MODERN METHODS OF REALISTIC LIGHTING IN REAL TIME

István Szentandrási
Master Degree Programme (2), FIT BUT
E-mail: xszent00@stud.fit.vutbr.cz

Supervised by: Adam Herout
E-mail: herout@fit.vutbr.cz

Abstract: Physically plausible illumination in real-time is often achieved using approximations. Recent methods approximate global illumination in the screen space by exploiting the capabilities of modern graphics cards. In this work I concentrated on screen-space ambient occlusion and screen-space directional occlusion. The main goal of this project was to further experiment with these methods and improve them. For uniform distribution of the sampling points, the Halton sequence is used. Methods are sped up by computing them in lower resolution. To reduce the noise and to upsample computed global illumination values, I used a geometry aware bilateral filtering.

Keywords: directional occlusion, ambient occlusion, Halton sequence, bilateral filtering

1 INTRODUCTION

Computing global illumination in real-time has been and still is a major challenge in computer graphics. Due to the complexity of light transport and some material properties, real-time frame rates can only be achieved at the cost of trade-offs and rough approximations. Perceptually among the most important optical phenomena belong soft shadows and indirect lighting. There have been many attempts to simulate either of these in real time. A handful of these attempts are based on ambient occlusion, which is very popular in the film industry as well as in games. The main advantage of these techniques lays in their speed and simple implementation.

2 SCREEN-SPACE GLOBAL ILLUMINATION TECHNIQUES

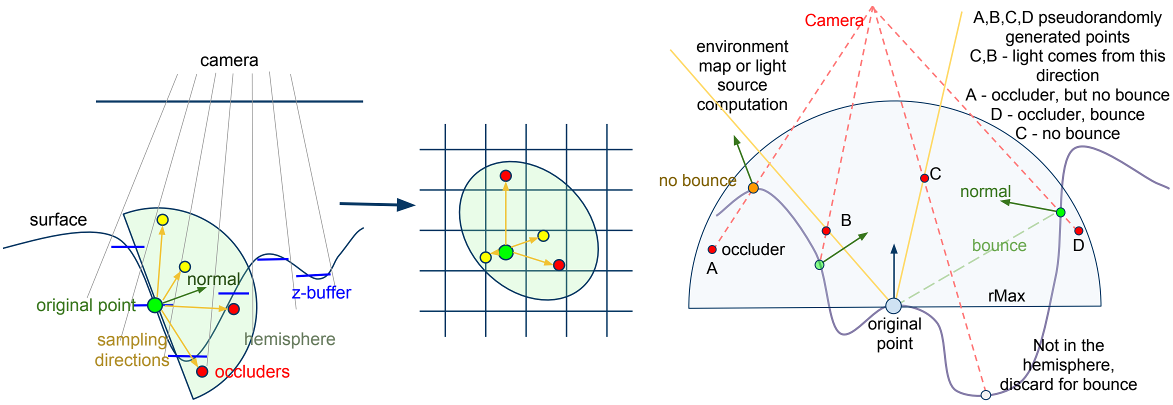
Ambient occlusion computes the visibility of the hemisphere at each point of the scene. The calculated visibility is used to modulate ambient component of the light, just as the name suggests. [1] Casting rays in every point still requires too much computing power, so a few alternative methods were introduced. Instead of computing occlusion over surfaces in 3D, these methods usually approximate AO in the screen space [2][3].

Screen-space ambient occlusion (SSAO) is very fast, it does not require any additional data and can be applied as a post-process to the scene. Screen-space directional occlusion (SSDO) [2] tries to combine the speed and simplicity of screen-space ambient occlusion methods with directional information of lighting and indirect color bleeding (indirect bounce). (Figure 1)

2.1 MODIFICATIONS AND IMPROVEMENTS

The aim of this project was to experiment with these methods. Although the following modifications were used for screen-space directional occlusion, some of them can be equally useful for screen-space ambient occlusion, too.

To avoid lighting computations for each sample per pixel in screen-space directional occlusion, I computed a modulation factor for each pixel similarly to ambient occlusion. These values may be



(a) Screen-space ambient occlusion. An area around the pixel is sampled in 2D (right part). Using the normal and depth only sampling points that should be in the hemisphere are used (left part). The generated depths for each sampling point are compared to the depth of the appropriate pixel in the z-buffer to determine, if the pixel corresponds to an occluder object (yellow (light) and red (dark) points).

(b) Screen-space directional occlusion. Random samples are generated in 3D in the hemisphere. Samples under the scene surface are classified as occluders. Otherwise, the incoming radiance can be computed from the direction defined by the sampling point. The sampling points are projected on the scene surface and based on the color and position of the pixel an indirect bounce is computed.

Figure 1: Screen-space directional and ambient occlusion principle.

further filtered to reduce noise. For this I used geometry-aware bilateral filtering described also by Reinbothe et al. [4].

Since directional occlusion values change slowly over surfaces (similarly to ambient occlusion values), they can be computed in lower resolutions. Upsampling values correctly based on the geometry properties of the scene is done using joint bilateral upsampling with the same modifications as for bilateral filtering.

2.2 Preprocessing step

To generate random samples with uniform distribution, I used values from Halton sequences [1]. The Halton sequence is a sequence of numbers in the interval $(0,1)$, which have a periodic property: points with indexes with a given offset (period) are from the same interval. This periodicity can be used to generate more sampling points for interesting regions in the image, and less for regions where the possibility of detecting an occluder is low, but still have a uniform distribution of samples ($k \cdot period, k \in N$ always covers the whole hemisphere).

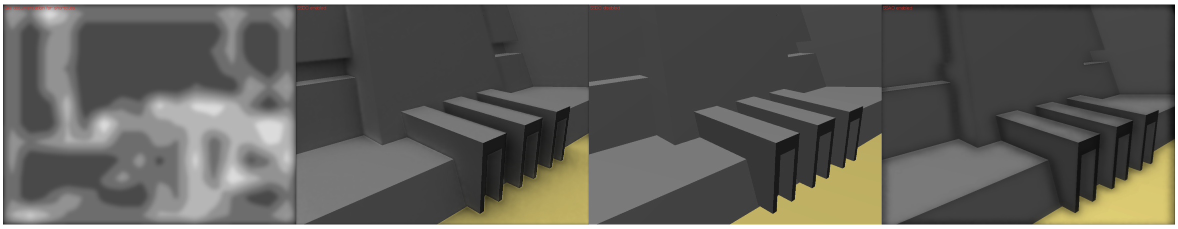
I added a preprocessing step to screen-space directional occlusion to filter out the more important regions by calculating the gradient over normals and depths of the pixels. Where the normals and depth values change significantly, there is a greater chance of detecting occluders. I applied this filter in much lower resolution and used the results to control the number of sampling point for screen-space directional.

3 RESULTS

The results of the achieved frame rates are summarized in Table 1. The methods were tested on four scenes with resolution 1024x768. These numbers are just exemplary. The speed of these methods depends in great extent also on the resolution, graphics hardware, as well as the degree of required smoothing. An example output of the used methods are on Figure 2.

Scene	Number of faces	SSAO [FPS]	SSDO [FPS]	SSDOv [FPS]
Dragon	201037	146	104	85
Buddha	290939	126	95	74
Cornell box	30	199	142	104
Boxes	2294	191	130	95

Table 1: Frame rates for each scene and method with 7x7 kernel for filtering and 3x3 kernel for upsampling. For SSAO and SSDO columns 10 sampling points were used per pixel, for the SSDOv column 20,30,40 or 50 sampling points were used based on the preprocessing step. For testing an ATI Radeon™ HD 4850 GPU was used.



(a) Preprocessing output. The lighter the pixels, the more samples are generated.

(b) Screen-space directional occlusion

(c) Without global illumination method

(d) Screen-space ambient occlusion

Figure 2: Example output for each technique and the output of the preprocessing step

4 CONCLUSIONS

Screen-space ambient occlusion was the fastest method for all scenes, but it has some limitations. Screen-space directional occlusion includes two generalizations that add directional occlusion and diffuse indirect bounces. Both extensions improve realism considerably for a minor computational cost.

Using the proposed modifications I was able to speed up screen-space directional occlusion. In the future, more experiments can be made to screen-space ambient occlusion and screen-space directional occlusion accompanied with more comprehensive testing based on the controllable features of each method. To get a clearer picture where these methods stand performance-wise, a few other methods could be explored, too.

REFERENCES

[1] Dutré, P., Bala, K., Bekaert, P.: Advanced Global Illumination (Second Edition). A K Peters, Ltd., Wellesley, MA, USA, 2006, ISBN: 1-56881-307-4

[2] Ritschel, T., Grosch, T., Seidel, H.-P: Approximating dynamic global illumination in image space. In I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games, 2009, 75-82.

[3] Shanmugam, P. and Arikan, O.: Hardware accelerated ambient occlusion techniques on GPUs. In Proceedings of the 2007 symposium on Interactive 3D graphics and games (I3D '07). ACM, New York, NY, USA, 2007, 73-80.

[4] Reinbothe, C., Boubekeur, T., Alexa, M.: Hybrid Ambient Occlusion. In EUROGRAPHICS 2009 Areas Papers, 2009.