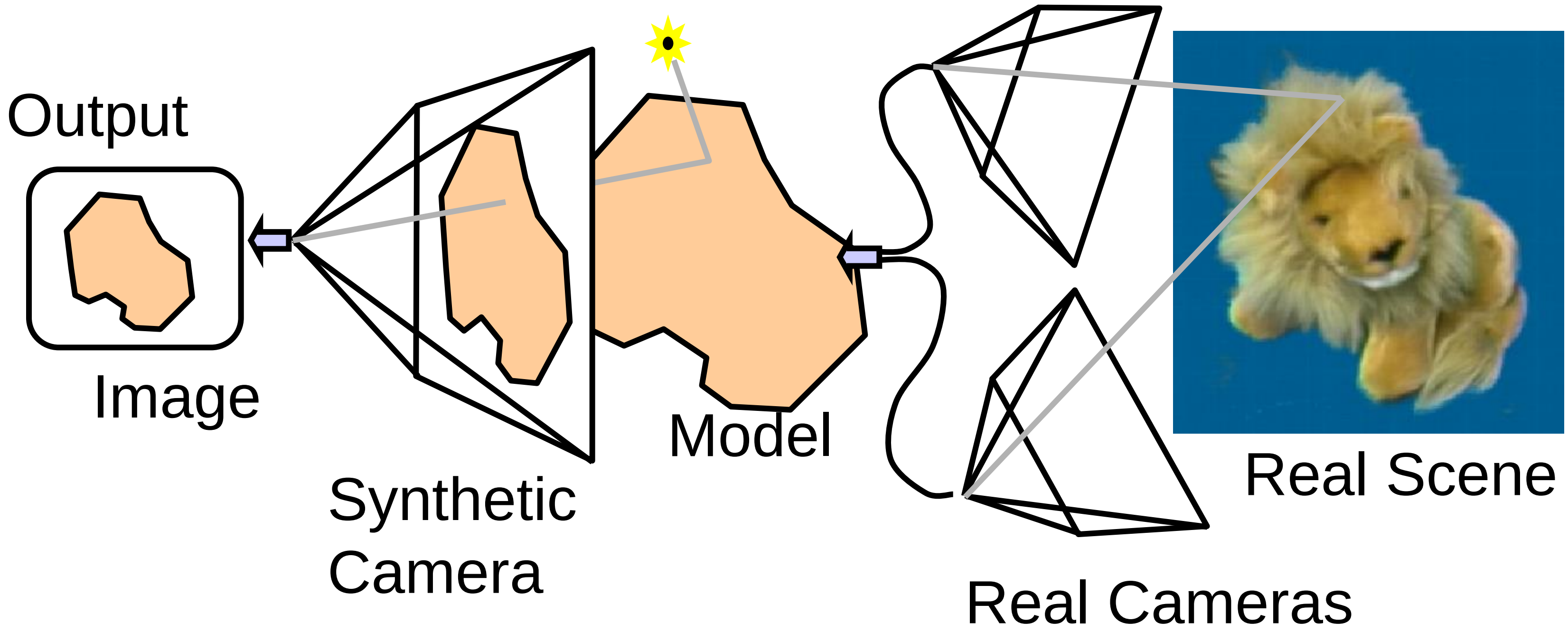
# Computer Vision



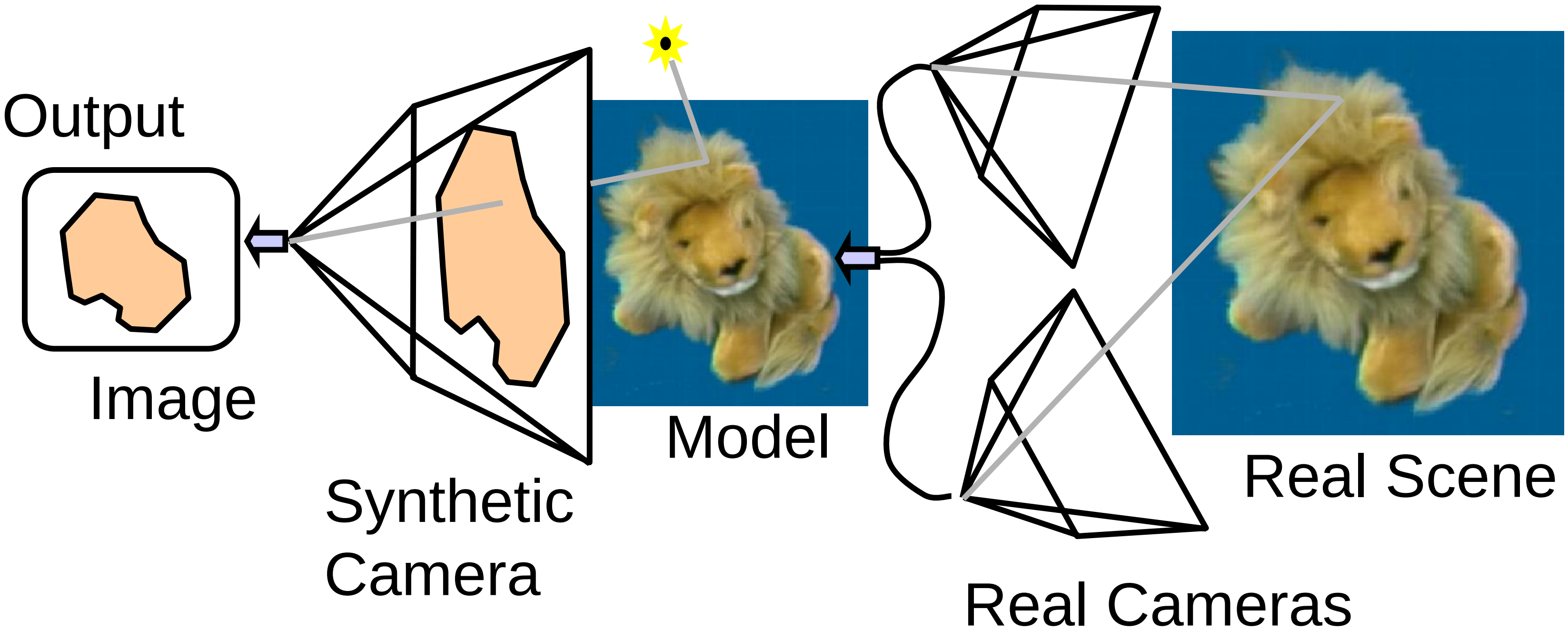
# Combined


Output
Image
Synthetic
Camera
Model
Real Scene
Real Cameras

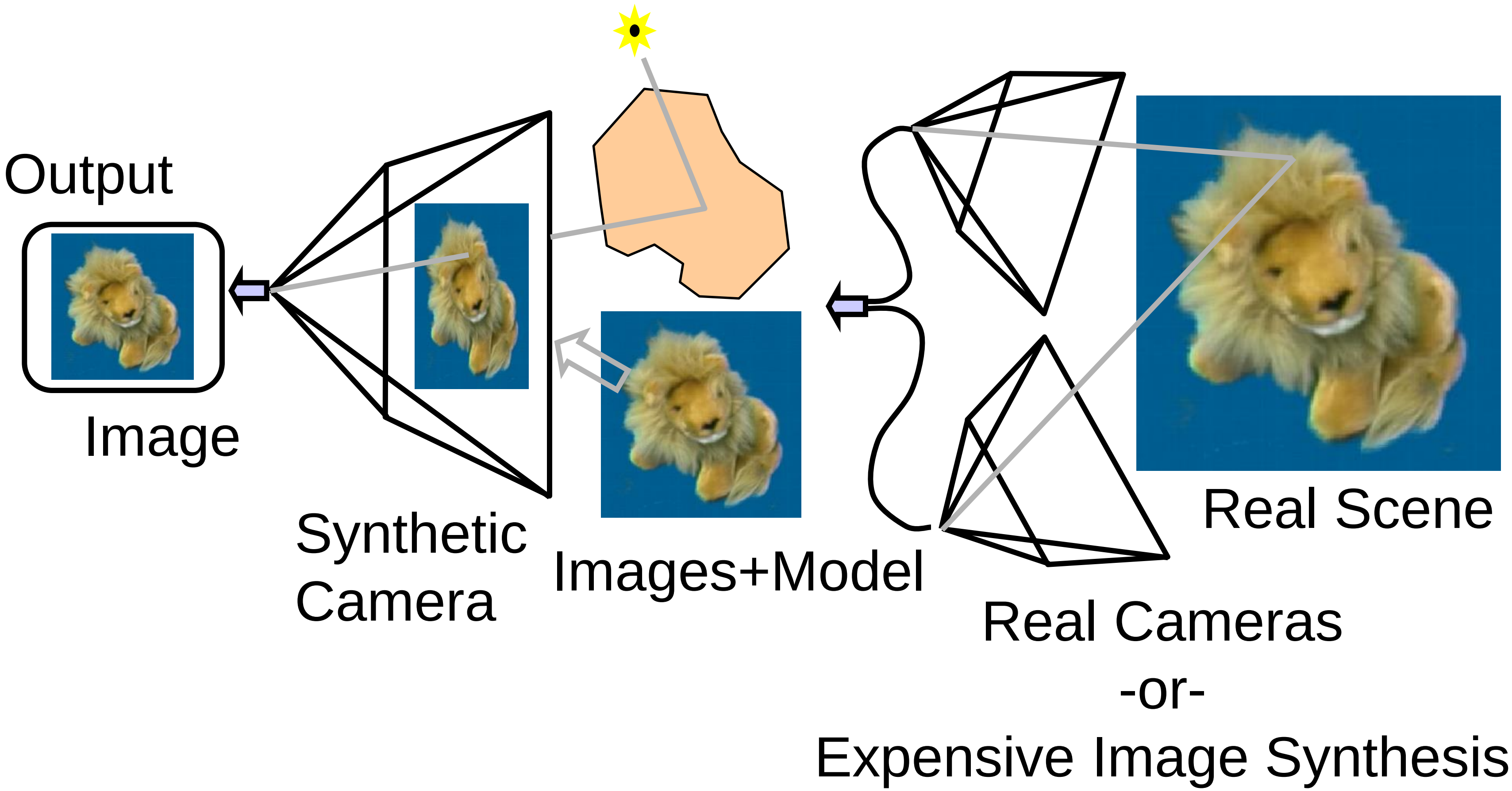
# But, vision technology falls short



# … and so does graphics.



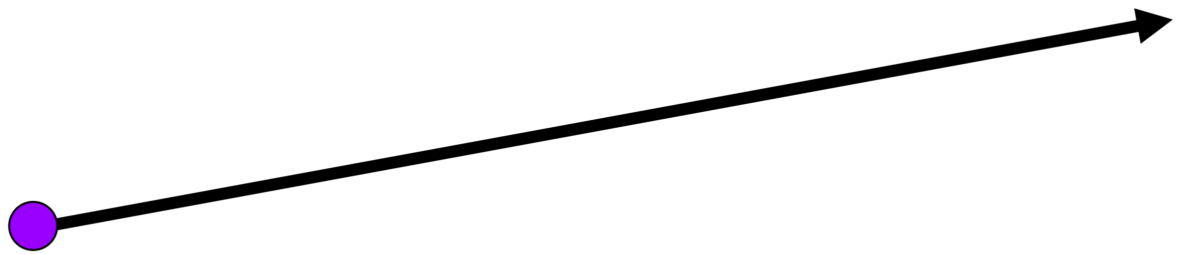
# Image Based Rendering



# Ray

Constant radiance

- time is fixed



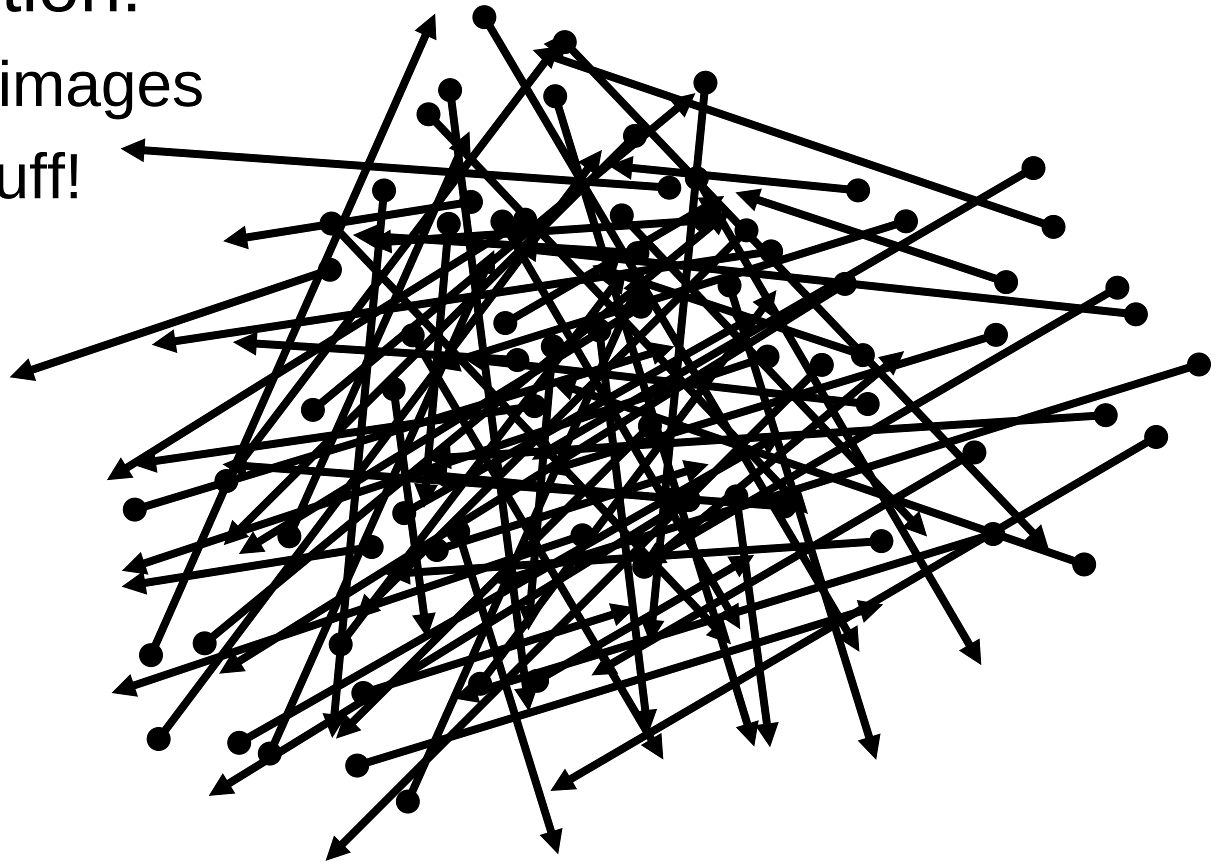
5D

- 3D position
- 2D direction

# All Rays

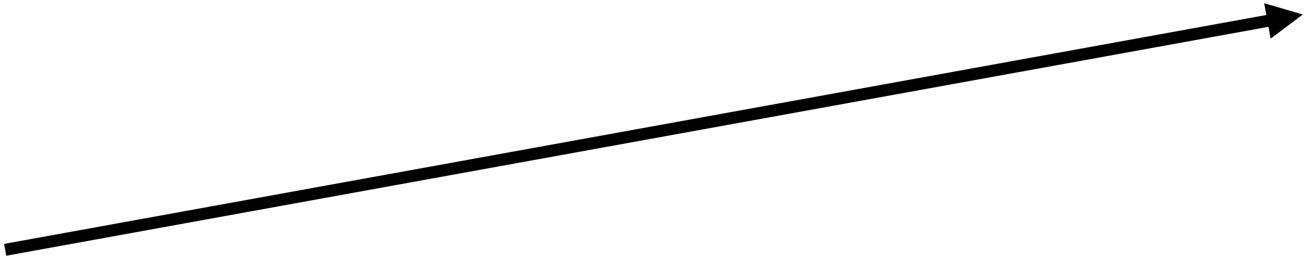
Plenoptic Function:

- all possible images
- too much stuff!



# Line

Infinite line

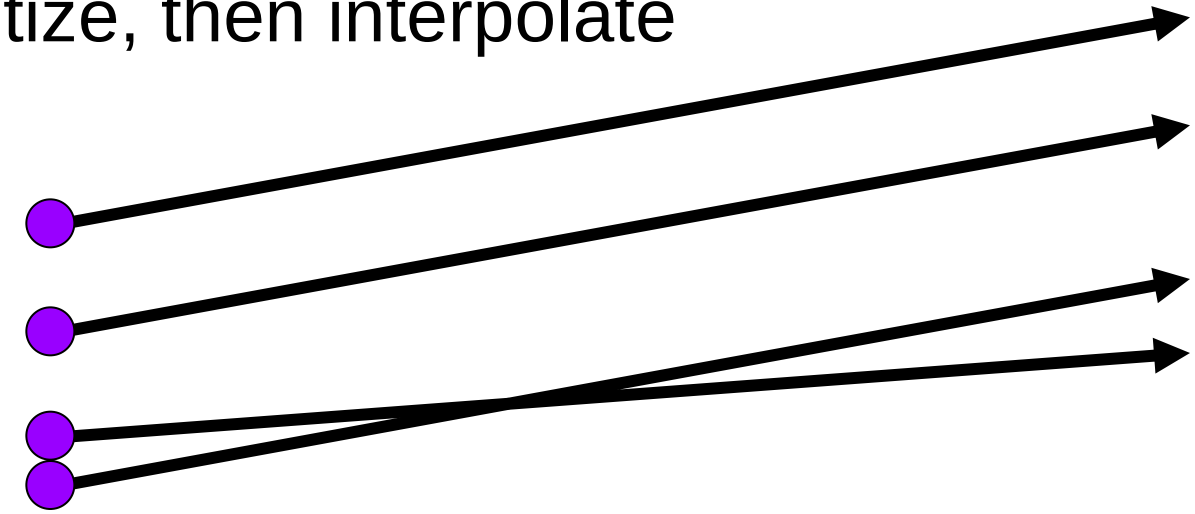


4D

- 2D direction
- 2D position
- non-dispersive medium

# Ray

Discretize, then interpolate

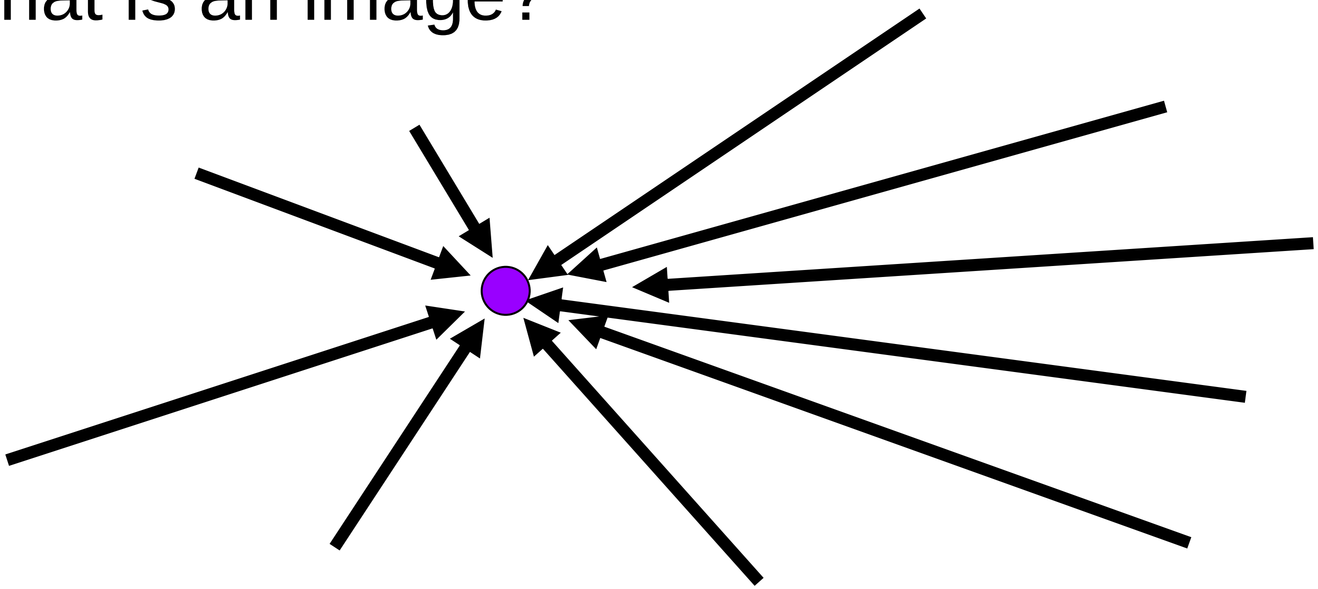


Distance between 2 rays

- Which is closer together?

# Image

What is an image?



All rays through a point

- Panorama

# Panoramic Mosaics

Convert panoramic image sequence into a cylindrical image

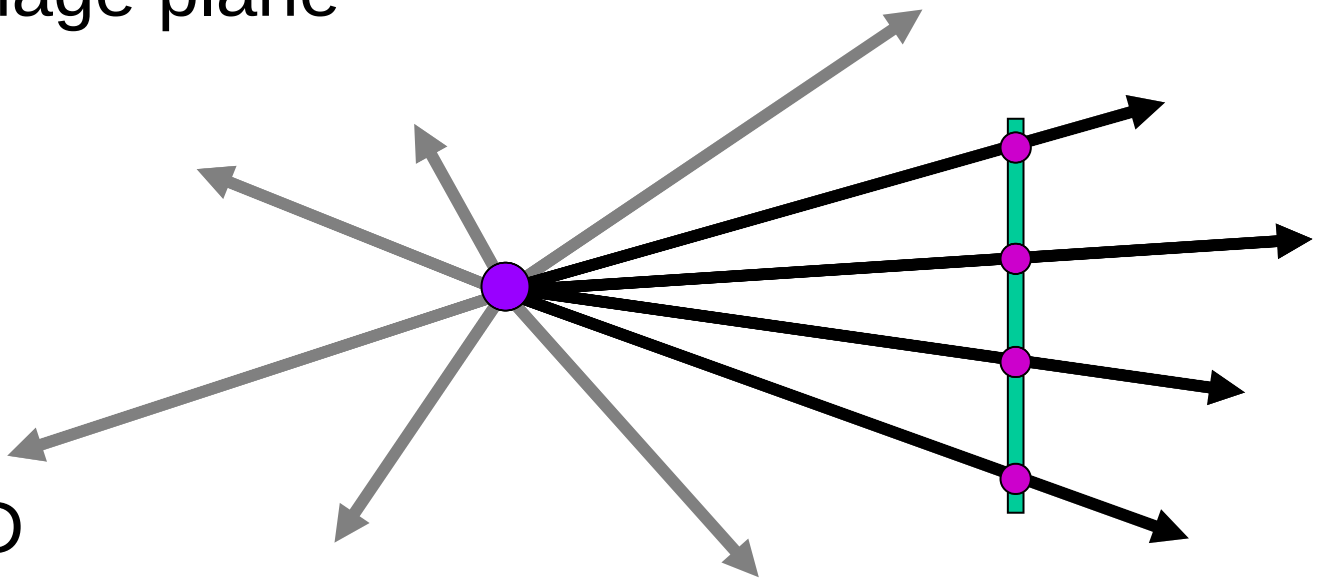
+

+ … +

=

# Image

Image plane

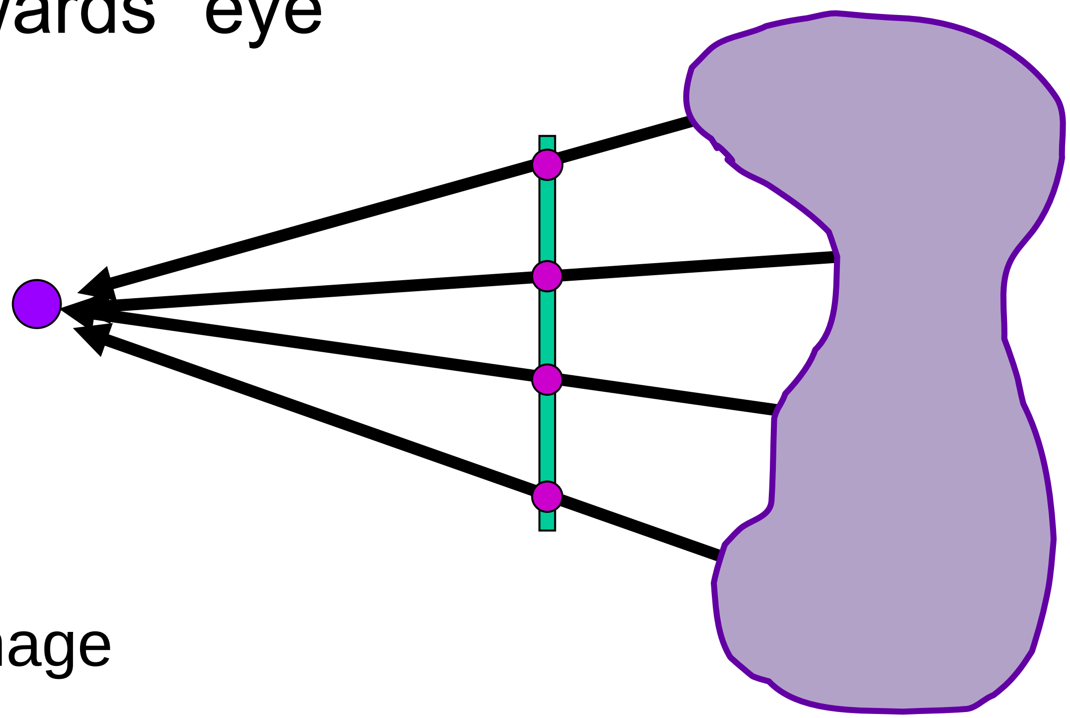


2D

- position in plane

# Object

Light leaving towards "eye"

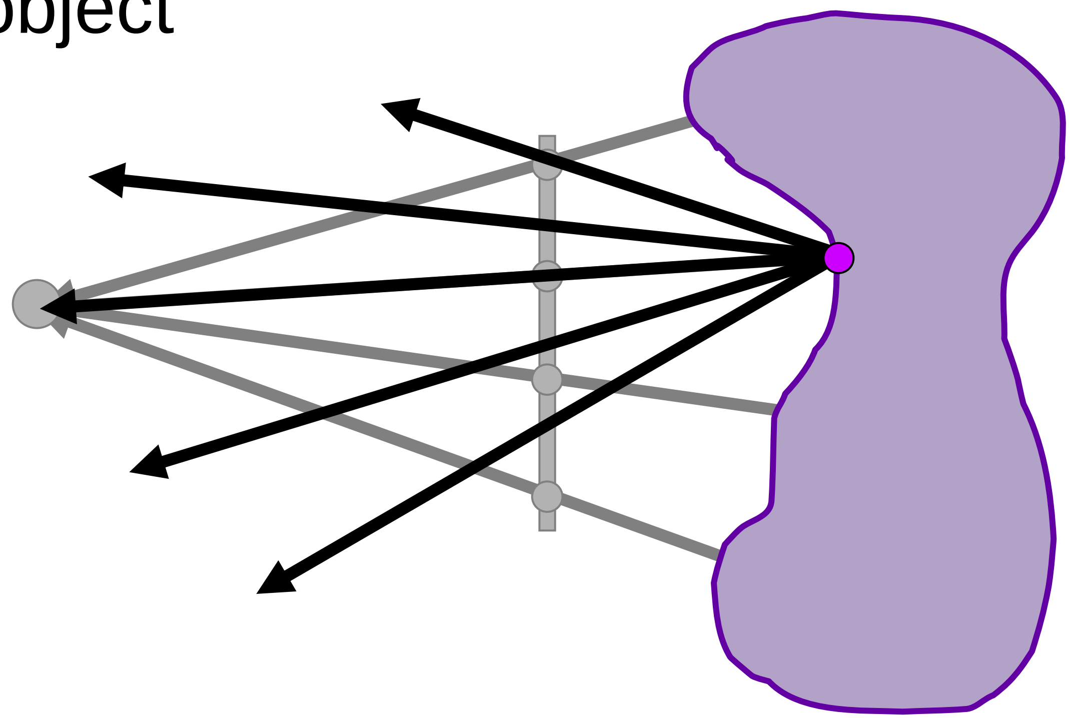


2D

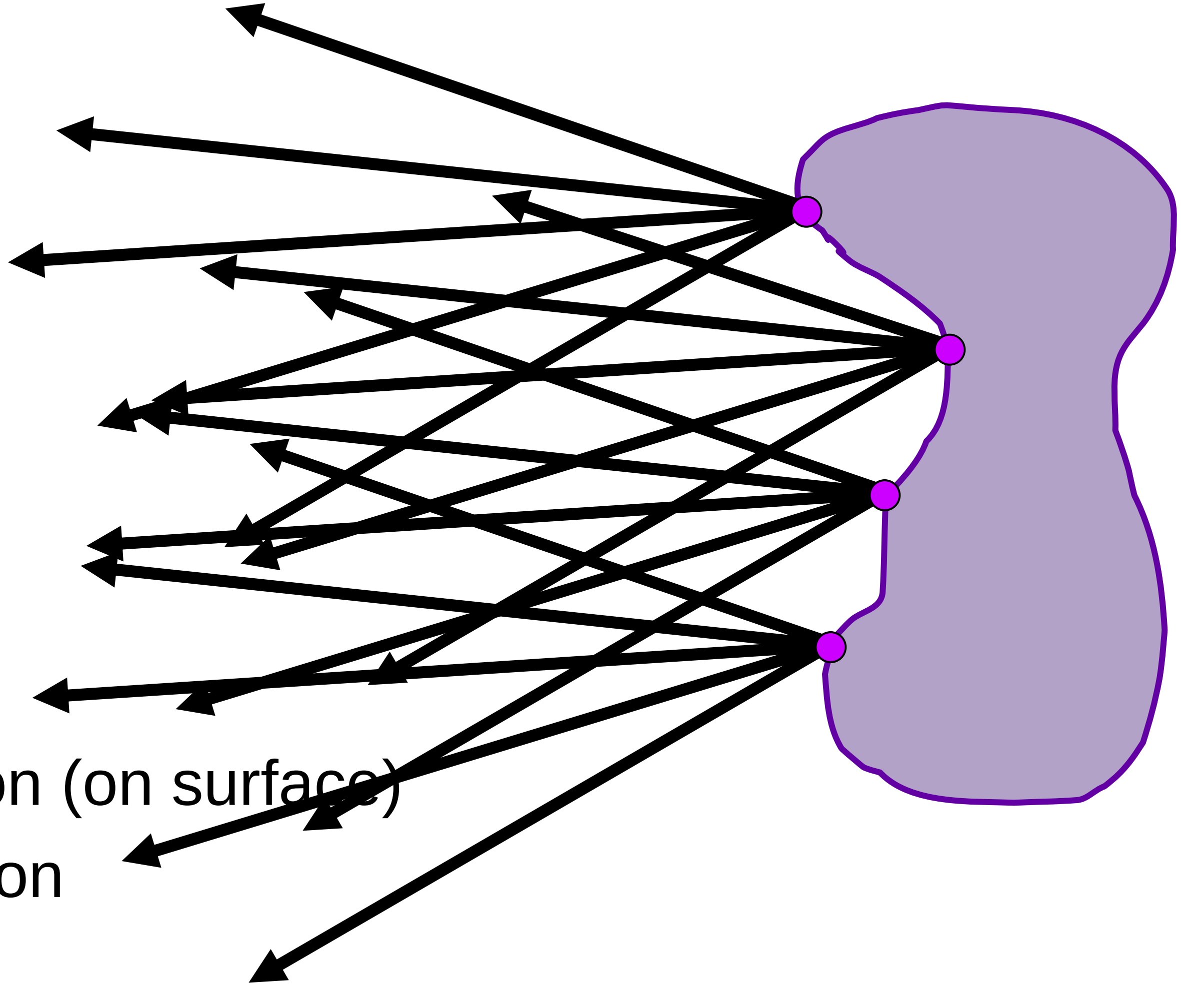
- just dual of image

# Object

All light leaving object



# Object

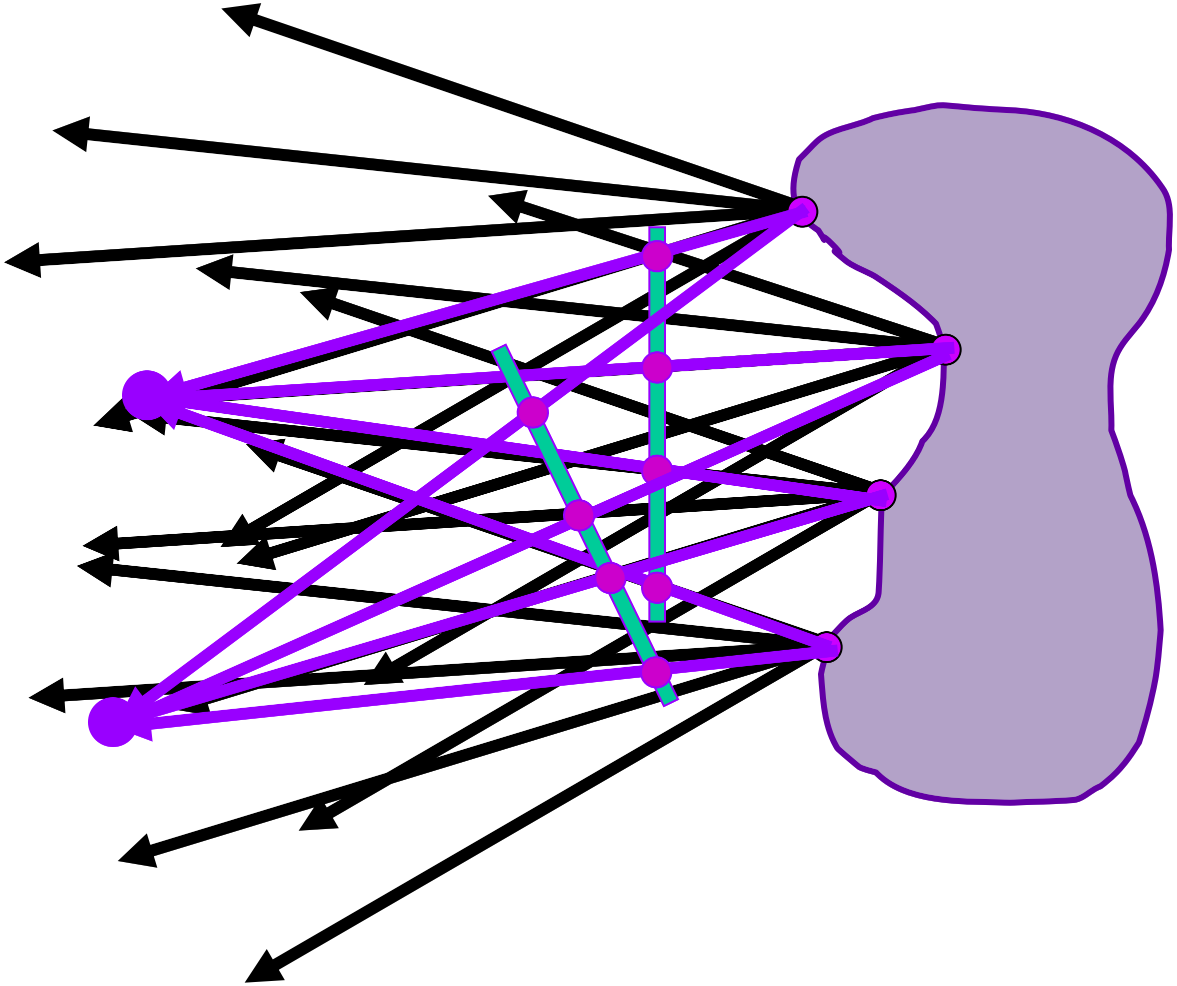


4D

- 2D position (on surface)
- 2D direction

# Object

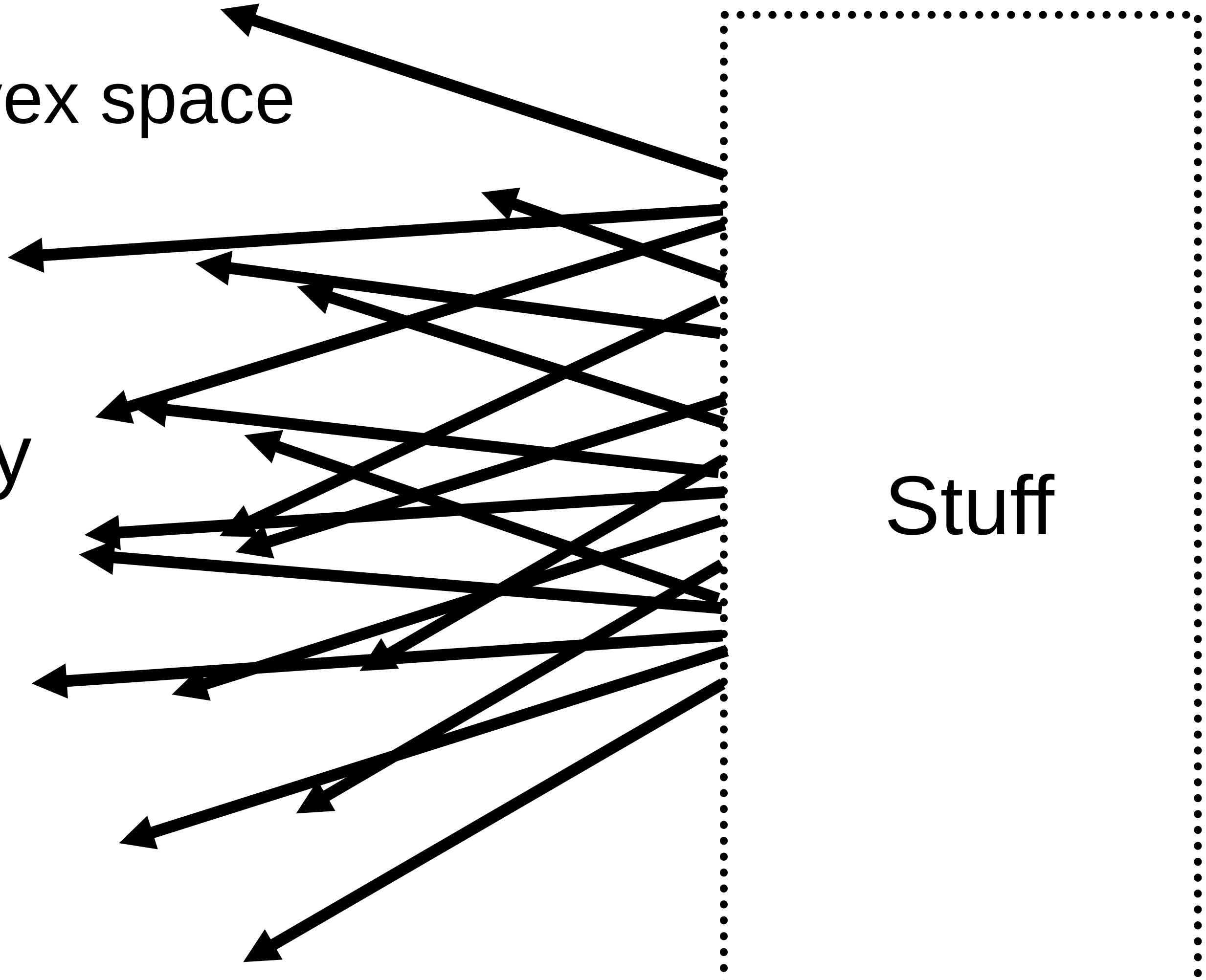
All images



# Lumigraph / Lightfield

Outside convex space

Empty

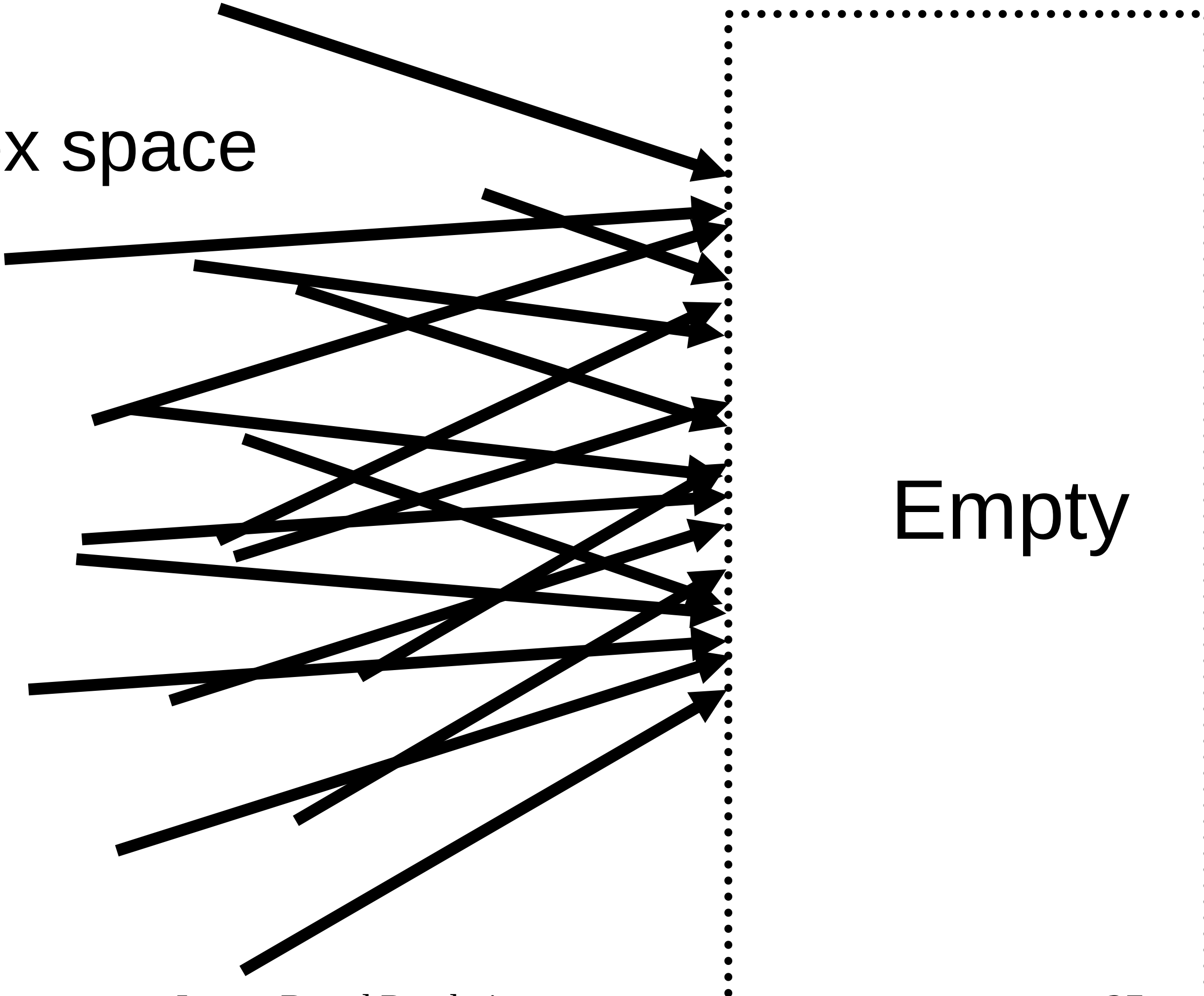



4D

# Lumigraph

Inside convex space

Stuff

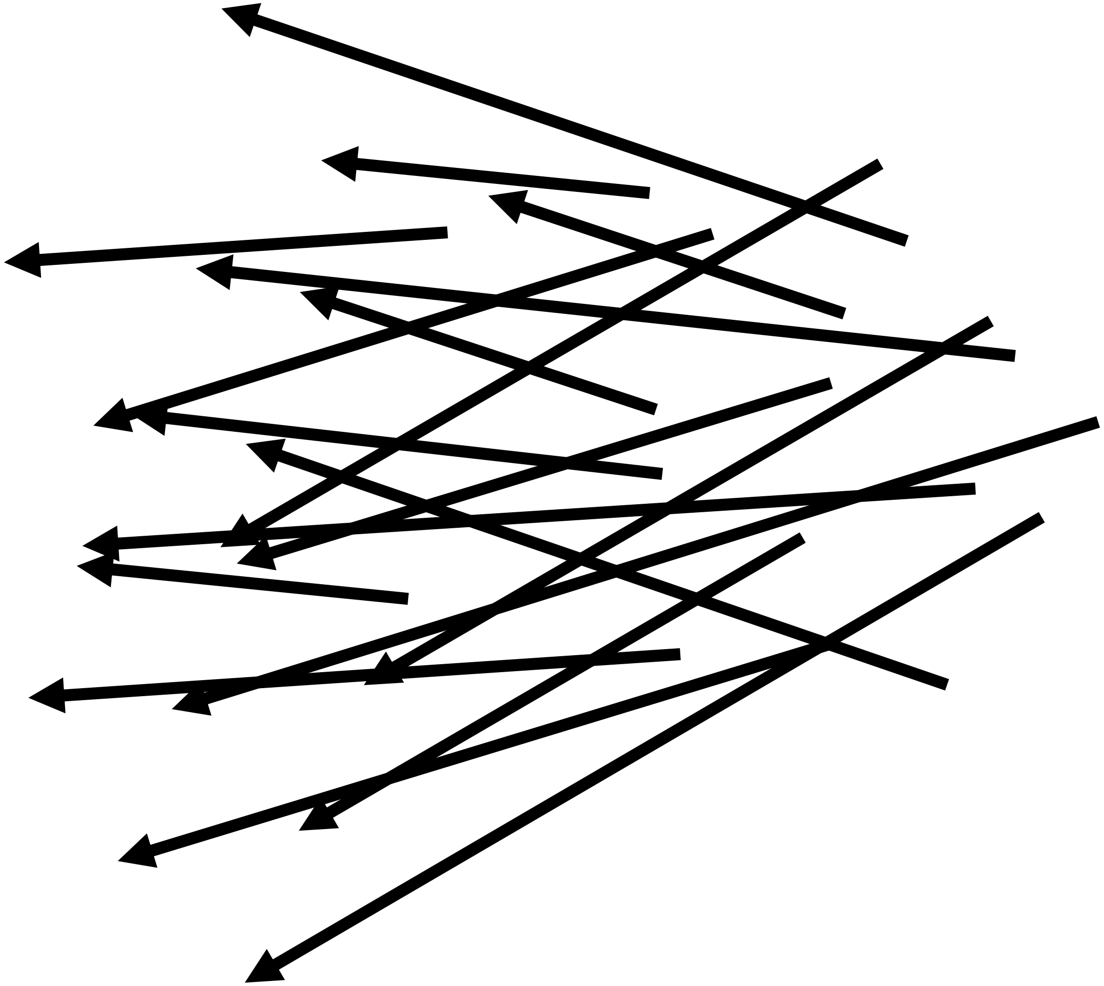



4D

# Lumigraph

How to ?

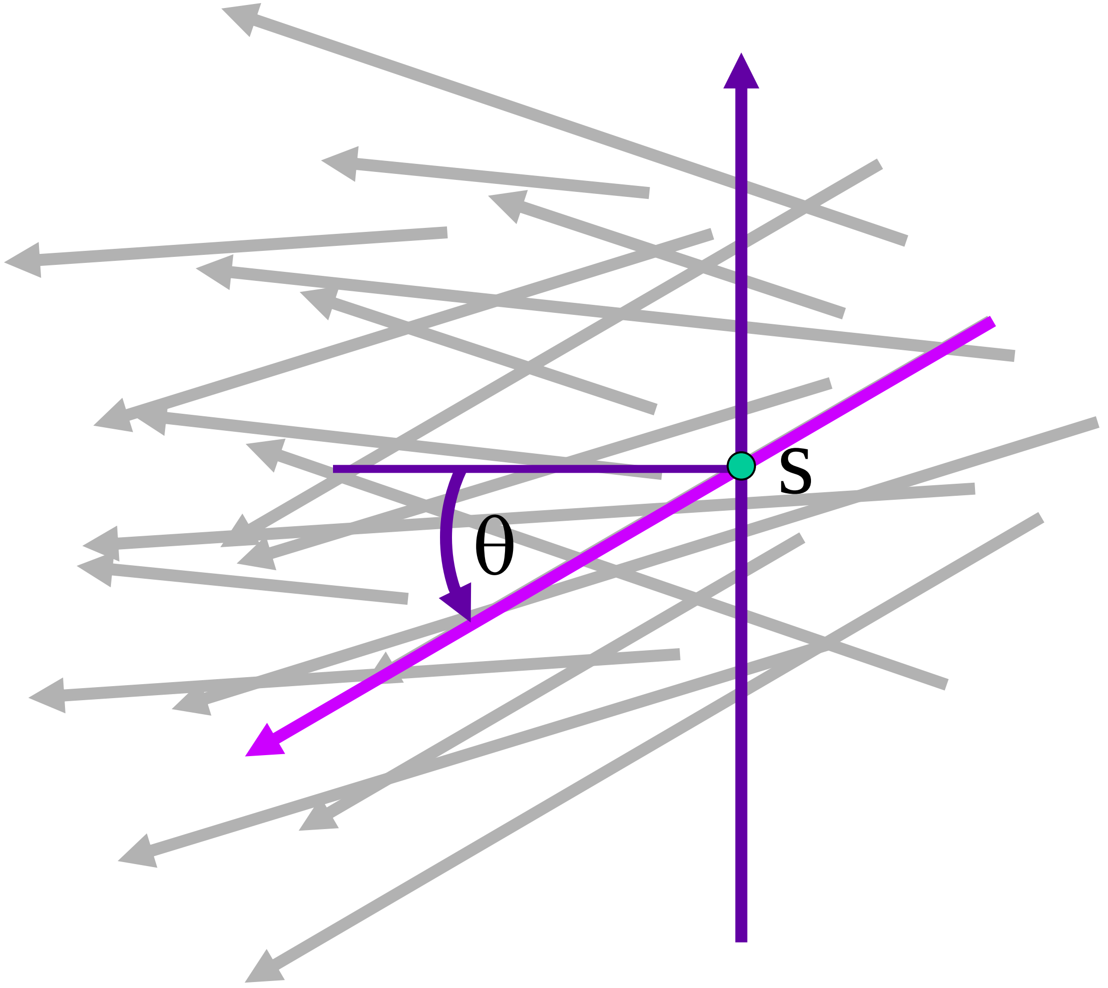
- organize
- capture
- render



# Lumigraph - Organization

2D position

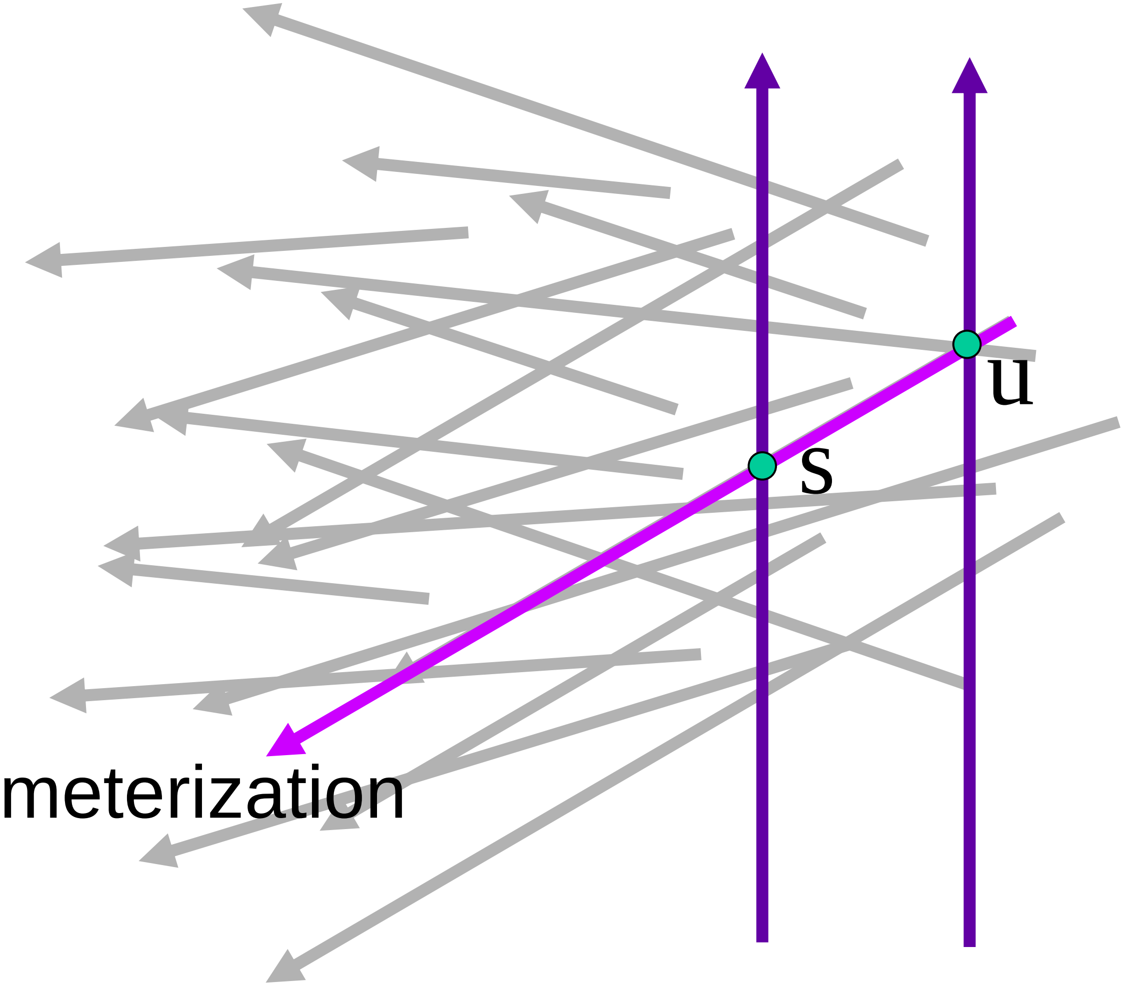
2D direction



# Lumigraph - Organization

2D position

2D position

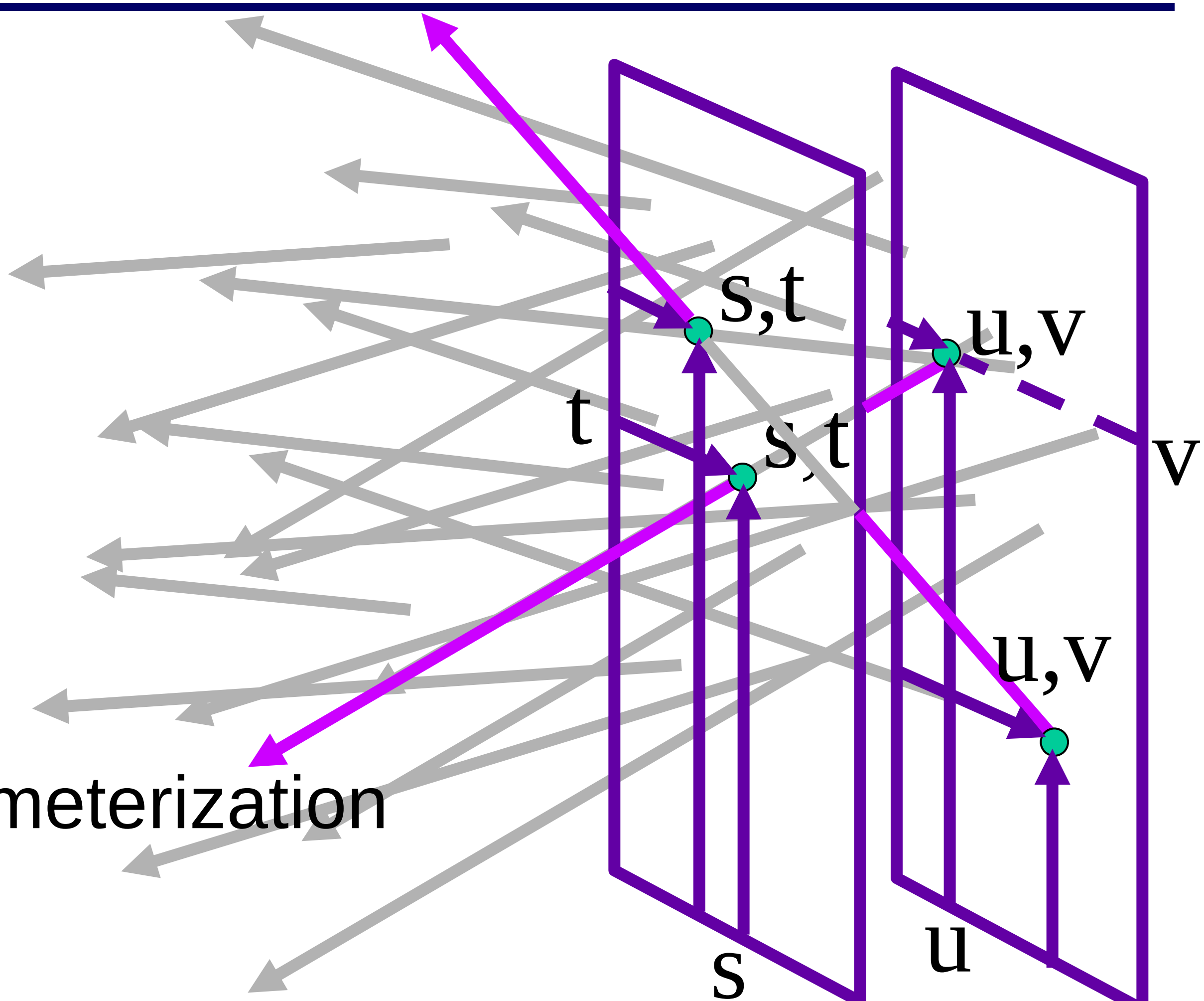



2 plane parameterization

# Lumigraph - Organization

2D position

2D position



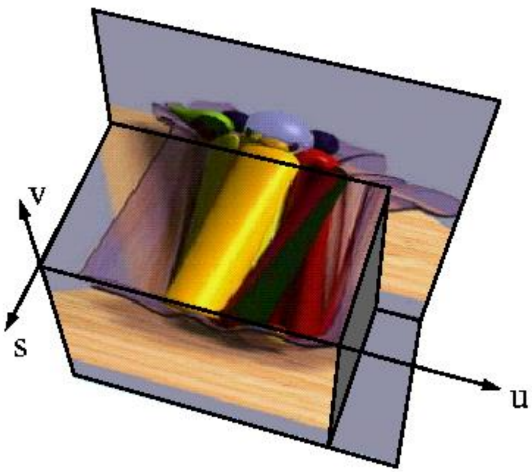

2 plane parameterization

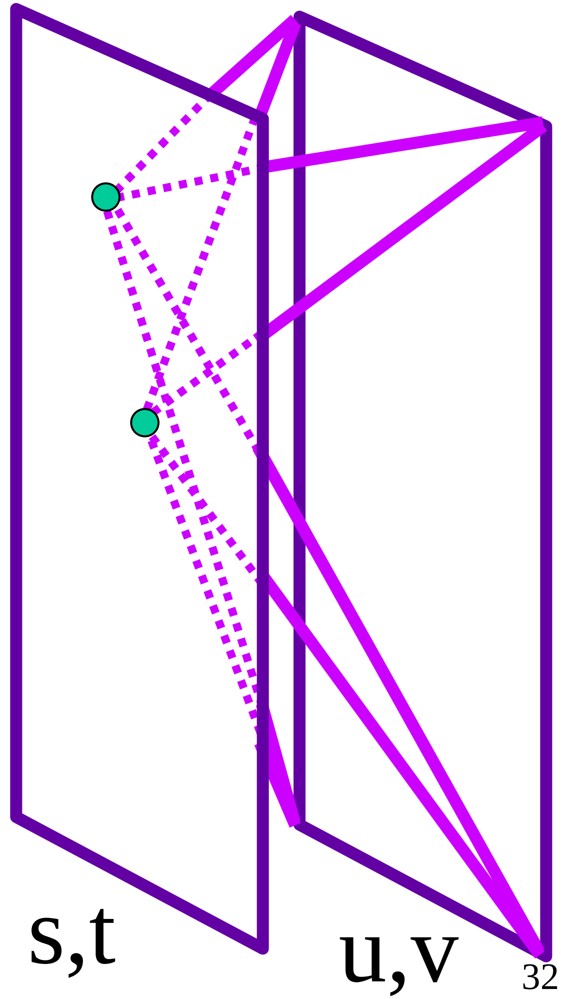
# Lumigraph - Organization

Hold s,t constant

Let u,v vary

An image

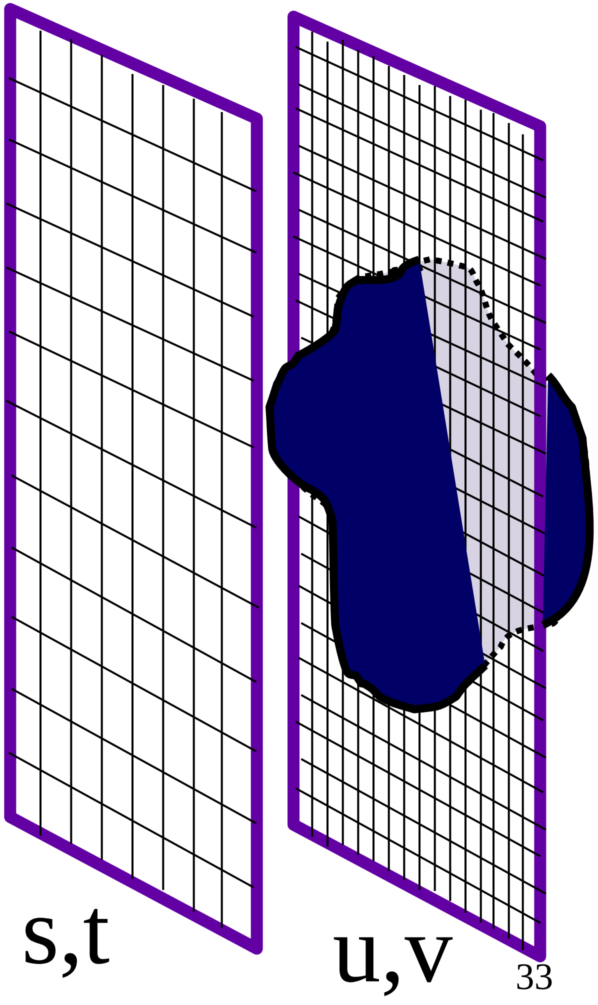


# Lumigraph - Organization

Discretization

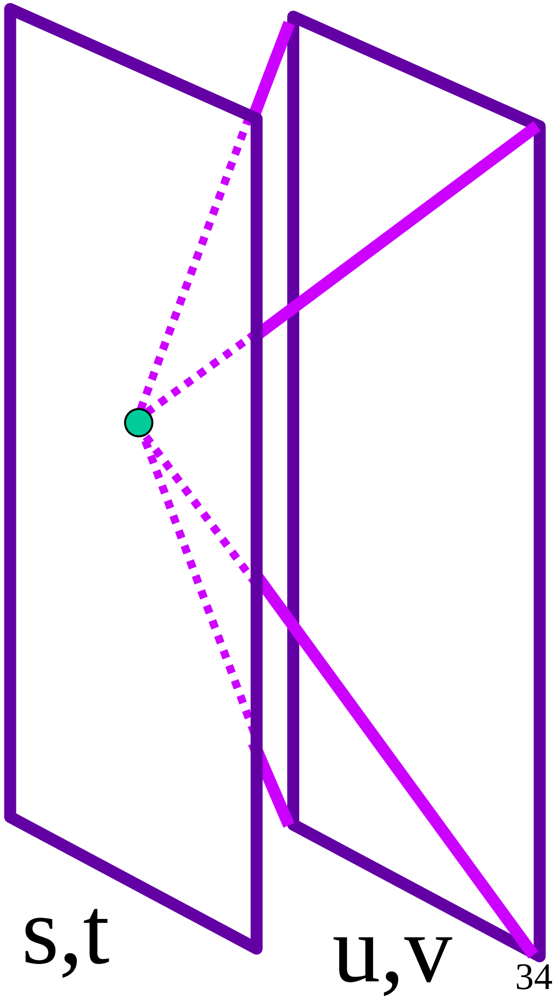
- higher res near object
  - if diffuse
  - captures texture
- lower res away
  - captures directions



# Lumigraph - Capture

Idea 1

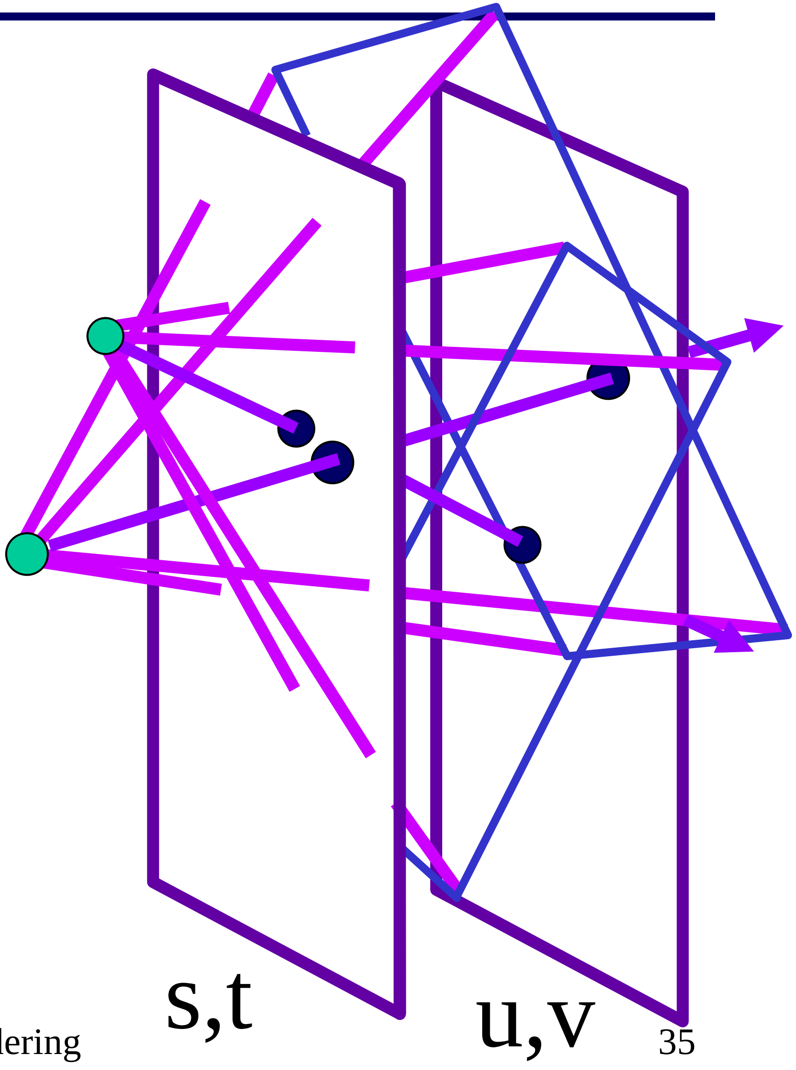
- Move camera carefully over s,t plane
- Gantry
  - see Light Field paper



# Lumigraph - Capture

Idea 2

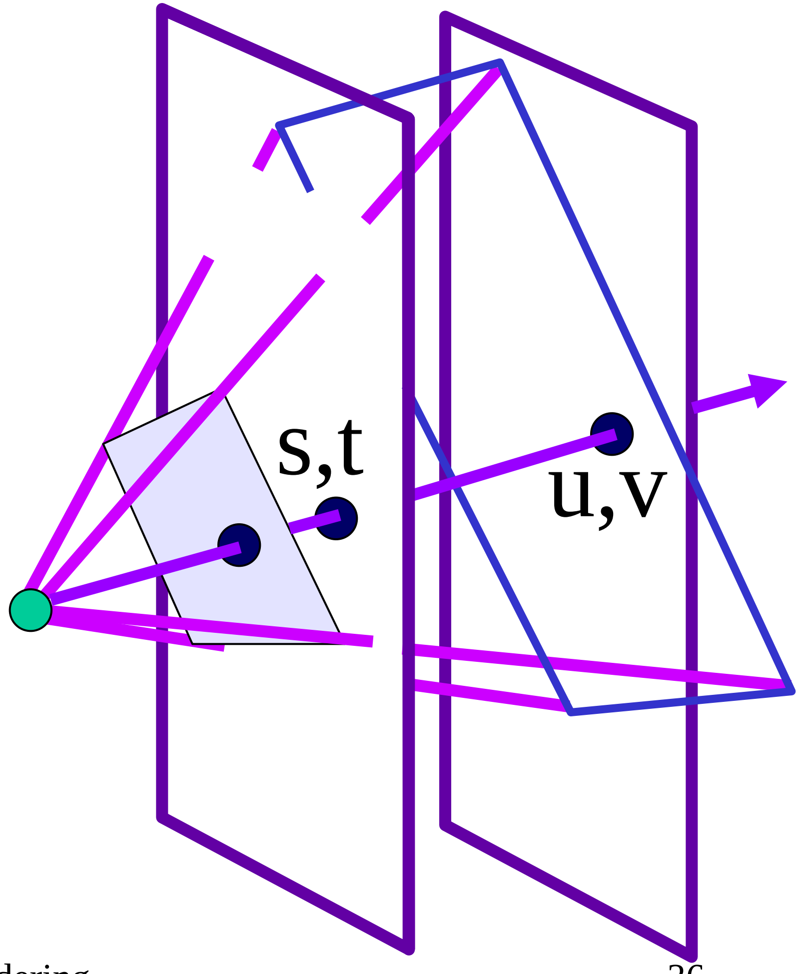
- Move camera anywhere
- Rebinning
  - see Lumigraph paper



# Lumigraph - Rendering

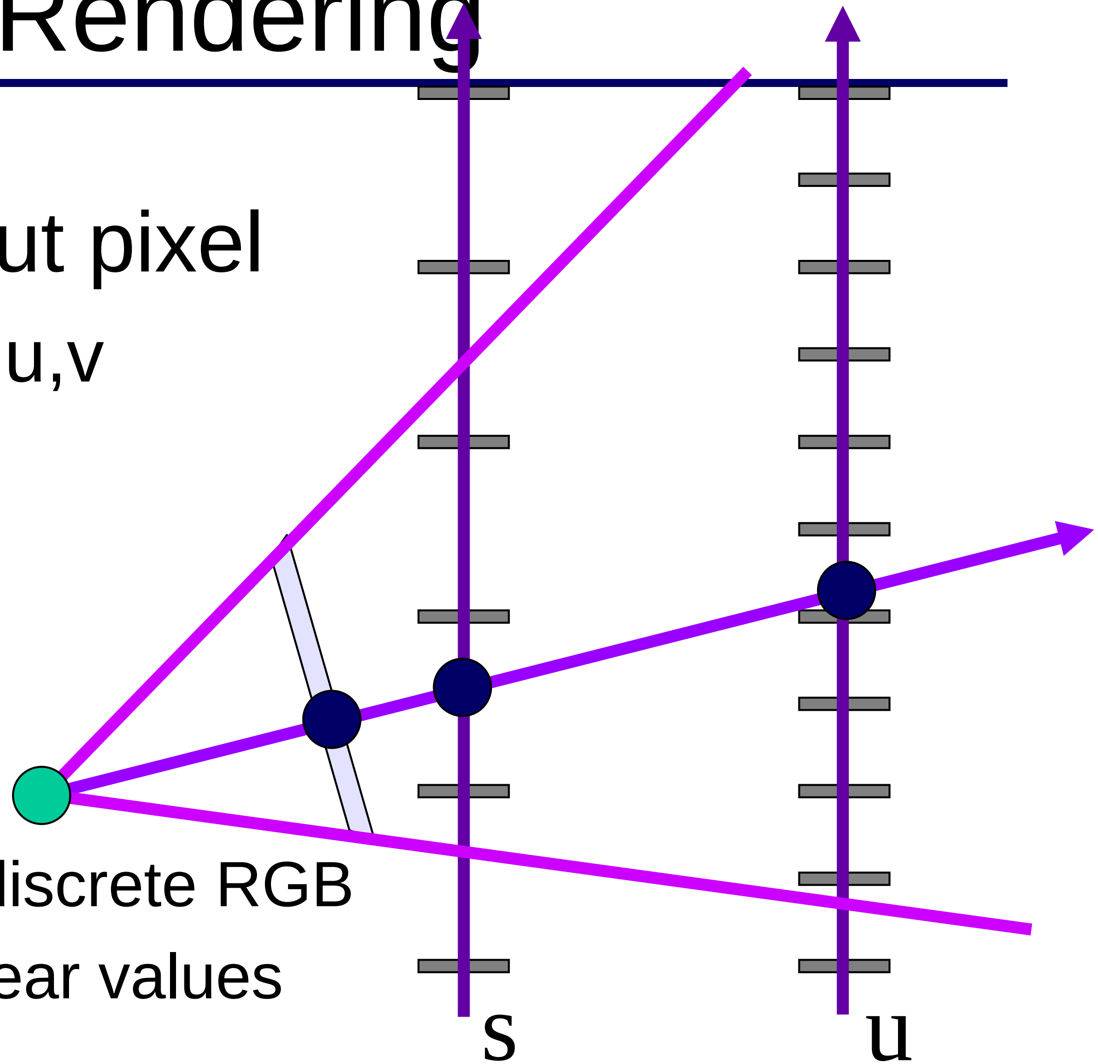
For each output pixel

- determine s,t,u,v
- either
  - find closest discrete RGB
  - interpolate near values



# Lumigraph - Rendering

- For each output pixel
  - determine s,t,u,v
  - either
    - use closest discrete RGB
    - interpolate near values



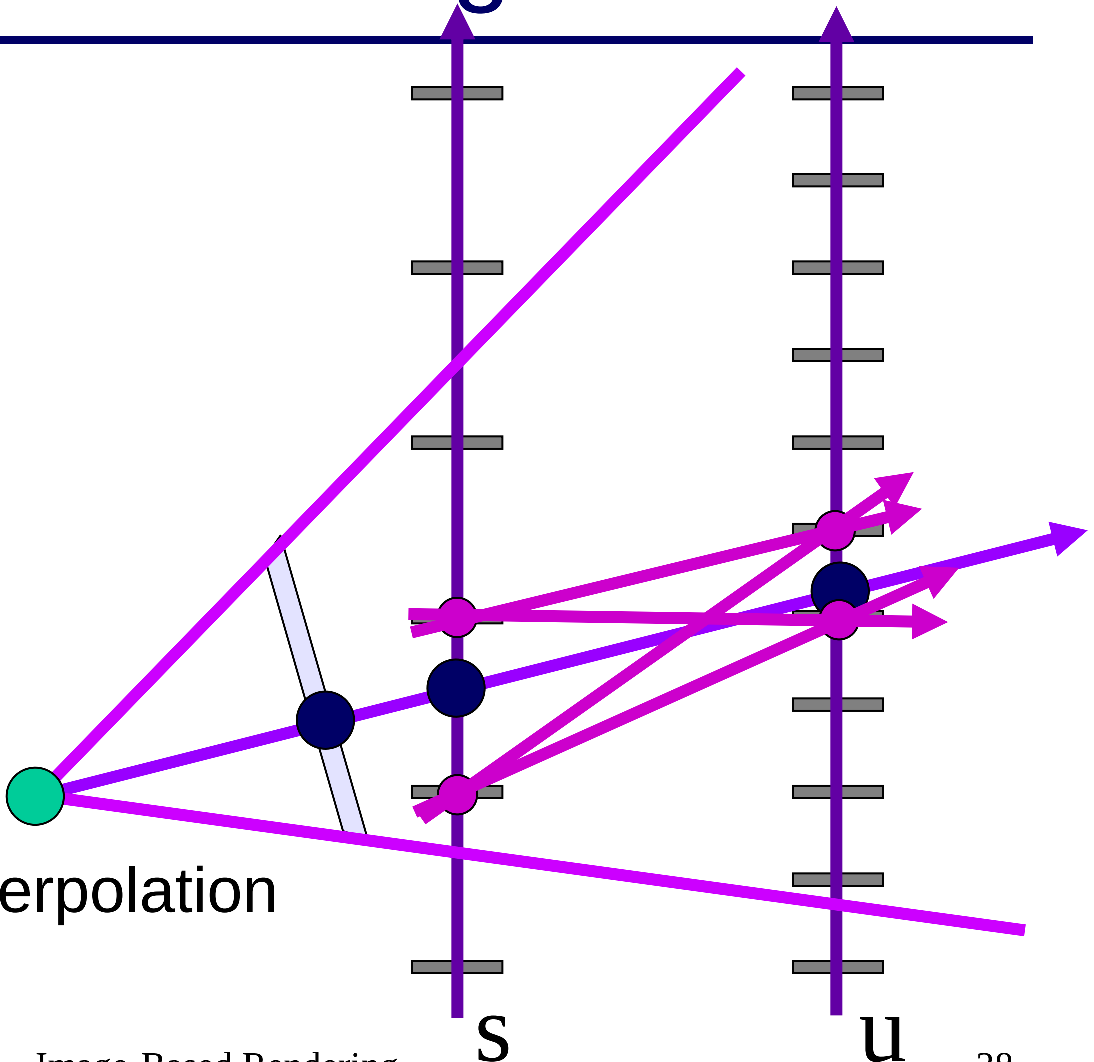
# Lumigraph - Rendering

Nearest

- closest s
- closest u
- draw it

Blend 16 nearest

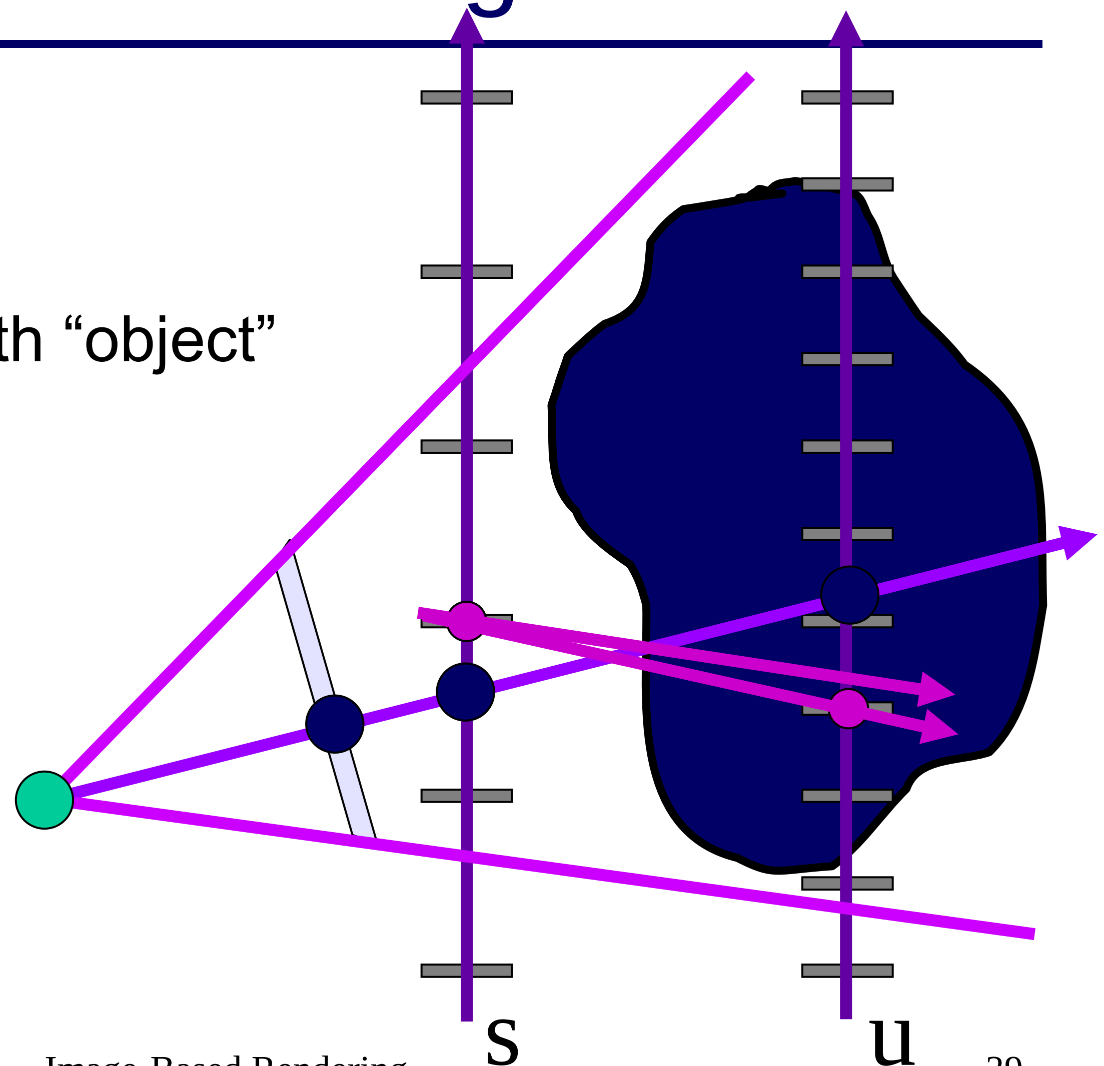
- quadrilinear interpolation



# Lumigraph - Rendering

Depth Correction

- closest s
- intersection with "object"
- best u
- closest u

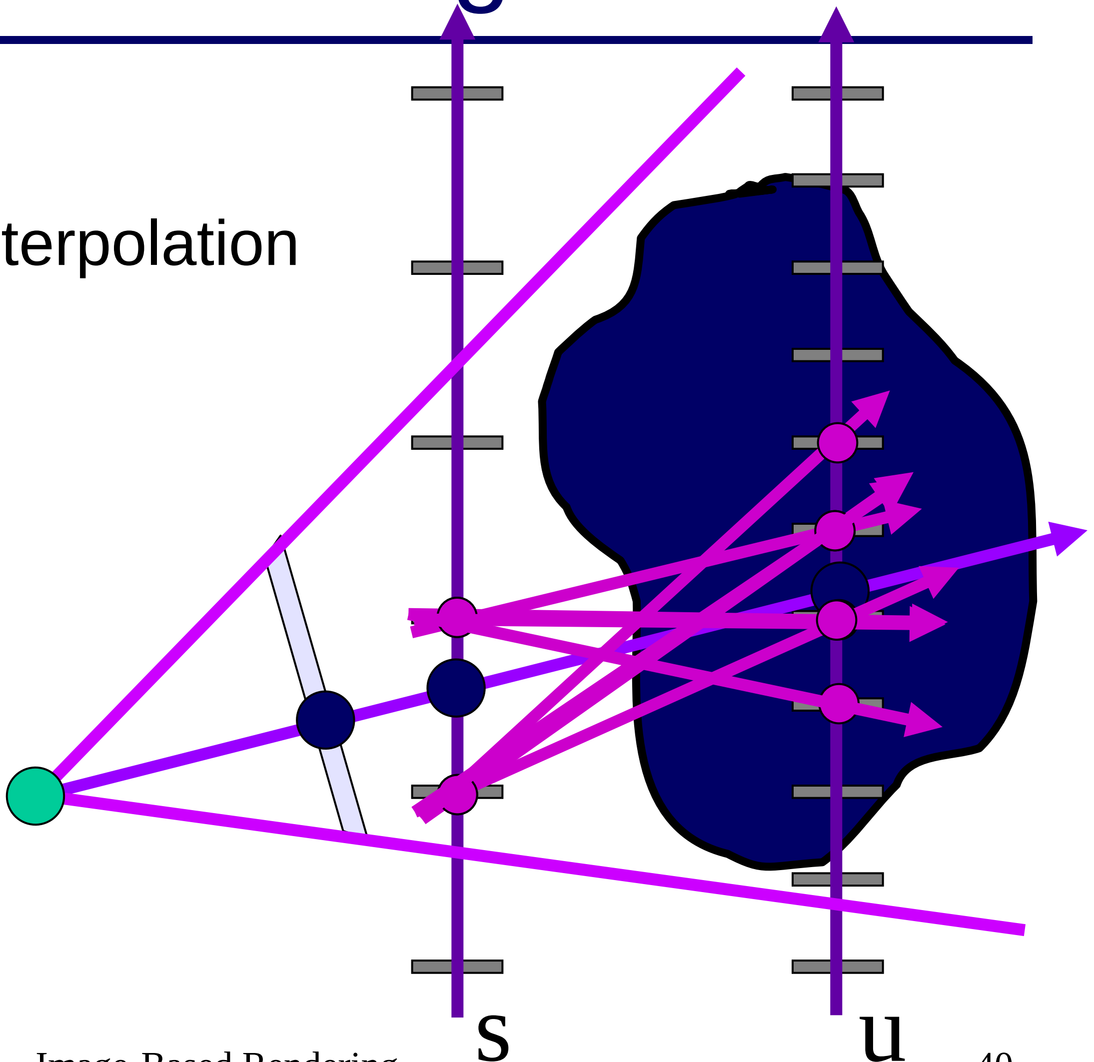


# Lumigraph - Rendering

Depth Correction

- quadralinear interpolation
- new “closest”
- like focus

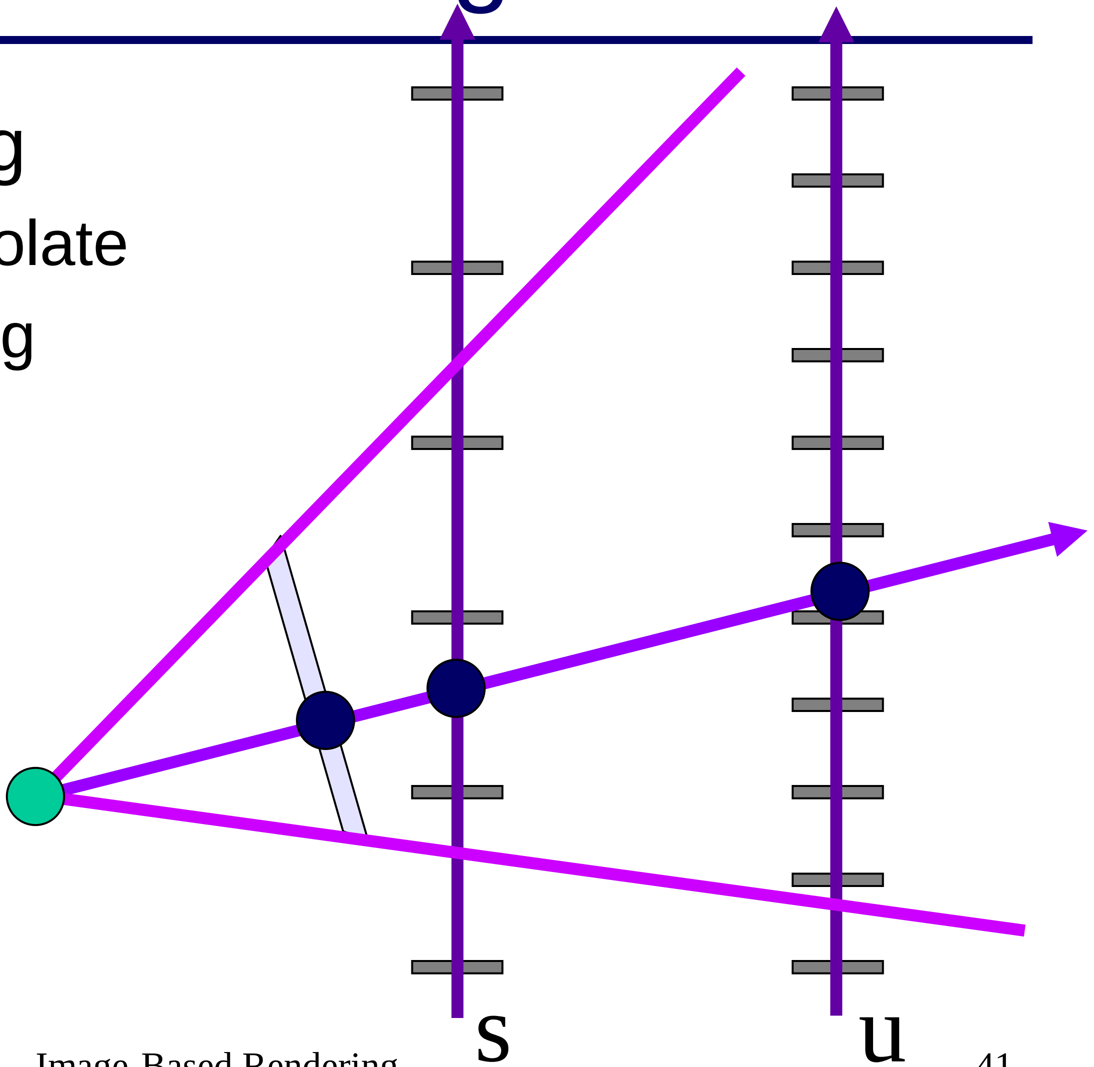
[Dynamically
Reparameterized
Light Fields,
Isaksen, SG’2000]



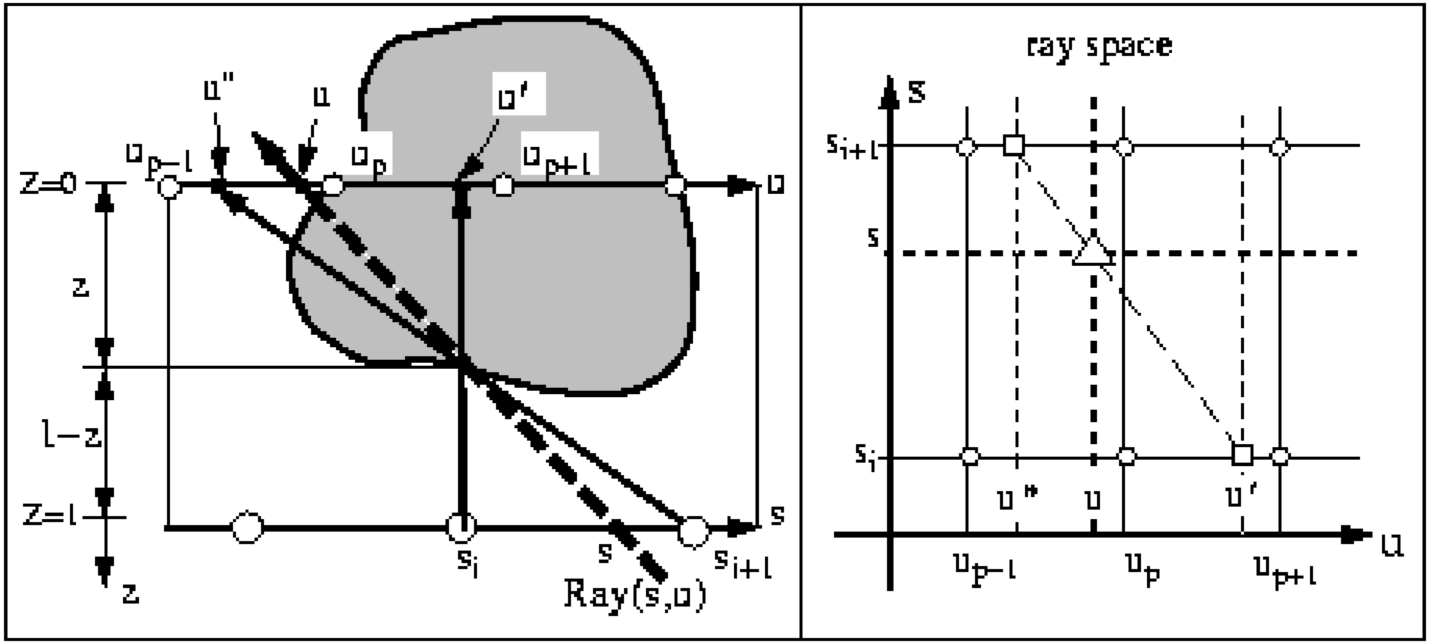
# Lumigraph - Rendering

Fast s,t,u,v finding

- scanline interpolate
- texture mapping
- shear warp



# Lumigraph - Ray Space



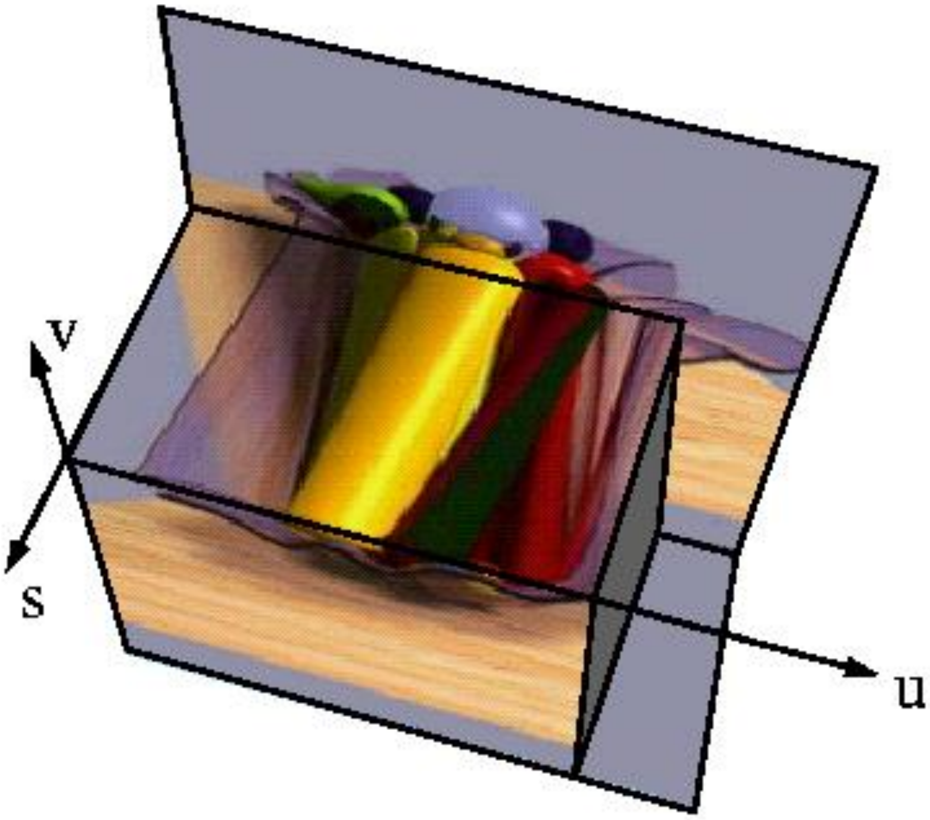

3D space ray space

surface depth ⇔ slope in ray space

# Lumigraph - Ray Space

Image effects:

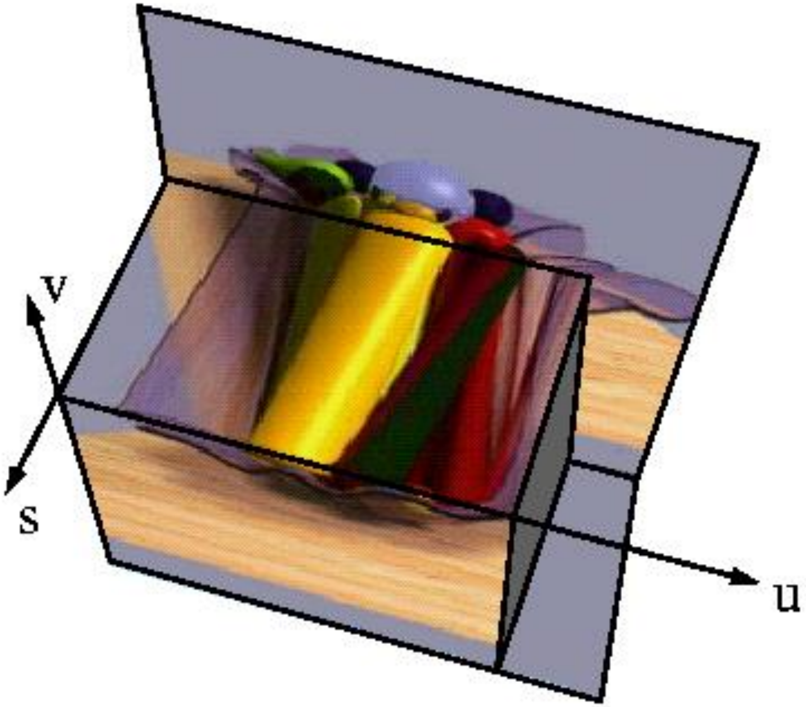
- parallax
- occlusion
- transparency
- highlights



# Lumigraph - Demo

## Lumigraph

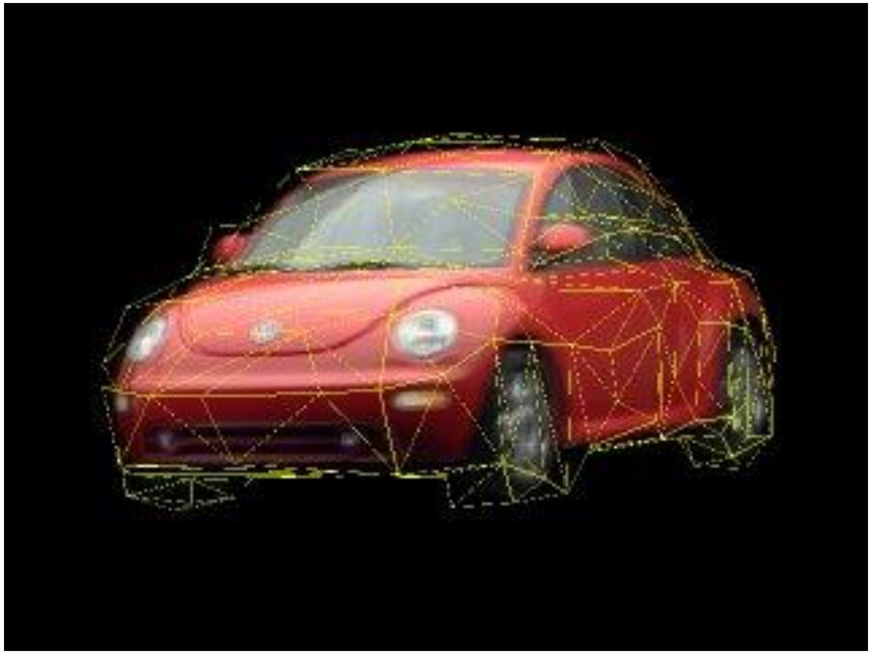
- Lion, Fruit Bowl



# Unstructured Lumigraph

What if the images aren't sampled on a regular 2D grid?

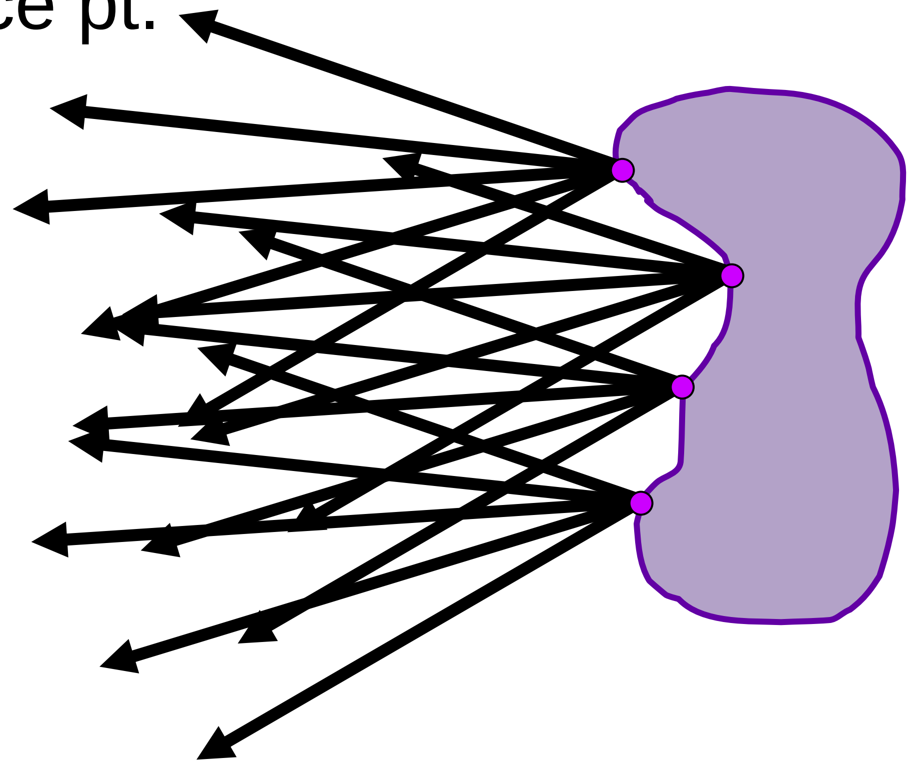
- can still re-sample rays
- ray weighting becomes more complex [Buehler *et al.*, SIGGRAPH'2000]



# Surface Light Fields

Turn 4D parameterization around:

- image @ every surface pt.



Leverage coherence:

- compress radiance fn (BRDF * illumination) after rotation by ***n***

# Surface Light Fields

[Wood et al, SIGGRAPH 2000]

# 3D Representations

Image (and panoramas) are 2D

Lumigraph is 4D

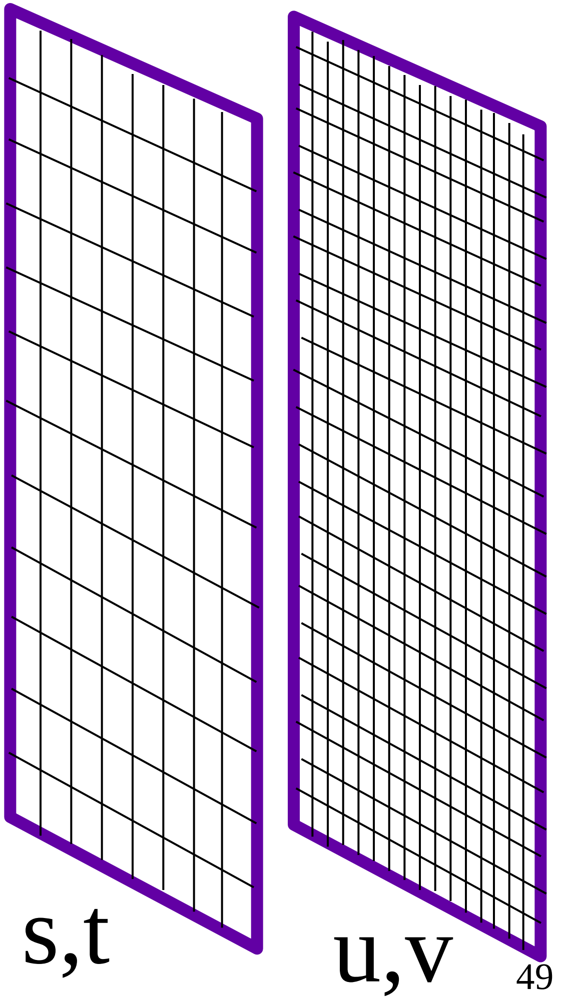
What happened to 3D?

- 3D Lumigraph subset
- Concentric mosaics

# 3D Lumigraph

One row of s,t plane

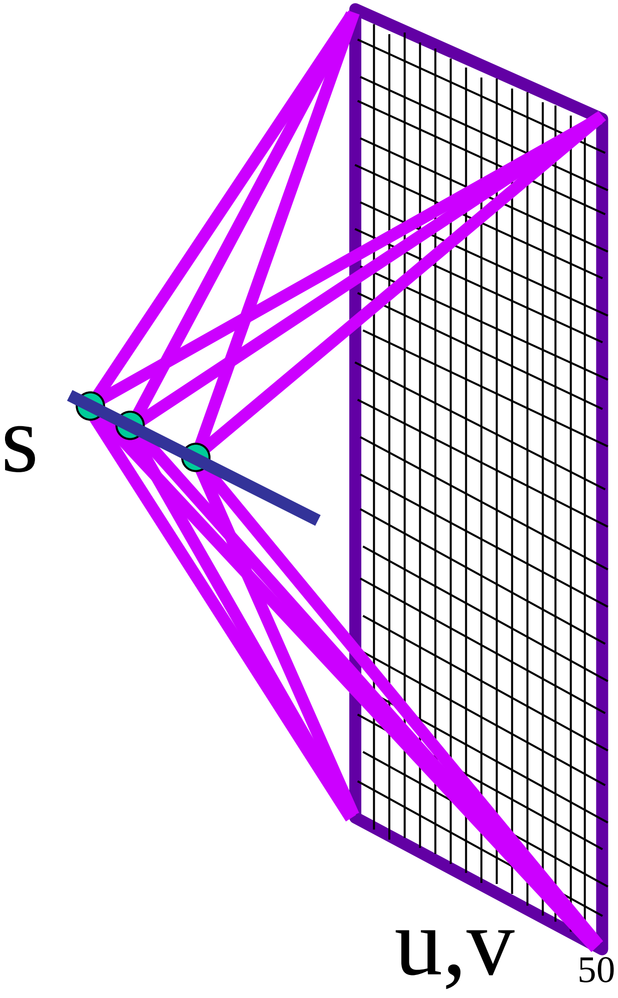
- i.e., hold t constant



# 3D Lumigraph

One row of s,t plane

- i.e., hold t constant
- thus s,u,v
- a “row of images”

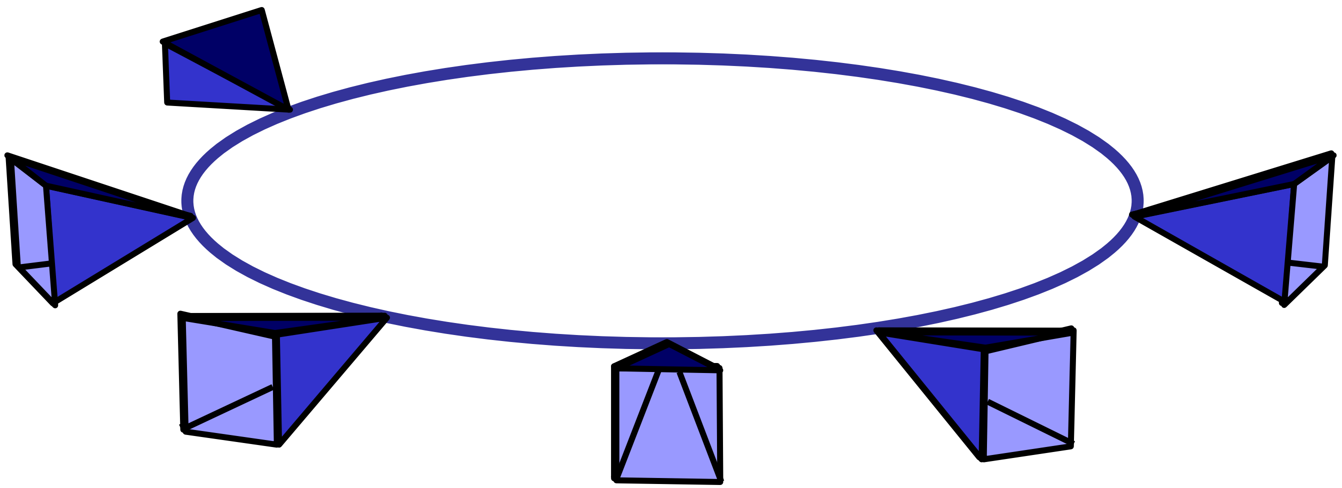



[Sloan *et al.*, Symp. I3DG 97]

# Concentric Mosaics

Replace “row” with “circle” of images

[Shum & He, SIGGRAPH’97]

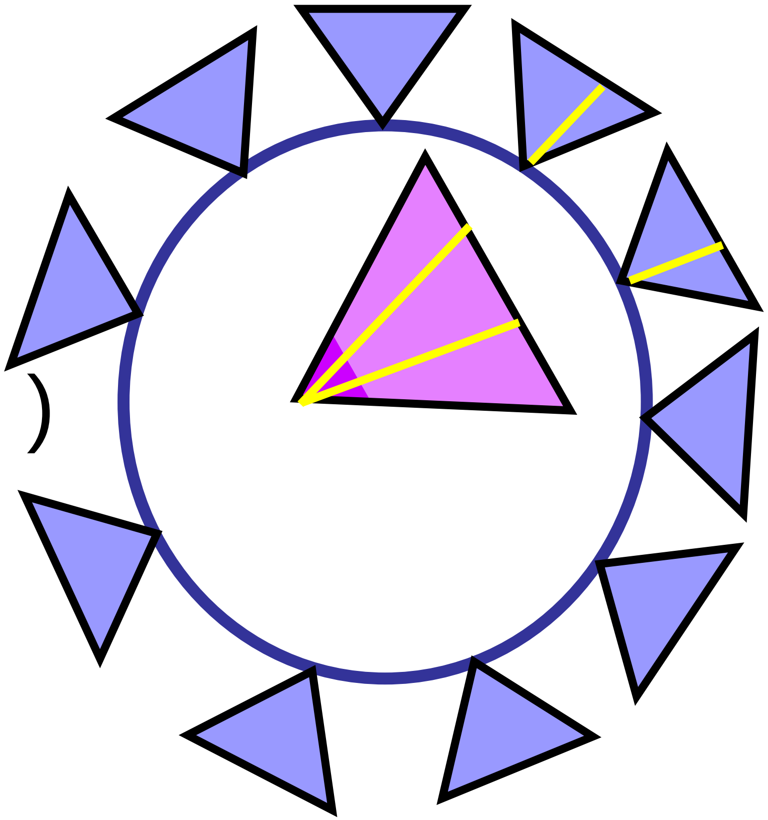


# Concentric Mosaics

# Concentric Mosaics

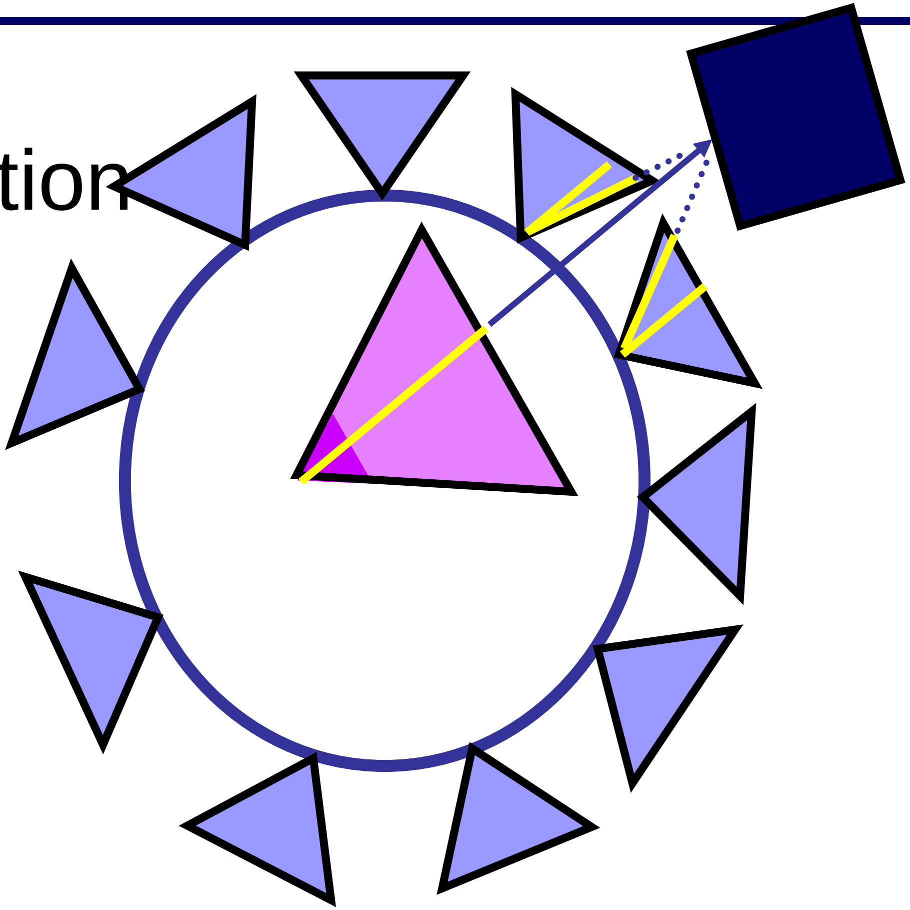
Rendering

( as seen
from above )



# Concentric Mosaics

Depth correction



# Concentric Mosaics

- *Demo*

# 2.5D Representations

Image is 2D

Lumigraph is 4D

3D

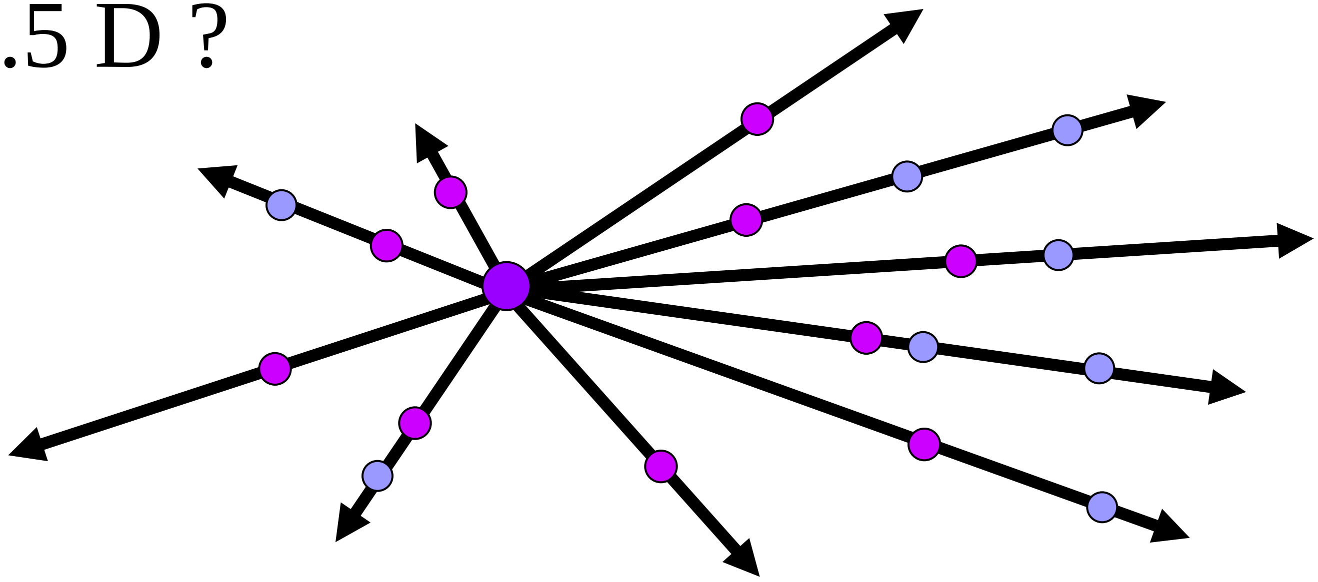
- 3D Lumigraph subset
- Concentric mosaics

2.5D

- Layered Depth Images
- Sprites with Depth (impostors)
- View Dependent Surfaces (see Façade)

# Layered Depth Image

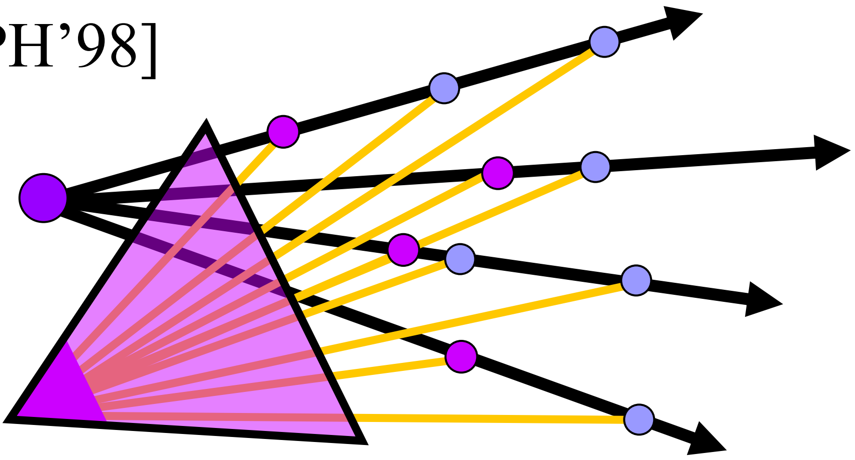
2.5 D ?



Layered Depth Image

# Layered Depth Image

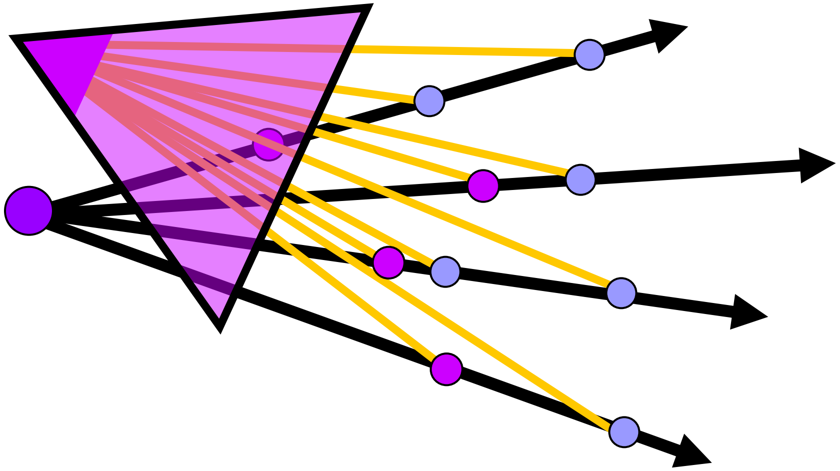
- Rendering from LDI
  [Shade *et al.*, SIGGRAPH'98]



- Incremental in LDI X and Y
- Guaranteed to be in back-to-front order

# Layered Depth Image

- Rendering from LDI



- Incremental in LDI X and Y
- Guaranteed to be in back-to-front order

# Sprites with Depth

Represent scene as collection of cutouts with depth (planes + parallax)

Render back to front with fwd/inverse warping [Shade *et al.*, SIGGRAPH'98]

# Environment matting and compositing

D. E. Zongker, D. M. Werner,
B. Curless and D. H. Salesin.
SIGGRAPH'99

# Environment Matting

Capture the *reflections* and *refractions* of a real-world object

Composite object over a novel background

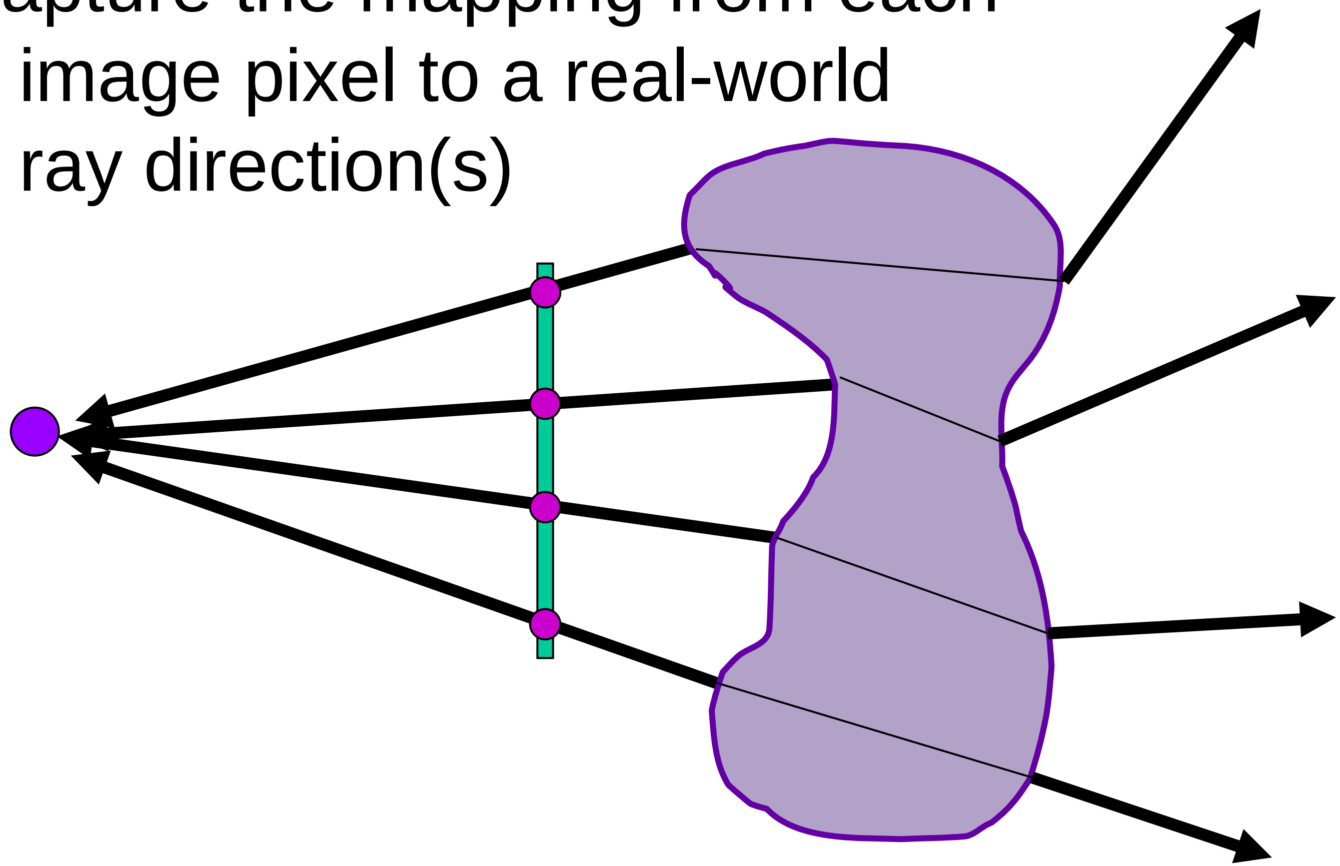
# Environment Matting - examples

**Figure 1** A water goblet, digitally composited onto background images, preserving the effects of refraction.

**Movie**

# Environment Matting

Capture the mapping from each image pixel to a real-world ray direction(s)



# Acquisition setup

Use several monitors with stripes

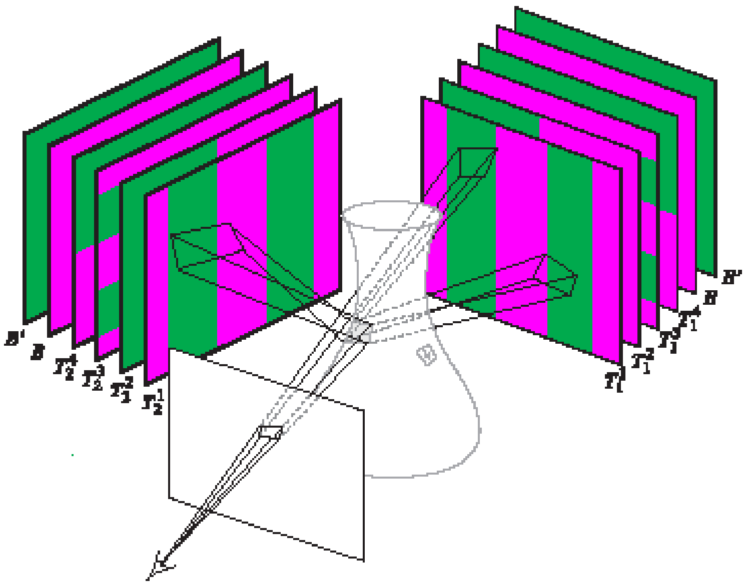
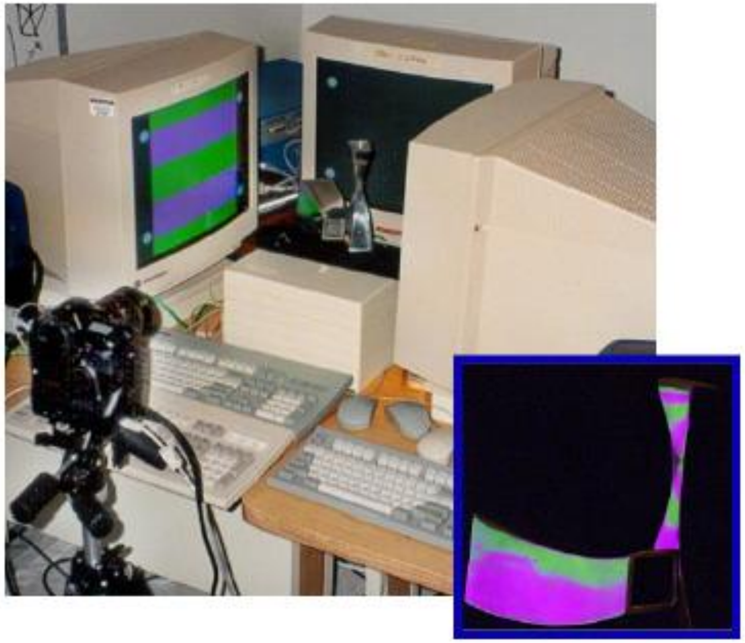


**Figure 2** The environment matting process uses structured textures to capture how light is reflected and refracted from a backdrop (right shaft), as well as from various sidedrops (left shaft). The process also captures light coming from the backdrop that is seen through uncovered portions of a pixel (center shaft).



# Environment matting equation

Captures foreground color
and background directions

$$C = F + (1 - \alpha) B + \sum_{i=1}^{m} R_i \mathcal{M}(T_i, A_i)$$

Most standard texture-mapping methods actually compute the *average* value of an axis-aligned region of a texture, so we'll let $R_i = K_i A_i$. Letting $\mathcal{M}(T, A)$ be a texture-mapping operator that returns the average value of an axis-aligned region $A$ of the texture $T$,

# Environment matting - result

**Figure 4** From left to right: an alpha matte composite, an environment matte composite, and a photograph of an object in front of a background image. The top row shows a ribbed glass candle holder; the bottom row shows a rough-surfaced glass bookend.

# Environment matting extensions

[Chuang *et al.*, SIGGRAPH'2001]

- accurate (multiple refractions):
  - soft stripes
- fast (video rate):
  - color ramp

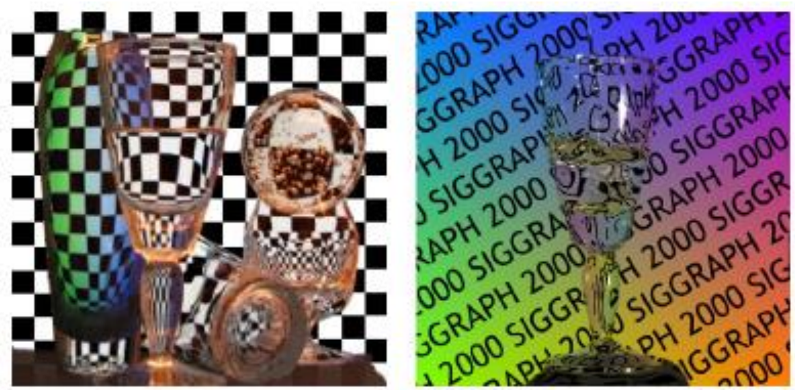



**Figure 1** Sample composite images constructed with the techniques of this paper: slow but accurate on the left, and a more restricted example acquired at video rates on the right.

# Image-Based Modeling

# Image Based Models

Modeling and Rendering Architecture from Photographs

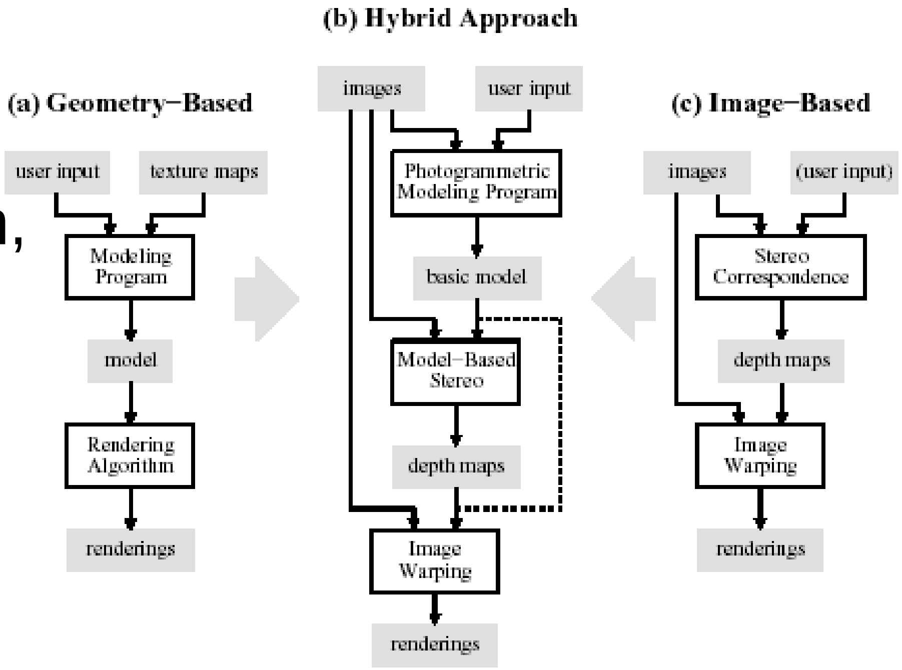
Debevec, Taylor, and Malik 1996

Original photograph with marked edges | Recovered model | Model edges projected onto photograph | Synthetic rendering

# Image-Based Modeling

Create 3D model (and texture maps) from images

- automated
  - (structure from motion, stereo)
- interactive
  - Façade system



# Façade

1. Select building blocks
2. Align them in each image
3. Solve for camera pose and block parameters (using constraints)

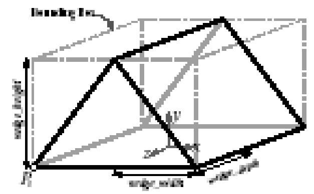
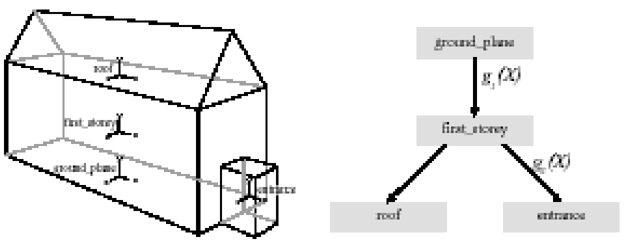


Figure 3: *A wedge block with its parameters and bounding box.*

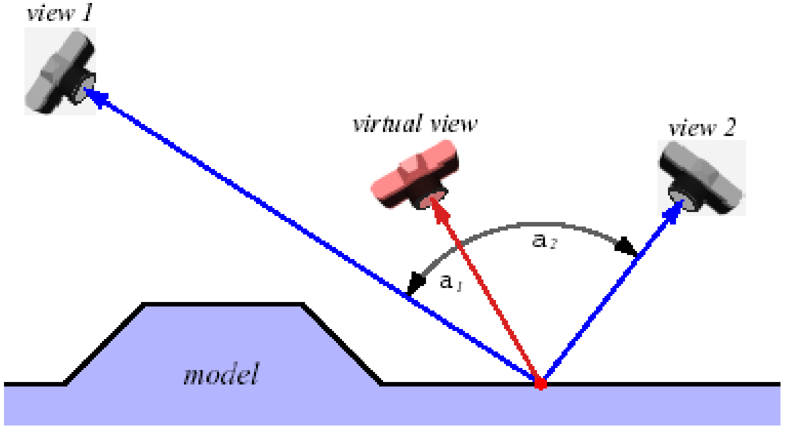


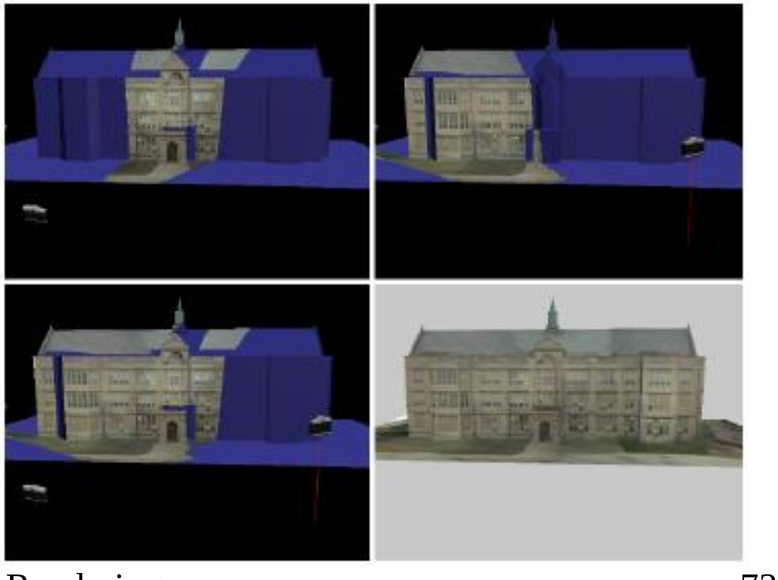

(a) (b)

Figure 4: **(a)** *A geometric model of a simple building.* **(b)** *The model's hierarchical representation. The nodes in the tree represent parametric primitives (called* blocks*) while the links contain the spatial relationships between the blocks.*

# View-dependent texture mapping

1. Determine visible cameras for each surface element
2. Blend textures (images) depending on distance between original camera and novel viewpoint



# Model-based stereo

Compute offset from block model

(a) *Key Image* (b) *Warped Offset Image* (c) *Offset Image* (d) *Computed Disparity Map*

Some more results:

# Image-Based Faces

Estimate shape from images

Match metrics to shape

Project video onto shape

- Texture map

Animate

[Z. Liu *et al.*, MSR-TR-2000-11]

# Hierarchy of Light Fields [Levoy]

8D: Refractive/reflective environment

5D: Plenoptic Function (Ray)

4D: Lumigraph / Lightfield

4D*: Environment Matte (single view)

3D: Lumigraph Subset

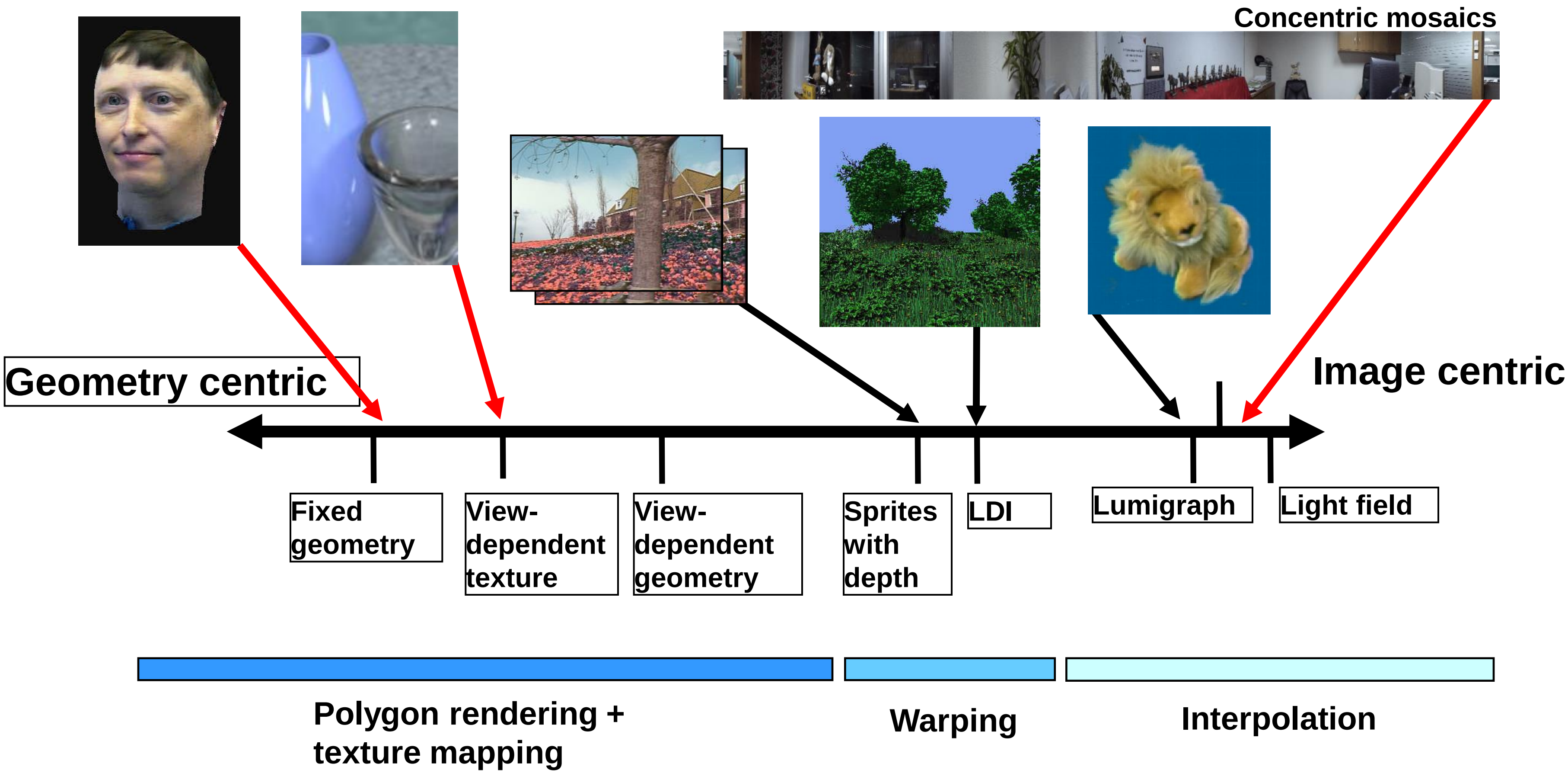
3D: Concentric Mosaics

2.5D: Layered Depth Image

2.5D: Image Based Models

2D: Images and Panoramas

# Graphics/Imaging Continuum



# What lies *beyond* Image-Based Rendering?

# Video-Based Rendering

Image-Based Rendering:

- render from (real-world) images for efficiency, quality, and photo-realism

Video-Based Rendering

- use *video* instead of still images for dynamic elements and source footage
- generate *computer video* instead of *computer graphics*

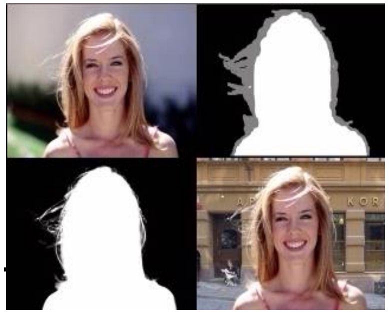
# VBR Examples

Facial animation

- Video Rewrite, …

Layer/matte extraction

- Video Matting, …



Dynamic (stochastic) elements

- Video Textures, …

3-D world navigation

- Image-Based Realities, …

# Facial animation

Modeling from still images

- [Pighin *et al.*, SG'98]

Lip-synching from video

- Video Rewrite
  [Bregler *et al.*, SG'97]
- [Ezzat *et al.*, SG'02]

# Matting and Compositing

# Digital Matting and Compositing


**http://grail.cs.washington.edu/projects/digital-matting/**

## Overview

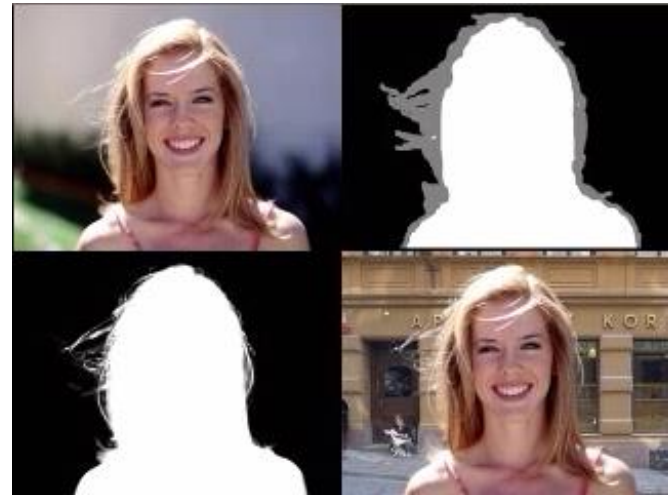
Matting and compositing are important operations in the production of special effects. These techniques enable directors to embed actors in a world that exists only in imagination, or to revive creatures that have been extinct for millions of years. During matting, foreground elements are extracted from a film or video sequence. During compositing, the extracted foreground elements are placed over novel background images.

Traditional approaches to matting include *blue-screen matting* and *rotoscoping*. The former requires filming in front of an expensive blue screen under carefully controlled lighting, and the latter demands talent and intensive user interaction. Our Bayesian matting algorithm (CVPR 2001) can pull alpha mattes of complex shapes from natural images. In the "video matting" paper (SIGGRAPH 2002), we extended this approach to video by interpolating user-drawn keyframes using optical flow. A novel technique for smoke matte extraction is also demonstrated.

Traditional compositing operations can only model color blending effects like anti-aliasing, motion blur and transparency. This model, however, can't model reflections, refractions and shadows. In SIGGRAPH 1999, we introduced environment matting which can captures how a foreground object refracts and reflects light. The foreground object can then be placed in a new environment using environment compositing, where it will refract and reflect light from that scene. We later developed more sophisticated sampling schemes to capture mattes with higher accuracies, and techniques requiring fewer images, to allow for real-time capture. Shadows are yet another effects that traditional approaches fail to model correctly. In SIGGRAPH 2003, we introduced a novel process called "shadow matting and compositing" to acquire the photometric and geometric properties of the background for making realistic shadow composites.

# Video Matting

Pull dynamic $\alpha$-matte from video with complex backgrounds



[Chuang *et al.* @ UW, SIGGRAPH’2002]

# Video Matting

# Shadow Matting

Transfer a shadow from one background to another:

- Extract and model photometry (darkening)
- Extract and model geometry (deformation)

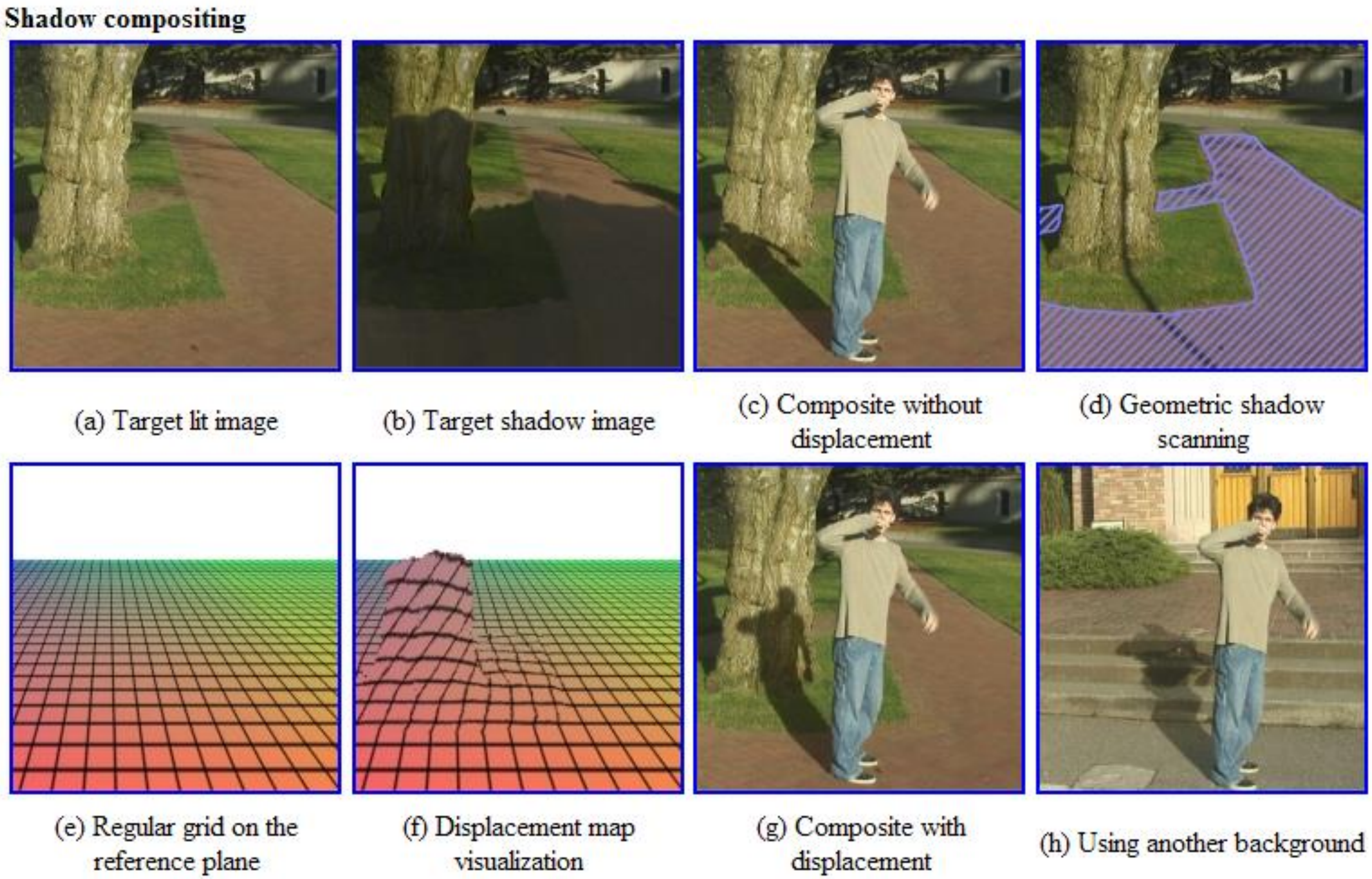
# Shadow Matting

**Shadow matting**

(a) Source frame (b) Source lit image (c) Source shadow image (d) Recovered beta

**Shadow matting.** Starting from a source image sequence (one frame shown in (a)), we first remove the foreground character using video matting. Our shadow matting algorithm recovers lit and shadow images (b,c) using max/min compositing. It then estimates â by projecting observed pixel colors onto the color lines between them (d).

# Shadow Matting



# Shadow Matting

Video

# Video Textures

# Video Textures

How can we turn a short *video clip*
into an ∞ amount of continuous video?

- dynamic elements in 3D games and presentations
- alternative to 3D graphics animation?

[Schödl, Szeliski, Salesin, Essa, SG'2000]

# Video Textures

Find cyclic structure in the video

(Optional) region-based analysis

Play frames with random shuffle

Smooth over discontinuities (morph)

# Region-based analysis

# Crossfading and morphing

Jump Cut　　Crossfade　　Morph

# Video portrait

# Dynamic scene element

Live waterfall in static panorama

# Interactive fish

# A complete animation

# Video-Based Tours

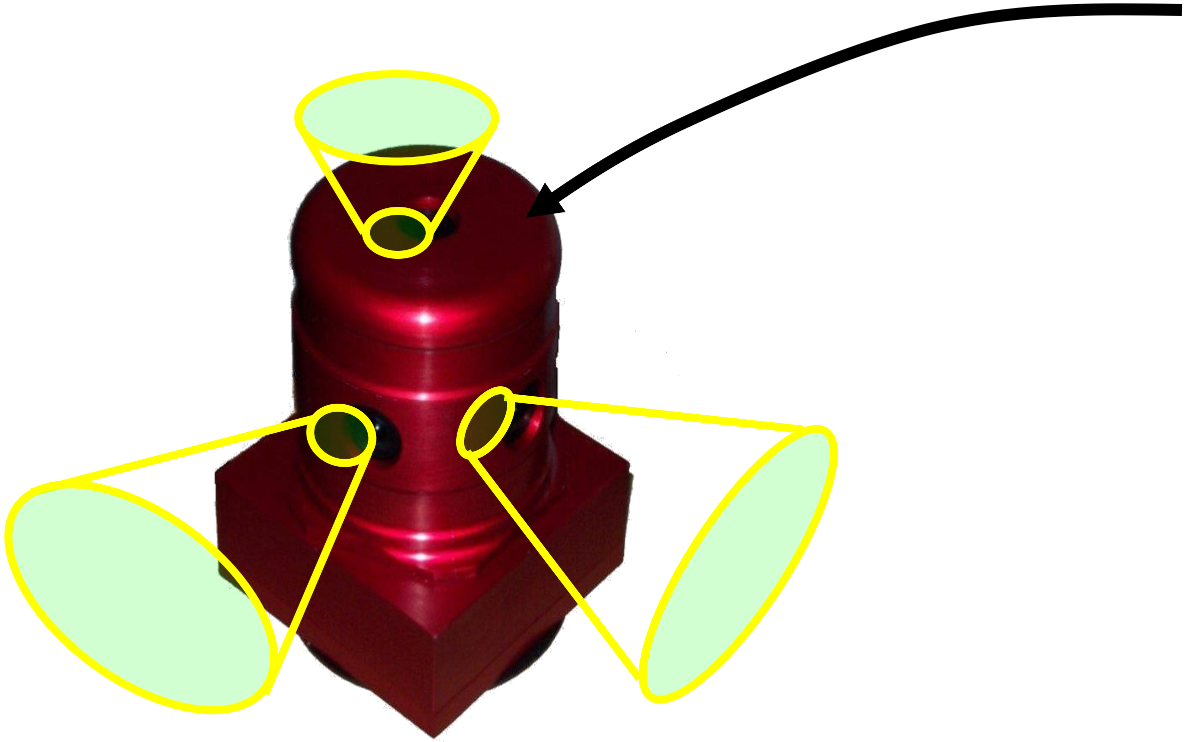
# Video-Based Walkthroughs

Move camera along a rail ("dolly track") and play back a 360° video

Applications:

- Homes and architecture
- Outdoor locations (tourist destinations)

# Surround video acquisition system



***OmniCam (six-camera head)***

# OmniCam

Built by Point Grey Research (*Ladybug*)

Six camera head

Portable hard drives, fiber-optic link

Resolution per image: 1024 x 768

FOV: ~100º x ~80º

Acquisition speed: 15 fps uncompressed

# Acquisition platforms

Robotic cart

Wearable

# Open issues

How to best sample and interpolate Light Field

- (sub-?) pixel accurate stereo
- reflections, refractions, …

Compositing

- how to insert Light Field into new environment
- relighting
- …?

# Summary

Image–Based Rendering

- Light Fields and Lumigraphs
- Panoramas and Concentric Mosaics
- Matting: natural, environment, and shadows
- Image-Based models

Video-Based Rendering

- Facial animation
- Video Textures and Animating Stills
- Video-based tours

# Summary

Image–Based Rendering
- Light Fields and Lumigraphs
- Panoramas and Concentric Mosaics
- Environment Matting
- Image-Based models

Video-Based Rendering
- Facial animation
- Video matting
- Video Textures and Animating Stills
- Video-based tours