CS 395/495-26: Spring 2002

IBMR: Image Based Modeling and Rendering

Jack Tumblin
jet@cs.northwestern.edu

Admin: How this course works

- Refer to class website:
  www.cs.northwestern.edu/~jet

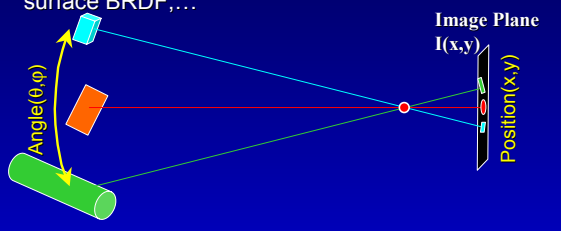
GOAL: First-Class Primitive

- **Want images as 'first-class' primitives**
  - Useful as BOTH **input** and **output**
  - Convert to/from traditional scene descriptions
- **Want to mix real & synthetic scenes freely**
- **Want to extend photography**
  - Easily capture scene:
    shape, movement, surface/BRDF, lighting …
  - Modify & Render the captured scene data
- **--BUT--**
- **images hold only PARTIAL scene information**
  - You can't always get what you want" –(Mick Jagger 1968)

Back To Basics: Scene & Image

Light + 3D Scene:
Illumination,
shape, movement,
surface BRDF,…

2D Image:
Collection of rays
through a point

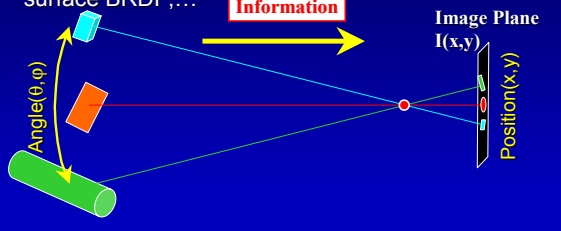


Trad. Computer Graphics

Light + 3D Scene:
Illumination,
shape, movement,
surface BRDF,…

Reduced, Incomplete Information

2D Image:
Collection of rays
through a point

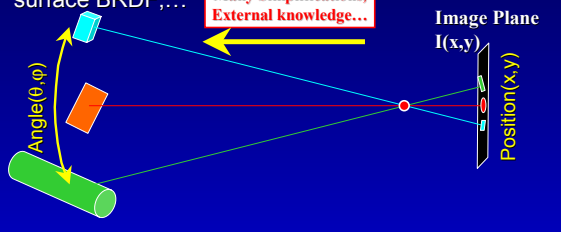


Trad. Computer Vision

Light + 3D Scene:
Illumination,
shape, movement,
surface BRDF,…

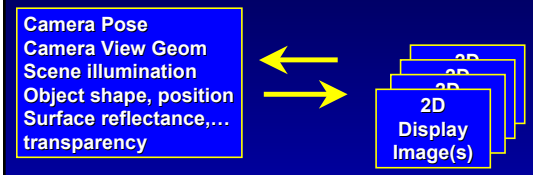
!TOUGH! 'ILL-POSED'
Many Simplifications, External knowledge…

2D Image:
Collection of rays
through a point



IBMR Goal: Bidirectional Rendering

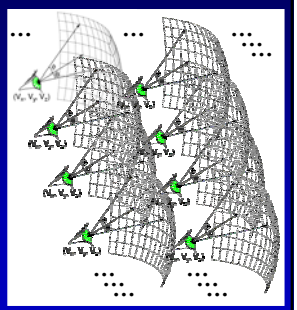
- Both forward and 'inverse' rendering!



A broad generalization of computer graphics

Plenoptic Function (Adelson, Bergen `91)

- for a given scene, describe:
  - ALL rays through
  - ALL pixels, of
  - ALL cameras, at
  - ALL wavelengths,
  - ALL time

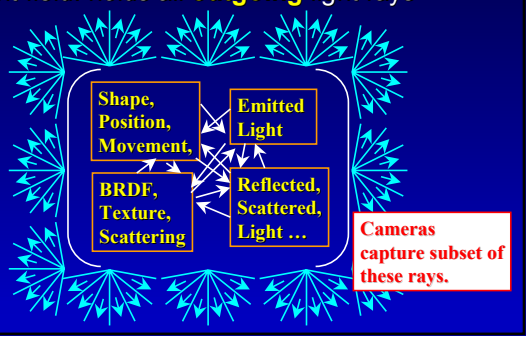


$F(x,y,z, \phi,\theta, \lambda, t)$

"Eyeballs Everywhere" function (7-D!)

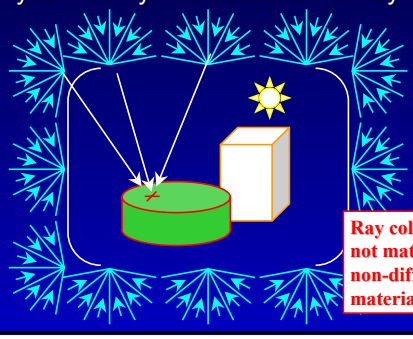
'Scene' causes Light Field

Light field: holds all **outgoing** light rays



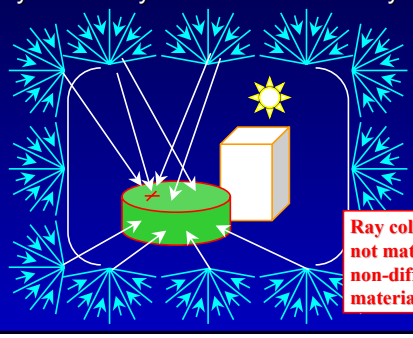
Shape Problems: Correspondence

Can you find ray intersections? Or ray depth?



Shape Problems: Correspondence

Can you find ray intersections? Or ray depth?



Early IBR: QuickTime VR (Chen, Williams '93)

1) Four Planar Images → 1 Cylindrical Panorama:

IN:

OUT:

Early IBR: QuickTime VR (Chen, Williams '93)

2) Windowing, Horizontal-only Reprojection:

OLDEST IBR: Shadow Maps (1984)

Fast Shadows from Z-buffer hardware:

1) Make the "Shadow Map":
   - Render image seen from light source, BUT
   - Keep ONLY the Z-buffer values (depth)

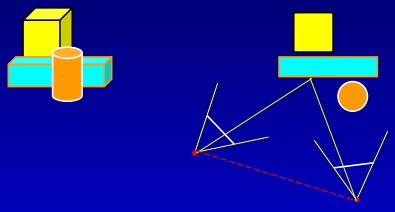
2) Render Scene from Eyepoint:
   - Pixel + Z depth gives 3D position of surface;
   - Project 3D position into Shadow map image
   - If Shadow Map depth < 3D depth, SHADOW!

Plenoptic Array: 'The Matrix Effect'

- Brute force!
  Simple arc, line, or ring array of cameras
- Synchronized shutter http://www.ruffy.com/firingline.html
- Warp/blend between images to
  change viewpoint on 'time-frozen' scene:

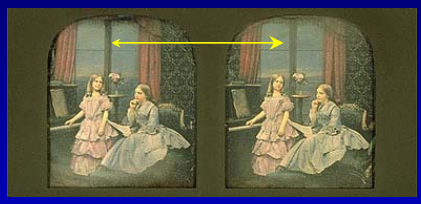

View Interpolation: How?

- Store Depth at each pixel: reproject
- Coarse or Simple 3D model:



View Interpolation: How?

- But what if no depth is available?
- Traditional Stereo Disparity Map:
  pixel-by-pixel search for correspondence



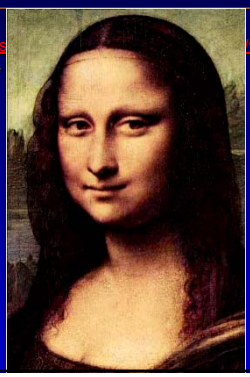
Seitz: 'View Morphing' SIGG`96

- http://www.cs.was /vmorph.htm

1)Manually set some corresp.points (eye corners, etc.)

2) pre-warp and post-warp to match points in 3D,

3) Reproject for Virtual cameras



Seitz: ‘View Morphing’ SIGG`96

- http://www.cs.wash vmorph.htm

Seitz: ‘View Morphing’ SIGG`96

- http://www.cs.wash vmorph.htm

Seitz: ‘View Morphing’ SIGG`96

- http://www.cs.wash vmorph.htm

Seitz: 'View Morphing' SIGG`96

- http://www.cs.wasi vmorph.htm

IBR-Motivating Opinions

"Computer Graphics: Hard"

- **Complex!** geometry, texture, lighting, shadows, compositing, BRDF, interreflections, etc. etc., etc., …
- **Irregular!** Visibility,Topology, Render Eqn.,…
- **Isolated!** Tough to use real objects in CGI
- **Slow!** compute-bound, off-line only,…

"Digital Imaging: Easy"

- **Simple!** More quality? Just pump more pixels!
- **Regular!** Vectorized, compressible, pipelined…
- **Accessible!** Use real OR synthetic (CGI) images!
- **Fast!** Scalable, Image reuse, a path to interactivity…

Practical IBMR

What useful partial solutions are possible?

- Texture Maps++:
- Image(s)+Depth: (3D shell)
- Estimating Depth & Silhouettes
- 'Light Probe' measures real-world light
- Light control measures BRDF
- Hybrids: BTF, stitching, …

Conclusion

- Heavy overlap with computer vision: careful not to re-invent & re-name!
- Elegant Geometry is **at the heart of it all**, even surface reflectance, illumination, etc. etc.
- THUS: we'll dive into geometry--all the rest is built on it!