

Image-based Lighting (Part 2)



T2

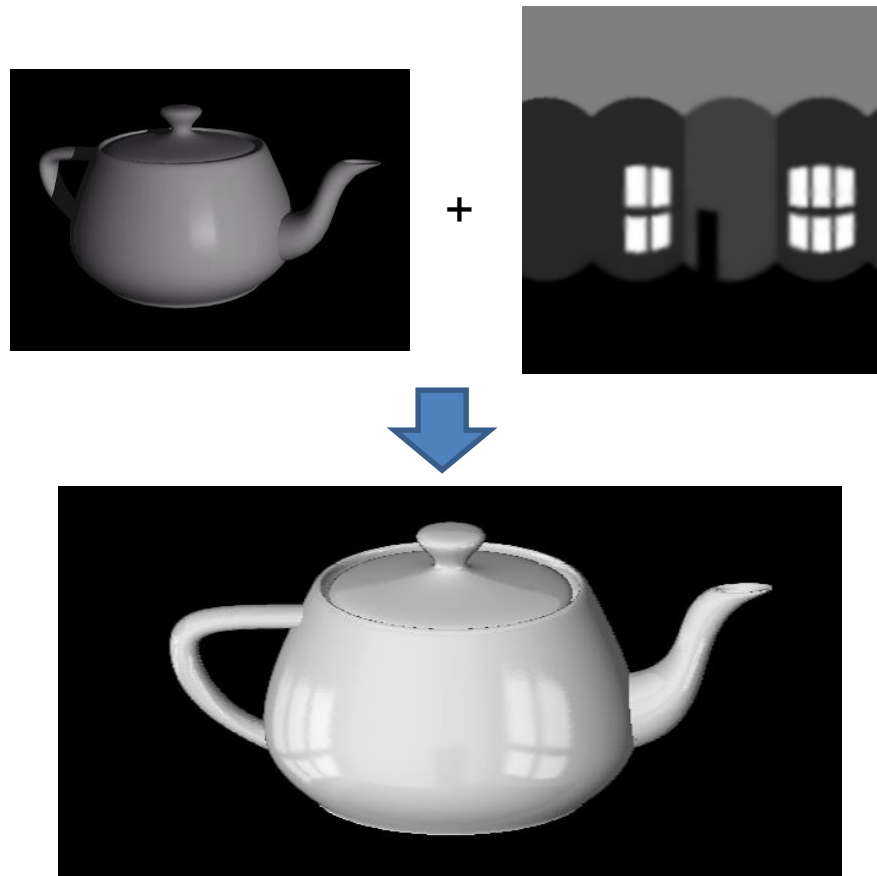
Computational Photography
Derek Hoiem, University of Illinois

Today

- Brief review of last class
- Show how to get an HDR image from several LDR images, and how to display HDR
- Show how to insert fake objects into real scenes using environment maps

Key ideas for Image-based Lighting

- Environment maps: tell what light is entering at each angle within some shell



Spherical Map Example

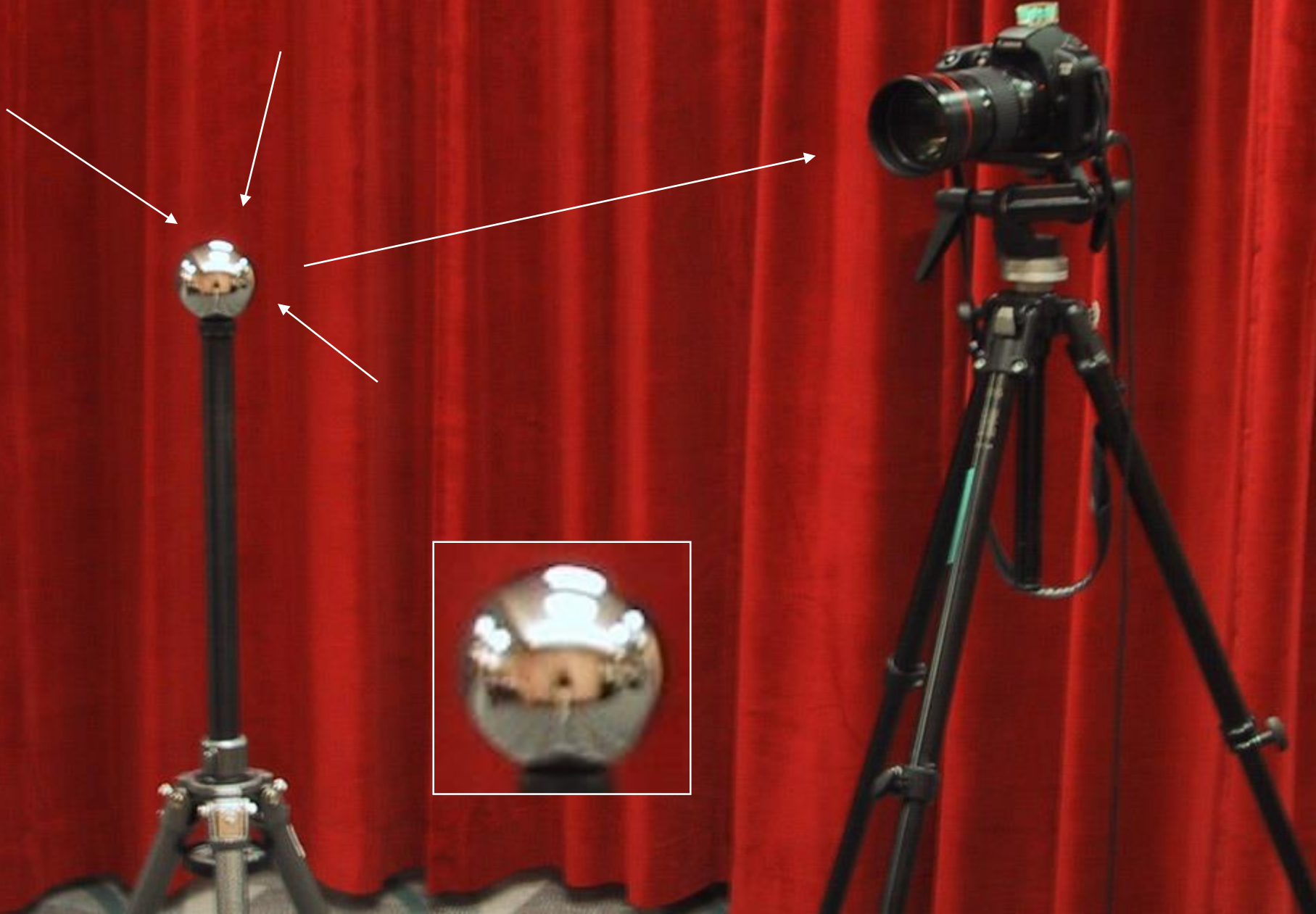


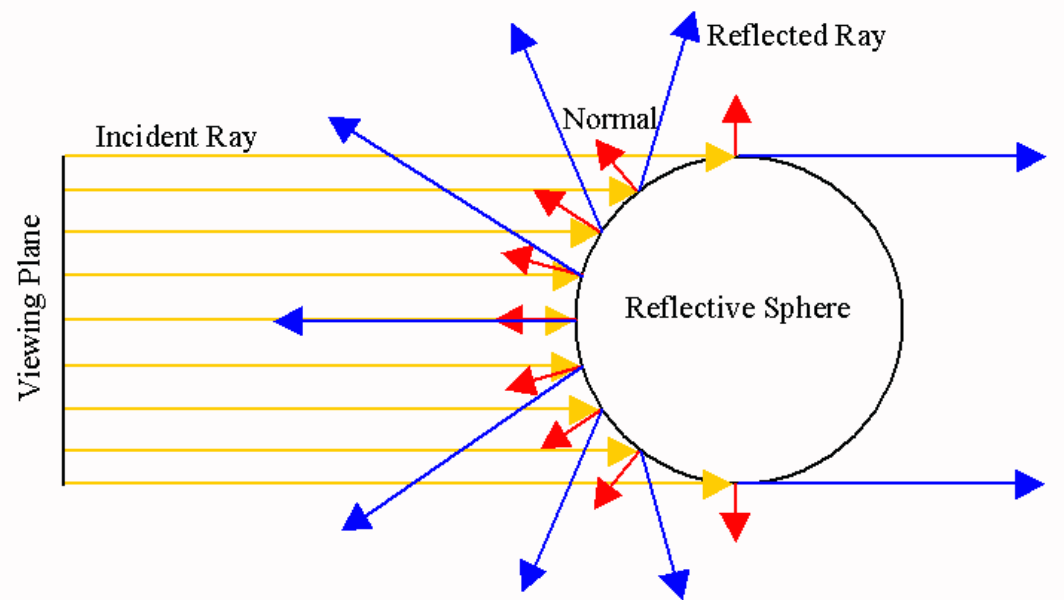
Key ideas for Image-based Lighting

- Light probes: a way of capturing environment maps in real scenes



Mirrored Sphere





- 1) Compute normal of sphere from pixel position
- 2) Compute reflected ray direction from sphere normal
- 3) Convert to spherical coordinates (θ , ϕ)
- 4) Create equirectangular image

Mirror ball -> equirectangular



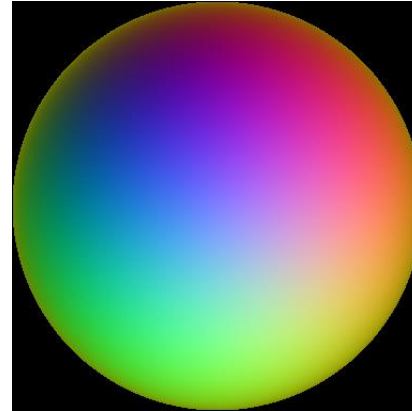
Mirror ball -> equirectangular



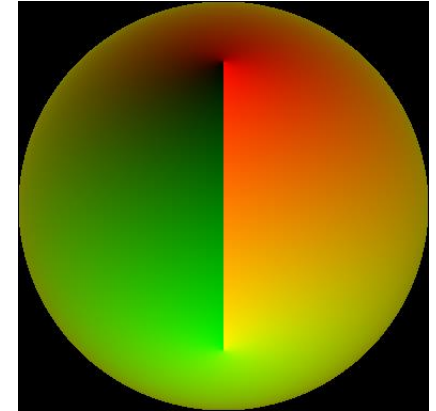
Mirror ball



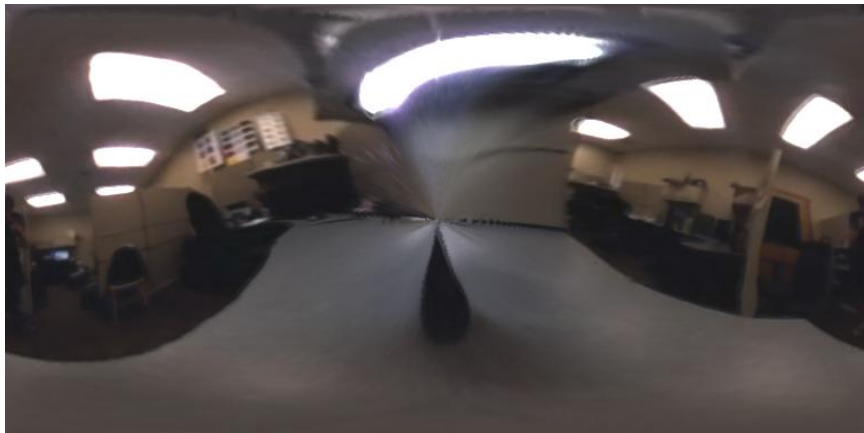
Normals



Reflection
vectors



Phi/theta of
reflection vecs



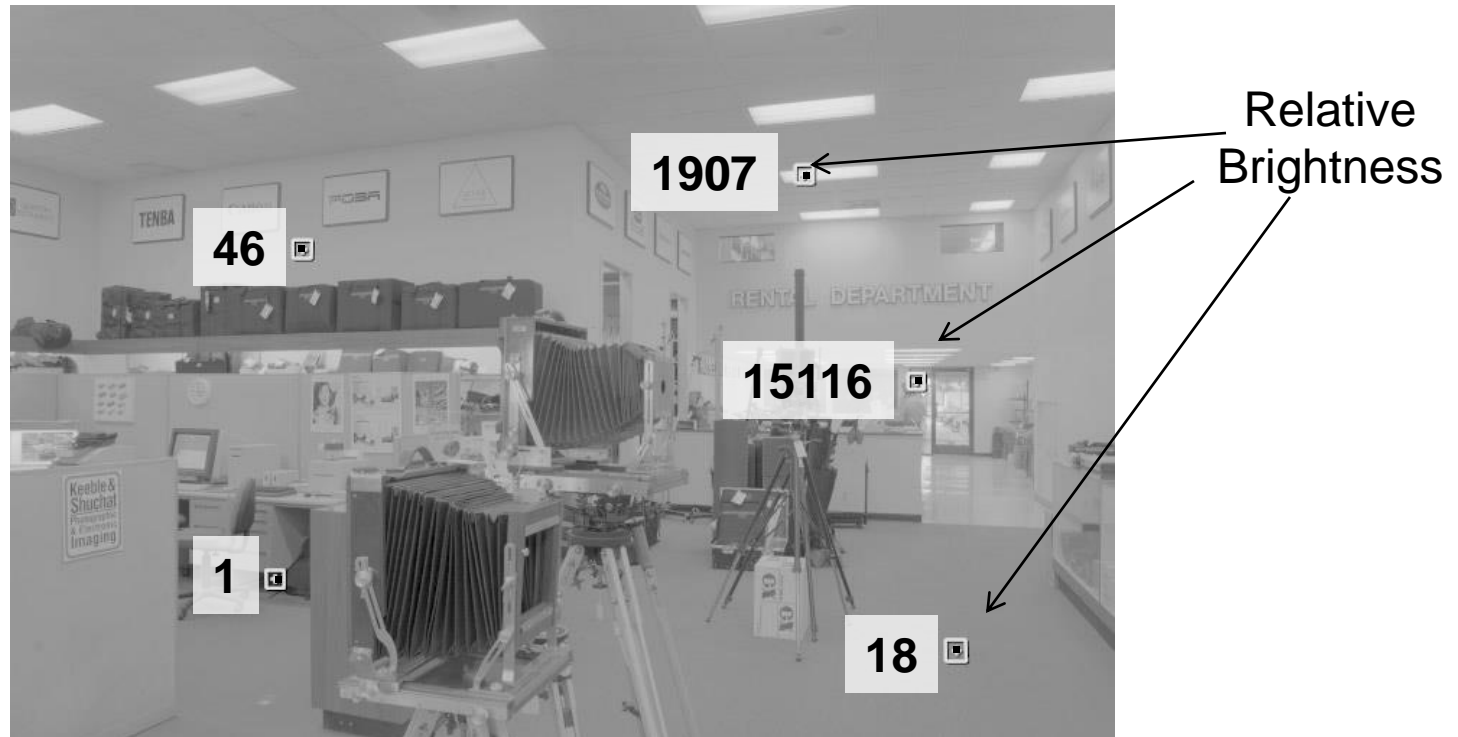
Equirectangular



Phi/theta equirectangular
domain

One small snag

- How do we deal with light sources? Sun, lights, etc?
 - They are much, much brighter than the rest of the environment



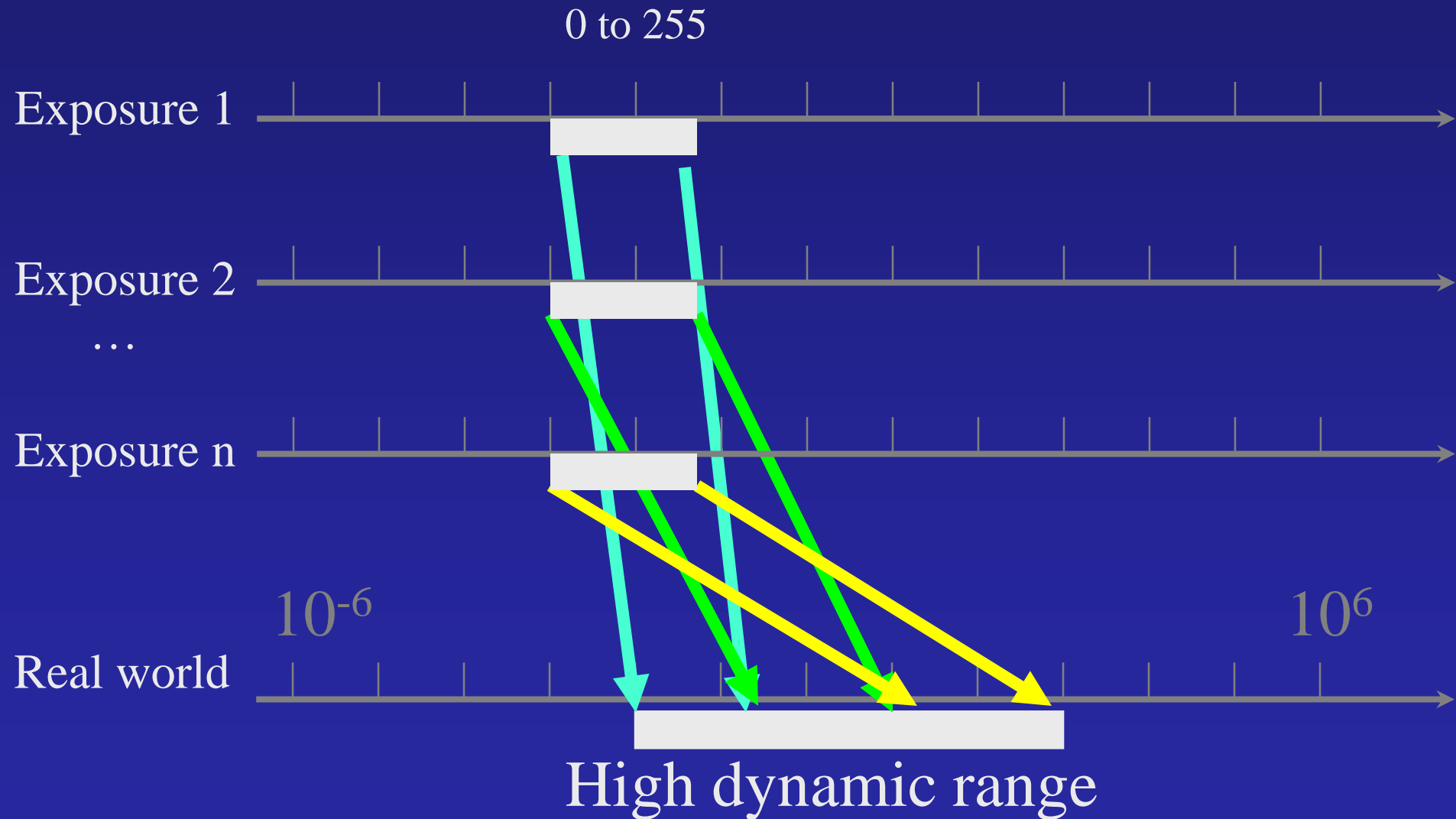
- Use High Dynamic Range photography!

Key ideas for Image-based Lighting

- Capturing HDR images: needed so that light probes capture full range of radiance



LDR->HDR by merging exposures



Ways to vary exposure

- Shutter Speed (*)
- F/stop (aperture, iris)
- Neutral Density (ND) Filters



Recovering High Dynamic Range Radiance Maps from Photographs



Paul Debevec
Jitendra Malik



Computer Science Division
University of California at Berkeley

August 1997

The Approach

- Get pixel values Z_{ij} for image with shutter time Δt_j (i^{th} pixel location, j^{th} image)

- Exposure is irradiance integrated over time:

$$E_{ij} = R_i \cdot \Delta t_j$$

- Pixel values are non-linearly mapped E_{ij} 's:

$$Z_{ij} = f(E_{ij}) = f(R_i \cdot \Delta t_j)$$

- Rewrite to form a (not so obvious) linear system:

$$\ln f^{-1}(Z_{ij}) = \ln(R_i) + \ln(\Delta t_j)$$

$$g(Z_{ij}) = \ln(R_i) + \ln(\Delta t_j)$$

The objective

Solve for radiance R and mapping g for each of 256 pixel values to minimize:

$$\sum_{i=1}^N \sum_{j=1}^P w(Z_{ij}) [\ln R_i + \ln \Delta t_j - g(Z_{ij})]^2 + \lambda \sum_{z=Z_{min}}^{Z_{max}} w(z) g''(z)^2$$

give pixels near 0
or 255 less weight

known shutter time
for image j

exposure should smoothly
increase as pixel intensity
increases

irradiance at particular
pixel site is the same for
each image

exposure, as a function of
pixel value

Matlab Code

```
%
% gsolve.m - Solve for imaging system response function
%
% Given a set of pixel values observed for several pixels in several
% images with different exposure times, this function returns the
% imaging system's response function g as well as the log film irradiance
% values for the observed pixels.
%
% Assumes:
%
%   Zmin = 0
%   Zmax = 255
%
% Arguments:
%
%   Z(i,j) is the pixel values of pixel location number i in image j
%   B(j)   is the log delta t, or log shutter speed, for image j
%   l      is lambda, the constant that determines the amount of smoothness
%   w(z)   is the weighting function value for pixel value z
%
% Returns:
%
%   g(z)   is the log exposure corresponding to pixel value z
%   lE(i)  is the log film irradiance at pixel location i
%
function [g,lE]=gsolve(Z,B,l,w)

n = 256;

A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
b = zeros(size(A,1),1);

%% Include the data-fitting equations

k = 1;
for i=1:size(Z,1)
    for j=1:size(Z,2)
        wij = w(Z(i,j)+1);
        A(k,Z(i,j)+1) = wij;  A(k,n+i) = -wij;      b(k,1) = wij * B(i,j);
        k=k+1;
    end
end

%% Fix the curve by setting its middle value to 0

A(k,129) = 1;
k=k+1;

%% Include the smoothness equations

for i=1:n-2
    A(k,i)=1*w(i+1);      A(k,i+1)=-2*1*w(i+1);  A(k,i+2)=1*w(i+1);
    k=k+1;
end

%% Solve the system using SVD

x = A\b;

g = x(1:n);
lE = x(n+1:size(x,1));
```

Matlab Code

```
function [g,lE]=gsolve(Z,B,l,w)

n = 256;
A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
b = zeros(size(A,1),1);

k = 1;                                %% Include the data-fitting equations
for i=1:size(Z,1)
    for j=1:size(Z,2)
        wij = w(Z(i,j)+1);
        A(k,Z(i,j)+1) = wij; A(k,n+i) = -wij; b(k,1) = wij * B(i,j);
        k=k+1;
    end
end

A(k,129) = 1;                          %% Fix the curve by setting its middle value to 0
k=k+1;

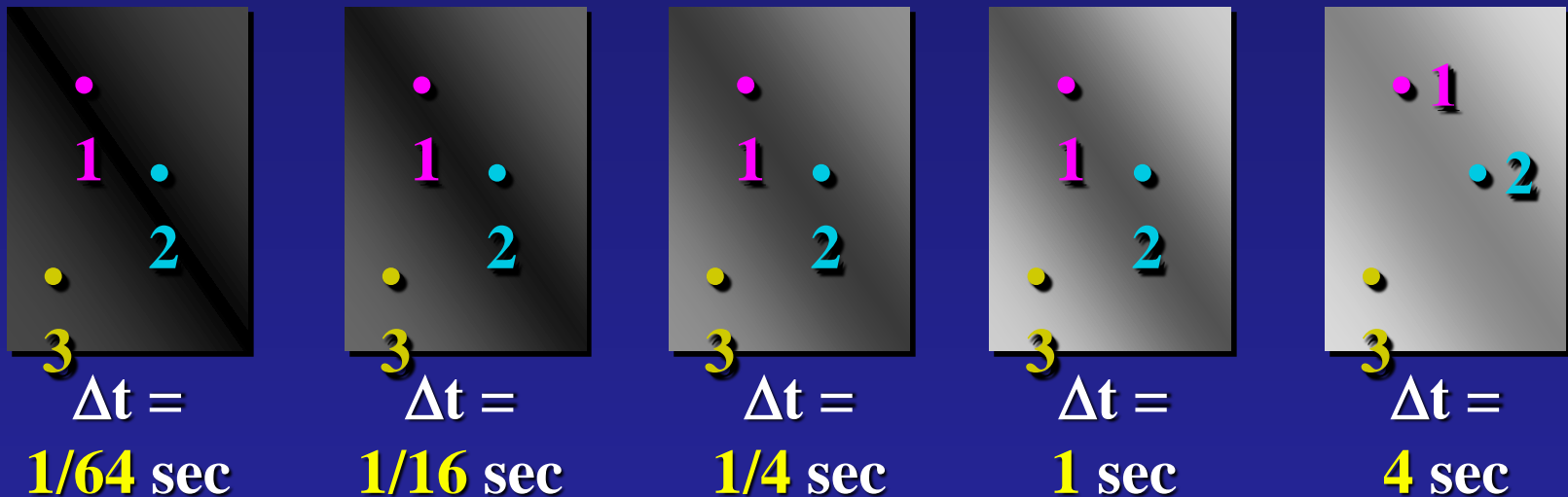
for i=1:n-2                            %% Include the smoothness equations
    A(k,i)=1*w(i+1); A(k,i+1)=-2*l*w(i+1); A(k,i+2)=1*w(i+1);
    k=k+1;
end

x = A\b;                                %% Solve the system using pseudoinverse

g = x(1:n);
lE = x(n+1:size(x,1));
```

Illustration

Image series



Pixel Value $Z = f(\text{Exposure})$

$\text{Exposure} = \text{Radiance} * \Delta t$

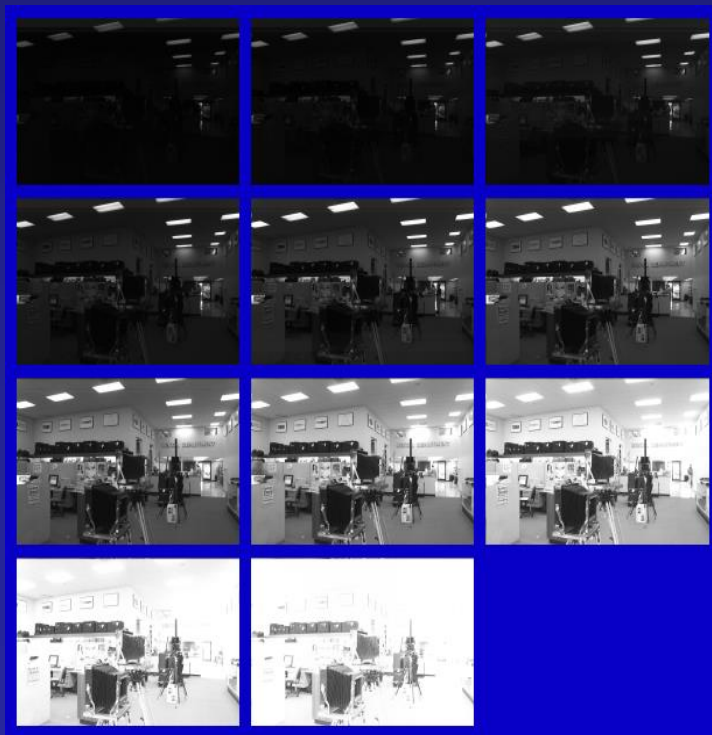
$\log \text{Exposure} = \log \text{Radiance} + \log \Delta t$

Results: Digital Camera

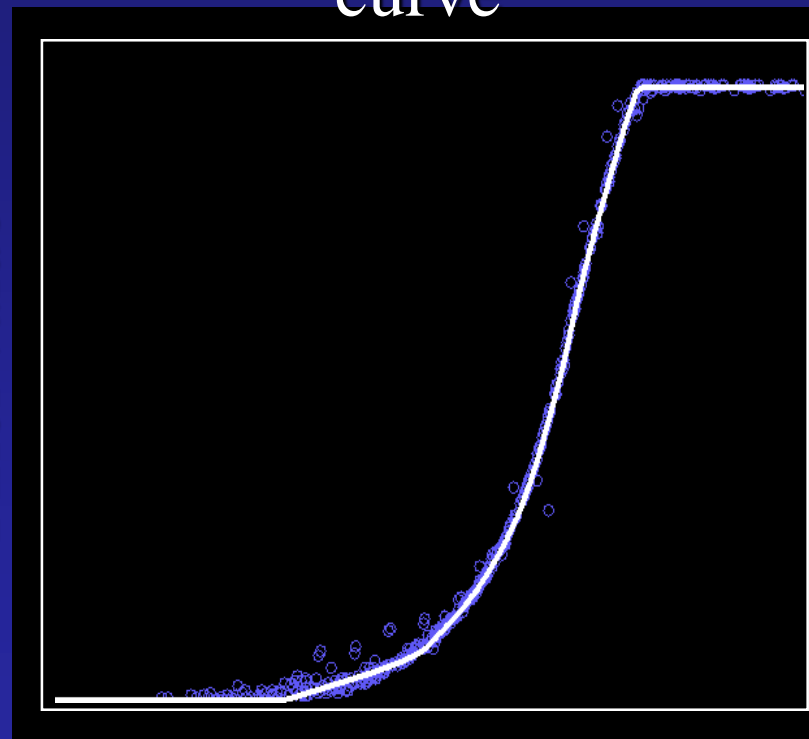
Kodak DCS460

1/30 to 30 sec

Recovered response
curve



Pixel value



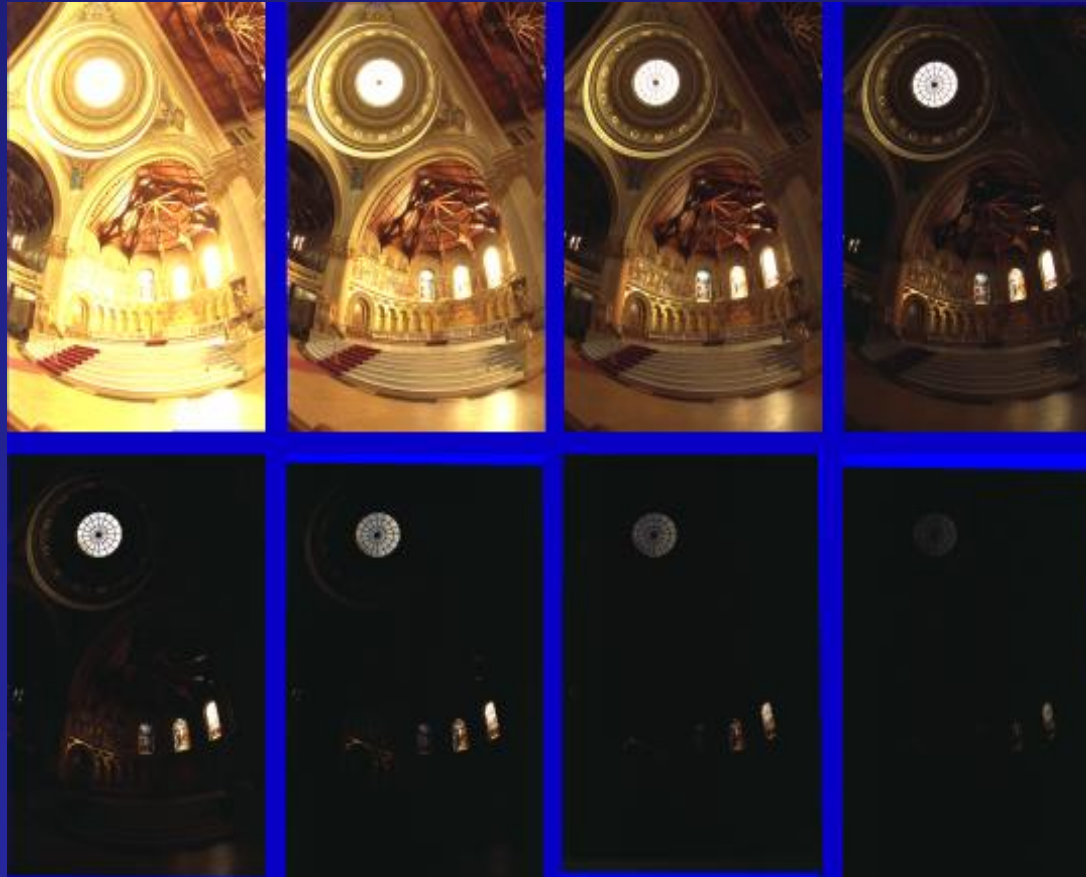
log Exposure

Reconstructed radiance map

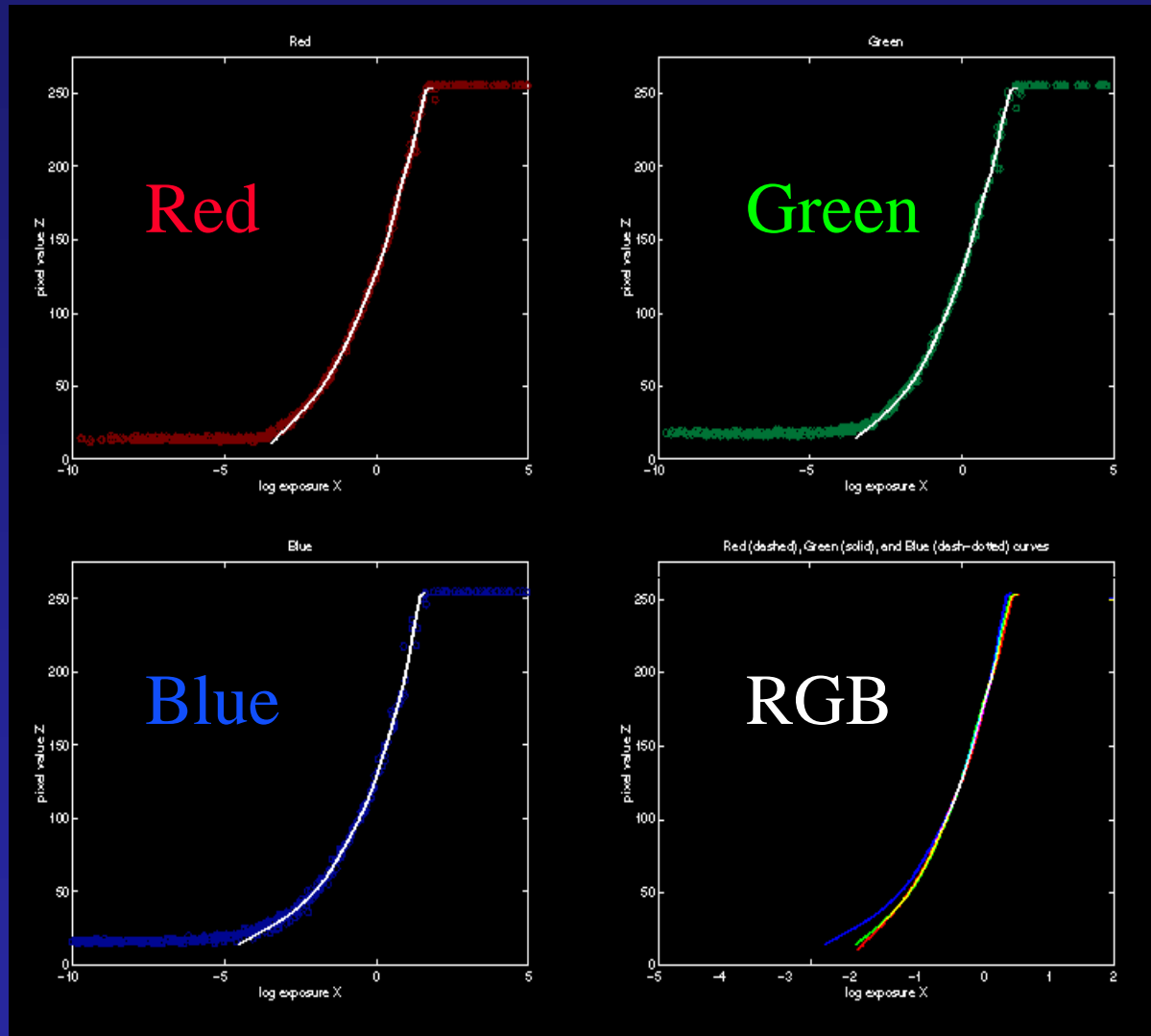


Results: Color Film

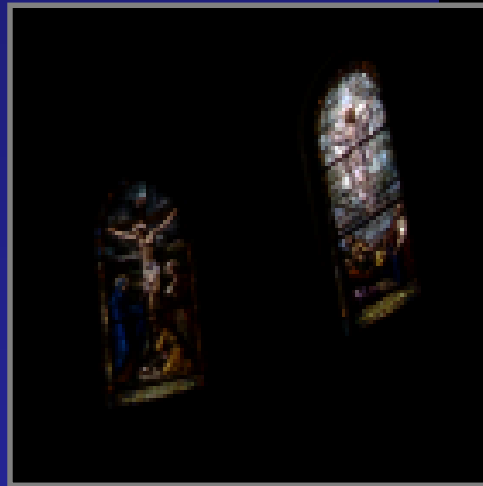
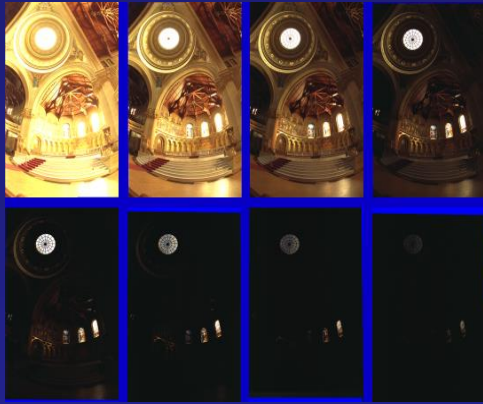
- Kodak Gold ASA 100, PhotoCD



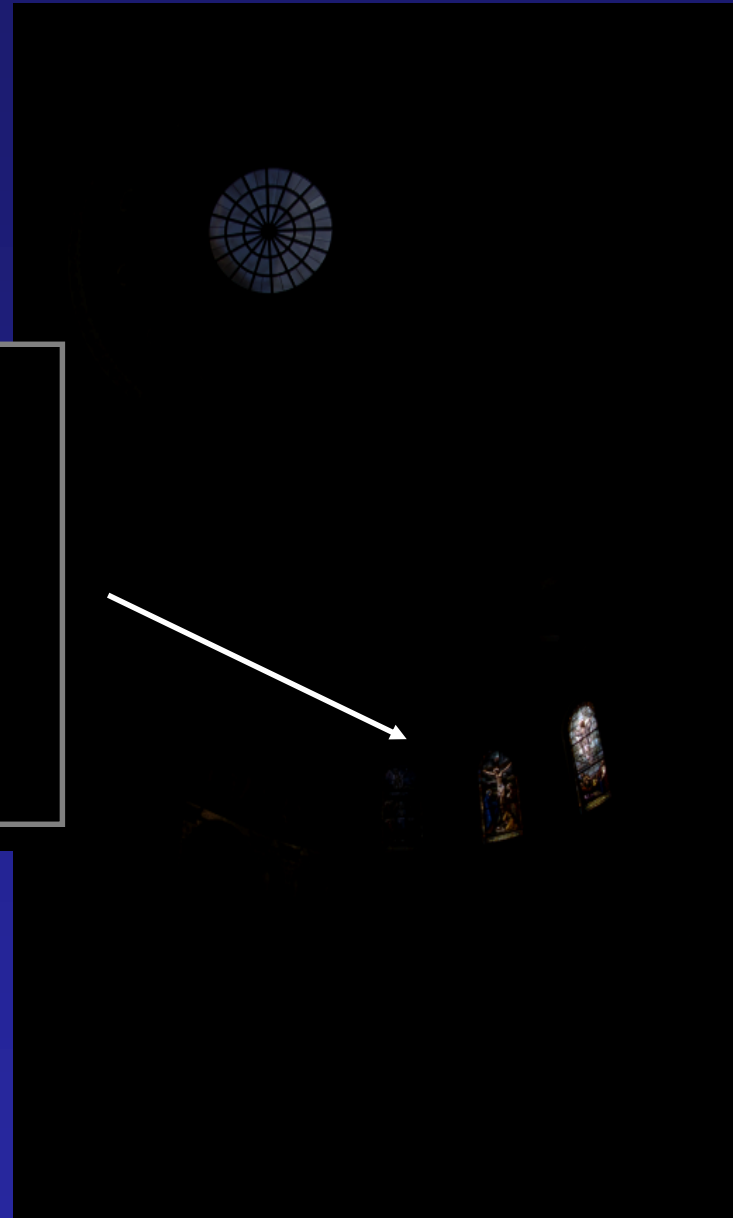
Recovered Response Curves



How to display HDR?

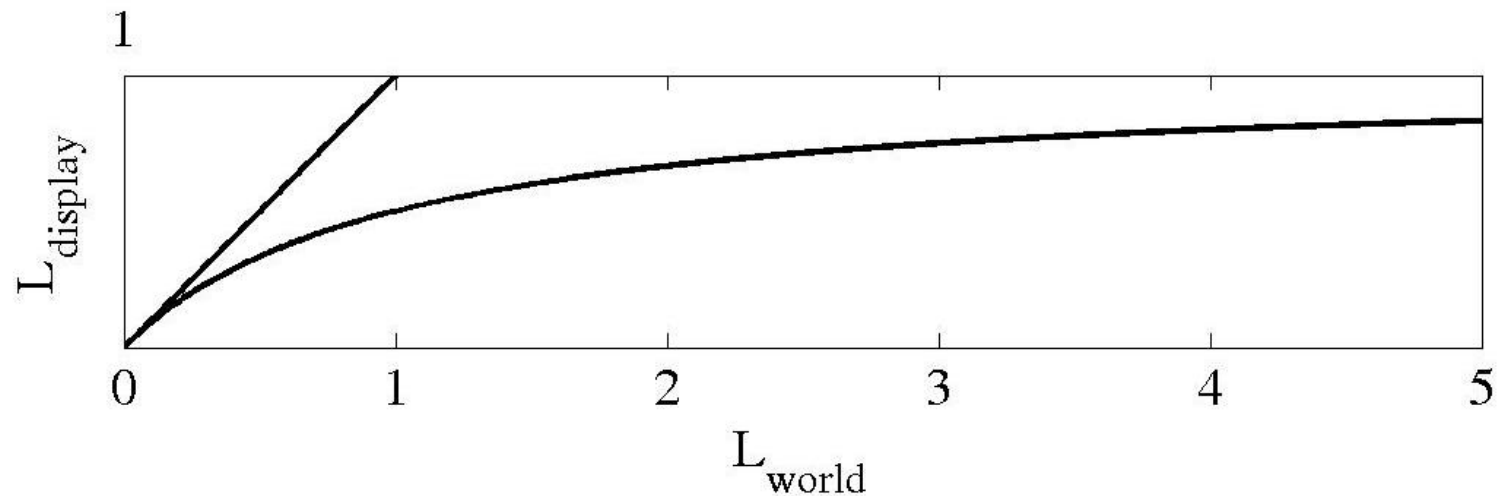


Linearly scaled to
display device



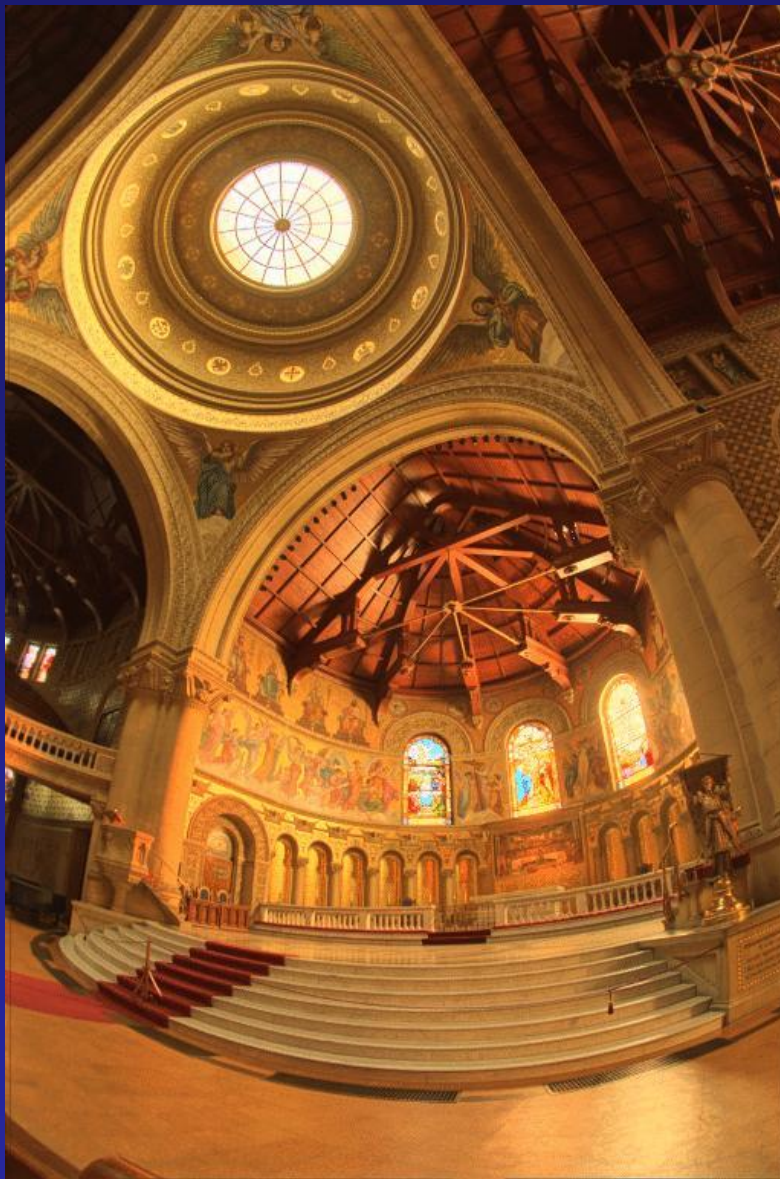
Global Operator (Reinhart et al)

$$L_{display} = \frac{L_{world}}{1 + L_{world}}$$

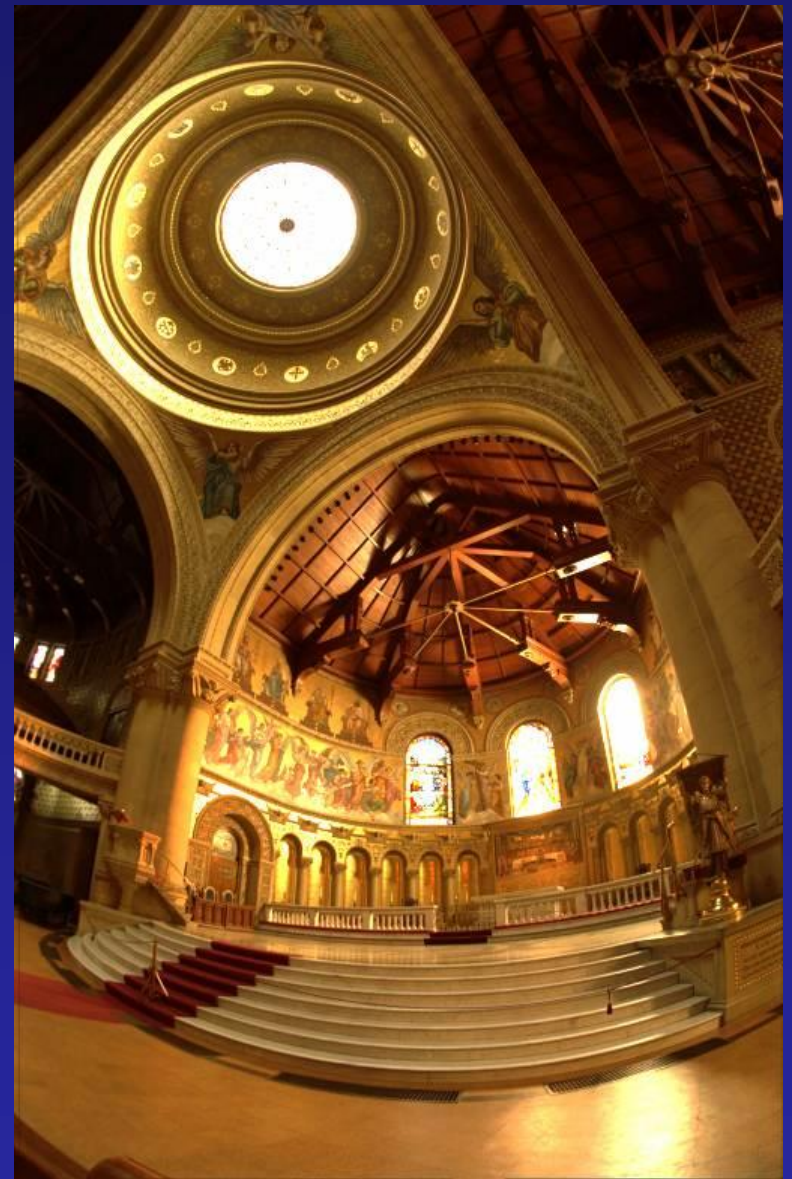


Global Operator Results



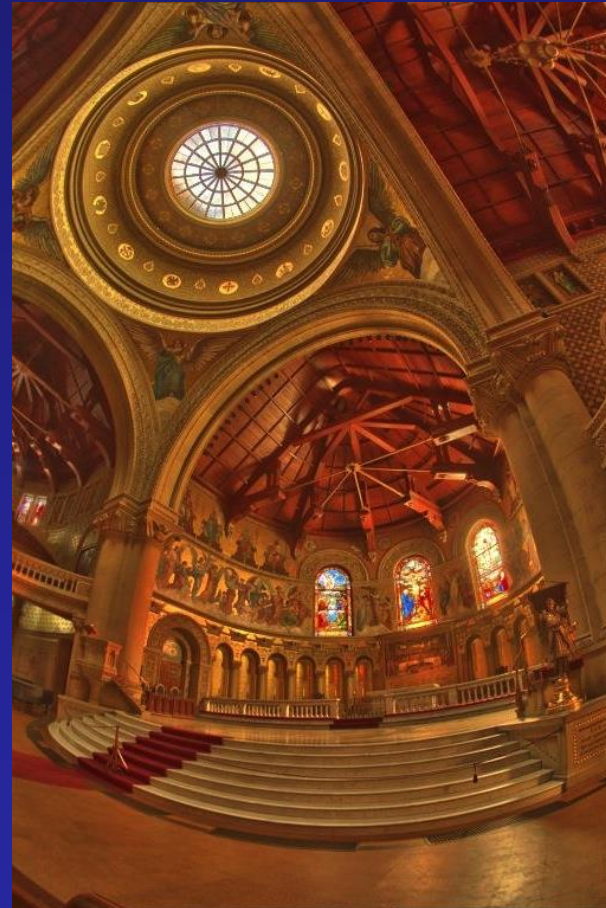
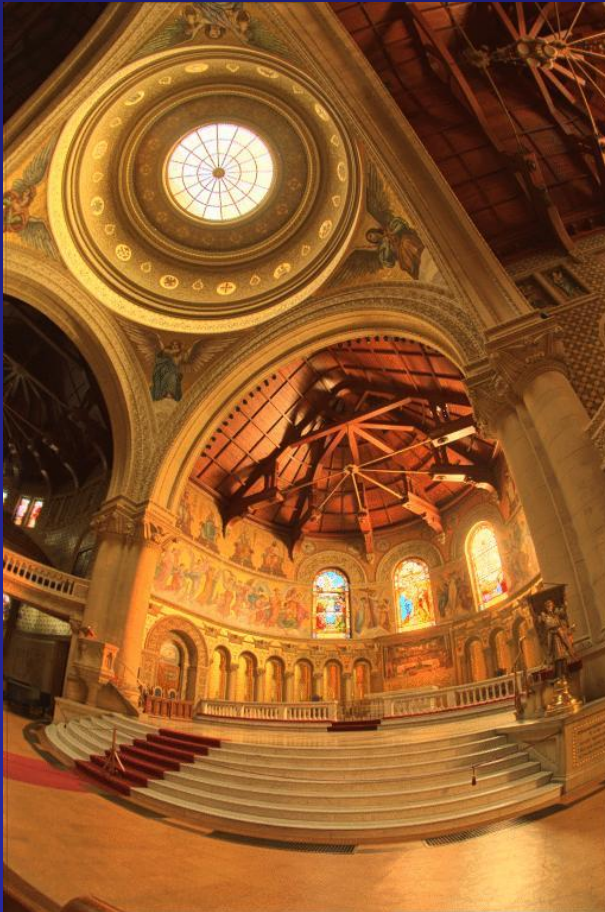


Reinhart Operator

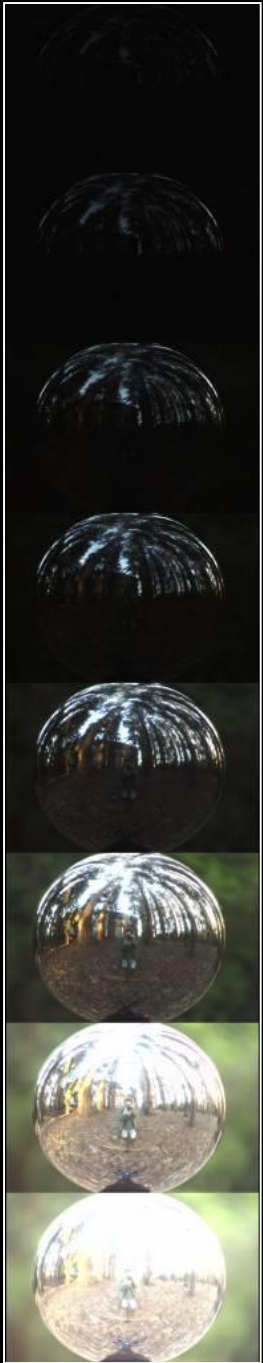
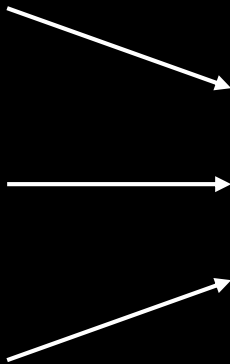
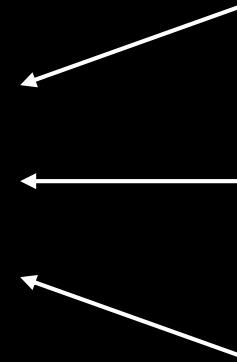


Darkest 0.1% scaled
to display device

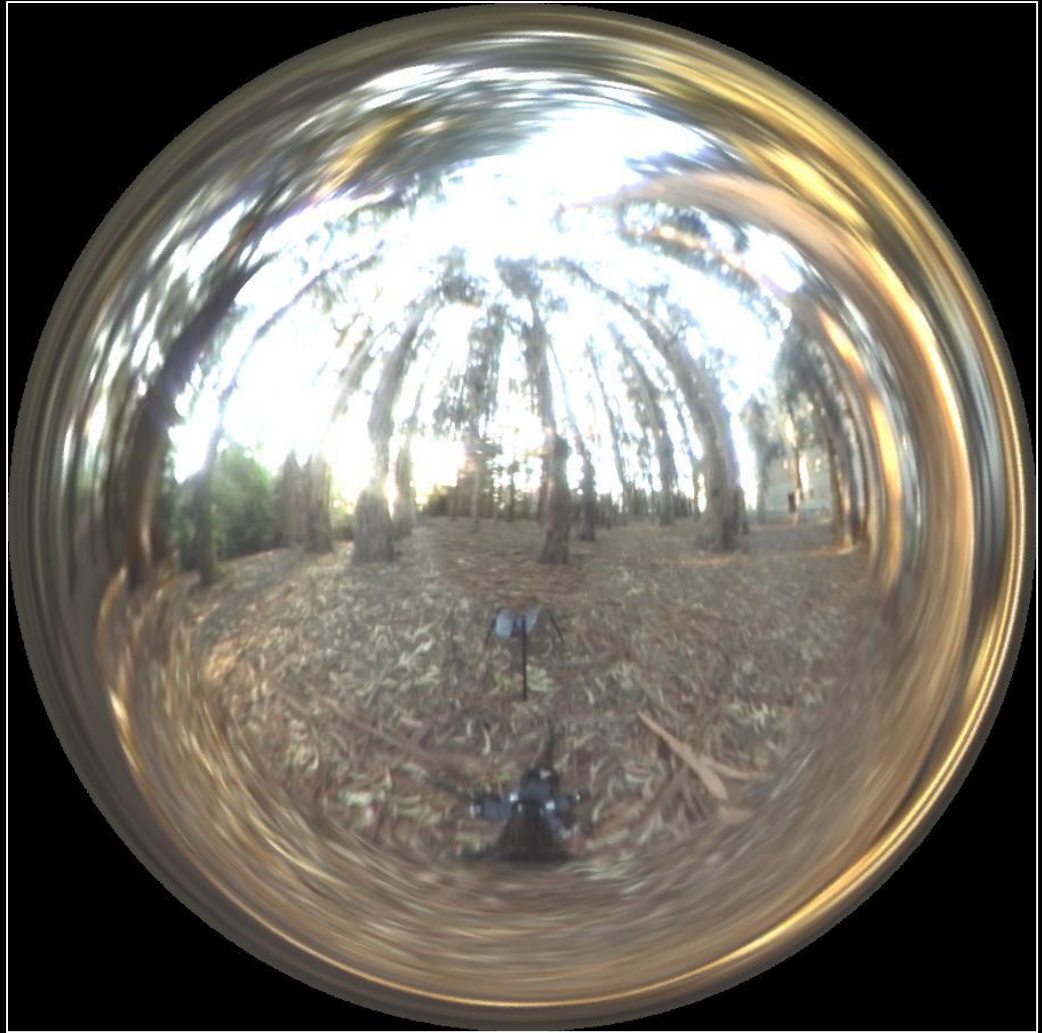
Local operator



Acquiring the Light Probe



Assembling the Light Probe



Real-World HDR Lighting Environments

Funston
Beach



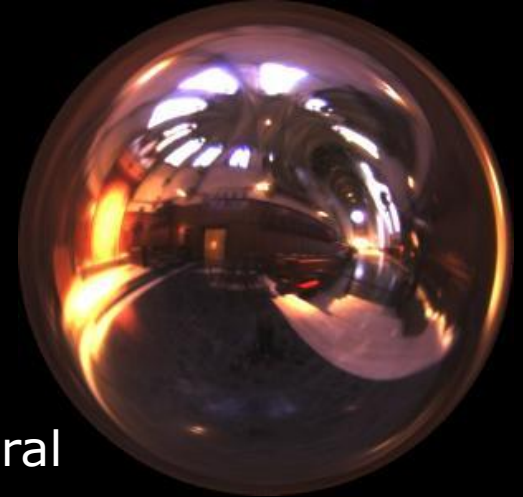
Eucalyptus
Grove



Uffizi
Gallery



Grace
Cathedral



Lighting Environments from the Light Probe Image Gallery:
<http://www.debevec.org/Probes/>

Illumination Results



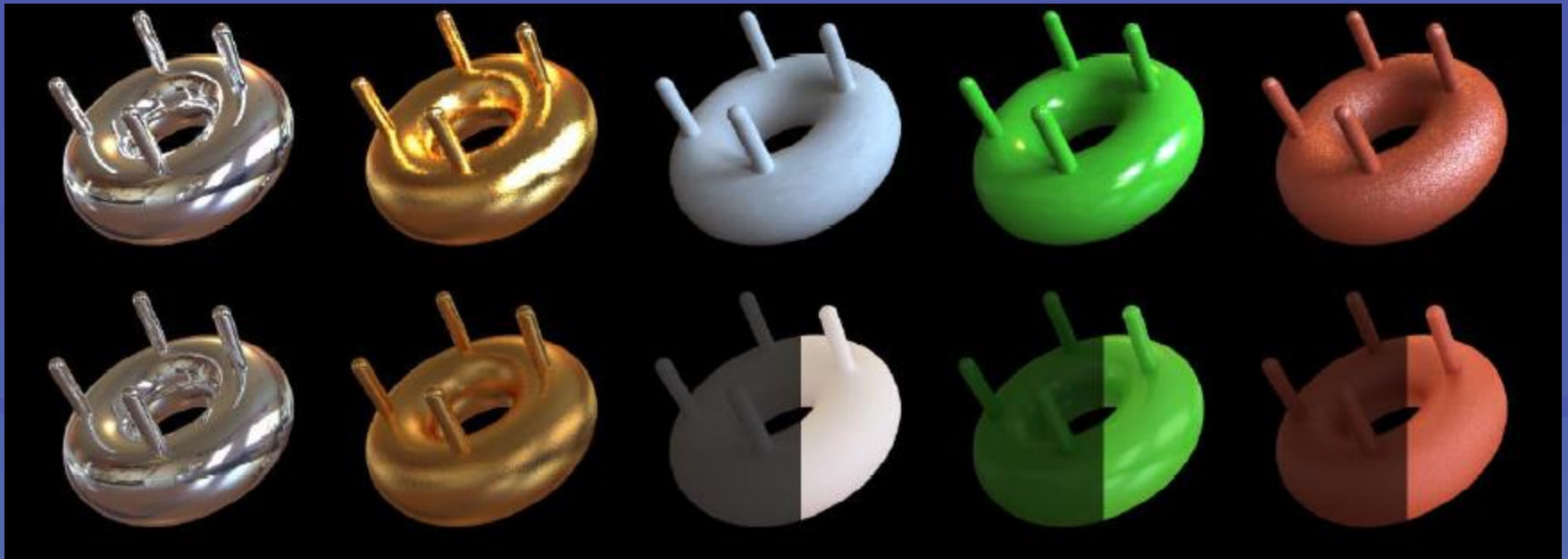
Comparison: Radiance map versus single image



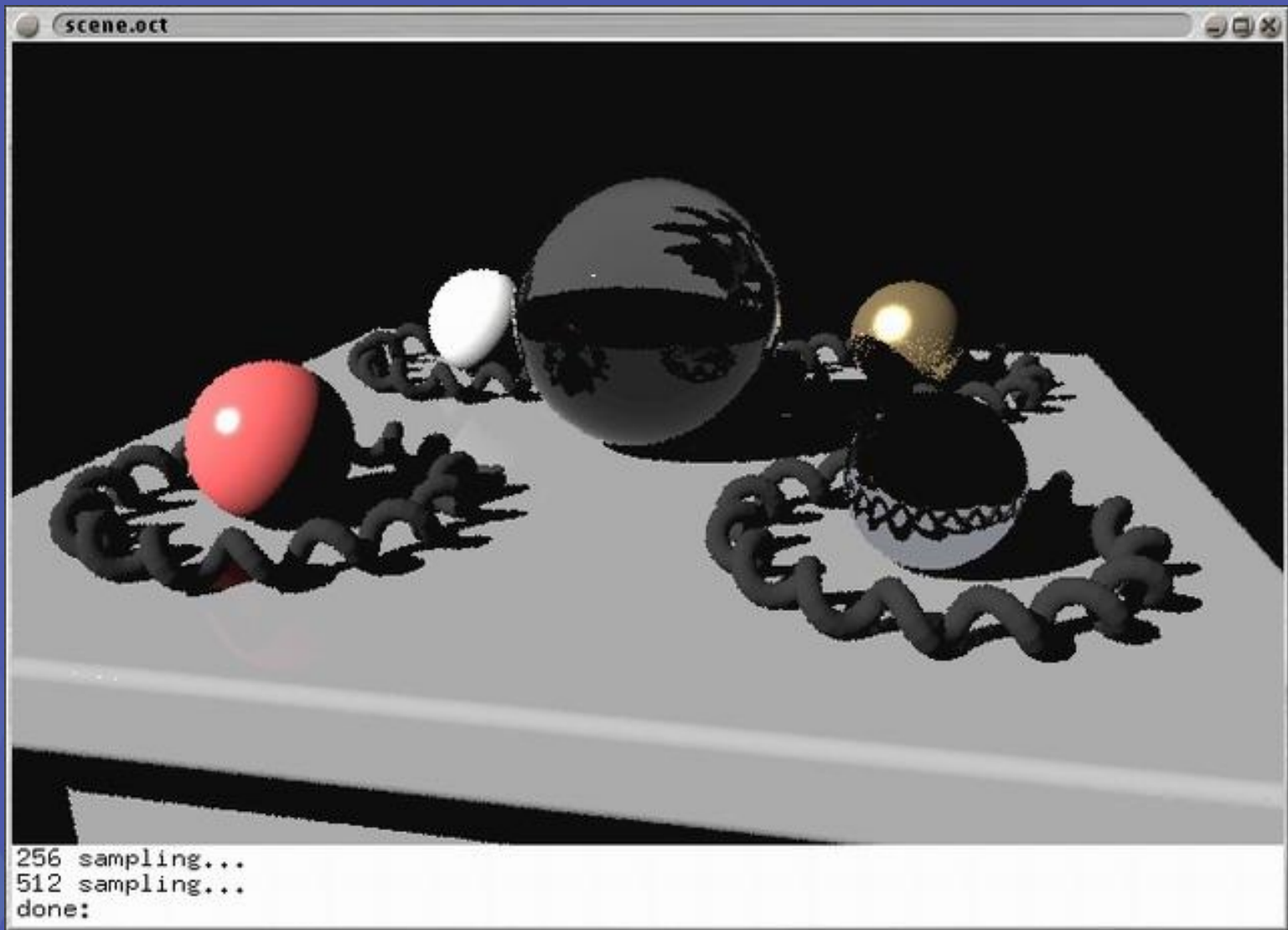
SIGGRAPH2004



HDR



LDR



H2004

CG Objects Illuminated by a Traditional CG
Light Source

Illuminating Objects using Measurements of Real Light

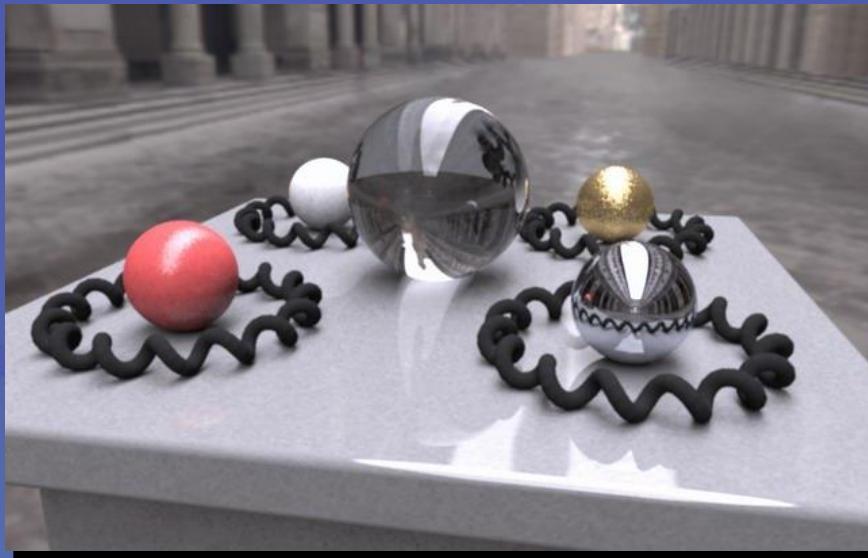


SIGGRAPH2004



Environment
assigned "glow"
material
property in
Greg Ward's
RADIANCE
system.

<http://radsite.lbl.gov/radiance/>



Paul Debevec. A Tutorial on Image-Based Lighting. IEEE Computer Graphics and Applications, Jan/Feb 2002.

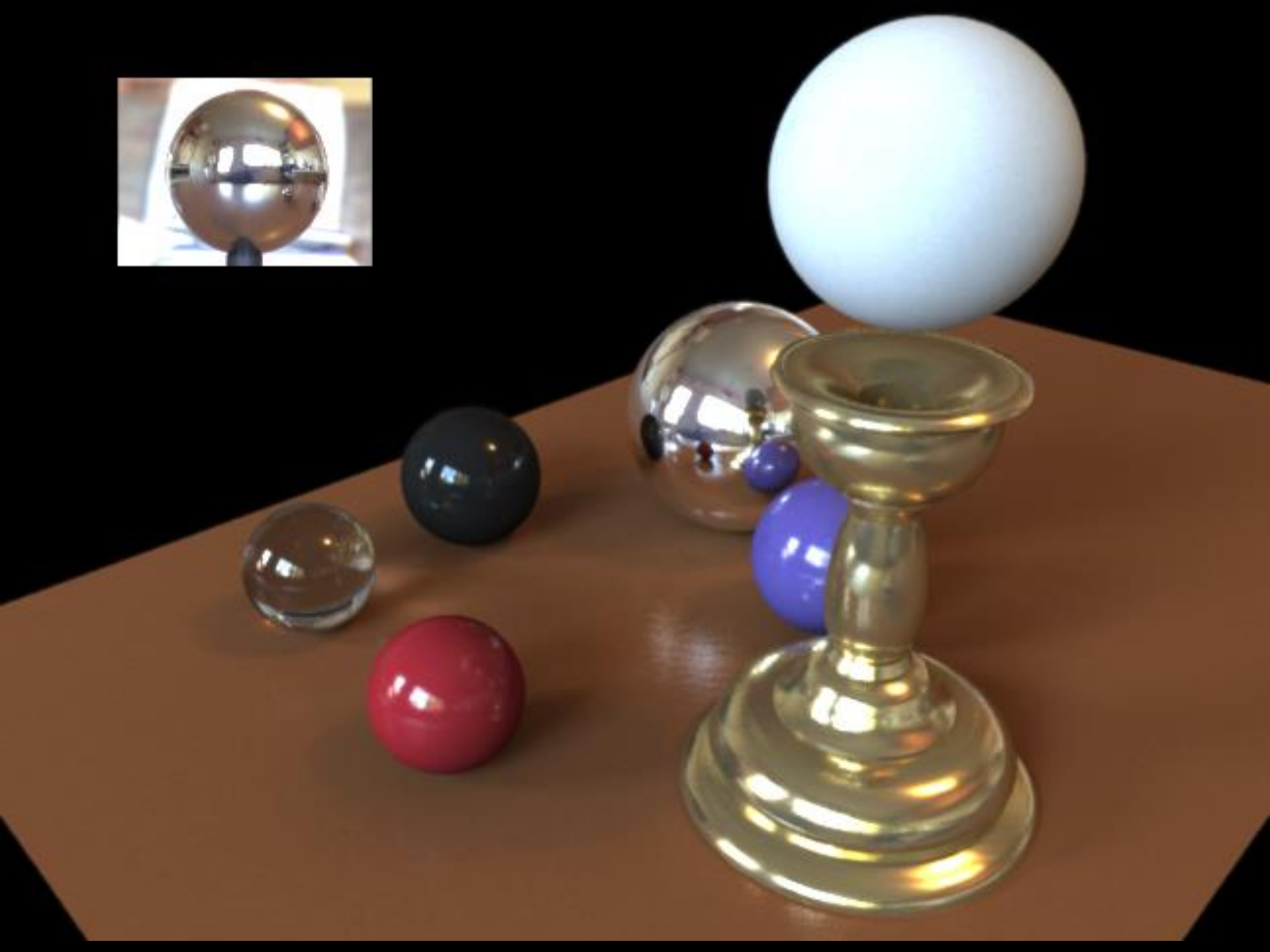
Rendering with Natural Light



SIGGRAPH 98 Electronic Theater

Movie

- <http://www.youtube.com/watch?v=EHBgkeXH9IU>





We can now illuminate
synthetic objects with real light.

- Environment map
- Light probe
- HDR
- Ray tracing

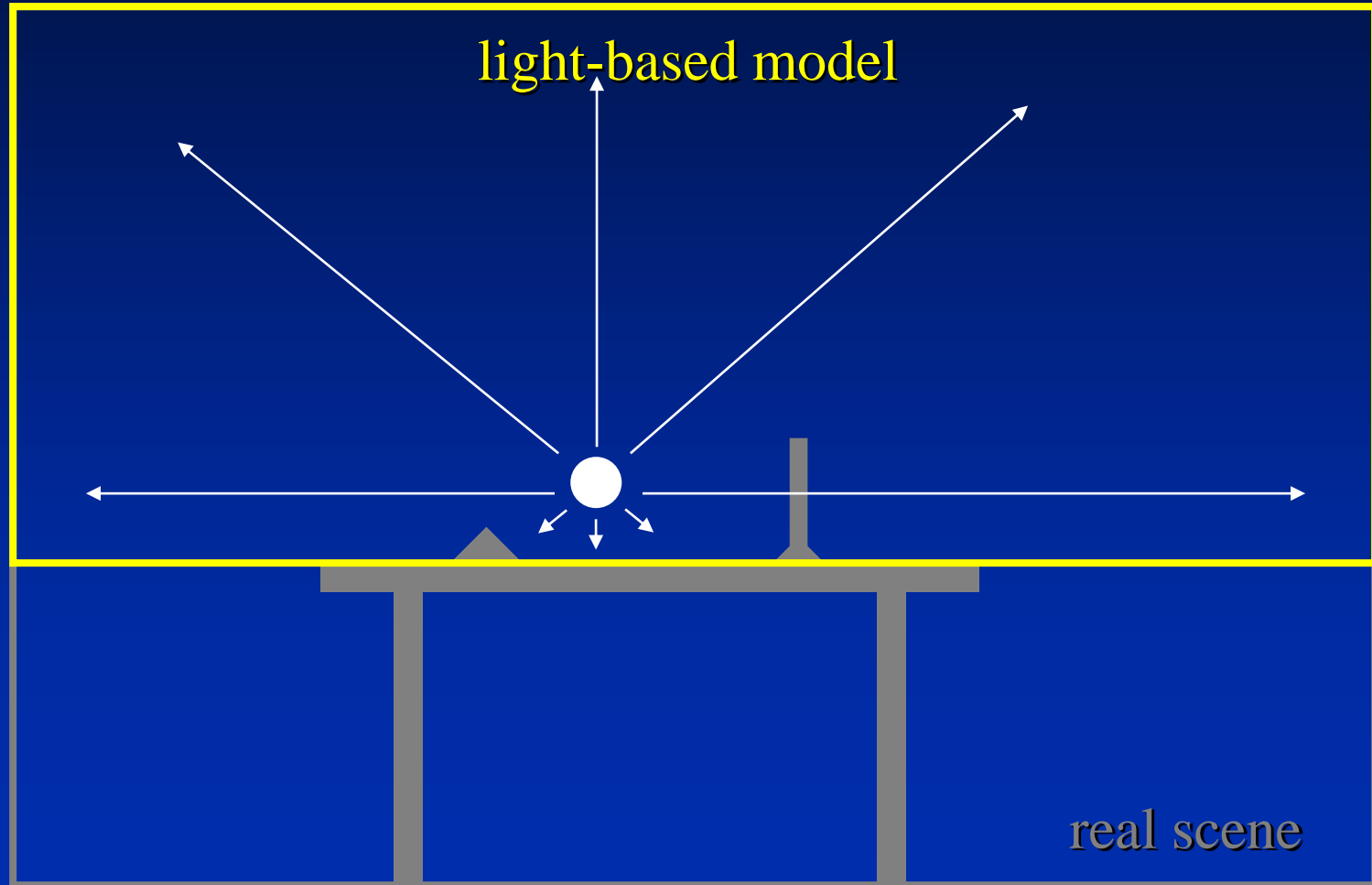
How do we add synthetic objects to a
real scene?

Real Scene Example

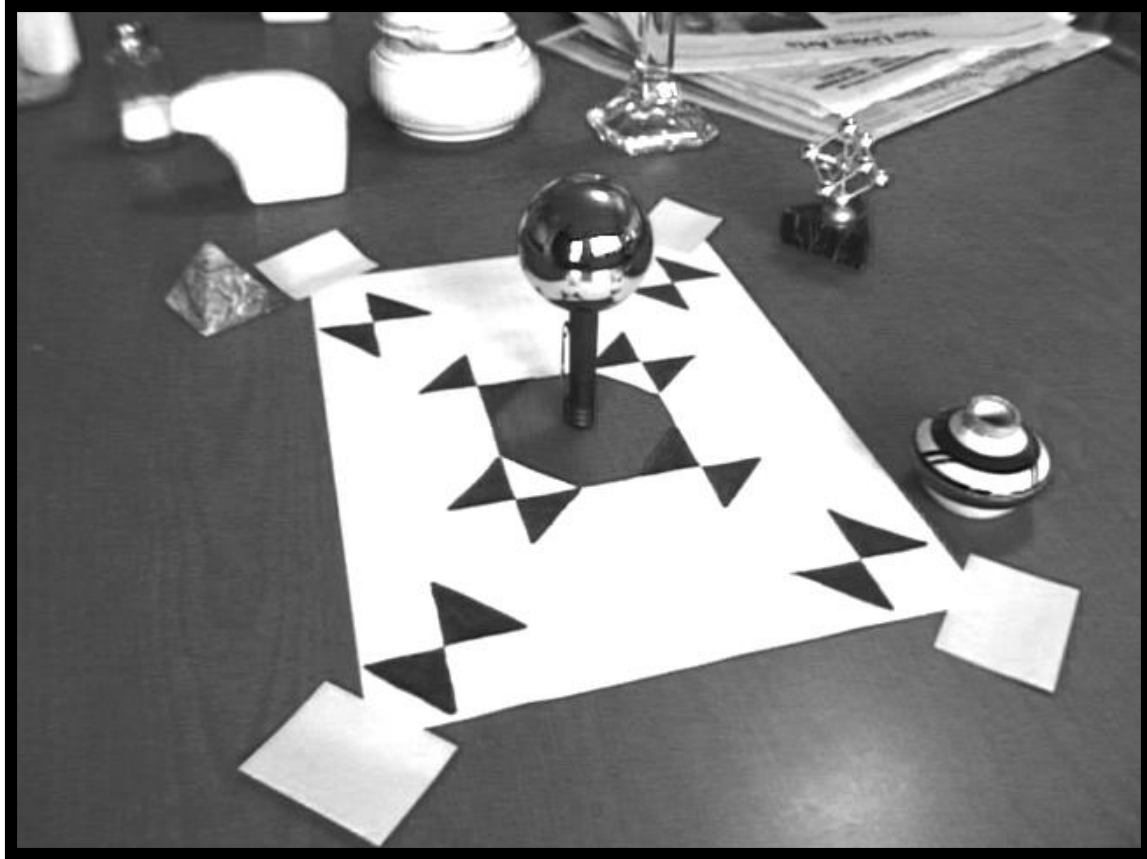


Goal: place synthetic objects on table

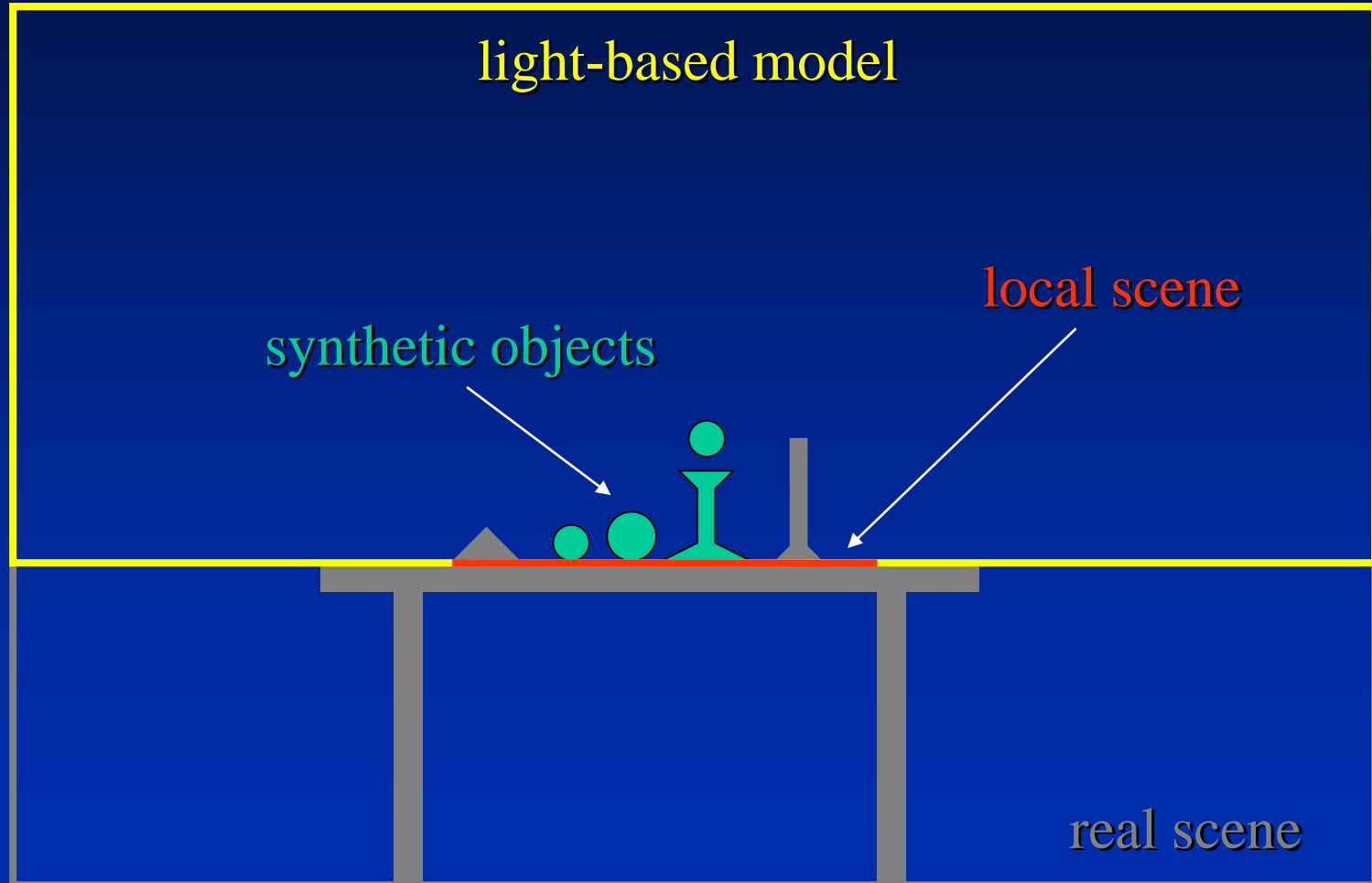
Modeling the Scene



Light Probe / Calibration Grid



Modeling the Scene

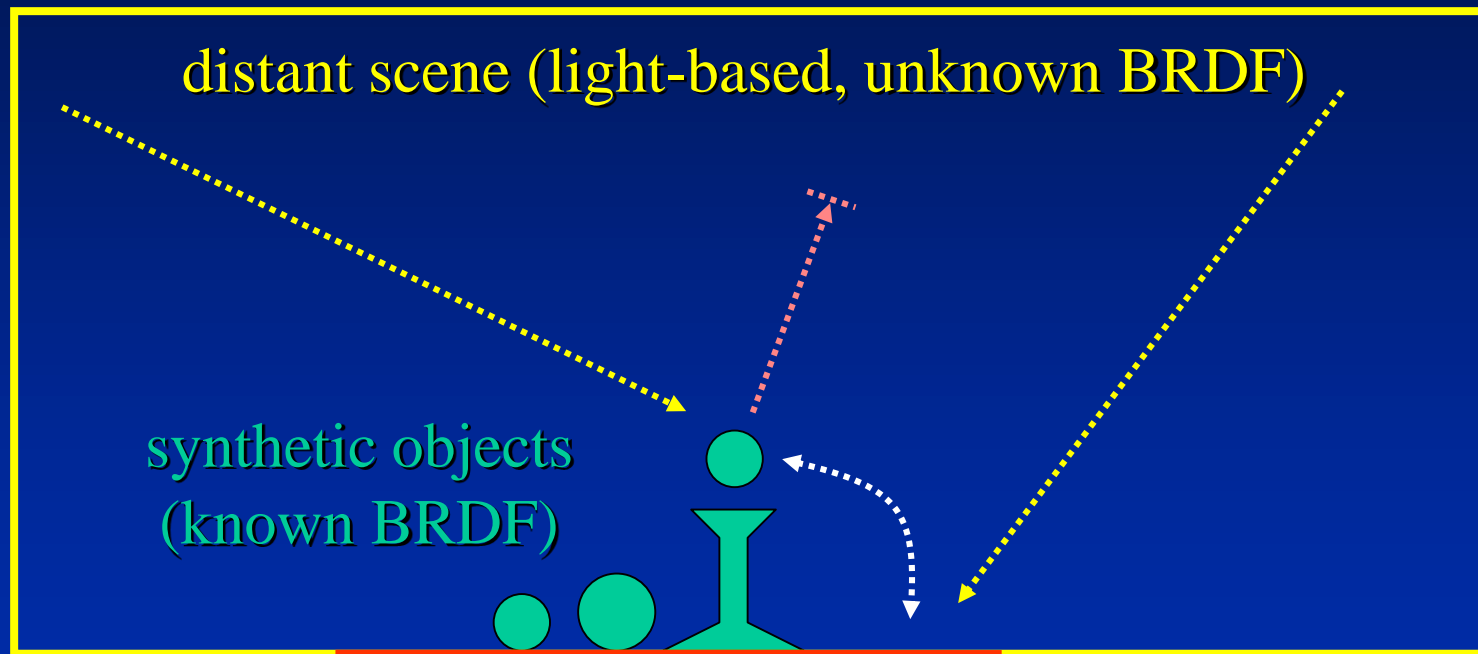


Differential Rendering



Local scene w/o objects, illuminated by model

The Lighting Computation



local scene
(estimated BRDF)

Rendering into the Scene



Background Plate

Rendering into the Scene



Objects and Local Scene matched to Scene

Differential Rendering

Difference in local scene

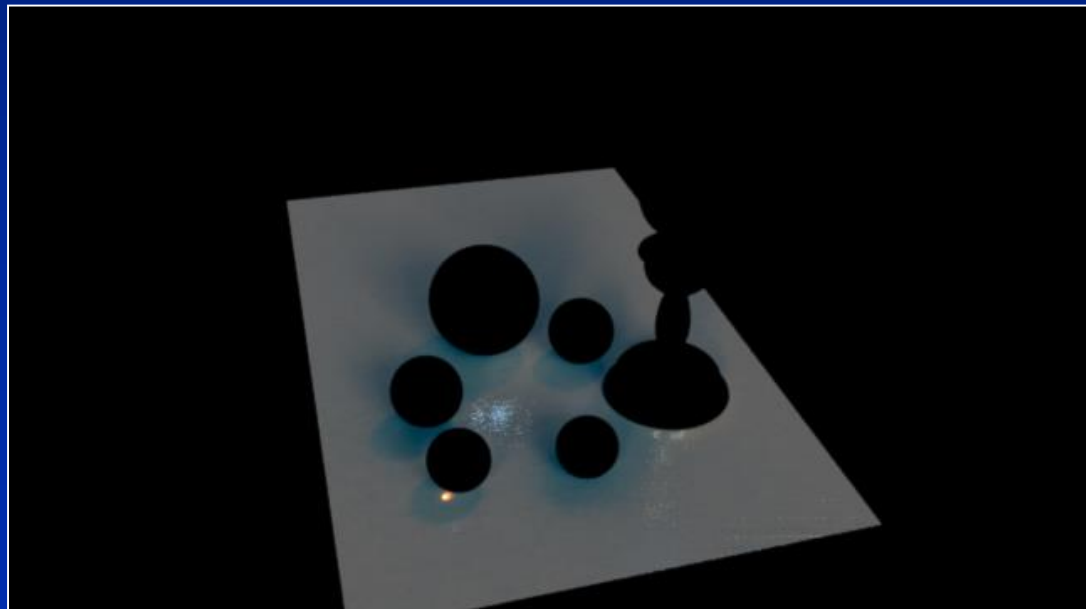
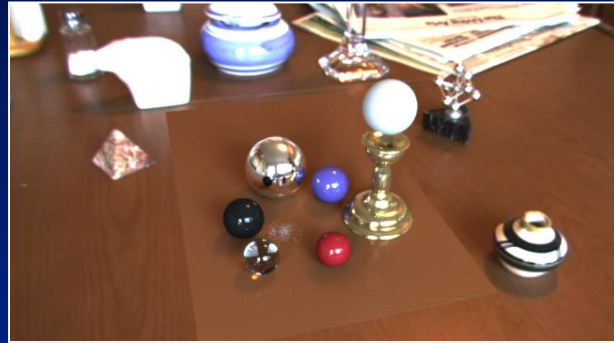






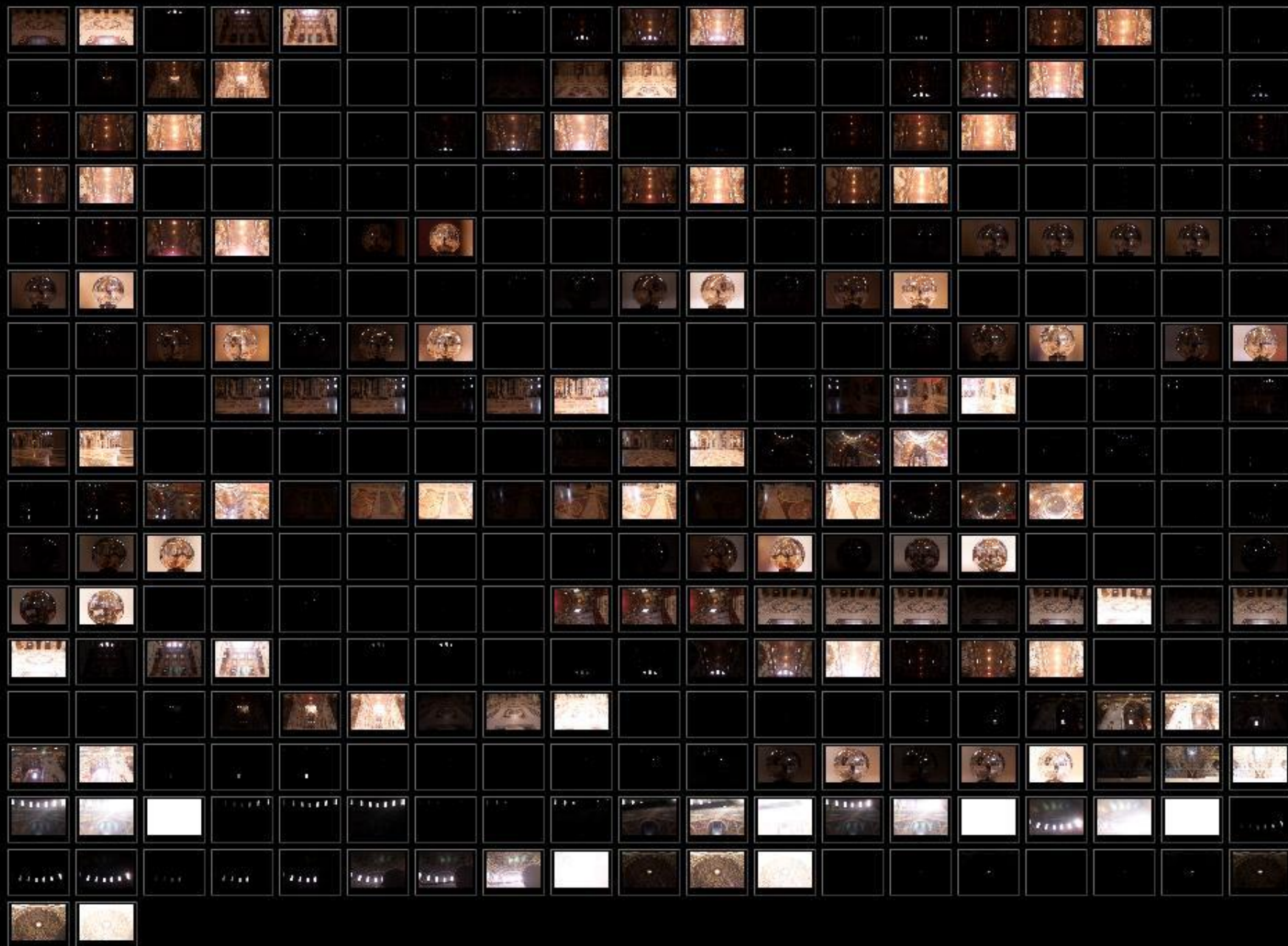
IMAGE-BASED LIGHTING IN *FIAT LUX*

Paul Debevec, Tim Hawkins, Westley Sarokin, H. P. Duiker, Christine Cheng, Tal Garfinkel, Jenny Huang

SIGGRAPH 99 Electronic Theater

Fiat Lux

- <http://ict.debevec.org/~debevec/FiatLux/movie/>
- <http://ict.debevec.org/~debevec/FiatLux/technology/>



HDR Image Series



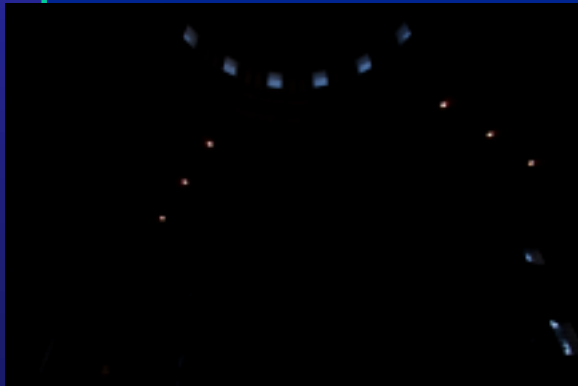
2 sec



1/4 sec



1/30 sec



1/250 sec



1/2000 sec



1/8000 sec



Assembled Panorama



Light Probe Images



Capturing a Spatially-Varying Lighting Environment



What if we don't have a light probe?

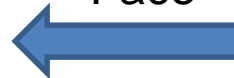


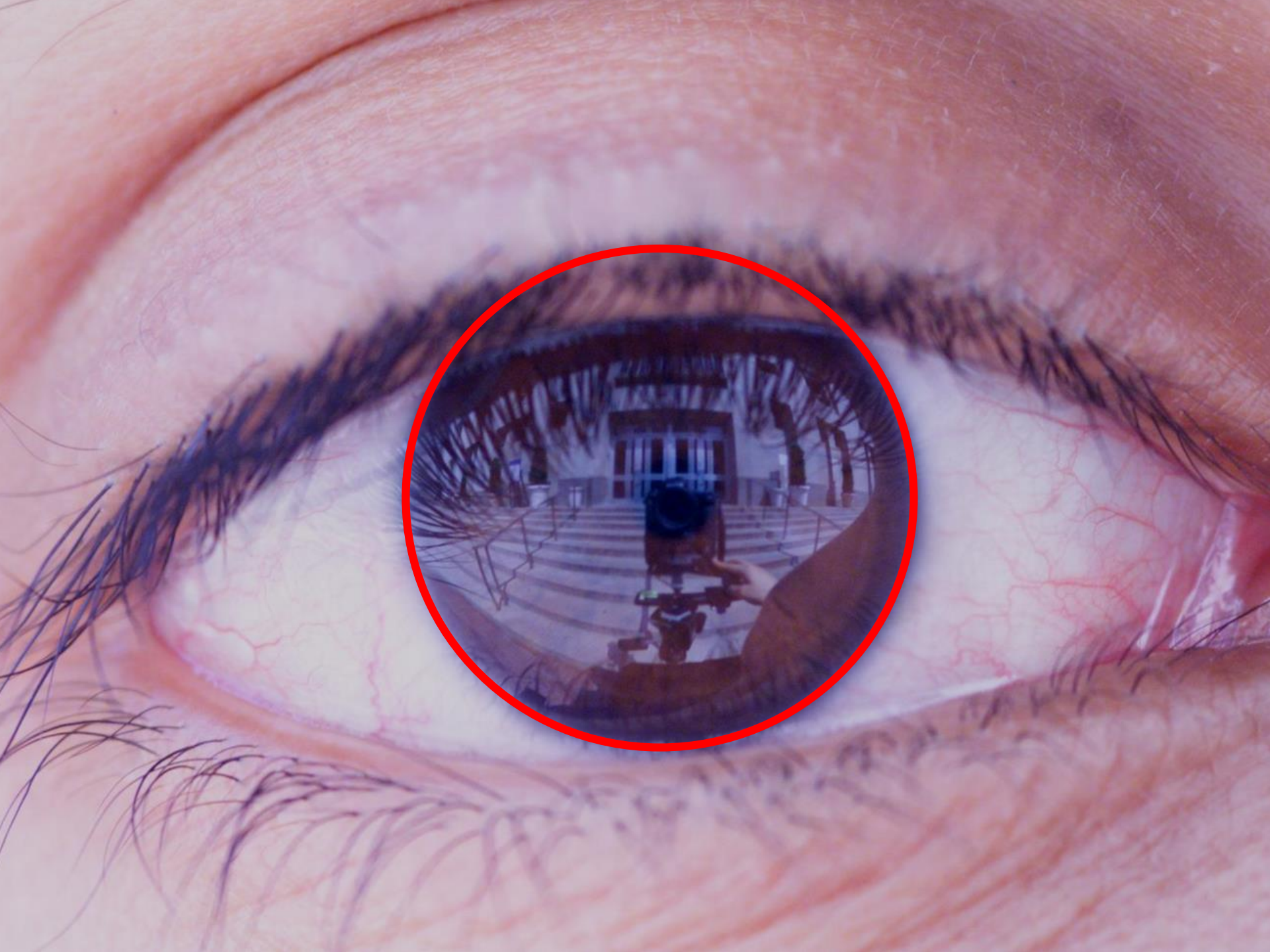
Zoom in on eye



Environment map

Insert Relit
Face





Environment Map from an Eye

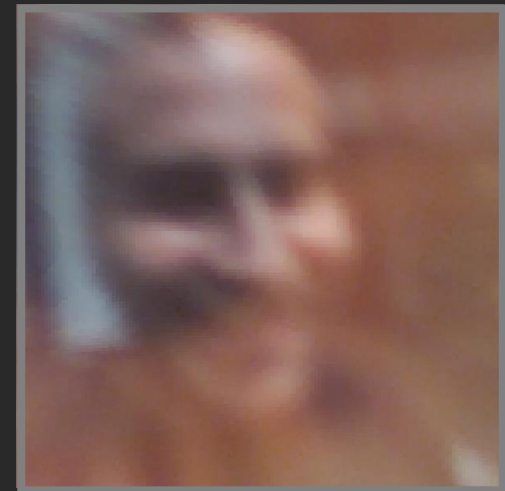


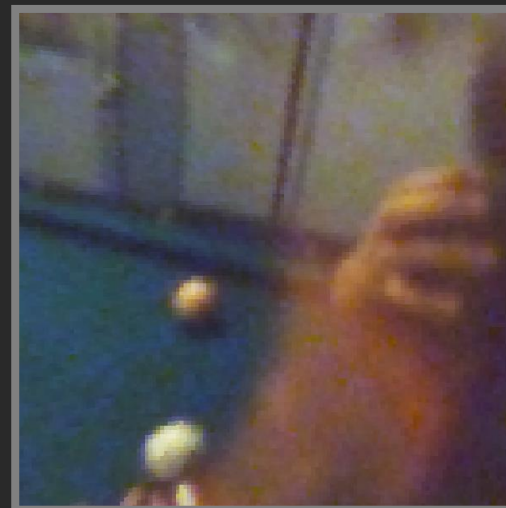
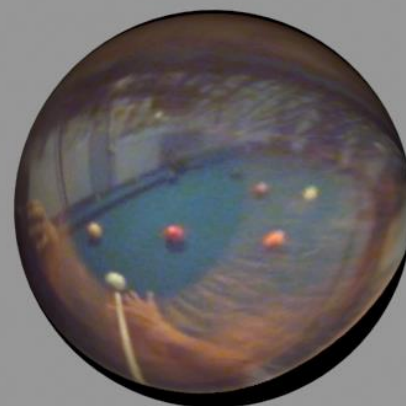
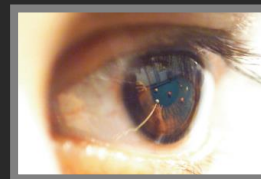
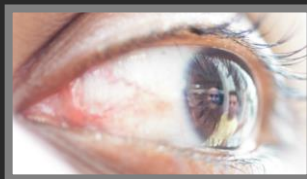
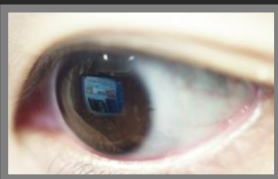
Can Tell What You are Looking At

Eye Image:



Computed Retinal Image:





Video

Summary

- Real scenes have complex geometries and materials that are difficult to model
- We can use an environment map, captured with a light probe, as a replacement for distance lighting
- We can get an HDR image by combining bracketed shots
- We can relight objects at that position using the environment map

