High Dynamic Range

Steve Rotenberg

CSE168: Rendering Algorithms

UCSD, Spring 2014

High Dynamic Range

- *High dynamic range imaging* (HDRI or HDR) refers to a set of techniques for rendering and displaying images with a wide range of brightness values
- So far, we’ve been dealing with pixel values from [0…1], but there is no upper limit to the radiance we could have in a pixel

High Dynamic Range

- A modern computer monitor has a pretty good contrast ratio, typically around 1000:1
- Film images displayed by a projector may get around 500:1
- A high quality printed page may provide a contrast ratio of up to 100:1 in a well lit room
- A newspaper would be closer to 15:1
- A view of the real world, with the sun visible could have a contrast ratio of 10^{15}:1

High Dynamic Range

- Human vision is capable of perceiving a remarkable range of brightness
- Some estimates put it around 10^{12}:1, however, this includes the effects of pupil contraction and visual adaptation
- In a single view, humans vision can still perceive a huge contrast ratio (maybe 10^{9}:1)

High Dynamic Range

- A central purpose of HDR is to provide solutions for representing high contrast images on low contrast display devices

HDR Encoding

- There are many ways to encode a HDR color
- A simple, intuitive method is to just use floating point values for R, G, and B, with a lower bound of 0 and no upper bound
- This uses 32*3=96 bits to represent a color (without compression), and is a reasonable approach for representing an HDR image in memory during the rendering process
- However, there are also various other encodings that are meant to use less data

RGBE Encoding

- The 32-bit RGBE format is one method for representing a high dynamic range color
- It uses 8 bit values for red, green, blue, and an exponent
- To convert from floating point RGB to the fixed point RGBE format:

$$E = \left\lceil \log_2\left(\max\left(R_f, G_f, B_f\right)\right) + 128 \right\rceil$$

$$R = \left\lfloor \frac{256 R_f}{2^{E-128}} \right\rfloor$$

$$G = \left\lfloor \frac{256 G_f}{2^{E-128}} \right\rfloor$$

$$\mathrm{B} = \left\lfloor \frac{256 B_f}{2^{E-128}} \right\rfloor$$

RGBE Encoding

- To convert from RGBE back to floating point RGB:

$$R_f = \frac{R+0.5}{256} 2^{E-128}$$

$$G_f = \frac{G+0.5}{256} 2^{E-128}$$

$$B_f = \frac{B+0.5}{256} 2^{E-128}$$

HDR Formats & Tools

- There are some standard formats for storing HDR images such as .hdr and .exr, plus there are extensions to .tif and tricks for storing HDRs as .jpg
- There are also some open source software packages such as OpenEXR and open source tools like HDR Shop

Color Spaces

- The human eye has rods and cones on the retina for detecting light
- The rods respond to low light levels and are responsible mainly for night vision
- The cones respond to brighter light and there are three distinct cone types that respond each to red, green, and greenish-blue light
- Because we perceive color based on the proportional response of the 3 cones, it allows us to represent colors as a mixture of 3 primary colors
- This is why the RGB color system works
- It also allows us to use various other 3-value color representations
- Some examples are the HSV, YUV, and XYZ formats
- There are various methods for converting between different formats

Rendering with HDR

- Rendering an image with high dynamic range is simple
- We already compute the radiance per pixel in RGB format, so all we need to do to generate an HDR image is to store the colors like that- instead of clamping them to [0…1] and converting to 24 bit, as we've been doing
- Once the image is rendered in full range, we can then apply various techniques for displaying the image on a low dynamic range display system
- This breaks the rendering into two passes: an actual rendering pass which simulates light, and then a *tone mapping* pass which takes the HDR data and maps it to a set of displayable values

Tone Mapping

Tone Mapping

- *Tone mapping* is a general term for converting an image of HDR values to an image of displayable colors
- For a 24 bit monitor, obviously this would involve converting the HDR colors to 24 bit values in the [0…1] range
- However, we might also want to tone map an image for a different display type, such as a color printer, or even a black-and-white display

Local / Global Tone Mapping

- Some tone mapping operators are global, meaning that they apply uniformly to all pixels in the scene. The same color in two different pixels of the input image will map to the same color in the final image
- Other, more sophisticated operators are local, meaning that they use local properties near the pixel to determine the final color. The same color in two input pixels will probably not map to the same color in the final image

Exposure

- A simple global approach is to have an ‘exposure’ setting in the camera that represents the upper limit and just divide all colors by the exposure to map them into a [0…1] range

Gamma Correction

- Human vision is sensitive to differences in radiance ratios, rather than differences in absolute radiance
- Therefore, monitors are designed to take the 0-255 values and map them to an exponential brightness scale
- The exponent used is called the *monitor gamma*, and is typically around 1.5-1.7
- This allows humans to perceive a change from 4 to 5 to be similar to a change from 249 to 250
- To take our linear measurement of radiance and convert it to a proper displayable 8-bit value for red, green, or blue, we should apply *gamma correction*

$$L_{out} = {L_{in}}^{1/\gamma}$$

- This maps a [0…1] value to a new [0…1] value, and should always be done as a final step in the tone mapping process

Exposure & Gamma

- A basic exposure scale and gamma correction are really essential elements of a renderer and should probably be included in the Camera class
- The combination of these results in an approximately linear mapping from computed radiance in the renderer to displayed radiance on the monitor

Global Tone Mapping

- There are some more sophisticated global and local tone mapping operations as well that attempt to compress higher dynamic range renderings into low dynamic range displays
- The goal is to simulate the appearance of a wider range of contrast ratios than the device is actually capable of

Exponential Mappings

- For medium dynamic range images, the following global mappings are sometimes used:

$$L_{out} = \frac{\log_{10}(1 + L_{in})}{\log_{10}(1 + L_{max})}$$

$$L_{out} = 1 - \exp\left(\frac{-L_{in}}{L_{ave}}\right)$$

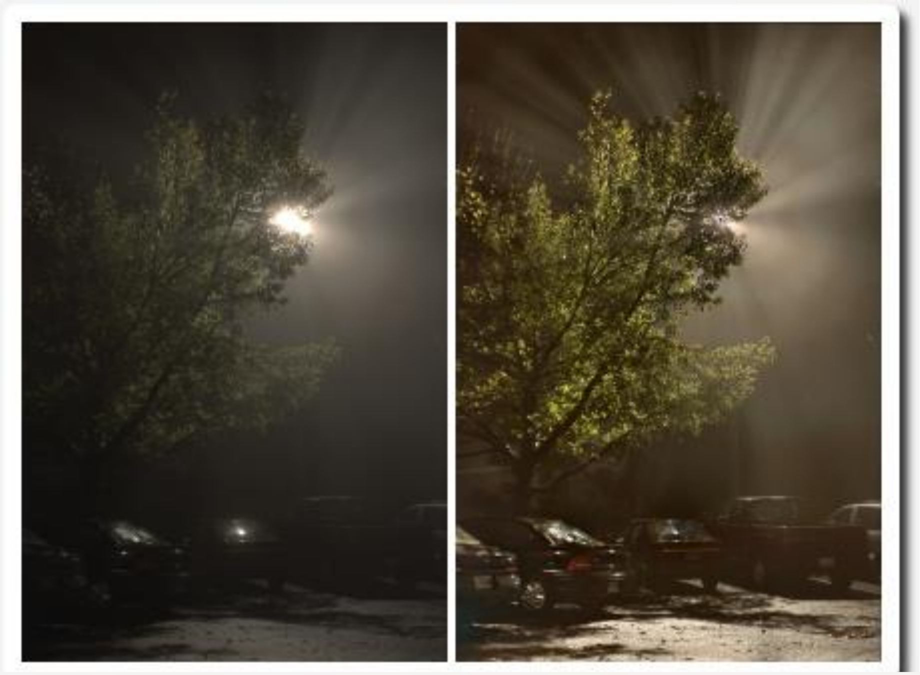
Global Tone Mapping

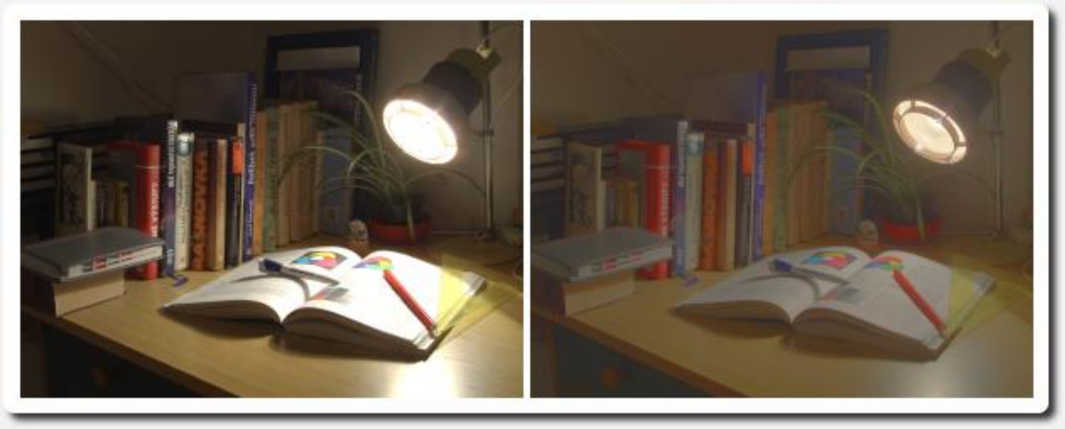
- "The Application of Computer Graphics in Lighting Design", Miller, Ngai, Miller, 1984
- "Tone Reproduction for Realistic Computer Generated Images", Tumblin, Rushmeier, 1993
- "A Contrast Based Scale Factor for Luminance Display", Ward, 1994
- "A Model for Visual Adaptation for Realistic Image Synthesis", Ferwerda, Pattanaik, Shirley, Greenberg, 1996

Local Tone Mapping

- “Photographic Tone Reproduction for Digital Images”, Reinhard, Stark, Shirley, Ferwerda, 2002
- “A Tone Mapping Algorithm for High Contrast Images”, Ashikhmin, 2002
- “Adaptive Gain Control for High Dynamic Range Display”, Pattanaik, Yee, 2002

Local Tone Mapping





Visual Adaptation

- Visual adaptation refers to the processes in the human visual system that adapt to different lighting scenarios
- For example, when we drive through a dark tunnel during the day, our eyes take a moment to adjust to the darkness, and then when we come back out, we are temporarily blinded by the brightness
- Visual adaptation also applies to colors. This can be observed by wearing a pair of very colored sunglasses for a while (like 10+ minutes)
- These types of effects are a form of tone mapping and have been modeled in various papers and are even included in some video games

Lens Diffusion

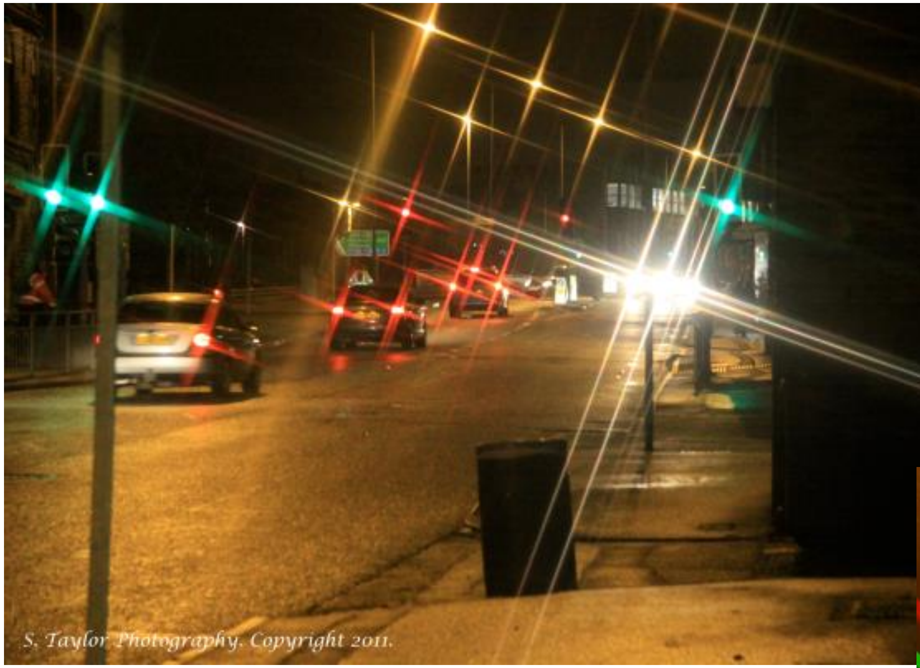
Lens Filters

- We discussed various lens effects in a previous lecture
- Some of these are geometric distortions that warp the image or cause focusing artifacts
- Some of them are caused by reflections and scattering within the lens system and lead to various lens flares such as blooms, halos, and stars
- These are particularly evident when viewing very bright light sources, and are often combined with HDR rendering

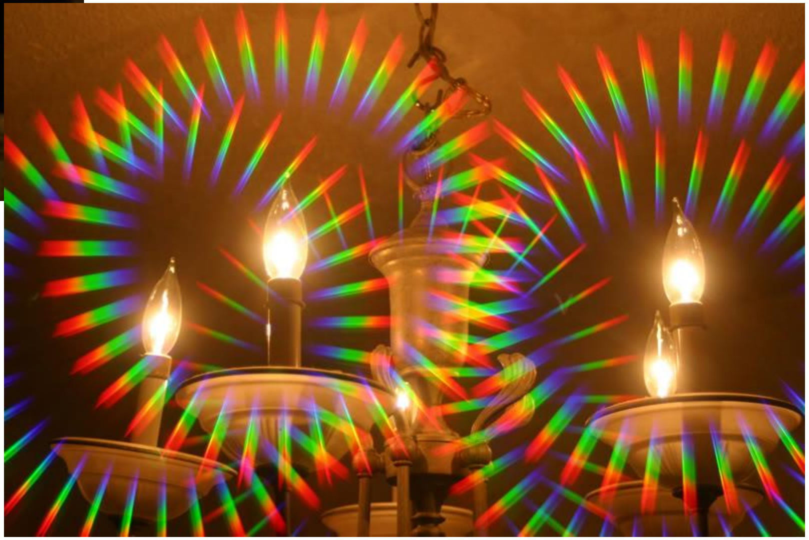
Bloom, Halos, & Stars

- Bloom, halos, and stars are forms of lens flares

Star Filters



S. Taylor Photography. Copyright 2011.



Lens Filters

- These effects can be achieved by applying *lens filters* to the image

- We start with a *kernel image* which is essentially an image of what a single bright point of light would look like (so for a star filter, it would be a single star centered on a black background)
  - The kernel image should be normalized so that all of the pixels together add up to exactly 1.0 in each color component
  - For a typical star or bloom filter, the pixel at the exact center would get most of the light (probably over 95%) and the entire remaining light would be spread out in the star or bloom pattern

- To apply the filter, we take our input HDR image and generate a new image, which is the sum of each pixel times the kernel image, centered at the pixel

Lens Filters

- For a 1000x1000 kernel and a 1000x1000 HDR image, this process would require looping through all 1000000 pixels of the HDR image and then for each one, looping through all 1000000 pixels of the kernel image
- This would require a trillion operations
- But there is a faster way…

Fourier Transformation

- We can apply 2D Fourier transforms to images
- For each color channel (RGB) of the input image, we generate a new image of complex numbers (i.e., a 1000x1000 pixel greyscale image would generate 2x1000x1000 floating point values)
- The inverse FFT operation converts the complex numbers back into an image
- There are various open source implementations of 2D FFTs including some very fast GPU versions

Convolution

- To perform a filter operation, we perform a convolution operation:
  - Apply FFT to filter kernel
  - Apply FFT to HDR image
  - Compute pixel-wise multiplication of two results
  - Apply inverse FFT to product

- Note that for the convolution, the images must be the same size, so one image is typically padded with black pixels to accommodate this
- Also, the operation will wrap around the edges of the image as if it were tiled, so additional padding is required to prevent this
- Also, most FFT implementations are recursive and require that the images be square and have resolutions in powers of two
- These padding operations are often implemented implicitly within an FFT software package

HDR Image Acquisition

HDR Image Acquisition

- Often when rendering, we need to use real-world imagery as inputs (i.e., for texture maps, environment maps, background images, etc.)
- We need some sort of photographic process for acquiring HDR imagery from the real world
- Unfortunately, most cameras are not capable of capturing a high dynamic range in a single image, and so we must use some special techniques to do this
- The process of acquiring HDR imagery with a camera is fairly mature and many good techniques exist, as well as several good off-the-shelf tools
- We will briefly examine these

HDR Image Acquisition

- The general approach is to take multiple photographs with different exposure settings and then algorithmically combine them into a single HDR image
- For each pixel, we have several measurements at different exposure settings
- In each of the images, some pixels will be dark (underexposed) and some will be bright (overexposed), but some will be in between
- We want to ignore overly bright or dark pixels and take the value from pixels that are in the middle range

HDR Image Acquisition

- There are several challenges that complicate this process:
  - Movement in the scene between shots (leaves blowing, people walking, clouds moving…)
  - Misalignment at the pixel level due to slight movement of the camera between shots
  - Removal of lens flares caused by overexposures
  - Removal of other lens effects
- There are good solutions for each of these issues

HDR Cameras

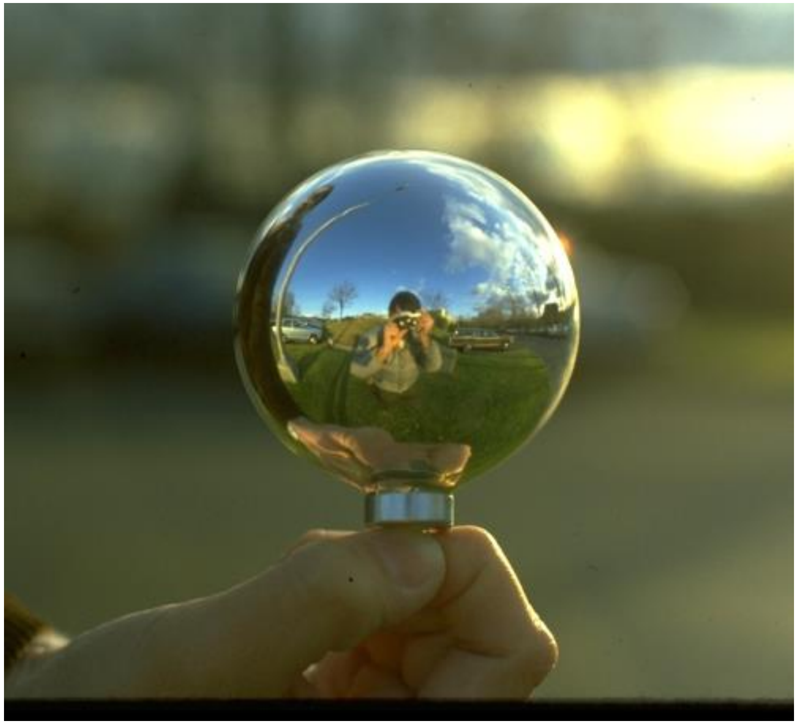
- Some modern cameras attempt to directly capture images with higher dynamic range than typical photos
- This can be done in several ways, but some of the more sophisticated cameras can open the shutter just once and measure the radiance on the sensor several times over a finite interval
- This data is then combined at the time of the photograph and a HDR image is saved out

Spherical Cameras

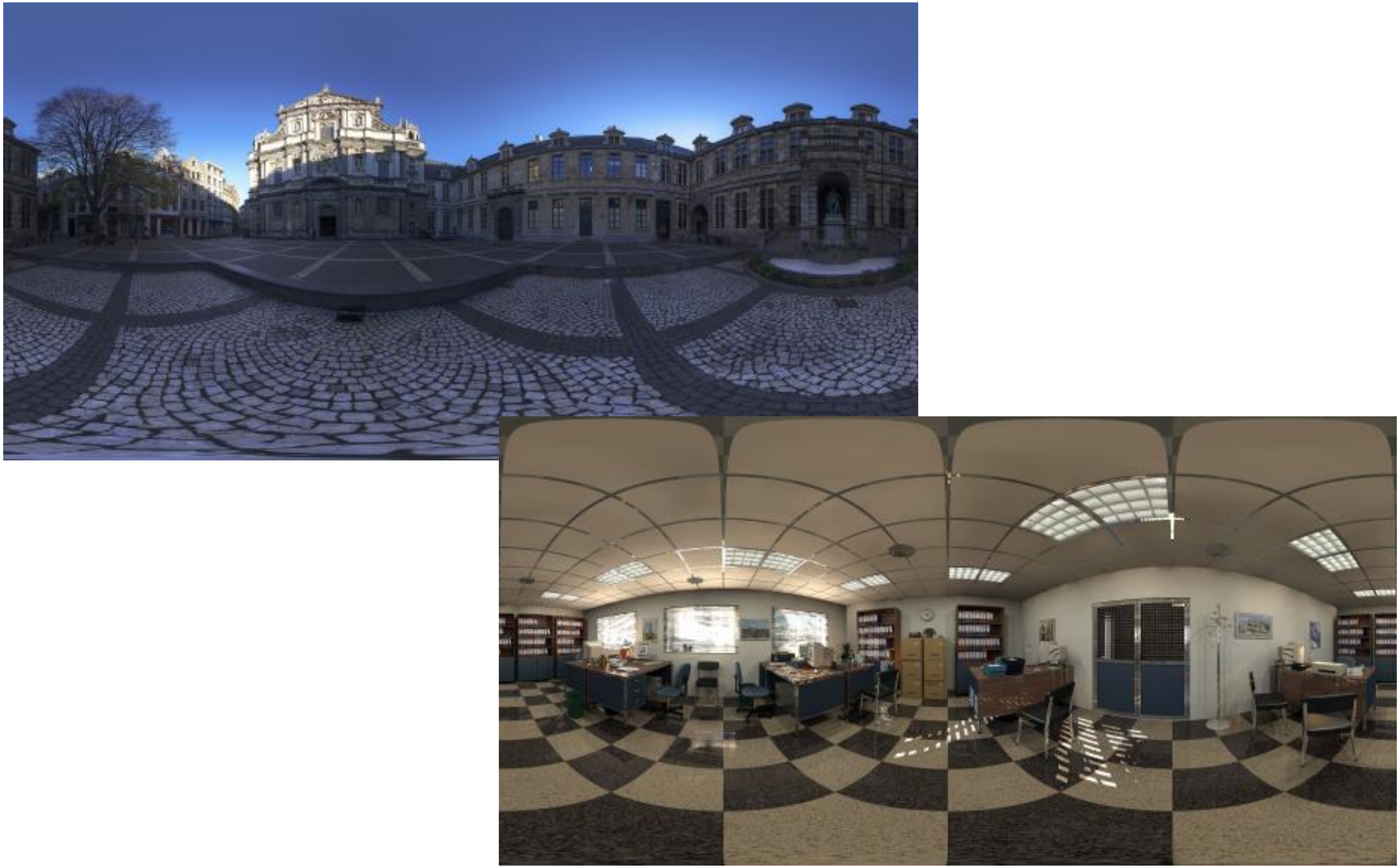
- HDR imagery is often used as a spherical environment map, and so there are some cameras that are designed to capture a full 360 degree spherical image
- The SpheronVR camera is a popular choice for HDR environment acquisition

Environment Acquisition

- There are also some lower-tech solutions that still work pretty well



HDR Environment Maps



Environment Lighting

Environment Lighting

- *Environment lighting* (also called *image based lighting* or IBL) is a technique for rendering objects that are lit by an spherical environment image
- Environment lighting is a very powerful technique that has many applications
- For example, it can be used to integrate computer generated objects into real world scenes (this is sometimes called *live action integration*)

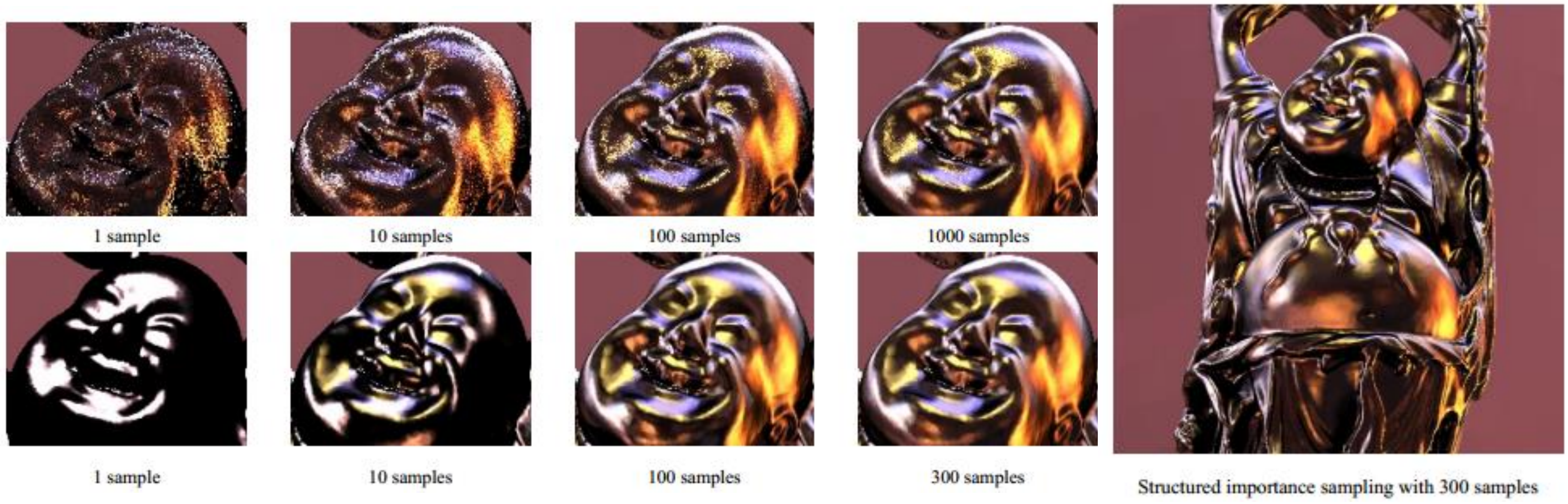
Path Tracing

- We can path trace a scene this way by creating a big textured spherical light source around the scene
- Or a reasonable way might be to modify the code that deals with rays that hit the ‘sky’ and use the ray direction to look up the image pixel
- This is all that is necessary to render objects in a virtual environment, however, it may suffer from the standard path tracing noise, particularly with images that have small, very bright light sources (like most real scenes…)

Live Action Integration

Structured Importance Sampling

- "Structured Importance Sampling of Environment Maps", Agarwal, Ramamoorthi, Belongie, Jensen, 2003



1 sample

10 samples

100 samples

1000 samples

1 sample

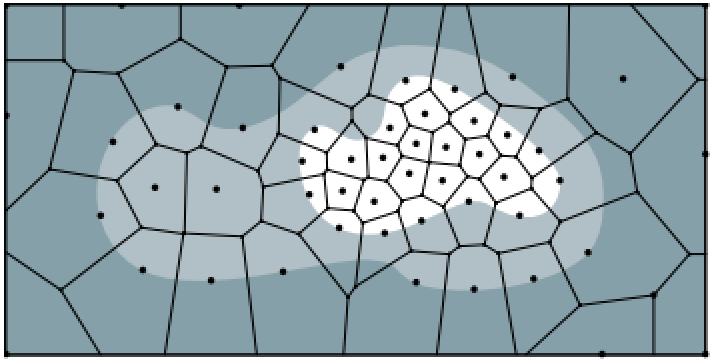
10 samples

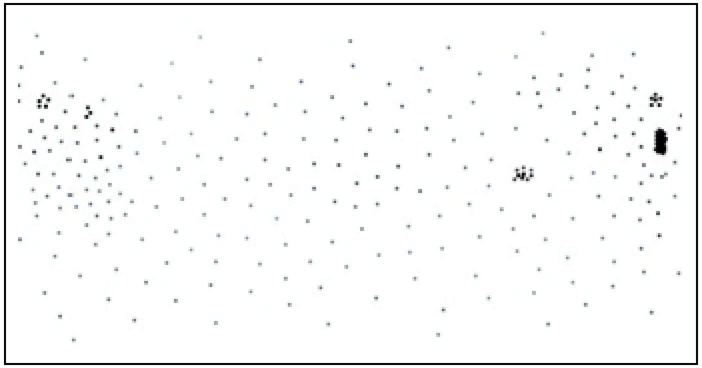
100 samples

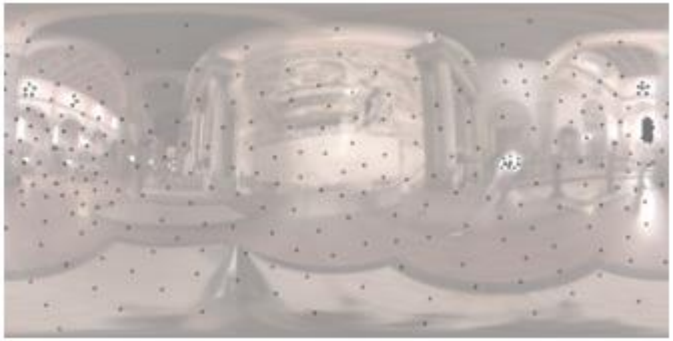
300 samples

Structured importance sampling with 300 samples

Structured Importance Sampling

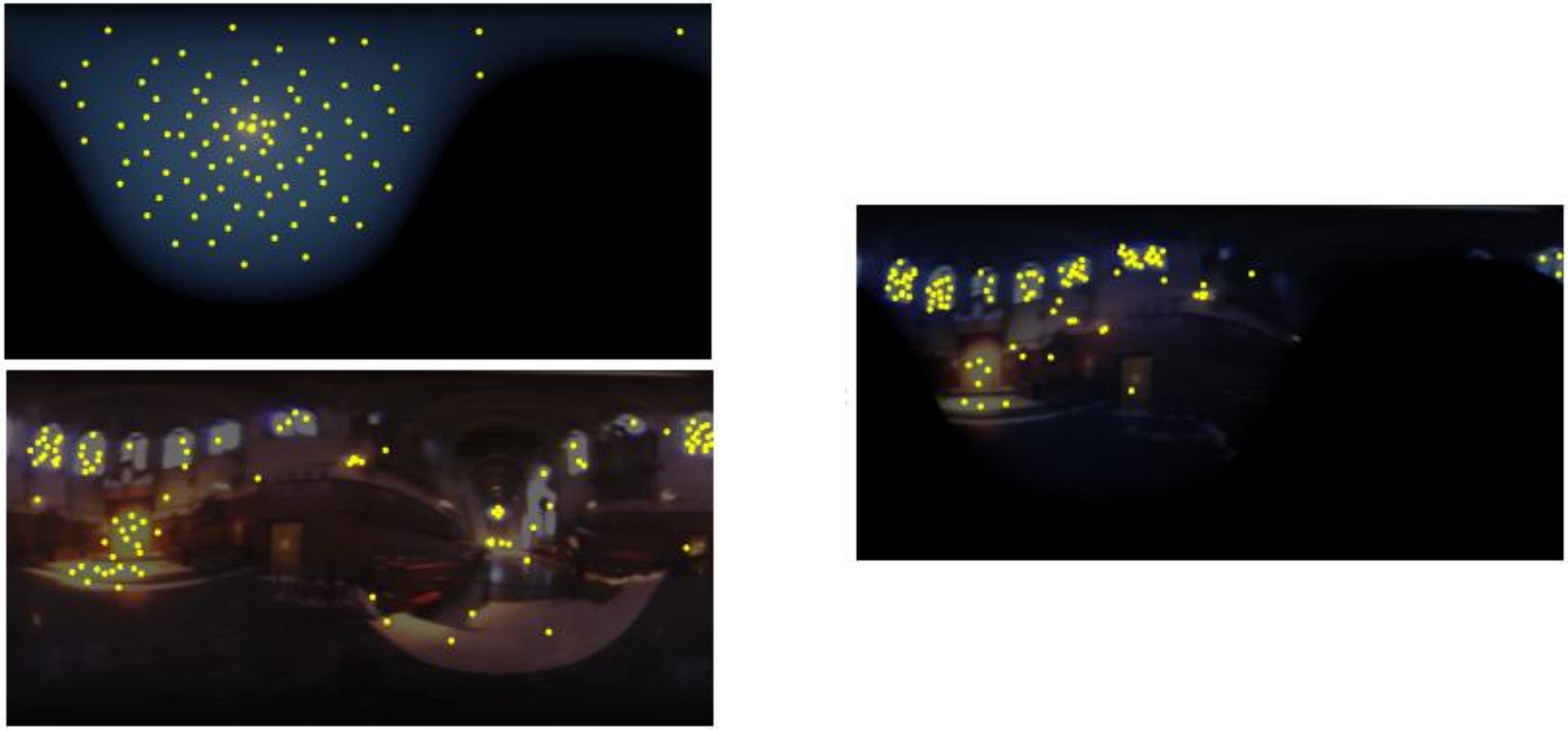






Wavelet Importance Sampling

- "Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions", Clarberg, Jarosz, Moller, Jensen, 2005



Wavelet Importance Sampling