To Do

- By Thu night, e-mail me brief project description (can be done in groups of 2). Time to discuss next week
- A good idea is to choose to present papers relating to your intended project and vice-versa.
- This lecture discusses shadow and environment mapping
- Remember: Jan 15 student presentations of papers
- Jan 20: no class (Siggraph deadline)

Shadow and Environment Maps

- Basic methods to add realism to interactive rendering
- Shadow maps: image-based way hard shadows
  - Very old technique. Originally Williams 78
  - Many recent (and older) extensions
  - Widely used even in software rendering (RenderMan)
  - Simple alternative to raytracing for shadows
- Environment maps: image-based complex lighting
  - Again, very old technique. Blinn and Newell 76
  - Huge amount of recent work (some covered in course)
- Together, give most of realistic effects we want
  - But cannot be easily combined!! Some of the course is about ways to get around this limitation
  - See Annen 08 [real-time all-frequency shadows dynamic scenes] for one approach: convolution soft shadows

Common Real-time Shadow Techniques

- Lance Williams: Brute Force in image space
  (shadow maps in 1978, but other similar ideas like Z buffer, bump mapping using textures and so on)

Problems

- Mostly tricks with lots of limitations
  - Projected planar shadows
    - works well only on flat surfaces
  - Stencilled shadow volumes
    - determining the shadow volume is hard work
  - Light maps
    - totally unsuited for dynamic shadows
  - In general, hard to get everything shadowing everything

Shadow Mapping

- Completely image-space algorithm
  - no knowledge of scene’s geometry is required
  - must deal with aliasing artifacts
- Well known software rendering technique
  - Basic shadowing technique for Toy Story, etc.
Phase 1: Render from Light

- Depth image from light source

Phase 2: Render from Eye

- Standard image (with depth) from eye

Phase 2+: Project to light for shadows

- Project visible points in eye view back to light source

Visualizing Shadow Mapping

- A fairly complex scene with shadows
Visualizing Shadow Mapping

- Compare with and without shadows

with shadows without shadows

Visualizing Shadow Mapping

- The scene from the light’s point-of-view

FYI: from the eye’s point-of-view again

Visualizing Shadow Mapping

- The depth buffer from the light’s point-of-view

FYI: from the light’s point-of-view again

Visualizing Shadow Mapping

- Projecting the depth map onto the eye’s view

FYI: depth map for light’s point-of-view again

Visualizing Shadow Mapping

- Comparing light distance to light depth map

Green is where the light planar distance and the light depth map are approximately equal
Non-green is where shadows should be

Visualizing Shadow Mapping

- Scene with shadows

Notice how specular highlights never appear in shadows
Notice how curved surfaces cast shadows on each other
Hardware Shadow Map Filtering

“Percentage Closer” filtering
- Normal texture filtering just averages color components
- Averaging depth values does NOT work
- Solution [Reeves, SIGGRAPH 87]
  - Hardware performs comparison for each sample
  - Then, averages results of comparisons
- Provides anti-aliasing at shadow map edges
  - Not soft shadows in the umbra/penumbra sense

GL_NEAREST: blocky
GL_LINEAR: antialiased edges

Problems with shadow maps

- Hard shadows (point lights only)
- Quality depends on shadow map resolution (general problem with image-based techniques)
- Involves equality comparison of floating point depth values means issues of scale, bias, tolerance
- Some of these addressed in papers presented

Reflection Maps

Blinn and Newell, 1976

Environment Maps

Miller and Hoffman, 1984

Interface, Chou and Williams (ca. 1985)
Environment Maps

- Cubical Environment Map
- 180 degree fisheye

Cylindrical Panoramas

Reflectance Maps

- Reflectance Maps (Index by N)
- Horn, 1977
- Irradiance (N) and Phong (R) Reflection Maps
- Miller and Hoffman, 1984

Mirror Sphere Chrome Sphere Matte Sphere

Irradiance Environment Maps

- Incident Radiance (Illumination Environment Map)
- Irradiance Environment Map

Assumptions

- Diffuse surfaces
- Distant illumination
- No shadowing, interreflection

Hence, Irradiance a function of surface normal

Diffuse Reflection

\[ B = \rho E \]

Radiosity (image intensity) \hspace{2cm} \text{Reflectance (albedo/texture)} \hspace{2cm} \text{Irradiance (incoming light)}

\[ \text{quake light map} \]

Analytic Irradiance Formula

Lambertian surface acts like low-pass filter

\[ E_{lm} = A_l L_{lm} \]

Ramamoorthi and Hanrahan 01
Basel and Jacobs 01
### 9 Parameter Approximation

**Order 0**

- 1 term
- RMS Error = 25%

**Order 1**

- 4 terms
- RMS Error = 8%

**Order 2**

- 9 terms
- RMS Error = 1%

For any illumination, average error < 3% [Basri Jacobs 01]

### Real-Time Rendering

\[ E(n) = n^T M n \]

Simple procedural rendering method (no textures)
- Requires only matrix-vector multiply and dot-product
- In software or NVIDIA vertex programming hardware

Widely used in Games (AMPED for Microsoft Xbox), Movies (Pixar, Framestore CFC, …)

```c
surface float1 irradmat (matrix4 M, float3 v) {
  float4 n = {v , 1} ;
  return dot(n , M*n) ;
}
```

### Environment Map Summary

- Very popular for interactive rendering
- Extensions handle complex materials
- Shadows with precomputed transfer
- But cannot directly combine with shadow maps
- Limited to distant lighting assumption

### Resources

- OpenGL red book (latest includes GLSL)
- Older books: OpenGL Shading Language book (Rost), The Cg Tutorial, …
- [Real-Time Rendering by Moller and Haines](http://www.realtimerendering.com)
- Debevec: [http://wwwDebevec.org/ReflectionMapping/](http://wwwDebevec.org/ReflectionMapping/)
- Links to Miller and Hoffman original, Haeberli/Segal
- [http://www.cs.berkeley.edu/~rvir/papers/envmap](http://www.cs.berkeley.edu/~rvir/papers/envmap)
- Also papers by Heidrich, Cabral, …
- Lots of information available on web…