Image-Based Rendering

- Produce a new image from real images.
- Combining images
- Interpolation
- More exotic methods

Why Image-Based Rendering?

- What's the most realistic image? A photograph.
- But photographs lack flexibility.
  - Can't change viewpoint.
  - Can't change lighting.

The need for correspondence

- Image-based rendering is mostly combining images to get a new image.
- Correspondences needed to sensibly combine images.
  - If viewpoint has changed this can be hard.
  - If not, it's trivial.

How to get correspondences

- By hand:
  - Works if few correspondences needed
- By matching intensities
  - This is really ~ ½ of computer vision.

Matching

- Simplest: SSD with windows.
  - Windows needed because pixels not informative enough.
- Compare windows

$$SSD = \sum_{[i,j] \in R} (f(i,j) - g(i,j))^2$$

f g

- Search for windows that match well

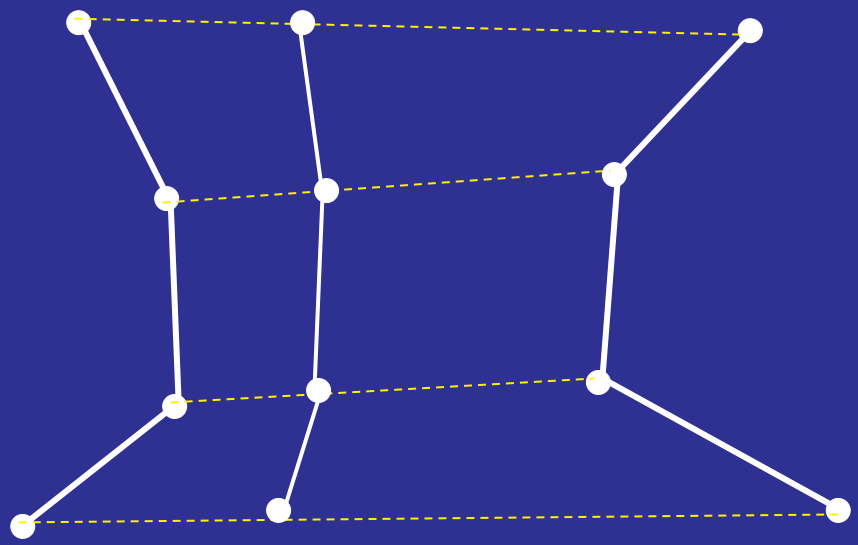
Morphing

- What happens if you interpolate images?
- Need corresponding points.

Morphing

- ***Corresponding points*** needed.
  - Often done by hand.
- Interpolate each point.
  - Position
  - and intensity.
- Also use interpolation for more correspondences.

Linear Interpolation of Position



Other Interpolation

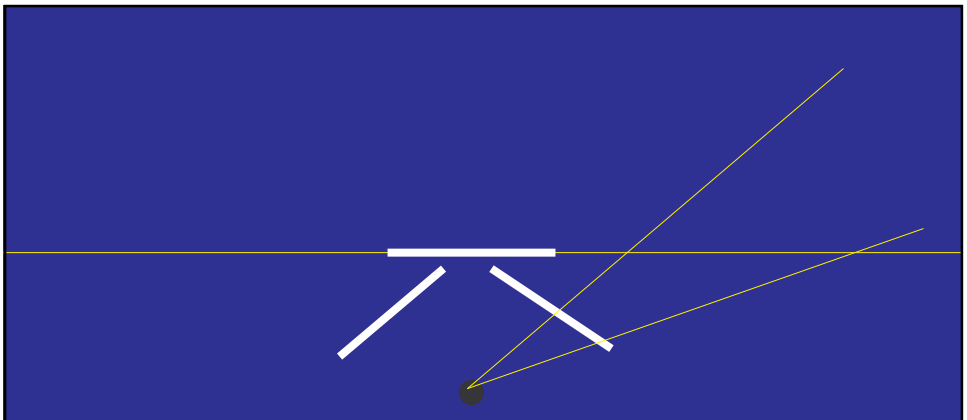
- Also interpolate intensities.
- Interpolate to find other point correspondences.

Light-Field Rendering

- Sample the set of light rays in the world.
- Then generate an image by selecting the right rays.
- Mosaicing: simpler, just sample rays through one focal point.
- If one has all rays then camera can also move.

Mosaics

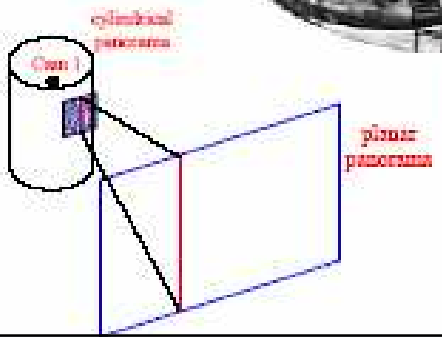
- Take multiple images and construct one big image.
  - Represented as image, cylinder or sphere.
- Allows panning and zooming.
  - Simplest kind of motion.



- Fixed focal point.
- Correspondence needed to align images.
- Image rectification

Cylindrical Panoramas for 360° Viewing

(Peleg, CVPR'99)



(a)

(b)

Figure 1: Construction of a cylindrical panorama: (a) two warped images; (b) part of cylindrical panorama composited from a sequence of images.

Figure 4: Tessellated spherical panorama covering the north pole (constructed from 54 images). The white triangles at the top are the parts of the texture map not covered in the 3D tesselated globe model (due to triangular elements at the poles).

Mosaics for Video Coding

Convert masked images into a background sprite for content-based coding

(slide from Mark Ollila)

Other mosaicing issues

- Pixel interpolation needed.
- Mosaicing can provide more information at each pixel.
  - Higher resolution images possible.
  - Higher dynamic range.

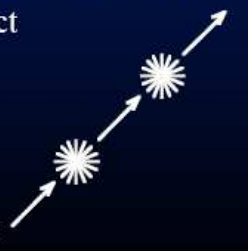
Interpolating Views

Light Fields

Gershun's and Moon's idea of a light field:

Radiance as a function of a ray or line: L (x, y, z, θ, ϕ)

- In "free space" (no occluders) 5D reduces to 4D
  - Exterior of the convex hull of an object
  - Interior of an environment
- Images are 2D slices
  - Insert acquired imagery
  - Extract image from a given viewpoint



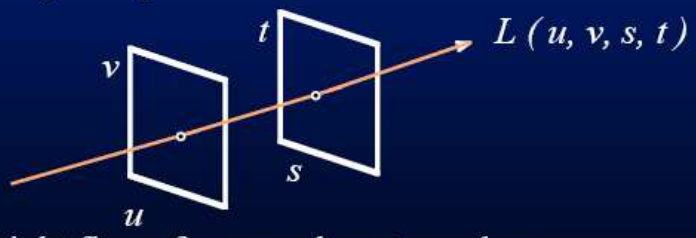
Light Field Rendering (Levoy and Hanrahan; paper and slides)

4D Light Field

Light Field Rendering (Levoy and Hanrahan; paper and slides)

Light Slab or Beam

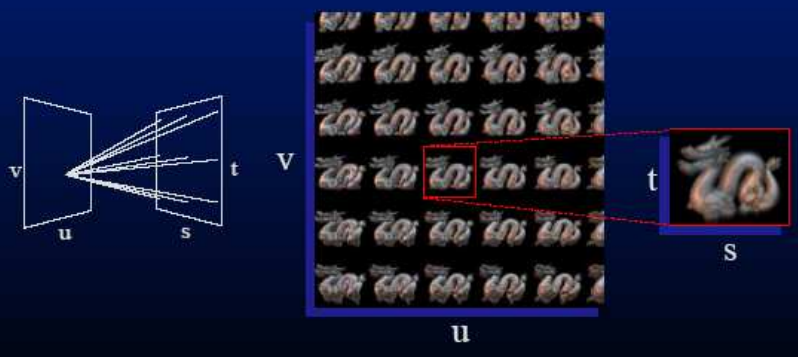
Two-plane parameterization



- Light flows from *uv* plane to *st* plane
- Planes in any position
- Parameters *uv* and *st* between 0 and 1

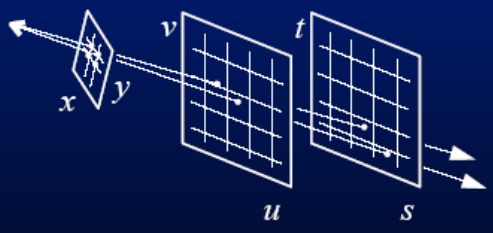
Light Field Rendering (Levoy and Hanrahan; paper and slides)

Array of Images



Light Field Rendering (Levoy and Hanrahan; paper and slides)

Fast Display



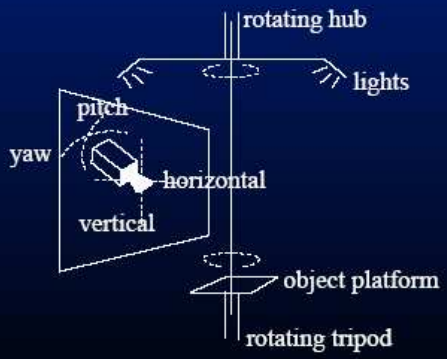

```
foreach x, y
    compute u,v,s,t
    I(x,y) = L(u,v,s,t)
```

Light Field Rendering (Levoy and Hanrahan; paper and slides)

Interpolation

- We have possibly non-uniform samples of a 4D space.
- Must interpolate to fill in.
- Worry about aliasing

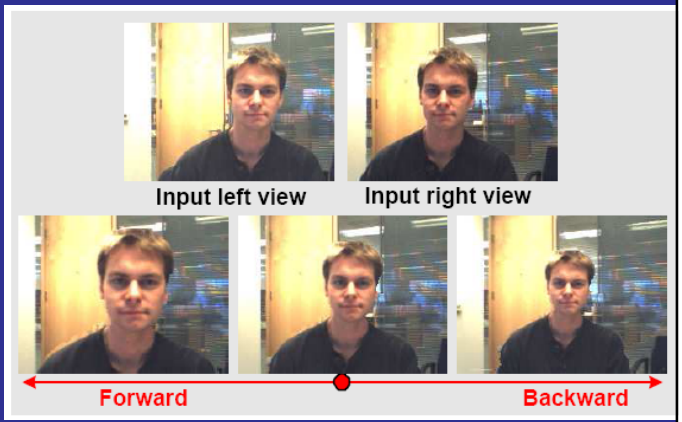
Experimental Light Field Gantry

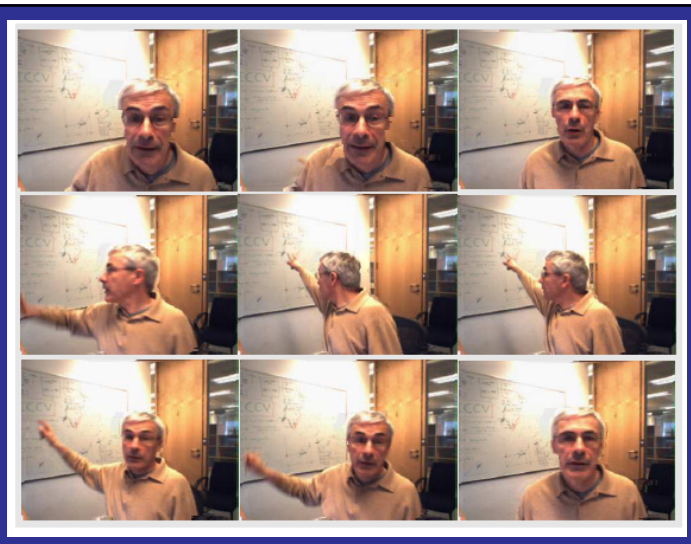


Light Field Rendering (Levoy and Hanrahan; paper and slides)

Using Depth in IBR

- Example, generating views for teleconferencing, with depth from stereo.





Generating using two views in which speaker isn't looking at camera.

Segmentation and compositing

Linear basis for lighting

λZ λX λY

Surface normal = (X,Y,Z) albedo = λ

Directional source = (Lx,Ly,Lz)

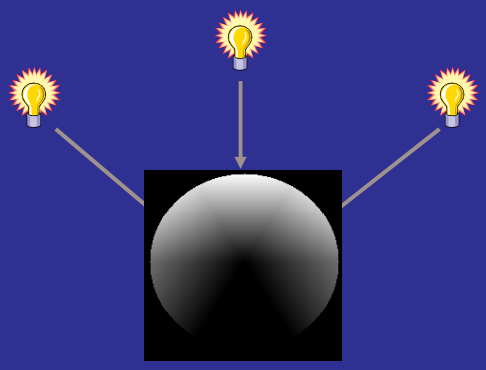
I = λ(Lx,Ly,Lz)(X,Y,Z) = Lx*λX + Ly*λY + Lz* λ Z

Take Max of this and 0

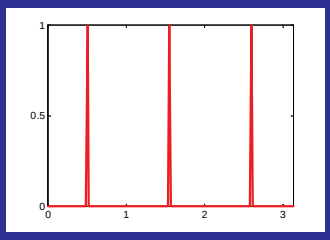
Using Linear Basis for Rendering

- Render three images
- Take linear combinations.
- Why can't we do this with three real images?

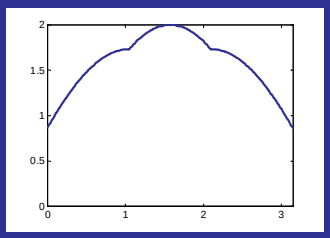
Reflectance smooths lighting



ℓ



r



Basis from diffuse lighting

λ

λZ

λX

λY

$2\lambda(Z^2-X^2-Y^2)$

$\lambda(X^2-Y^2)$

λXY

λXZ

λYZ

- Note, this can also be done with 9 real images, because this is a basis that contains real images
  - In 3D, real images aren't in the 3D space, we have to take the max with 0 to get real images.

Non-Photorealistic Rendering

- Take a photo and turn it into a different kind of image.

De Carlo and Santella

Video

Image Analogies

Given A, A’ and B, generate B’

A bit like Efros and Leung texture synthesis.