Advanced Post Processing
Agenda

• Efficient Use of Blur

• Horizon-Based Ambient Occlusion
Efficient Use of Blur
Many Forms of Blur

Bloom

Motion Blur

Depth of Field

God Rays

And more...
Issues of Blur Effect

• Full-screen operation

• Texture intensive
  • Large texture
  • Large filter kernel

• Artifacts caused by undersampling
Increase Effective Samples

• Approach 1: Multiple-pass blurring

• Blurring between two textures iteratively. Each rendering pass uses previous blurring result as input
  • Easily increase the effective samples to 512

• Can be applied to many blur effects:
  • Motion blur
  • Radical blur
  • Light streak
  • Star, cross, glare effects, etc.
Multiple-pass Blurring

Illustration of god-ray effect in Crysis
Increase Effective Samples

• Approach 2: Utilize hardware linear filtering

• Given two neighbor point samples at location \( s_0 \) and \( s_1 \), weight \( w_0, w_1 \), we can use one linear sample instead:
  • Position: \( (s_0 w_1 + s_1 w_0) / (w_0 + w_1) \)
  • Weight: \( w_0 + w_1 \)
  • Half samples required

• Bilinear filtering can be used for cubic filtering
  • 2D cubic filtering requires 16 point samples for each pixel. Using 4 bilinear samples instead
  • Check NVSDK sample “Fast Third Order Filtering”
  • When downscaling the frame buffer, e.g. 1600x1200 -> 800x600 -> 400x300…, using cubic filtering can significantly improve image quality.
Utilize hardware linear filtering

- Separable kernel is much more effective

- 31x31 Gaussian filter. 32 actual samples: 16 for horizontal pass and 16 for vertical pass.
Questions?

calin@nvidia.com
Horizon-Based Ambient Occlusion
Sky Light

- Simplest form of Ambient Occlusion
- Light source = sky (sphere light)
- Two definitions of AO
  - AO = diffuse illumination from the sky [Landis 02] [Christensen 03]
  - AO = shadow from the sky illumination [Pharr and Green 04] [Hegeman et al. 06]
- Limited to outdoor scenes
Ambient Occlusion

- **Light = local hemisphere**
  - Centered at current surface point
  - Radius = user parameter
- **Can be rendered with ray-tracing**
  - [Gelato] [Mental Ray]

![Diagram of local sphere light]
Ambient Occlusion

• Gives perceptual clues of depth, curvature and spatial proximity – “Contact Shadow”
Screen Space Ambient Occlusion

- Approach introduced by
  - [Shanmugam and Orikan 07]
  - [Mittring 07] [Fox and Compton 08]
- Input = Z-Buffer + normals
  - Render approximate AO for dynamic scenes with no precomputations
- Z-Buffer = Heightfield
  - $z = f(x,y)$
• Given a 1D heightfield
Finding the Horizon

- Marching on the heightfield

- Z
- P
- $S_0$
- horizon angle
- +X
- sampling direction
Finding the Horizon

- Marching on the heightfield
Finding the Horizon

- Marching on the heightfield

Diagram showing points P, S₀, S₁, and S₂, with sampling direction and horizon angle.
Finding the Horizon

- Marching on the heightfield
Tangent Plane

- Given point P and its normal n
Horizon-Based AO

Horizon vector $H$

Tangent vector $T$

Horizon angle in $[-\pi/2, \pi/2]$

$$h(H) = \arctan(H.z / \|H.xy\|)$$

Tangent angle in $[-\pi/2, \pi/2]$

$$t(T) = \arctan(T.z / \|T.xy\|)$$

$$AO = \sin h - \sin t$$
Ambient Occlusion Radius

- Ambient occlusion radius defined in eye space
  - Scene = depth image
- Project light sphere into texture space
  - Approximate projection of the sphere by a disk
  - Project disk onto uv space
Sampling the Depth Image

- Use uniform distribution of directions per pixel
  - Fixed num samples / dir
  - Per-pixel randomization
  - Rotate directions by random per-pixel angle
  - Jitter samples by a random offset

Example with 4 directions / pixel
• We store per-pixel normals
  • Not interpolated normals
    • Would generate false occlusion
  • But face normals
    • Using ddx/ddy instructions on eye-space coordinates in the geometry pass
Core Algorithm

- Integrate AO in 2D
  - Average AO over multiple 2D directions $\theta$
  - $\text{AO}(\theta) = \sin h(\theta) - \sin t(\theta)$
Ambient Occlusion in Creases
Low-Tessellation Problem

- Z

θ

XY plane

AO > 0

false occlusion

correct sampling direction

tangent plane
Solution: Angle Bias

- Similar to “spread” parameter in [Mental Ray]
- Ignore occlusion near the tangent plane
The Angle Bias in Action

Without angle bias

With angle bias = 30 deg
Sampling Outside the Screen

- No scene information outside view frustum
  - We remove false occlusion by using clamping to edge and an angle bias

angle bias = 0
angle bias = 30 deg
Discontinuity Problem

AO(P₀) = \sin h - \sin t = \sin 0 - \sin 0 = 0

AO(P₁) = \sin h - \sin t = \sin(45\text{deg}) - \sin 0 = 0.7

→ Large AO discontinuity between P₀ and P₁
• Weight AO by a radial function $W(r)$
  - Similar to obscurances [Zhukov et al. 98]
  - “Falloff” in [Gelato] and [Mental Ray]

Normalized distance
$r = \frac{||S - P||}{R}$

We use the attenuation
$W(r) = 1 - r^2$
**Per-Sample Attenuation**

- Initialize $WAO = 0$
- After sample $S_1$
  - $AO(S_1) = \sin \Phi(S_1) - \sin t$
  - $WAO += W(S_1) AO(S_1)$
- After sample $S_2$
  - If $\Phi(S_2) > \Phi(S_1)$
    - $AO(S_2) = \sin \Phi(S_2) - \sin t$
    - $WAO += W(S_2) (AO(S_2) - AO(S_1))$
  - sampling direction

![Diagram showing the sampling process and equations related to per-sample attenuation.](image)
With and Without Attenuation

With Attenuation
\[ W(r) = 1 - r^2 \]

Without Attenuation
\[ W(r) = 1 \]
• Per-pixel randomization generates noise

AO with 6 directions x 6 steps/dir
Cross Bilateral Filter

- We blur the ambient occlusion

- Depth-dependent Gaussian blur
  - [Petschnigg et al. 04]
  - [Eisemann and Durand 04]
  - Reduces blurring across edges

- Although it is a non-separable filter, we apply it separately in the X and Y directions
Cross Bilateral Filter

• Depth-dependent Blur

Without Blur

With 15x15 Blur
Half-Resolution AO

• AO is mostly low frequency
  • Can render the AO in half resolution
    • Source half-resolution depth image
  • Still do the blur passes in full resolution
    • To avoid bleeding across edges
    • Source full resolution eye-space depths
  • [Kopf et al. 07]
Rendering Pipeline

- Render opaque geometry
- Render AO (Half or Full Res)
  - eye-space normals
  - eye-space depths
- Blur AO in X
- Blur AO in Y
- Modulate Color

Unprojection parameters (fovy and aspect ratio)
- Eye-space radius R
- Number of directions
- Number of steps / direction
- Angle bias

Kernel radius
- Spatial sigma
- Range sigma
Demo
Performance

• Depends on
  • Screen Resolution
  • Ambient Occlusion Resolution
  • Number of samples (directions * steps)
  • Blur Size
Half-Resolution AO

- Image Size: 1600x1200
- AO Resolution: 800x600
- Blur Resolution: 1600x1200

<table>
<thead>
<tr>
<th>Half-Res AO</th>
<th>GeForce GTX 280</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry AO</td>
<td>1.0 ms</td>
</tr>
<tr>
<td>Blur</td>
<td>3.5 ms</td>
</tr>
<tr>
<td>Total</td>
<td>7.0 ms</td>
</tr>
</tbody>
</table>

- 6 directions per pixel
- 6 steps per direction
- 15x15 Blur Size
- 143 fps
## Full-Resolution AO

### Image Size
- 1600x1200

### AO Resolution
- 1600x1200

### Blur Resolution
- 1600x1200

### NVIDIA GeForce GTX 280

<table>
<thead>
<tr>
<th>Feature</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry</td>
<td>1.0 ms</td>
</tr>
<tr>
<td>AO</td>
<td>30.0 ms</td>
</tr>
<tr>
<td>Blur</td>
<td>2.5 ms</td>
</tr>
<tr>
<td>Total</td>
<td>33.5 ms</td>
</tr>
</tbody>
</table>

- 6 directions per pixel
- 6 steps per direction
- 15x15 Blur Size
- 30 fps
Half-Resolution AO
6x6 samples / AO pixel
No Blur

AO = 3.5 ms @ 800x600
On GeForce GTX 280
Half-Resolution AO
6x6 samples / AO pixel
15x15 Blur

AO = 3.5 ms @ 800x600
Blur = 2.5 ms @ 1600x1200
On GeForce GTX 280
Full-Resolution AO
6x6 samples / AO pixel
15x15 Blur

AO = 30 ms @ 800x600
Blur = 2.5 ms @ 1600x1200
On GeForce GTX 280
Full-Resolution AO
16x16 samples / pixel
No Blur
Full-Resolution AO
16x32 samples / pixel
No Blur
Conclusion

• DirectX10 SDK sample
  • Now available on developer.nvidia.com
  • Including video and brief whitepaper
• Easy to integrate into a game engine
  • Input Data = eye-space depths and normals
  • Rendered in a post-processing pass
• More details in ShaderX7 (to appear)
Acknowledgments

• NVIDIA
  • Rouslan Dimitrov, Samuel Gateau, Michael Thompson, Ignacio Castano, the demo team

• Models
  • Dragon - Stanford 3D Scanning Repository
  • Sibenik Cathedral - Marko Dabrovic
Questions?

calin@nvidia.com
References


• [Christensen 03] Christensen, P. H. 2003. “Global illumination and all that”. In ACM SIGGRAPH Course 9.
References

References