Texture

Texture maps

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

Level of detail hierarchy

Procedural shading and texturing

Texture synthesis and noise

Texture Maps

How is texture mapped to the surface?

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

What does texture control?

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

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

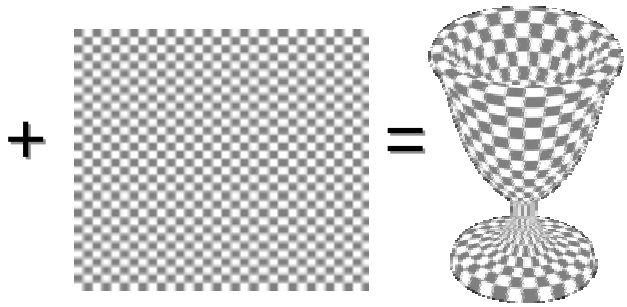
Miller and Hoffman 1984 - illumination and reflectance

Perlin 1985, Peachey 1985 - solid textures

Greene 1986 - environment maps/world projections

Akeley 1993 - Reality Engine

Texture Mapping



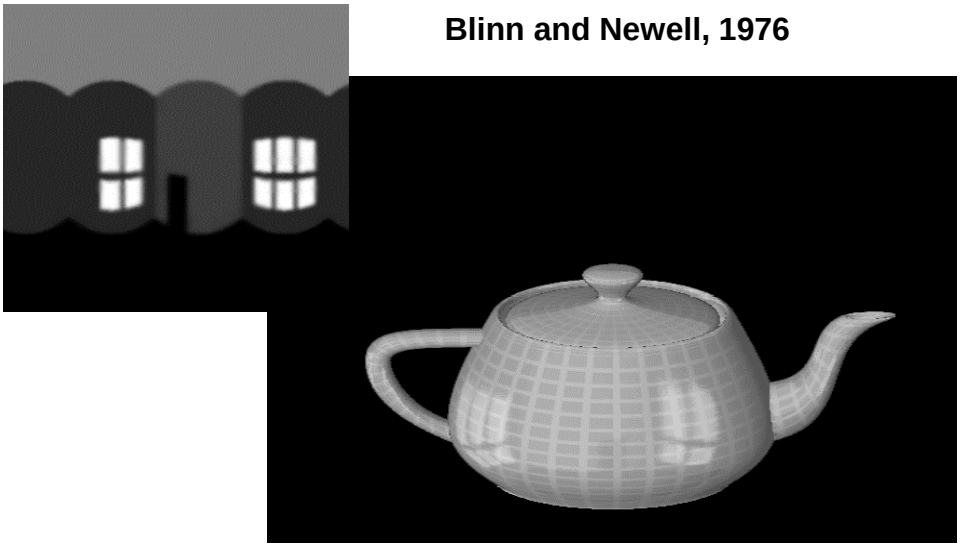
3D Mesh **2D Texture** **2D Image**

Surface Color and Transparency

Tom Porter's Bowling Pin

Source: RenderMan Companion, Pls. 12 & 13

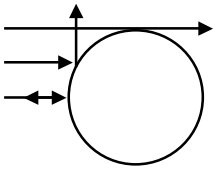
Reflection Maps



Blinn and Newell, 1976

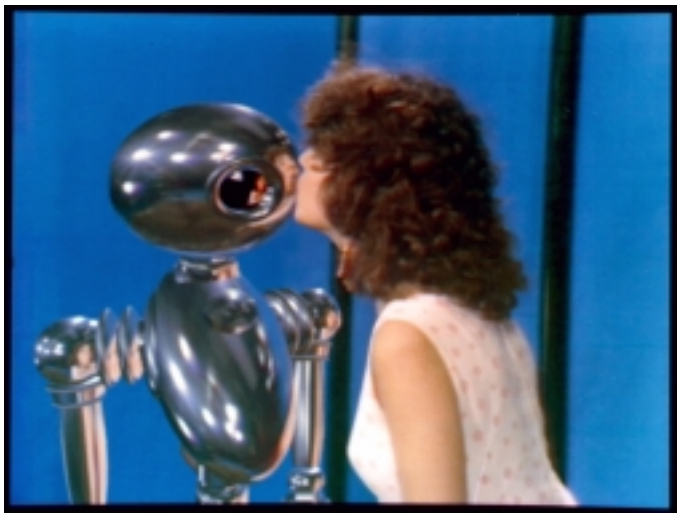
Gazing Ball

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

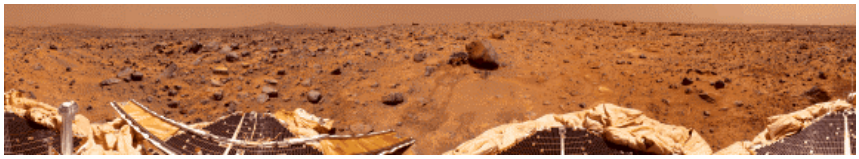
Ray Traced

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

Many ways to map directions to images...

Methods:

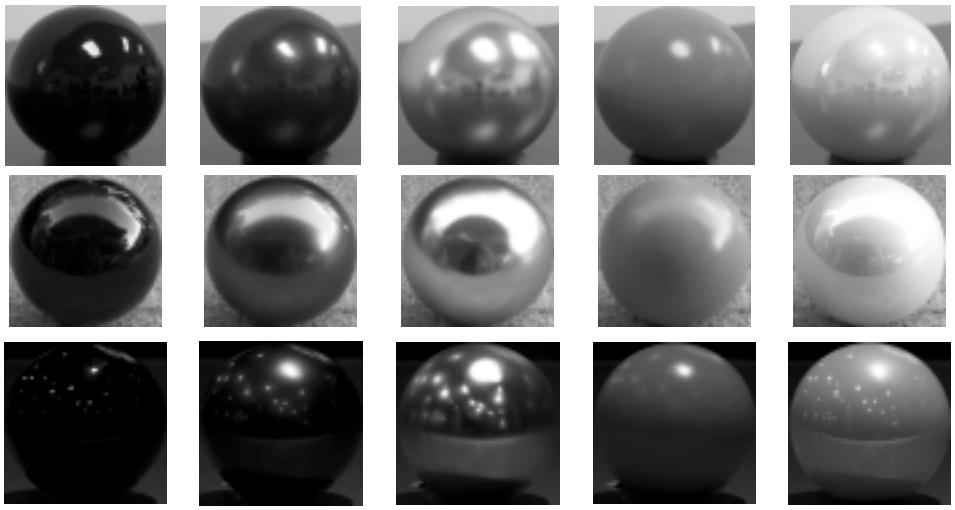
- **Lattitude-Longitude (Map Projections) [Newell and Blinn]**
  Create by painting
- **Gazing Ball (N) [Miller and Hoffman]**
  Create by photographing a reflective sphere
- **Fisheye Lens**
  Standard camera lens
- **Cubical Environment Map (R)**
  Create with a rendering program, photography...

Issues:

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

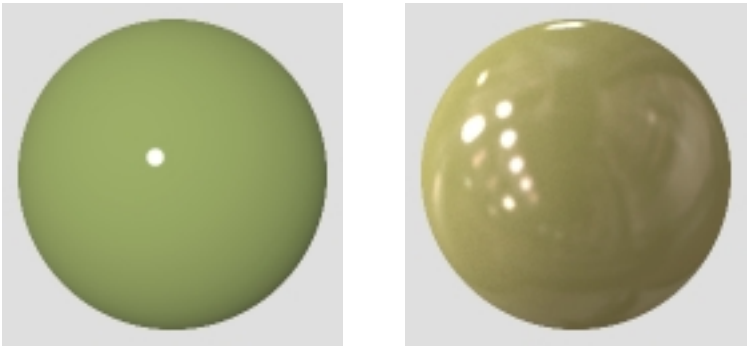
Combining Reflectance & Illumination

Photographs of 5 spheres in 3 environments (Adelson and Dror)



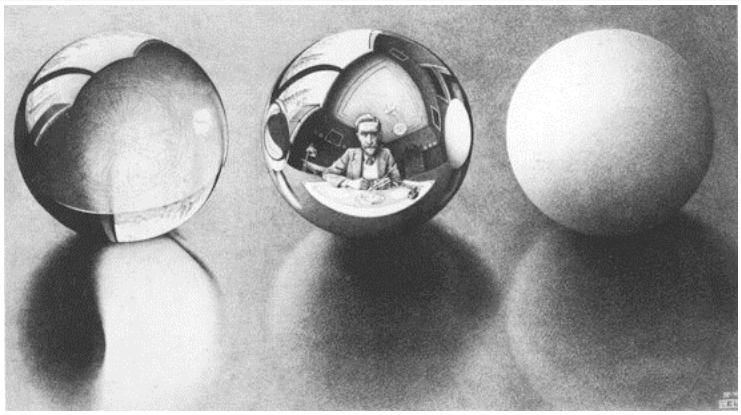
Material Recognition

People recognize materials more easily under natural illumination than simplified illumination.



Illusion due to Ted Adelson

Reflectance Maps



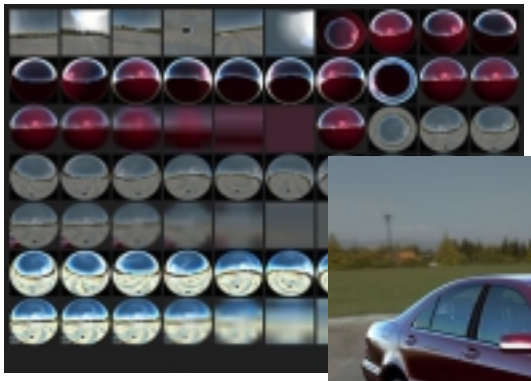
For a given viewing direction
For each normal direction
For each incoming direction (hemispherical integral)
Evaluate reflection equation

Example: Phong Model

Rough surfaces blur highlight

σ

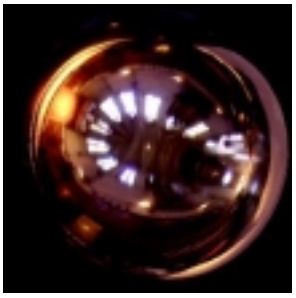
Reflectance Space Shading



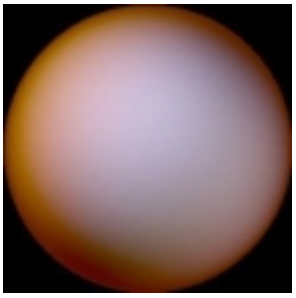
12 directions

Cabral, Olano, Nemic 1999

Example: Lambertian Reflectance



Incident Lighting

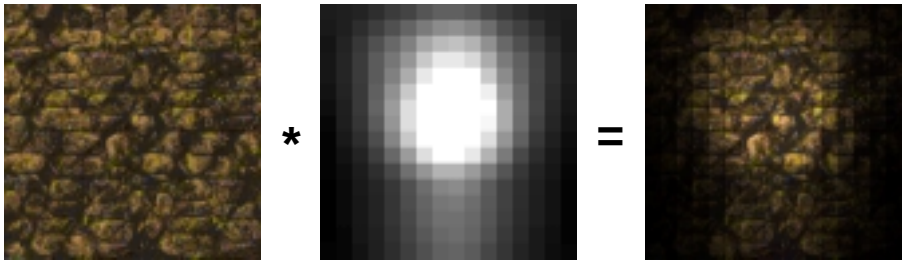


Reflected Light

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

Radiosity or Irradiance Map

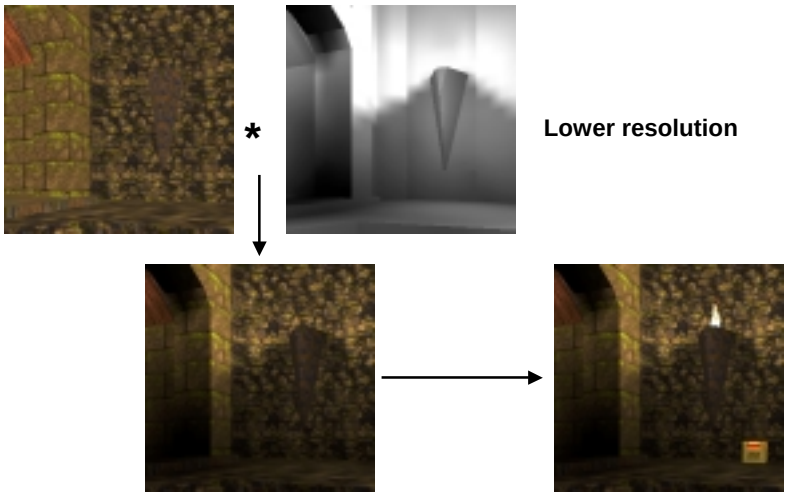
Illumination Maps



Reflectance **Irradiance** **Radiosity**

$\rho(x)$ $E(x)$ $B(x)$

Quake Light Maps



Shadow Maps

May incorporate shadow maps into lighting calculations

Correct Shadow Maps

Step 1:

Create z-buffer of scene as seen from light source

Step 2.

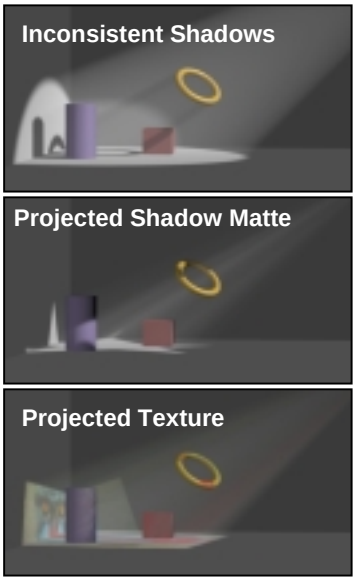
Render scene as seen from the eye

For each light

Transform point into light coordinates

return (zl < zbuffer[xl][yl]) ? 1 : 0

Barzel's *UberLight.sl*



Example of a complex shader

```
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
}
```

Displacement/Bump Mapping

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

$$\mathbf{T}_u(u,v) = \frac{\partial \mathbf{P}(u,v)}{\partial u} \quad \mathbf{T}_v(u,v) = \frac{\partial \mathbf{P}(u,v)}{\partial v}$$

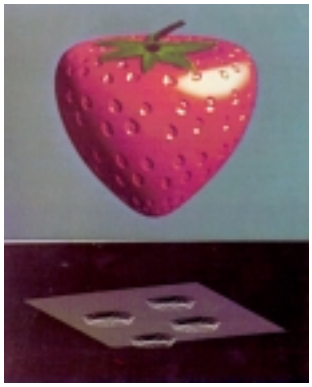
$$\mathbf{N}(u,v) = \mathbf{P}_u \times \mathbf{P}_v$$

- **Displacement**

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

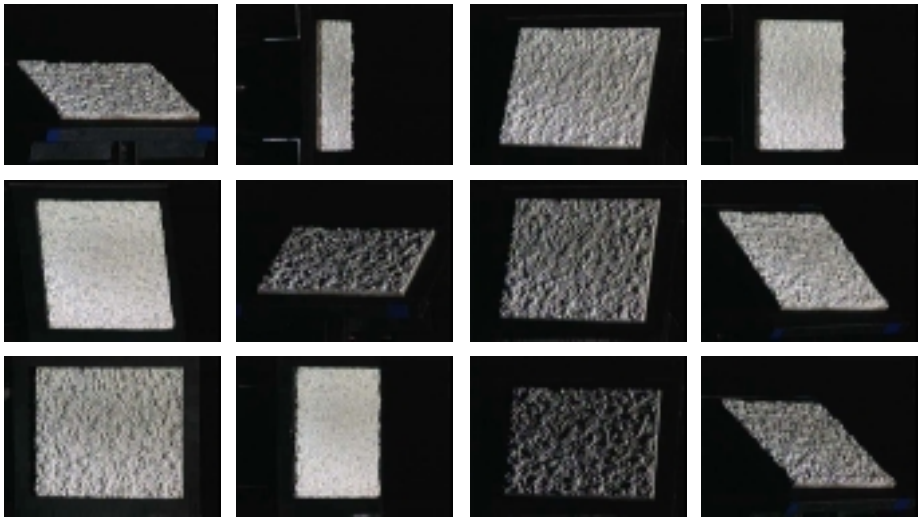
- **Perturbed normal**

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



From Blinn 1976

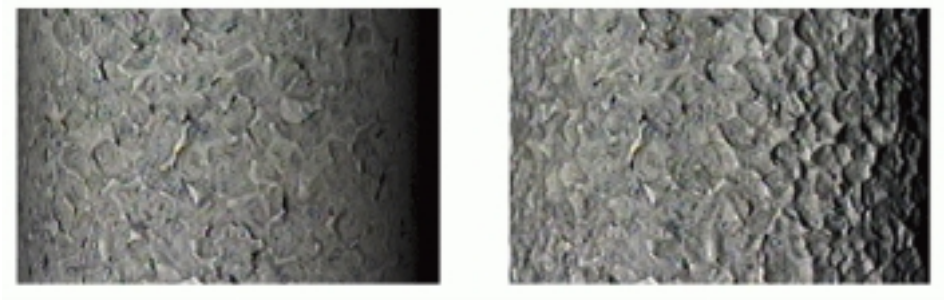
Bidirectional Texture Function (BTF)



Plaster

BTF Mapping

Complex interplay between texture and reflection



Hierarchy

Physics		Computer Graphics
Geometrical optics	↑	Geometry
■ Macro-structures		Displacement (P) maps
Transport		Bump (N) maps
■ Micro-structures		Reflection
Microfacets	—	Texture
Physical optics		
Kirchoff approx.		
Quantum optics	↓	