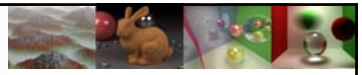


Ambient Occlusion

Slides modified from:
Patrick Cozzi
University of Pennsylvania
CIS 565 - Fall 2013

Ambient Occlusion

- *A*mbient *O*cclusion (*AO*)
  - "shadowing of ambient light"
  - "darkening of the ambient shading contribution"

Image from Bavoil and Sainz. http://developer.download.nvidia.com/SDK/10.5/direct3d/Source/ScreenSpaceAO/doc/ScreenSpaceAO.pdf

Ambient Occlusion

- Ambient Occlusion
  - "the crevices of the model are realistically darkened, and the exposed parts of the model realistically receive more light and are thus brighter"
  - "the soft shadow generated by a sphere light of uniform intensity surrounding the scene"

Ambient Occlusion

Evenly lit from all directions | Ambient Occlusion | Global Illumination

Images courtesy of A K Peters, Ltd. http://www.realtimerendering.com/

Ambient Occlusion

- "the integral of the occlusion contributed from inside a hemisphere of a given radius **R**, centered at the current surface point **P** and oriented towards the normal **n** at **P**"

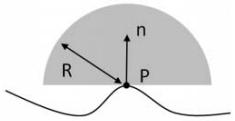


Figure 2. Hemisphere Ω around a surface point P.

Image from Bavoil and Sainz. http://developer.download.nvidia.com/SDK/10.5/direct3d/Source/ScreenSpaceAO/doc/ScreenSpaceAO.pdf

Ambient Occlusion Math

$$E(\mathbf{p}, \mathbf{n}) = \int_{\Omega} L_A \cos\theta_i d\omega_i = \pi L_A,$$

- E – surface irradiance
- L_A – incoming radiance

Ambient Occlusion Math

$$E(\mathbf{p}, \mathbf{n}) = L_A \int_\Omega v(\mathbf{p}, \mathbf{l}) \cos\theta_i d\omega_i,$$

- Cook, Torrance added a visibility term
- AO:

$$k_A(\mathbf{p}) = \frac{1}{\pi} \int_\Omega v(\mathbf{p}, \mathbf{l}) \cos\theta_i d\omega_i.$$

- 0 if fully occluded, 1 if fully visible

Ambient Occlusion Math

- AO:

$$k_A(\mathbf{p}) = \frac{1}{\pi} \int_\Omega v(\mathbf{p}, \mathbf{l}) \cos\theta_i d\omega_i.$$

- 0 if fully occluded, 1 if fully visible
- K_A means surface irradiance changes with position

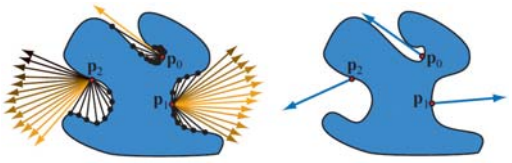
$$E(\mathbf{p}, \mathbf{n}) = k_A(\mathbf{p}) \pi L_A.$$

$$E(\mathbf{p}, \mathbf{n}) = \int_\Omega L_A \cos\theta_i d\omega_i = \pi L_A,$$

Object Space Ambient Occlusion

- AO does not depend on light direction
- Precompute AO for static objects using *ray casting*
  - How many rays?
  - How far do they go?
  - Local objects? Or all objects?

Object Space Ambient Occlusion



- Cosine weight rays
  - or use *importance sampling*: cosine distribute number of rays

Image courtesy of A K Peters, Ltd. http://www.realtimerendering.com/

Object Space Ambient Occlusion

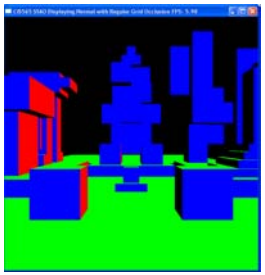
- Depends on scene complexity
- Stored in textures or vertices
- How can we
  - Support dynamic scenes
  - Be independent of scene complexity

Screen Space Ambient Occlusion

- Apply AO as a post processing effect using a combination of *depth*, *normal*, and *position* buffers
- Not physically correct but plausible
- Visual quality depends on
  - Screen resolution
  - Number of buffers
  - Number of samples

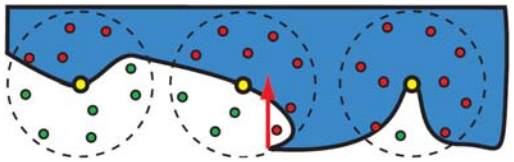
Depth Buffer

Normal Buffer



View Space Eye Position Buffer

Screen Space Ambient Occlusion



Sample a sphere and integrate the samples

Images courtesy of A K Peters, Ltd. http://www.realtimerendering.com/

Screen Space Ambient Occlusion

Figure 15. Screen-Space Ambient Occlusion in a complete ambient lighting situation (note how occluded areas darken at any distance)

Image from Martin Mittring. http://developer.amd.com/documentation/presentations/legacy/Chapter8-Mittring-Finding_NextGen_CryEngine2.pdf

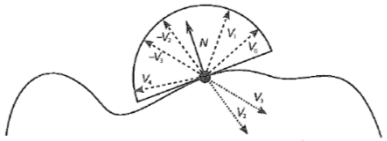
SSAO

Figure 9.39. The effect of screen-space ambient occlusion is shown in the upper left. The upper right shows the albedo (diffuse color) without ambient occlusion. In the lower left the two are shown combined. Specular shading and shadows are added for the final image, in the lower right. *(Images from "Crysis" courtesy of Crytek.)*

Images courtesy of A K Peters, Ltd. http://www.realtimerendering.com/

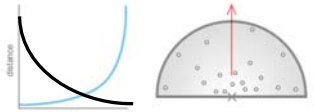
SSAO Improvement

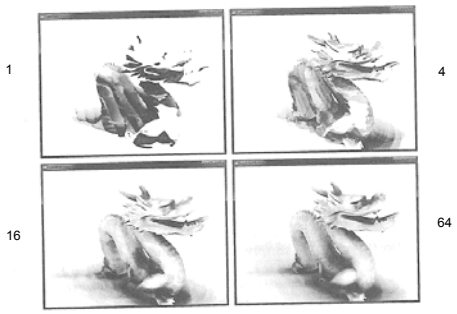
SSAO Improvement

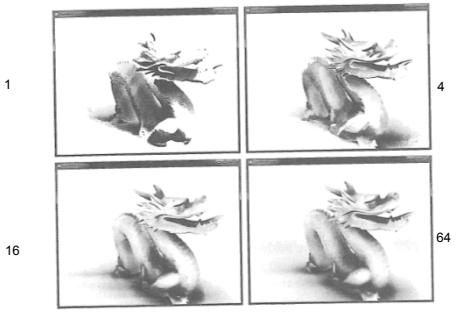


SSAO Improvement

- Poisson Sampling



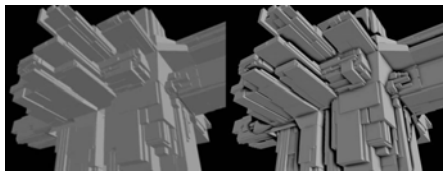




Demo


```
uniform sampler2D uTexInput;
uniform int uBlurSize = 4; // use size of noise texture
noperspective in vec2 vTexcoord; // input from vertex shader
out float fResult;
void main() {
    vec2 texelSize = 1.0 / vec2(textureSize(uInputTex, 0));
    float result = 0.0;
    vec2 hlim = vec2(float(-uBlurSize) * 0.5 + 0.5);
    for (int i = 0; i < uBlurSize; ++i) {
        for (int j = 0; j < uBlurSize; ++j) {
            vec2 offset = (hlim + vec2(float(x), float(y))) * texelSize;
            result += texture(uTexInput, vTexcoord + offset).r; } }
    Result = result / float(uBlurSize * uBlurSize); }
```

Another SSAO Method



Z-unsharp Mask
- Blur depth buffer
- Subtract it from original depth buffer
- Scale and clamp image, then subtract from original
- Superficially resembles AO but fast

Image from Mike Pan. http://mikepan.com

Z-unsharp Mask

Image Enhancement by Unsharp Masking the Depth Buffer: Luft et al. SIGGRAPH 2006

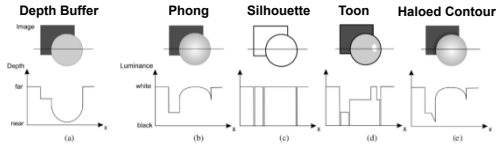
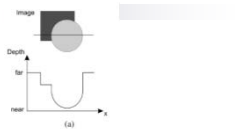


Figure 2: Example scene: (a) depth buffer along a scanline; (b-e) shading functions along a scanline: (b) Phong shading); (c) silhouette rendering; (d) toon shading; (e) haloed contour.

Z-unsharp Mask



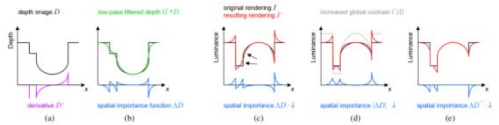
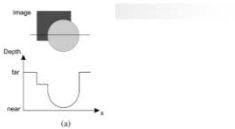
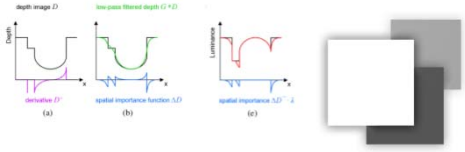


Figure 3: Depth functions: (a) depth function and its derivative; (b) difference between original and low-pass filtered depth buffer. Resulting shading: (c) adding the spatial importance function $\Delta D \cdot \lambda$ with $\lambda < 0$; (d) linear combination of the original input image and a high contrast version weighted by the absolute value of the spatial importance function $|\Delta D| \cdot \lambda$ with $\lambda > 0$; (e) depth darkening by using the negative fraction of the spatial importance function $\Delta D^{-} \cdot \lambda$ with $\lambda > 0$, which proved to be a natural image enhancement operator.

$\Delta D = G * D - D$

Z-unsharp Mask





$I' = I + \Delta D^{-} \cdot \lambda$

D^{-} is the negative fraction of the spatial importance
Which means, don't lighten the luminance

Z-unsharp Mask

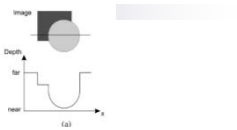
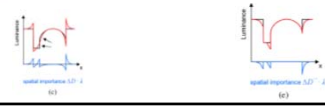


Figure 4: 3D example scene: (a) original rendering using a standard flat shading; (b) adding the spatial importance function ($\lambda < 0$); (c) linear combination of the original and a high contrast rendering; (d) depth darkening ($\lambda > 0$).

$I' = I + \Delta D^{-} \cdot \lambda$



Z-unsharp Mask

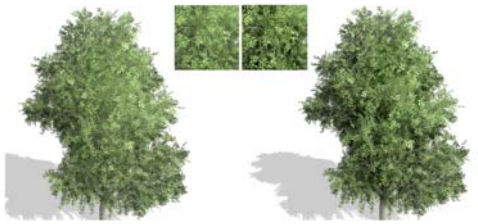


Figure 5: Enhancement of a complex botanical object using depth darkening.

SSAO

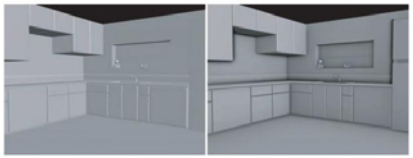


Figure 9.40. The Z-buffer unsharp mask technique for approximate ambient occlusion. The image on the left has no ambient occlusion; the image on the right includes an approximate ambient occlusion term generated with the Z-buffer unsharp mask technique. *(Images courtesy of Mike Fun.)*

Images courtesy of A K Peters, Ltd. http://www.realtimerendering.com/