Now Playing:

Baba O'Riley
The Who
from *Who's Next*
Released 1971

Image-Based Rendering (IBR)

Rick Skarbez, Instructor
COMP 575
November 27, 2007

Announcements

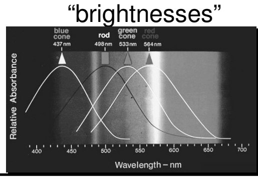
- You need to arrange to talk to me before December 1 for your project update
- I am going to attempt to reserve a room/time on December 11 for project presentations
- The final deadline for project submissions will be the evening of December 12
- The final exam is Friday, December 14 at 4:00pm in this room (SN 011)

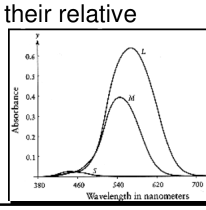
Human Visual Perception

- The human eye has two types of detectors
  - Rods and cones
- Rods, basically, only detect luminance and are the dominant detector in low light
- Cones detect color
  - Three types of cones: red, green, and blue (more or less)

Human Visual Perception

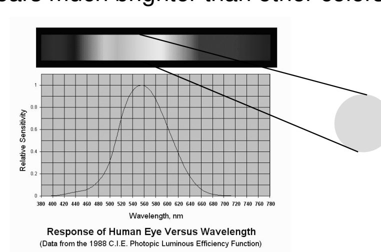
- As you can see in the images below,the "red" cone doesn't just see red, etc.
- The left image shows the range of wavelengths each cone responds to
- The right image shows their relative "brightnesses"





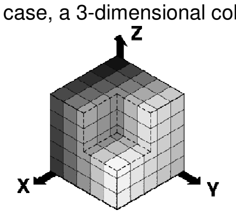
Human Visual Perception

- To humans (with normal vision), green appears much brighter than other colors



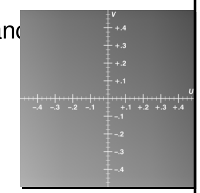
RGB Color Space

- We can think of RGB defining a <u>color space</u>
  - In this case, a 3-dimensional color space



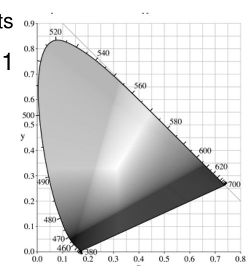
Luv (or Yuv) Color Space

- L: a single value representing luminance
- u,v: two colors for chrominanc
  - Define a position in the color space to the right
  - u = Blue - L
  - v = Red - L
- Component (HD) video cables use YPbPr, which is a scaled version of Yuv



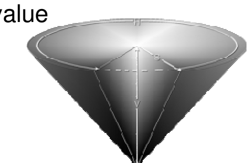
CIELab Color Space

- Very perceptually linear
  - Great for measurements
- Derived from the CIE 1 space, shown here



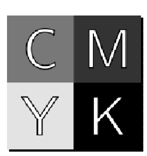
HSV Color Space

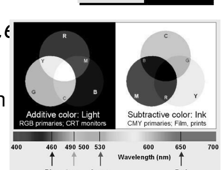
- Hue, saturation, and value
  - Hue: color
  - Saturation: "vividness"
  - Value: brightness
- Popular in graphics applications



CMYK Color Space

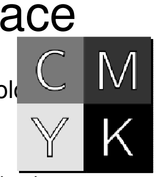
- Not really used for computer graphics
  - However, far and away the most popular for printed materials
- CMYK is a subtractive (*i.e* like paints) color space
  - As opposed to RGB, which additive (*i.e.* like light)



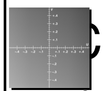


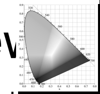
CMYK Color Space

- Why do we need subtractive col for printing?
  - Paper is already white (maximum value), so adding ink can only make the image darker
- Black (K) is separate
  - Because no combination of cyan, magenta, and yellow can generate a true black

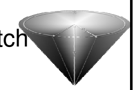


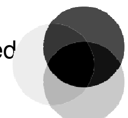
Color Spaces Review





- These were just a sampling of possible color spaces
- There are equations to easily switch between spaces
  - However, some colors that are within the gamut of one space may not be in the gamut of another
- Consider what properties you need when choosing a color space





Sample HDR Image

Image from Wikipedia Commons

Dynamic Range

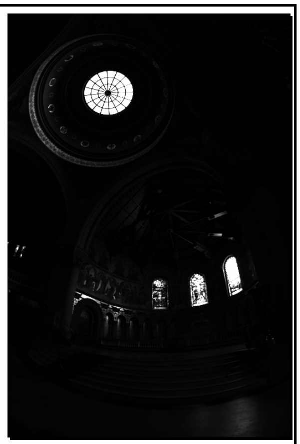
- Computer monitors and digital cameras have limited dynamic range
  - 8 bits [0,255] to 12 bits [0,4095]
- The real world has MUCH greater dynamic range
  - The difference between sunlight and moonlight is on the order of 10000x
  - Some scenes can contain even a wider range

Capturing Greater Dynamic Range

- To capture this greater dynamic range with digital cameras, we can capture multiple <u>bracketed</u> images
  - <u>Bracketing</u> means taking multiple pictures of the same scene with different camera settings
    - *i.e.* different exposure times or aperture sizes
- To capture it with computer graphics, can just do lighting calculations with more bits

Stanford Chapel

Shortest Exposure



All HDR Data from http://debevec.org/

Stanford Chapel

Short Exposure

Stanford Chapel
Longer Exposure

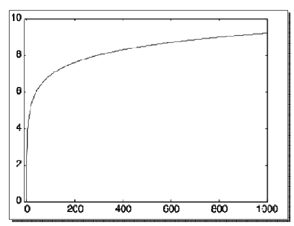
Stanford Chapel
Longest Exposure

HDR Image Generation

- We have a problem here
  - Does anyone see it?
    - These images have too much dynamic range to be drawn on our display!
- The process of fixing this is called tone mapping

What Can We Do?

Solution #1:
Log Mapping



What Can We Do?

Solution #2:
Low Frequency Attenuation

- Reduce Low Frequencies
  - i.e. gradual changes
- Keep High Frequencies

What Can We Do?

Solution #2:
Low Frequency Attenuation

- Reduce Low Frequencies
  - i.e. gradual changes
- Keep High Frequencies

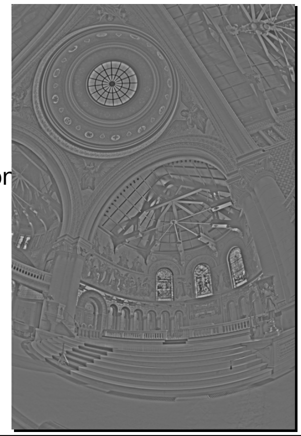
Low Frequency Mask

What Can We Do?

Solution #2:
Low Frequency Attenuation

- Reduce Low Frequencies
  - i.e. gradual changes
- Keep High Frequencies

High Frequency Mask

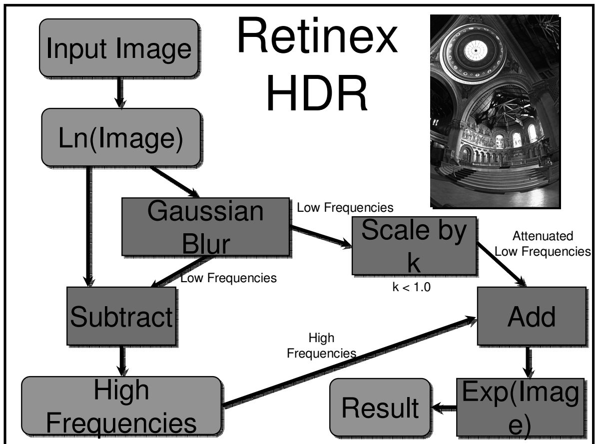


What Can We Do?

Solution #2:
Low Frequency Attenuation

- Reduce Low Frequencies
  - i.e. gradual changes
- Keep High Frequencies

Retinex HDR



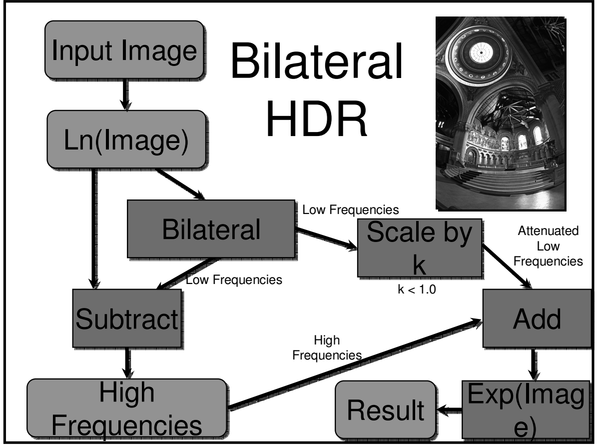
How can we do even better?

- Maybe Gaussian filters aren't the best tool
  - Blur across edges, obscuring high frequency detail
- Can use an edge-preserving filter
  - I won't go into the math
  - Basically, the filter can recognize when it encounters an edge, and not blur across it

Bilateral Filter Example

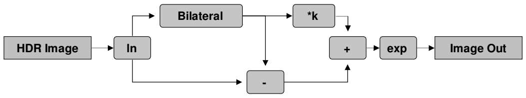
Original Gaussian Bilateral

Bilateral HDR



Bilateral Filter HDR

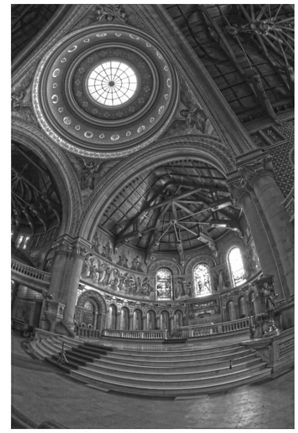
Durand & Dorsey, SIGGRAPH 02



What Can We Do?

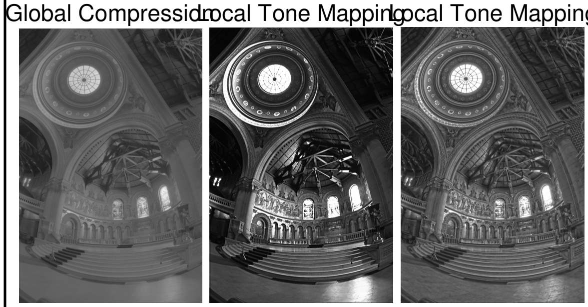
Solution #3:
Bilateral Tone Mapping

Durand & Dorsey 02



HDR Results

Global Compression | Local Tone Mapping | Local Tone Mapping



Log Mapping | Retinex Tone Mapping | Bilateral Tone Mapping

HDR Review Over

- Any questions?

The Pursuit of Photorealism

- Almost the entire story of computer graphics has been the attempt to generate more and more photorealistic images

The Pursuit of Photorealism

- So what if you wanted to render a real place?
  - Say, the Cathedr of Notre Dame in Paris

Example drawn from Debevec's SIGGRAPH 99 Course

The Pursuit of Photorealism

- You could:
  - Acquire accurate measurements of the building
  - Use these measurements to construct a geometric model
  - Apply the appropriate material properties to every surface
  - Use some advanced global illumination technique to simulate light bouncing around the cathedral

The Pursuit of Photorealism

- Alternatively, you could:
  - Take a picture of the cathedral from the desired viewpoint
    - This would be much easier
    - Also, it would <u>look better</u>
      - Pictures are by definition photorealistic

The Pursuit of Photorealism

- So why even bother with computer graphics?
  - For one, you can generate imagery for scenes that don't actually exist
  - Also, you generally want a user to be able to move through your virtual scene
    - It would be a huge pain to take pictures from every possible viewpoint
      - Right?

Image-Based Rendering

- Not necessarily
- Image-based rendering (IBR) grew out of the desire to bypass the manual modeling stage
  - Allows you to retain much of the realism of photographs, while also gaining the flexibility of computer graphics

What is IBR?

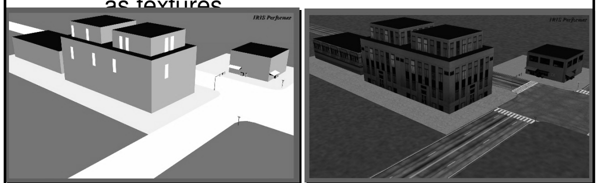
- It can mean any number of things
  - As a short definition, we can say that it is any technique that uses images (of some kind), either directly rendering with them or using them to create models

Texturing as IBR?

- If we accept this definition, then we've already seen one "IBR" technique
  - Texture mapping
- In general, this wouldn't be considered IBR
  - At least, as long as it still uses manually-created geometry

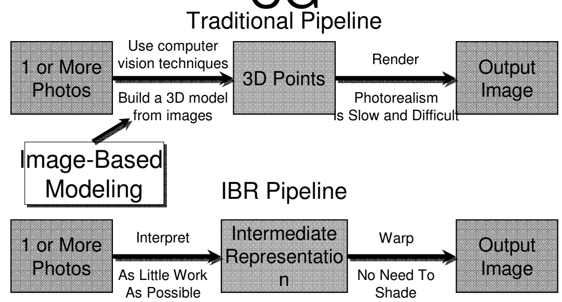
Remember Texture Mapping?

- Rather than trying to render complex appearance with just geometry, use simple geometry and complex images as textures



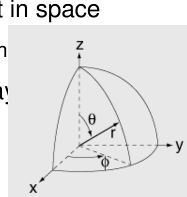
Before Texturing After Texturing

IBR vs. Traditional CG



The Math Behind Photographs

- Can think of a photograph as a “catalog” of the colors of the rays that pass through a single point in space
  - *i.e.* a pinhole, or a camera len
- We can parametrize any ra as [Φ, θ, x, y, z]
  - The ray through point (x,y,z) in direction (Φ, θ)



The Plenoptic Function

Adelson & Bergen,
“The Plenoptic Function and Elements of Early Vision”

- Describes the light received
  - At any position,
  - From any direction,
  - At any time

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

The Plenoptic Function

- Simplifications:
  - Ignore changes over time
  - Use 3-component color instead of wavelength
  - Left with a 5D function:
    - P(Φ, θ, x, y, z)
      - 3D position
      - 2D orientattion

Panoramas

- A panorama stores a 2D plenoptic function
  - Always a single center of projection
  - Can be stitched (that is, put together from multiple smaller images) or taken by a single camera with complicated lensing

Stitched Panorama

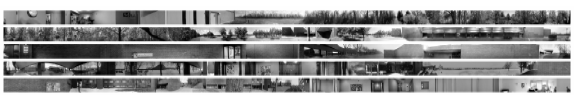
Kansas City Country Club
Library of Congress

Digitally Stitched Panorama

Great Wall of China

"All This Useless Beauty"

Paul Kry, SIGGRAPH 2004



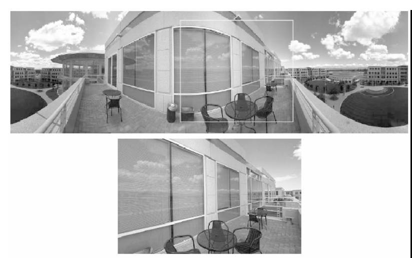
3" x <u>600"</u> "panorama"

200 megapixels

Extending Panoramas

Eric Chen "An Image-Based Approach to Virtual Environment Navigation", SIGGRAPH 95

Quicktime VR

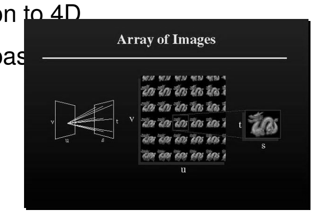


Panoramas as Virtual Environments

- Pros:
  - Easy to make
- Cons:
  - No sense of 3D
  - Fixed viewpoint
  - Hard to navigate

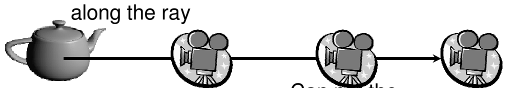
Function: Light Fields and Lumigraphs

- Can take advantage of empty space around the camera to reduce the plenoptic function to 4D
- Build up a databas of ray values
  - How and why?

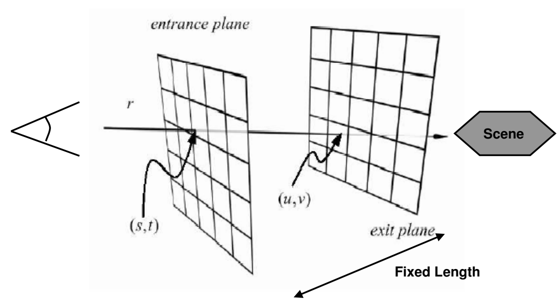


Why 4D from Empty Space?

- If we assume that the camera is in empty space (*i.e.* all objects in the scene are distant)
  - We no longer have to worry about occlusions
  - A ray has the same color at every point along the ray



4D => 2 Plane Parametrization

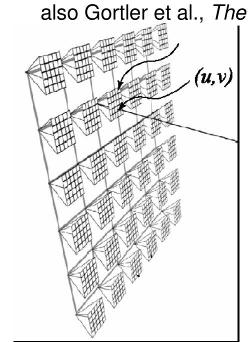


So, how do we use it?

- Now we know (at least for certain types of scenes) you can move the virtual camera without taking new pictures
- But how do we find and store all the rays we need?

Light Fields

Levoy & Hanrahan, *Light Field Rendering* (SIGGRAPH 96)
also Gortler et al., *The Lumigraph* (SIGGRAPH 96)



- "Box of photons"
- Captured either by a moving camera or an array of cameras
- 6x6 5x5 images shown
- 16x16 512x512 in the original paper
  - 256 MB / image

Lumigraph Mapping

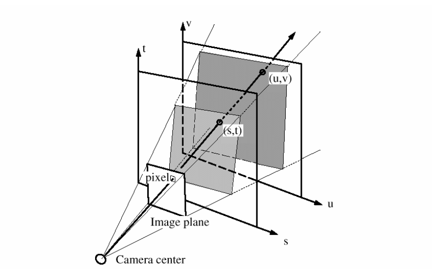
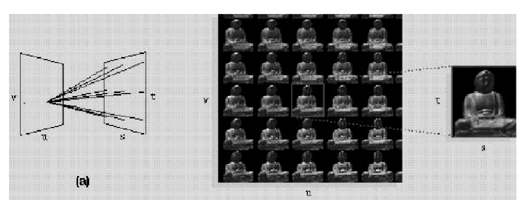
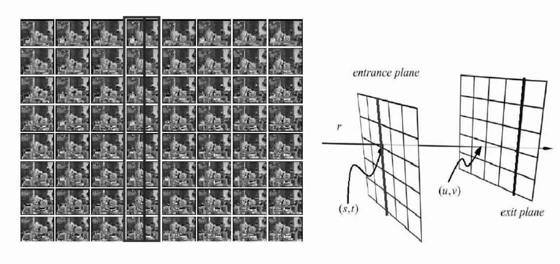


Figure 3: Relationship between Lumigraph and a pixel in an arbitrary image

Images Form (u,v) Plane

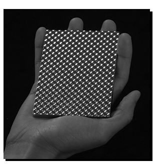


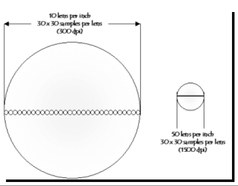
Light Field Example

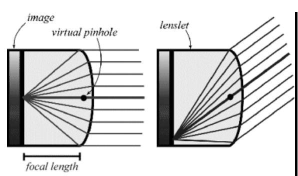


Stanford Multi-Camera Array

Stanford Light Field Camera

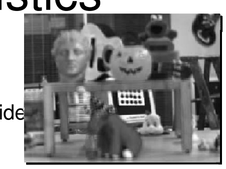


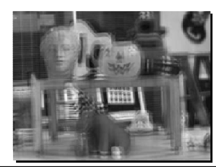




Light Field Characteristics

- Pros:
  - Very attractive
  - Can be used to capture vide
- Cons:
  - Huge memory footprint
  - Difficult sampling issues
  - Only one plane in focus
  - Difficult to capture





No Geometry

- Note that neither of these techniques make any assumptions at all about geometry
  - Just show images
- Another technique in this vein is Concentric Mosaics, from Shum & He (SIGGRAPH 99)

Facade

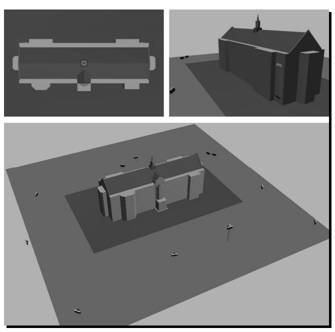
Debevec, SIGGRAPH 96

- Use a small number of images to generate a “blocks” model
  - Establish edge correspondences
  - Reconstruct by minimizing error
  - Do view-dependent texture mapping

Facade Example

Images with Edges Marked

Model



Novel View

IBR Review

- Attempts to use real photographs to generate high-quality images without manual modeling
- Can include:
  - Automatically building geometry from images
  - Rendering a dynamic scene with no geometry
  - Something in between
- Any questions?

Quick Aside: Computational Photography

- Digital cameras are great
- Can afford to take a whole bunch of images and throw many away
- This gives us new opportunities

Quick Aside: Computational Photography

- For example, we can use elements of multiple images to create an image that is different from any that actually exist
- Can mess with:
  - Time
  - Space
  - Perspective
- This is the essence of computational photography

Pictures vs. "Moments"

Photograph

A "Moment"

Images from Michael Cohen, MSR

Next Time

- Our last real lecture
- I'll try to do a bit of a grab bag
  - Filtering and image processing
  - Computer graphics and video games
  - Particle effects
  - And much, much more