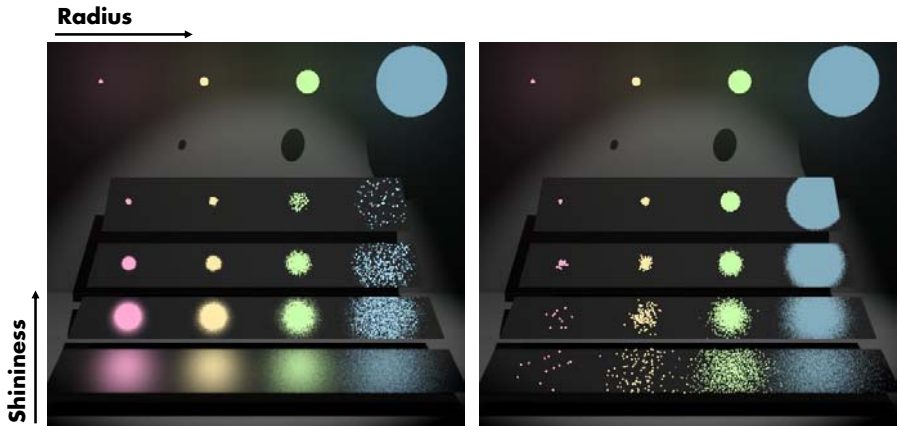
Multiple Importance Sampling

Multiple Importance Sampling

Reflection of a circular light source by a rough surface



Sampling the light source **Sampling the BRDF**

$$\int f(x)g(x)\,dx$$

Multiple Importance Sampling

Two sampling techniques

$$X_{1,i} \sim p_1(x) \qquad X_{2,i} \sim p_2(x)$$

$$Y_{1,i} = \frac{f(X_{1,i})}{p_1(X_{1,i})} \qquad Y_{2,i} = \frac{f(X_{2,i})}{p_2(X_{2,i})}$$

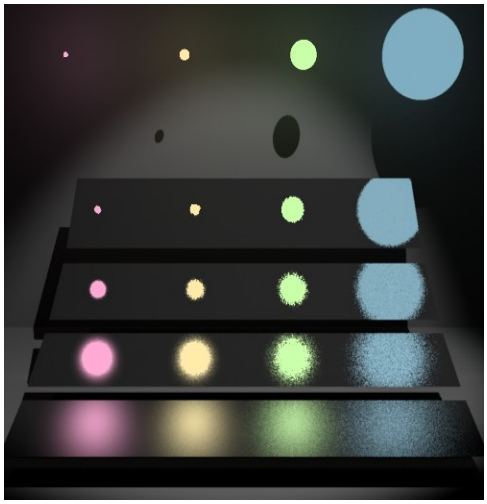
Form weighted combination of samples

$$Y_i = w_1 Y_{1,i} + w_2 Y_{2,i}$$

The balance heuristic

$$w_i(x) = \frac{p_i(x)}{p_1(x) + p_2(x)} \Rightarrow p(x) = w_1(x)p_1(x) + w_2(x)p_2(x)$$

Multiple Importance Sampling



Source: Veach and Guibas

Texture / Image-Based Rendering

Texture maps

- **Surface color and transparency**
- **Environment and irradiance maps**
- **Reflectance maps**
- **Shadow maps**
- **Displacement and bump maps**

Texture Maps

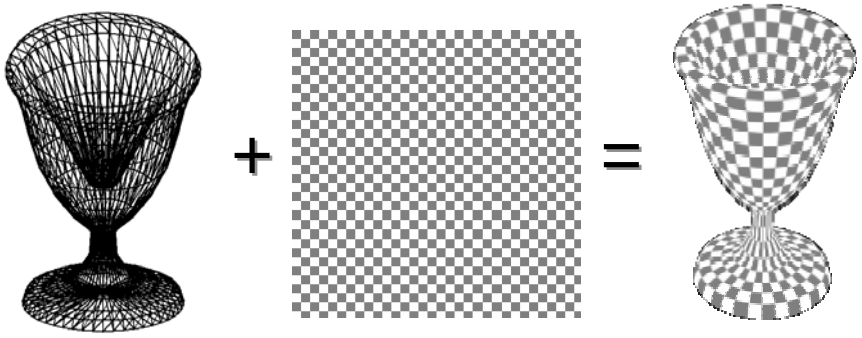
How is texture mapped to the surface?

- **Dimensionality: 1D, 2D, 3D**
- **Texture coordinates (s,t)**
  - **Surface parameters (u,v)**
  - **Direction vectors: R, N, H**
  - **Projection: cylinder**
  - **Developable surface: polyhedral net**
  - **Reparameterize a surface: old-fashion decal**

What does texture control?

- **Surface color and opacity**
- **Illumination functions: environment maps, shadow maps**
- **Reflection functions: reflectance maps**
- **Geometry: bump and displacement maps**

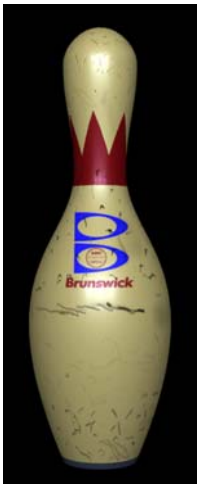
Texture Mapping



3D Mesh **2D Texture** **2D Image**

Surface Color and Transparency

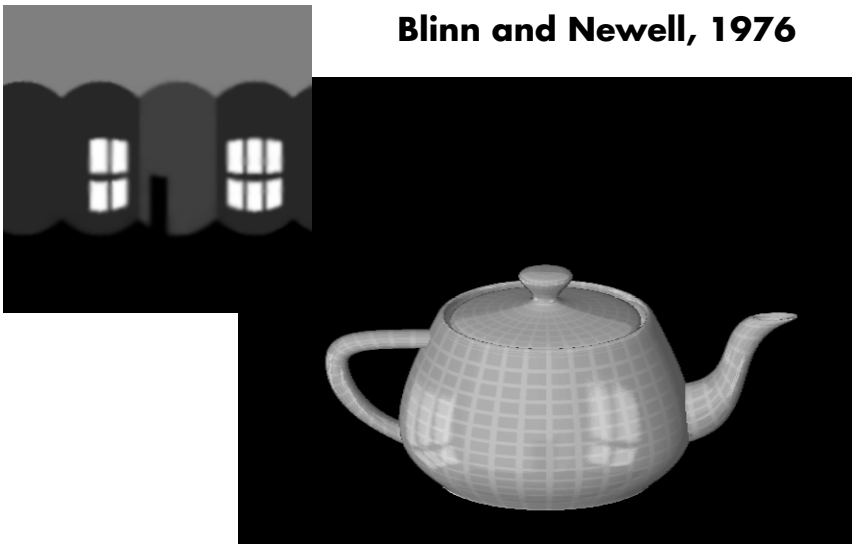
Tom Porter's Bowling Pin



Source: RenderMan Companion, Pls. 12 & 13

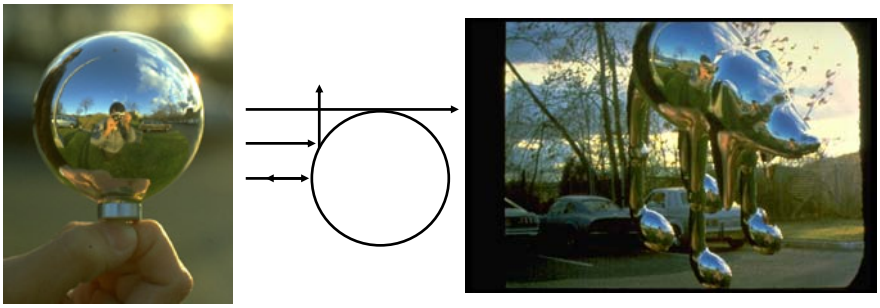
Reflection Maps

Blinn and Newell, 1976



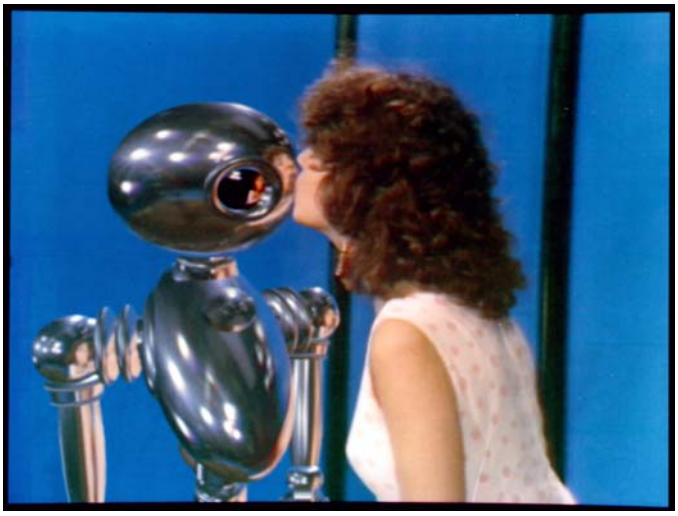
Gazing Ball (Light Probe)

Miller and Hoffman, 1984



- **Photograph of mirror ball**
- **Maps all directions to a to circle**
- **Resolution function of orientation**
- ***Reflection indexed by normal***

Environment Maps



***Interface*, Chou and Williams (ca. 1985)**

Environment Map Approximation

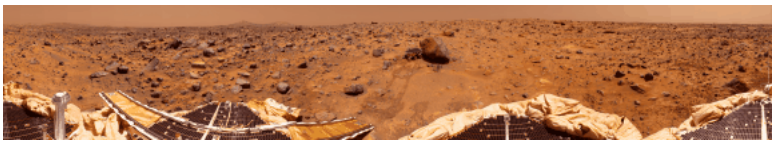
Ray Traced

Environment Map

Self reflections are missing in the environment map

Cylindrical Panoramas

QuickTime VR



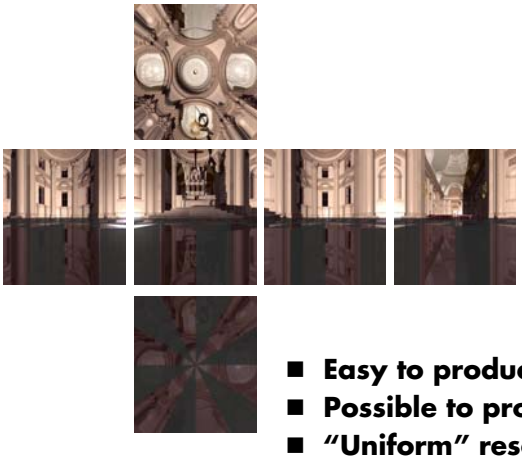
Mars Pathfinder

Memorial Church (Ken Turkowski)

Fisheye Lens

Pair of 180 degree fisheye
Photo by K. Turkowski

Cubical Environment Map



- Easy to produce with rendering system
- Possible to produce from photographs
- "Uniform" resolution
- Simple texture coordinates calculation

Direction Maps

Methods:

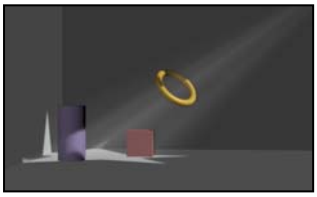
- Lattitude-Longitude (Map Projections)
  - Create by painting
- Gazing Ball
  - Create by photographing a reflective sphere
- Fisheye Lens
  - Standard camera lens
- Cubical Environment Map
  - Create with a rendering program, photography...

Issues:

- Non-linear mapping - expensive, curved lines
- Area distortion - spatially varying resolution
- Convert between maps using image warp

Shadow Mattes

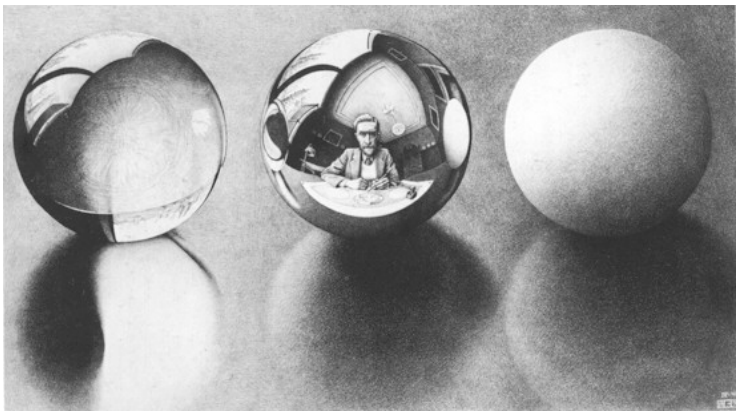
Projected Texture



Shadow Matte

```
UberLight( )
{
  Clip to near/far planes
  Clip to shape boundary
  foreach superelliptical blocker
      atten *= …
  foreach cookie texture
      atten *= …
  foreach slide texture
      color *= …
  foreach noise texture
      atten, color *= …
  foreach shadow map
      atten, color *= …
  Calculate intensity fall-off
  Calculate beam distribution
}
```

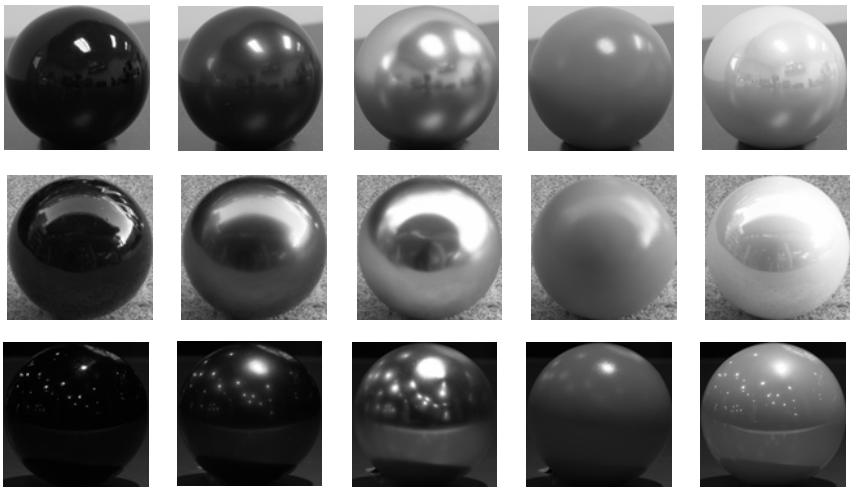

Reflectance Maps



M. C. Escher

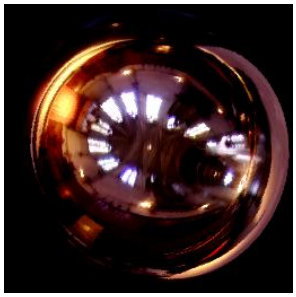
Capturing Reflectance Maps

Photographs of 5 spheres in 3 environments

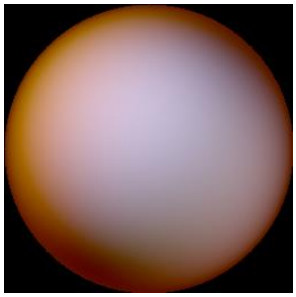


[Adelson and Dror]

Creating Irradiance Map



Incident Lighting



Reflected Light

$$B(\hat{\mathbf{N}}) = \rho E(\hat{\mathbf{N}})$$

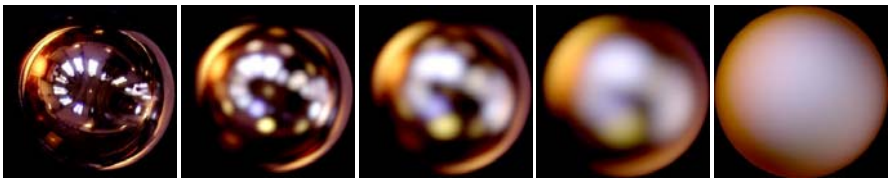
For each normal direction (defines hemisphere)

For each incoming direction

Compute contribution to irradiance

Creating Reflectance Map

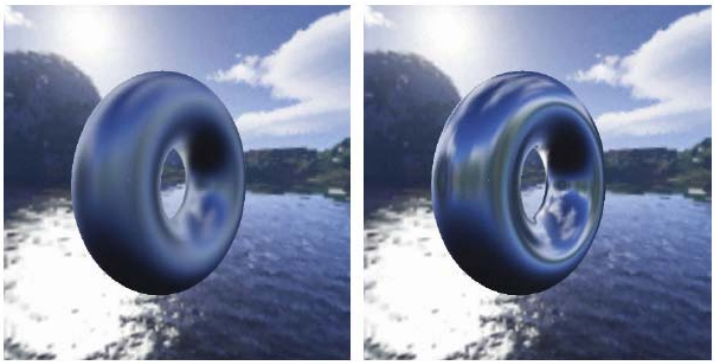
For any material



σ

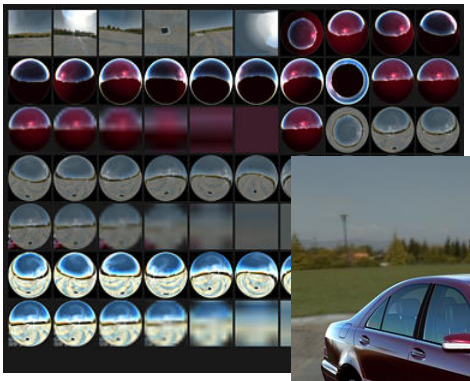
For each normal direction (defines hemisphere)
For each incoming direction
Compute contribution to reflection

Filtered Environment Maps



From W. Heidrich

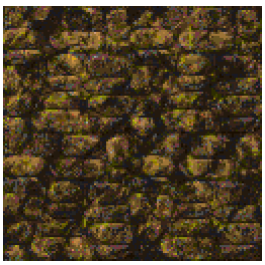
Reflectance Space Shading



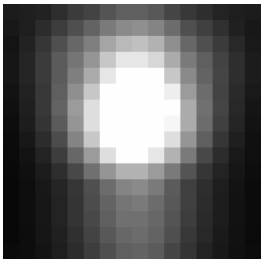
Cabral, Olano, Nemic 1999

Reflectance maps for 12 directions

Illumination Maps



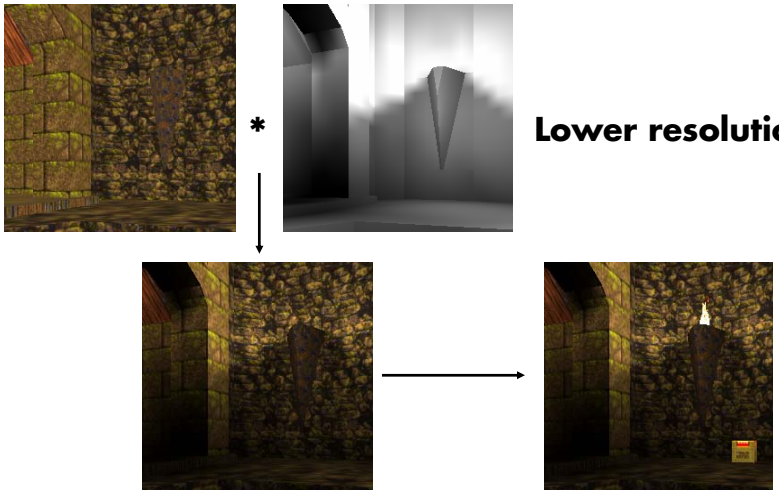
*



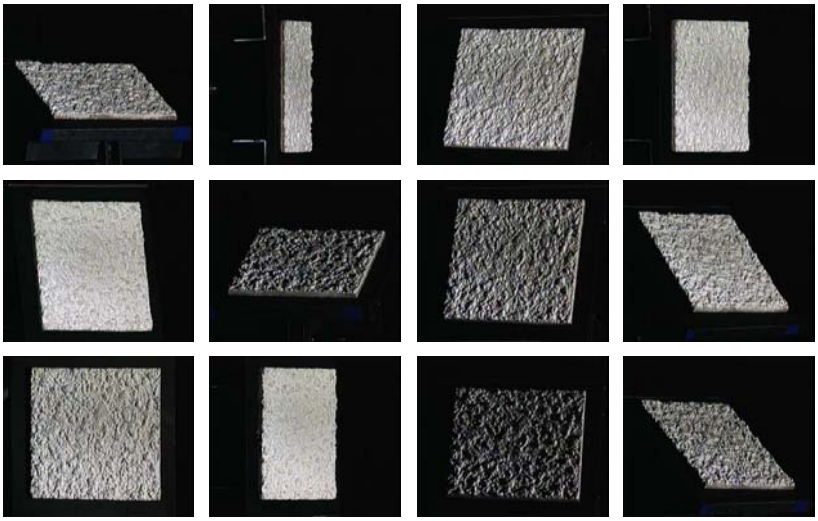
=

Reflectance	Irradiance	Radiosity
$\rho(x)$	$E(x)$	$B(x)$

Quake Light Maps



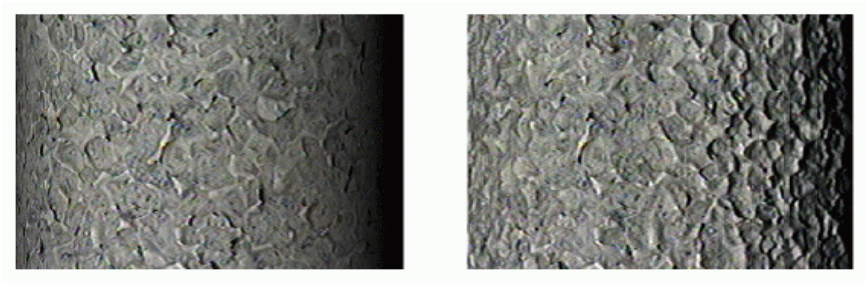
Bidirectional Texture Function (BTF)



Plaster

BTF Mapping

Complex interplay between texture and reflection



http://www1.cs.columbia.edu/CAVE/projects/btf/btf.php

Shadow Maps

Shadow maps = depth maps from light source

Correct Shadow Maps

Step 1:

- **Create z-buffer of scene as seen from light source**

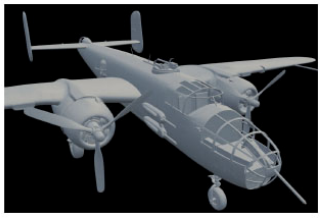
Step 2.

- **Render scene as seen from the eye**

```
For each light
    Transform point into light coordinates
        return (zl < zbuffer[xl][yl] ) ? 1 : 0
```

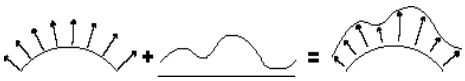

Ambient Occlusion Maps

Percentage of hemisphere visible



From Production ready global illumination, Hayden Landis, ILM

Displacement/Bump Mapping



$$\mathbf{P}(u,v)$$

$$\mathbf{S}(u,v) = \frac{\partial \mathbf{P}(u,v)}{\partial u} \qquad \mathbf{T}(u,v) = \frac{\partial \mathbf{P}(u,v)}{\partial v}$$

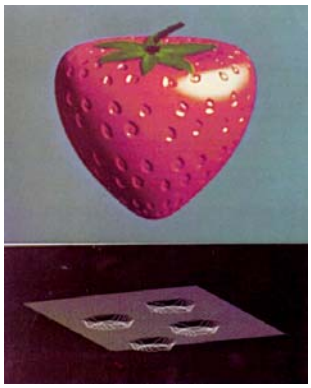
$$\mathbf{N}(u,v) = \mathbf{S} \times \mathbf{T}$$

- **Displacement**

$$\mathbf{P}'(u,v) = \mathbf{P}(u,v) + h(u,v)\mathbf{N}(u,v)$$

- **Perturbed normal**

$$\mathbf{N}'(u,v) = \mathbf{P}'_u \times \mathbf{P}'_v$$
$$= \mathbf{N} + h_u(\mathbf{T} \times \mathbf{N}) + h_v(\mathbf{S} \times \mathbf{N})$$



From Blinn 1976

Normal Maps

(nx,ny,nz) = (r,g,b)

http://members.shaw.ca/jimht03/normal.html

Classic History

Catmull/Williams 1974 - basic idea
Blinn and Newell 1976 - basic idea, reflection maps
Blinn 1978 - bump mapping
Williams 1978, Reeves et al. 1987 - shadow maps
Smith 1980, Heckbert 1983 - texture mapped polygons
Williams 1983 - mipmaps
Miller and Hoffman 1984 - illumination and reflectance
Perlin 1985, Peachey 1985 - solid textures
Greene 1986 - environment maps/world projections
Akeley 1993 - Reality Engine
Levoy and Hanrahan 1996 - Light Field
Dana, van Ginneken, Nayar, Koenderink 1996 - BTF