Global Illumination & HDRI Formats

Greg Ward
Anyhere Software

Global Illumination

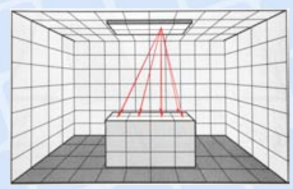
- Accounts for most (if not all) visible light interactions
- Goal may be to maximize realism, but more often "visual reproduction"
- Visible light often outside the range of standard displays and formats
  - HDRI format is required
  - HDR display would be nice

GI Techniques

- Radiosity
  - Divide surfaces into patches
  - Compute diffuse light exchange
  - Final pass can add specular effects
- Ray-tracing
  - Follow individual light paths each pixel
  - Monte Carlo sampling for everything
  - Filter resulting image to reduce noise

Radiosity Method

1. Subdivide surfaces into patches
2. Compute form factor (mutual visibility) between patch pairs
3. Shoot (or gather) radiant flux to (or from) patches
4. Interpolate result and render image from desired view, adding specular reflections if desired



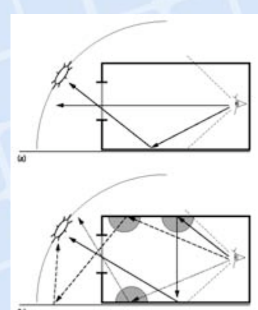
+ Handles simple, diffuse environments very efficiently
+ Provides a view-independent solution
- Resource problems for complicated environments
- Cannot handle glossy surfaces except via ray-tracing in final pass

Radiosity Example

- *Lightscape* rendering of SFMOMA from *View by View*
- Specular effects added in final pass

Ray-tracing Method

1. Trace light paths from a viewpoint
2. At each surface, find sources to compute reflection and transmission
3. Compute diffuse interreflection using Monte Carlo path tracing (unbiased) or irradiance caching (biased)
4. Multiple samples averaged together at each pixel to reduce noise



+ Handles complicated geometry well
+ Supports all surface types
- View-dependent solution
- Can be expensive

Ray-tracing Example

- *Radiance* rendering of Mandalay-Luxor Retail Complex from *Arup Lighting*
- Fritted glass on roof partially scatters sunlight

GI and HDRI

- Traditional CG rendering constrains input to fit within 24-bit RGB gamut
- Global Illumination algorithms attempt to simulate the real world, which has no such constraints
  - CG rendering parallels TV broadcasting
  - GI rendering parallels film photography
- What we need is a digital film format

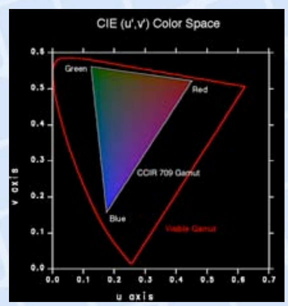
The 24-bit Red Green Blues

- Although 24-bit *sRGB* is reasonably matched to CRT displays, it is a poor match to human vision
  - People can see twice as many colors
  - People can see twice the log range

Q: Why did they base a standard on existing display technology?

A: Because signal processing *used* to be expensive…

24-bit *sRGB* Space

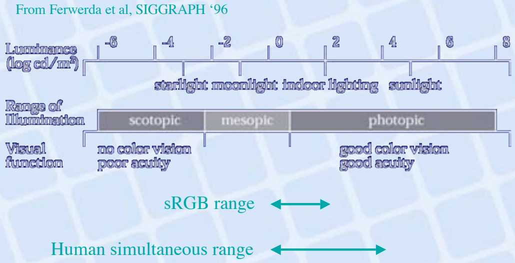


Covers roughly half of the visible color gamut

Lost are many of the saturated blue-greens and violets

Dynamic Range

From Ferwerda et al, SIGGRAPH '96



What is HDRI?

- High Dynamic Range Images have a wider gamut and contrast than 24-bit RGB
  - Preferably, the gamut and dynamic range covered exceed those of human vision

Advantage 1: an image standard based on human vision won't need frequent updates

Advantage 2: floating point pixels open up a vast new world of image processing

Some HDRI Formats

- *Pixar* 33-bit log-encoded TIFF
- *Radiance* 32-bit RGBE and XYZE
- IEEE 96-bit TIFF & Portable FloatMap
- 16-bit/sample TIFF (I.e., RGB48)
- LogLuv TIFF (24-bit and 32-bit)
- *ILM* 48-bit OpenEXR format
- Others??

Pixar Log TIFF Codec

Purpose: To store film recorder input

- Implemented in Sam Leffler's TIFF library
- 11 bits each of log red, green, and blue
- 3.8 orders of magnitude in 0.4% steps
- ZIP lossless entropy compression
- Does not cover visible gamut
- Dynamic range marginal for image processing

Radiance RGBE & XYZE

Purpose: To store GI renderings

- Simple format with free source code
- 8 bits each for 3 mantissas + 1 exponent
- 76 orders of magnitude in 1% steps
- Run-length encoding (20% avg. compr.)
- RGBE format does not cover visible gamut
- Color quantization not perceptually uniform
- Dynamic range at expense of accuracy

Radiance Format (.pic, .hdr)

(145, 215, 87, 149) =
(145, 215, 87) * 2^(149-128) =
(1190000, 1760000, 713000)

(145, 215, 87, 103) =
(145, 215, 87) * 2^(103-128) =
(0.00000432, 0.00000641, 0.00000259)

Ward, Greg. "Real Pixels," in Graphics Gems IV, edited by James Arvo, Academic Press, 1994

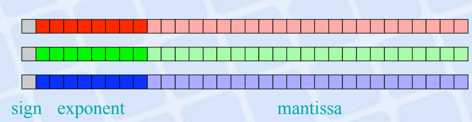
IEEE 96-bit TIFF & Portable FloatMap

Purpose: To minimize translation errors

- Most accurate representation
- Files are enormous
  - 32-bit IEEE floats do not compress well

Portable FloatMap (.pfm)

- 12 bytes per pixel, 4 for each channel



Text header similar to Jeff Poskanzer's .ppm image format:

```
PF
768 512
1
<binary image data>
```

Floating Point TIFF similar

16-bit/sample TIFF (RGB48)

Purpose: Higher resolution than 8-bit/samp

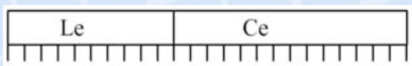
- Supported by Photoshop and TIFF libs
- 16 bits each of log red, green, and blue
- 5.4 orders of magnitude in < 1% steps
- LZW lossless compression available
- Does not cover visible gamut
- Good dynamic range requires gamma=2.2, not linear, and white much less than 1
  - Photoshop treats 1 as white, which is useless

24-bit LogLuv TIFF Codec

Purpose: To match human vision

- Implemented in Leffler's TIFF library
- 10-bit LogL + 14-bit CIE (u',v') lookup
- 4.8 orders of magnitude in 1.1% steps
- Just covers visible gamut and range
- No compression

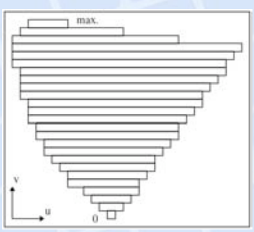
24 -bit LogLuv Pixel



$$L_e = \lfloor 64(\log_2 L + 12) \rfloor$$

$$u' = \frac{4x}{-2x + 12y + 3}$$

$$v' = \frac{9y}{-2x + 12y + 3}$$



32-bit LogLuv TIFF Codec

Purpose: To surpass human vision

- Implemented in Leffler's TIFF library
- 16-bit LogL + 8 bits each for CIE (u',v')
- 38 orders of magnitude in 0.3% steps
- Run-length encoding (30% avg. compr.)
- Allows negative luminance values

32-bit LogLuv Pixel

$$L_e = \lfloor 256(\log_2 L + 64) \rfloor$$

$$u' = \frac{4x}{-2x + 12y + 3}$$

$$v' = \frac{9y}{-2x + 12y + 3}$$

$$u_e = \lfloor 410u' \rfloor$$

$$v_e = \lfloor 410v' \rfloor$$

ILM OpenEXR Format

Purpose: HDR lighting and compositing

- 16-bit/primary floating point (sign-e5-m10)
- 9.6 orders of magnitude in 0.1% steps
- Wavelet compression of about 40%
- Negative colors and full gamut RGB
- Open Source I/O library released Fall 2002

ILM's OpenEXR (.exr)

- 6 bytes per pixel, 2 for each channel, compressed

- Several lossless compression options, 2:1 typical
- Compatible with the "half" datatype in NVidia's Cg
- Supported natively on GeForce FX and Quadro FX
- Available at http://www.openexr.net/

Conclusions

- Global Illumination benefits greatly from HDR on input *and* output
- Good and useful HDRI formats exist
- Rendering software and camera hardware will catch up, eventually
- In the meantime, check out *Radiance*, *Photosphere* and *HDRshop*

Photosphere Demo

- Mac OS X version available from:
  - http://www.anyhere.com/gward/pickup/photosphere.tar.gz