Real-Time Horizon-Based Reflection Occlusion

Xin Zhao*, Xubo Yang†‡
Shanghai Jiao Tong University

Figure 1: *From left to right: Chinese dragon's neck and the occlusion of chin. The occlusion of the left crus on the body(pointed out by the black arrow). Chinese dragon. The cathedral.*

1 Introduction

Reflection occlusion(RO) method can yield the self-occluded effect when using environment maps to produce a reflection scene. In most modern renderers the common approaches to achieve RO are the ray-trace method and the static reflection maps method. We propose *screen-space reflection occlusion*(SSRO) method to compute RO in screen-space like *screen-space ambient occlusion* in [Bavoil et al. 2008] and *screen-space directional occlusion* in [Ritschel et al. 2009]. Our different strategies for both calculating the occlusion value and sampling in the depth image enable SSRO to calculate RO in real-time and to deal with dynamic scenes, variational lighting environments and changing views.

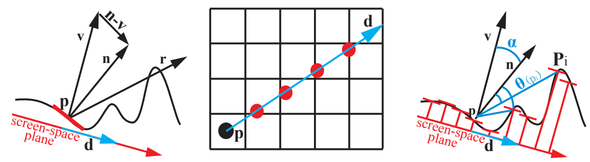
2 Reflection Occlusion Value Calculation

For the optically flat surface point $\mathbf{p}$, we sample the environment textures through an outgoing radiance vector $\mathbf{r}$, which is calculated from a unit vector $\mathbf{v}$ pointing to eye and the normal $\mathbf{n}$(see Figure 2a). In ordinary case, we retrieve color from texture data [Akenine-Möller et al. 2008]. But in the scene with RO, we employ a visibility function $\nu(\mathbf{p}, \mathbf{r})$ to check whether ray $\mathbf{r}$ cast from $\mathbf{p}$ is blocked by some objects prior to retrieving color. If no obstacle is detected($\nu(\mathbf{p}, \mathbf{r})$=1) then the sampled color is set. Otherwise($\nu(\mathbf{p}, \mathbf{r})$=0), a *DARK* color is set instead.

3 Visibility Function in Screen-Space

In our method the visibility function $\nu(\mathbf{p}, \mathbf{r})$ is approximated in screen-space. So our algorithm takes per-pixel linear depths and view-space normals as input.

Under the continuous heightfield assumption, we use a Monte Carlo approach to calculate $\nu(\mathbf{p}, \mathbf{r})$. We project the vector $\mathbf{n} - \mathbf{v}$ onto the screen-space ($\mathbf{d}$ on Figure 2a). Then we start sampling from pixel p in the direction of $\mathbf{d}$ as illustrated in Figure 2b. We can easily reconstruct the sampled pixel p_i's eye-space position $\mathbf{P}_i$ by sampling p_i's depth value like [Bavoil et al. 2008]. We then calculate the angle $\theta(p_i)$ between $\mathbf{n}$ and $\mathbf{P}_i - \mathbf{p}$. For checking the occurrence of blocking, we compare the angle α between $\mathbf{v}$ and $\mathbf{n}$ with $\theta(p_i)$ in Figure 2c. With indication function $\chi(x)$ equivalent to 1 if x ≤ 0,



(a) (b) (c)

Figure 2: *(a) Depths texture in 1D case. (b) Sample on the screen-space. (c) Compare angles α and $\theta(p_i)$ by sampling the depths.*

and 0 otherwise, we can integrate the visibility function:

$$\nu(\mathbf{p}, \mathbf{r}) = \chi(\sum(1 - \chi(\alpha - \theta(p_i)))). \quad (1)$$

Finally, we calculate the reflective color depending upon the visibility function value in screen-space from Equation 1.

We tested our algorithm on GeForce 8800 GT with a 1280×1024 resolution. The Chinese dragon case(see Figure 1c) was run at 77.56 FPS with 12 pixels' sample step. The Cathedral case(see Figure 1d) was run at 45.31 FPS with 4 pixels' sample step. Sample step indicates the interval between two samplings. The size of sample step in Monte Carlo method can be switched as a trade-off between effeciency and effects. SSRO can be applied to renderers and games. It could be extended by using multi-layer depth image like [Ritschel et al. 2009]. And we could improve it to cross-objects-occluded case by using several render targets.

References

AKENINE-MÖLLER, T., HAINES, E., AND HOFFMAN, N. 2008. *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA, 297.

BAVOIL, L., SAINZ, M., AND DIMITROV, R. 2008. Image-space horizon-based ambient occlusion. In *SIGGRAPH '08: ACM SIGGRAPH 2008 talks*, ACM, New York, NY, USA, 1–1.

RITSCHEL, T., GROSCH, T., AND SEIDEL, H.-P. 2009. Approximating dynamic global illumination in image space. In *I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ACM, New York, NY, USA, 75–82.

*e-mail: hatevol@gmail.com
†xubo.yang@gmail.com
‡Project 60970051 supported by NSFC

SIGGRAPH Asia 2009, Yokohama, Japan, December 16–19, 2009.
ISBN 978-1-60558-858-2/09/0012