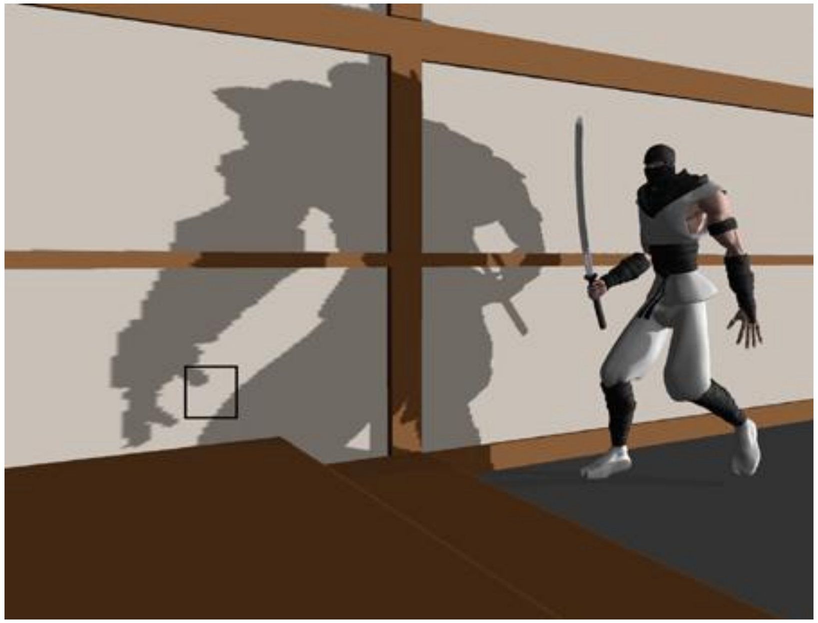
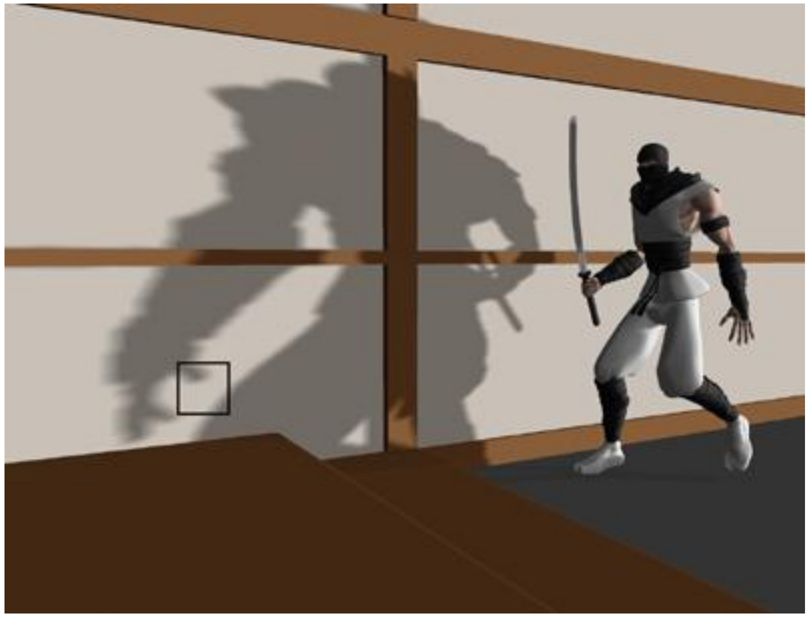
Percentage Closer Filtering

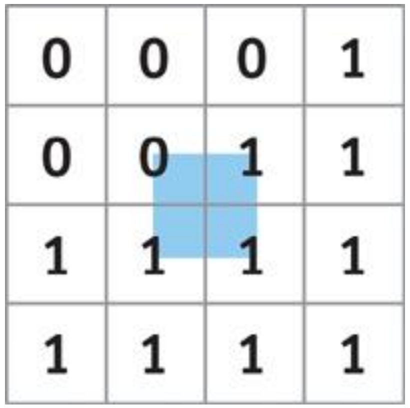
- Antialiasing of shadow maps
- http://http.developer.nvidia.com/GPUGems/gpugems_ch11.html
  - Close, but uses box filter instead of Gaussian





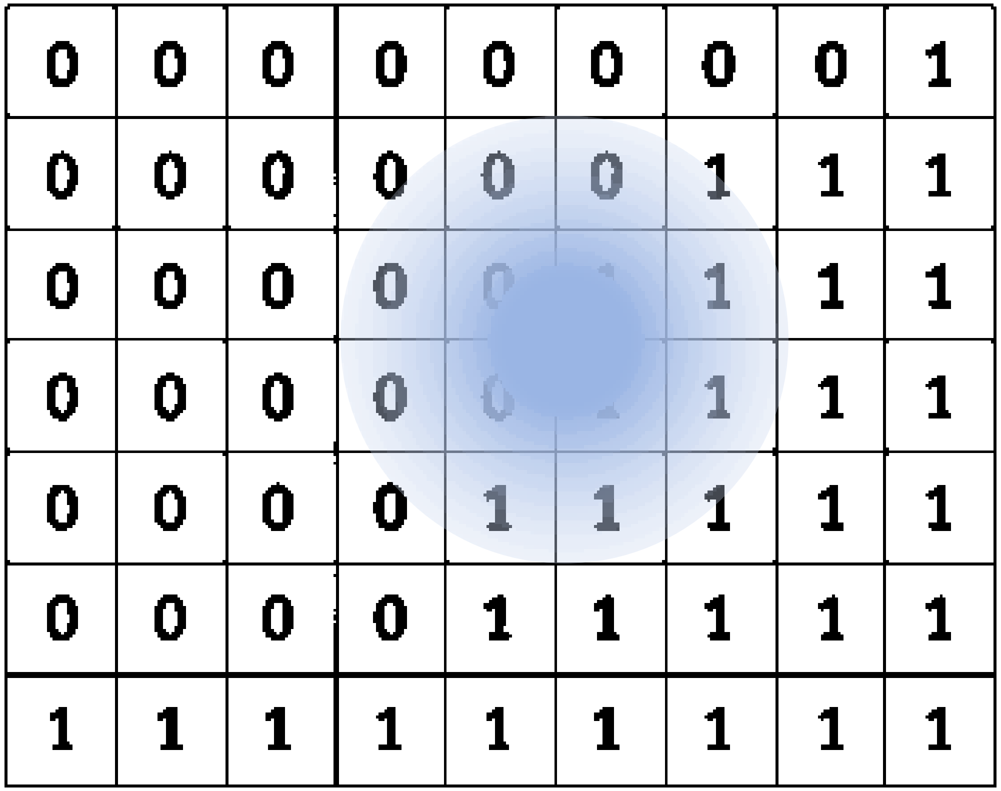
Source: http://http.developer.nvidia.com/GPUGems/gpugems_ch11.html

Built-in vs. Our Method



0	0	0	1
0	0	1	1
1	1	1	1
1	1	1	1

Bilinear interpolate 2x2 texels



0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1	1
0	0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Guassian weighted average of N nearest texels

Assn 3 PCF

- Use a continuous Gaussian function
  - General form:

$$f(x) = e^{-distance^2}$$

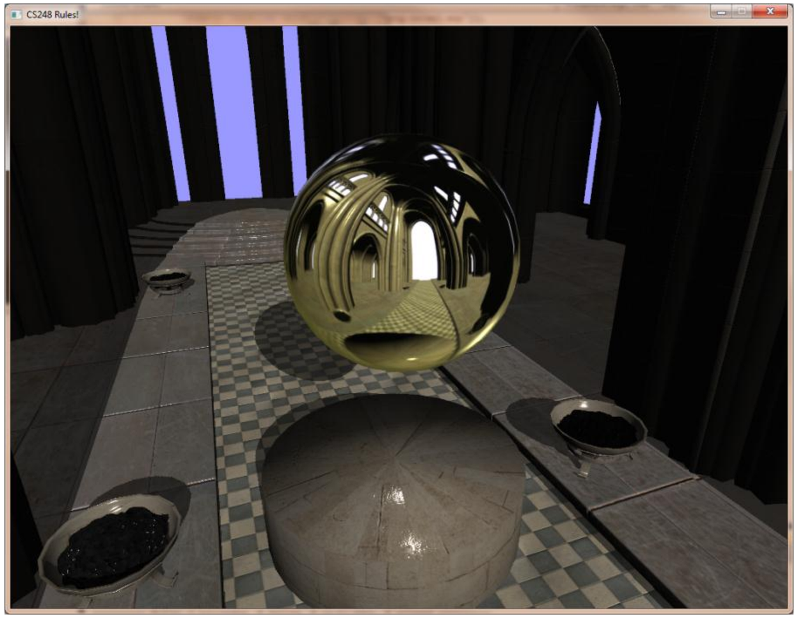
  - **distance** between sample point and texel
  - Tweak coefficients as you please
- Sample 3x3 nearest texels

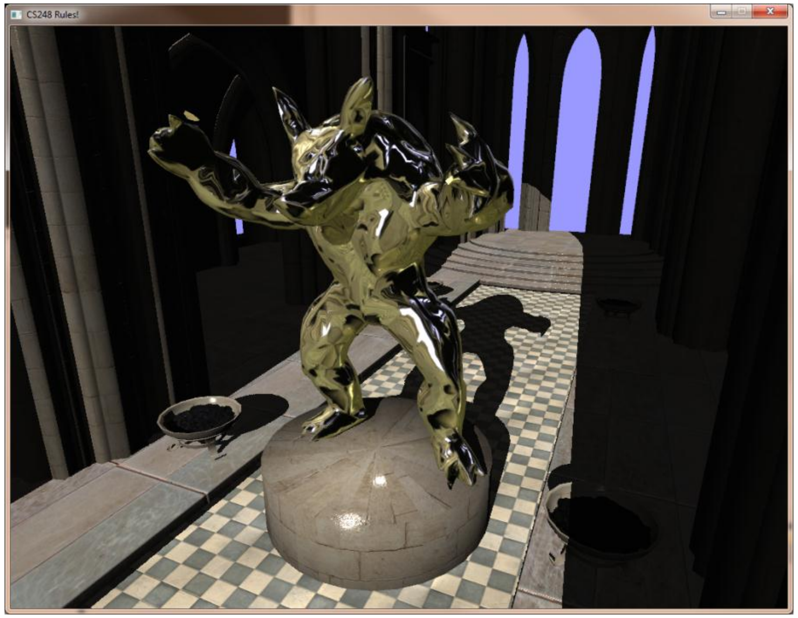
Box Filter

```
float sum = 0, x, y;

for (y = -1.5; y <= 1.5; y += 1.0)
    for (x = -1.5; x <= 1.5; x += 1.0)
        sum += offset_lookup(shadowmap,
        shadowCoord, float2(x, y));

shadowCoeff = sum / 16.0;
```

Environment Mapping

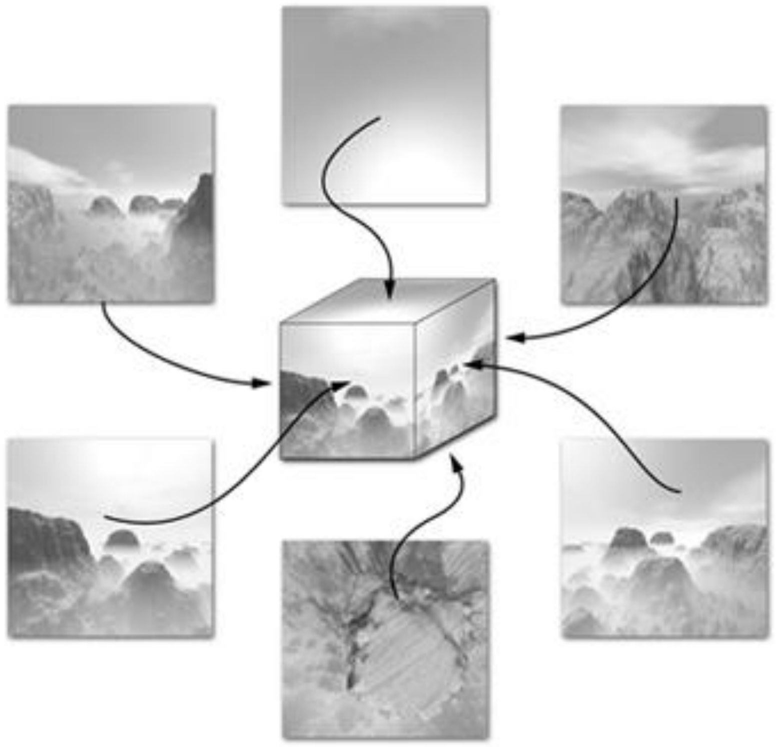
- Simulate reflective/refractive surfaces
- Assn 3: Shiny, metallic armadillo





High-Level Overview

- Set up an environment cube map
- Choose a point as the "center"
- Render six images in $(\pm x, \pm y, \pm z)$
- Sample environment texture from surface normal



Source: Nvidia GPU Gems

Set up an environment cube map

- `glGenTextures(`**`numTextures`**`, &`**`texId`**`)`
- `glBindTexture(GL_TEXTURE_CUBE_MAP,` **`texId`**`)`
- `glTexImage2D(`**`face`**`, 0, GL_RGBA,` **`width`**`,` **`height`**`, 0, GL_RGBA, GL_UNSIGNED_BYTE, 0)`
  - `GL_TEXTURE_CUBE_MAP_POSITIVE_X`, etc.

Set up an environment cube map

- `glGenFramebuffers(`**`numFbos`**`, &`**`fboId`**`)`

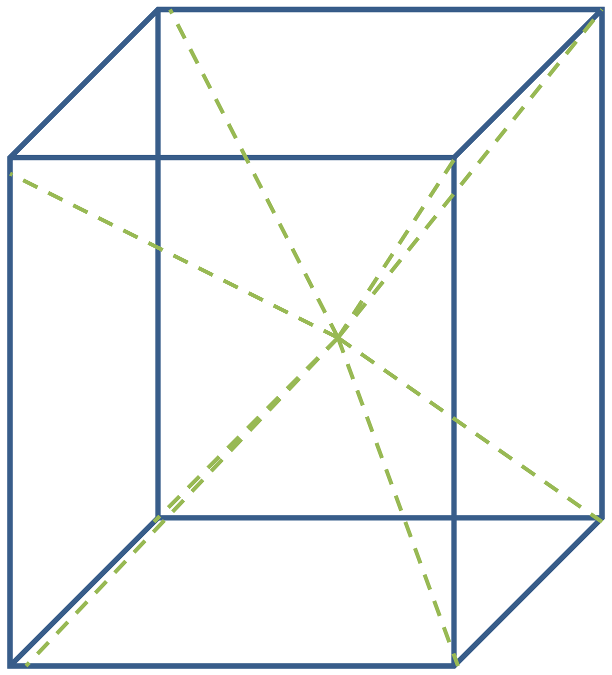
- `glBindFramebuffer(GL_FRAMEBUFFER,` **`fboId`**`)`

- `glFramebufferTexture2D(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0,` **`face`**`,` **`texId`**`, 0);`

*Note: May require vendor suffix, e.g. glGenFramebuffers**EXT,** GL_FRAMEBUFFER**_EXT**

Choose a point as the “center”

- Choose somewhere near the middle of the sphere/armadillo

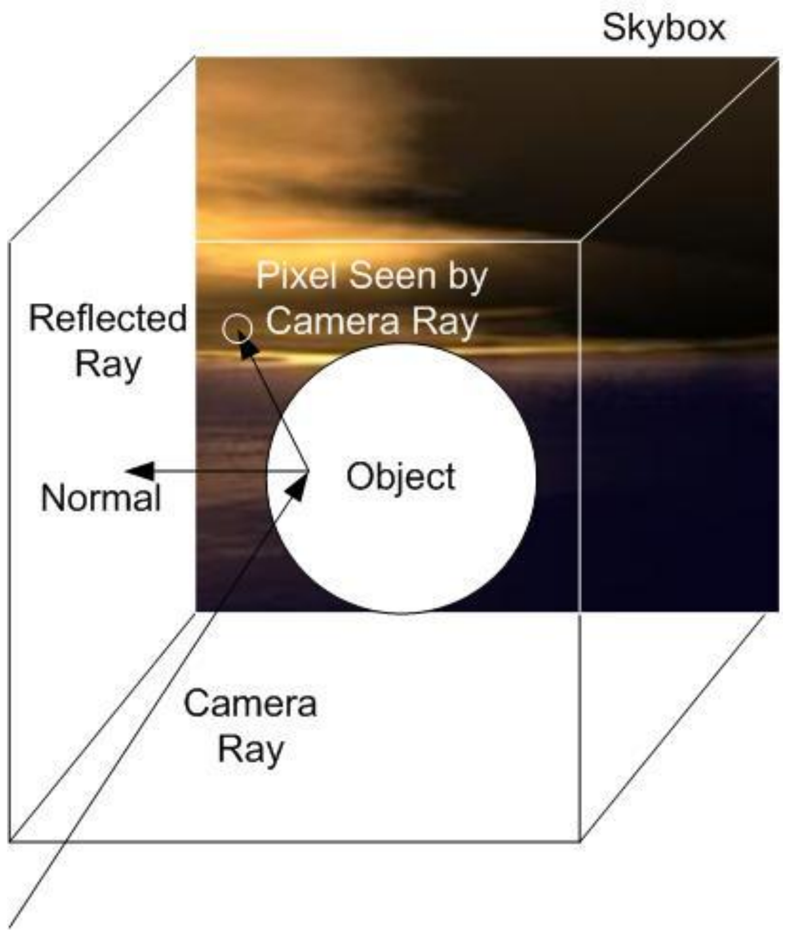


Render six images in $(\pm x, \pm y, \pm z)$

- Ignore the armadillo when rendering (you can hard-code this)
- Set wrap to GL_CLAMP
- Remember to change viewport to a square
- Field of view = 90°, aspect ratio = 1.0
- `glCopyTexSubImage2D(`**`face`**`, 0, 0, 0, 0, 0, `**`width`**`, `**`height`**`)`

Sample environment texture

- Similar to sampling a texture
- Find R, the view vector reflected by the normal in **world space**
- uniform samplerCube **tex**
- textureCube(**tex**, **R**)



Debugging Tips

- Render textures to file or fullscreen quad
  - Six environment maps
  - Light depth map from light POV
  - Light depth map from camera POV
- sf::Image can be useful for loading/manipulating images
  - glReadPixels
  - LoadFromMemory
  - SaveToFile