

# The Algorithm for Generating Ambient Occlusion Maps Based on the Differential Algorithm

Jiaru Li\*

Chengdu University of Information Technology, Chengdu, Sichuan 610225, China

\*Corresponding Author: 2870982152@qq.com

## ABSTRACT

This paper introduces an algorithm for generating ambient occlusion maps based on the differential algorithm, aiming to enhance the environmental occlusion effects in graphic rendering through advanced computational techniques. The algorithm is inspired by the principles of the differential algorithm and the lighting information stored in normal maps. By inputting a differential step size, it cleverly samples and computes the gradient information of the normal map, resulting in the generation of high-quality ambient occlusion maps. The unique advantage of the algorithm lies in its sensitivity to lighting information and its flexible application of the differential step size. By effectively capturing subtle gradient variations in the normal map, it provides a more realistic and refined effect for the final physically based rendering (PBR) material graphics rendering.

## KEYWORDS

Differential Algorithm; Normal Map; Ambient Occlusion Map; PBR

## 1. INTRODUCTION

The emergence of photoreceptor cells allowed organisms to perceive this colorful and vibrant world for the first time. Humans, often referred to as the "essence of the universe, the spirit of all things" by Shakespeare, have always been deeply fascinated by the wonders of light and shadow. From ancestors worshipping the sun and moon as deities for good weather, to modern scientists discovering photons, from chemists exploring the mysteries of photocatalysts to materials scientists attempting to create stealth materials, humans have never given up the pursuit of the connection between light and the world.

A core research problem in computer graphics is how to efficiently simulate lighting in virtual three-dimensional space. With the introduction of radiosity theory, BRDF (Bidirectional Reflectance Distribution Function), ray tracing, and path tracing, based on physical optical models, computers can simulate lighting and render results indistinguishable from real scenes. On the other hand, while methods based on physical lighting can realistically simulate lighting, their enormous computational cost limits them to specific scenarios and applications. How to represent graphics more realistically with limited computing resources is also an important direction in graphics research.

An empirical observation of lighting is that in the absence of obvious direct light sources, concave areas generally have lower visibility, while convex areas have higher visibility (e.g., on overcast days). By measuring local visibility, areas with fewer occlusions are assigned higher ambient light intensity, and areas with more occlusions are the opposite. The ambient occlusion model is a way of "describing" visibility.

The concept of ambient occlusion was simultaneously proposed in literature[1]and [2]at SIGGRAPH in 2002. Literature[1] applied ambient occlusion in cases where only partially visible regions exist due to self-occlusion. This method removes the occluded part from the lighting equation, using the center of the remaining visible part as the new normal vector direction (Bent normal). Subsequently, the ambient occlusion map calculates the lighting intensity based on this normal vector. Literature[2] used two large area light sources to simulate ambient occlusion in large buildings. By saving the result of ambient occlusion as a texture map, it can extract the ambient occlusion result for a specific position in different viewpoints. These two works are considered the embryonic form of ambient occlusion.

In computer graphics rendering, given the geometric information of a scene, some algorithms directly and quickly calculate ambient occlusion on the depth image of the scene at the current viewpoint[3][4]. These methods estimate the visible angle along various directions by sampling between adjacent pixels and calculating ambient occlusion based on the depth variation between pixels. For real scenes with unknown geometry, some computer vision algorithms use images of the scene under different lighting conditions to calculate ambient occlusion[5][6]. Park[7] proposed a method to handle non-occluded surfaces by highlighting corners where ambient occlusion is likely to occur. By checking the corners of surfaces in screen space, potential illumination occlusions are highlighted. This achieved a performance improvement of 15% to 28% compared to previous works. Engel[8] introduced the Deep Volumetric Ambient Occlusion (DVAO) method, which can predict per-voxel ambient occlusion in volumetric datasets, considering global information provided through the transfer function. The proposed neural network only needs to be executed upon a change in this global information, thus supporting real-time volume interaction.

This paper introduces a new ambient occlusion map generation algorithm based on the differential algorithm and normal map. By sampling the gradient information of the normal map with a differential step size, it generates a high-quality ambient occlusion map. The algorithm in this paper is simpler compared to other algorithms.

## 2. RELATED WORK

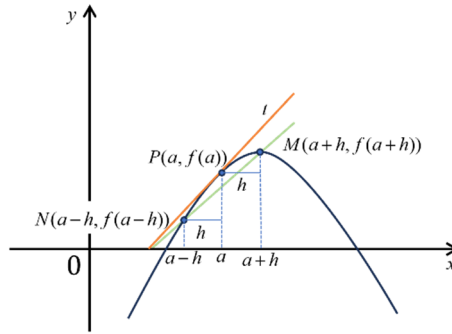
### 2.1. Center differential algorithm

Differential is an essential concept in discrete mathematics, representing the application of calculus theory to discrete numerical values. When a set of continuous values is represented in discrete form, differential equations are often expressed as a system of discrete equations. Central differencing is a specific type of differential used to approximate the derivative of a variable. Similar to derivatives in calculus, differentials are equivalent to the slope of the tangent line at a specific point. Central differencing involves calculating the average difference between the left and right points of a desired point, thereby obtaining an approximate derivative value for that point. In practical numerical calculations, the format of the central differencing scheme is often represented as Eq.1, where  $h$  is the predetermined step size. The advantage of this differencing scheme lies in its more accurate results, as it utilizes information from both sides of the target point. In contrast, forward differencing (using information from the target point and its right side) and backward differencing (using information from the target point and its left side) may introduce larger errors.

$$f'(x) \approx [f(x+h) - f(x-h)] / 2h \quad (1)$$

In two-dimensional space, for a point  $P(a, f(a))$  on the graph of the function  $y=f(x)$ , calculating the equation of the tangent line at that point allows us to determine the function perpendicular to the tangent line at that point. In mathematics, the slope of the tangent line at this point can be calculated using the central difference algorithm. By taking a small increment  $h$  as the step size, we obtain the point  $M(a+h, f(a+h))$  in the positive  $x$ -axis direction and the point  $N(a-h, f(a-h))$  in the negative  $x$ -

axis direction. The slope of the line MN, denoted by  $k$ , is given by  $k = (f(a+h)-f(a-h))/(2a)$ . As  $h$  approaches zero, the points M and N infinitely approach the point P. At this limit, the slope  $k$  of line MN approaches the slope  $t$  of point P. Therefore, we can approximately determine the slope of point P by taking a small enough  $h$ .



**Figure 1.** Central Difference Method Schematic

In computer graphics, when computing the gradient changes of a certain point in the  $uv$  direction in a two-dimensional image, this kind of thinking can also be adopted. The specific approach is to establish a coordinate system with the position of the pixel as the center, and consider the slope distribution in the  $u$  and  $v$  directions. For the current pixel point  $(u, v)$  with a certain differential step size  $step$ , four points are taken, namely  $(u+step, v)$ ,  $(u-step, v)$ ,  $(u, v+step)$ , and  $(u, v-step)$ . Then, the gradient change rates in the  $u$  and  $v$  directions are calculated according to Eq.3.

$$\begin{cases} du = (C(u_{i-step}, v_j) - C(u_{i+step}, v_j)) / 2 \\ dv = (C(u_i, v_{j-step}) - C(u_i, v_{j+step})) / 2 \end{cases} \quad (2)$$

$$\begin{cases} dux = (step, 0, du) \\ dvy = (0, step, dv) \end{cases} \quad (3)$$

## 2.2. Algorithm Design

To calculate the potential ambient occlusion (AO) for each point on a non-smooth surface, the algorithm considers the normal vector information at each point on the object's surface. The use of normal maps for AO map calculation is justified by the following reasons:

- (1) **Microscopic Geometry Detail Simulation:** Normal maps capture subtle variations in surface geometry, which are crucial for computing ambient occlusion. Microscopic details on surfaces, such as bumps and crevices, play a significant role in how they receive and reflect light. The complexity of these details directly affects the AO values on the surface.
- (2) **Directional Lighting Computation:** Ambient occlusion is essentially an estimate of the cumulative effect of ambient light in a local region. The vectors in the normal map provide precise data on how lighting directionally influences any given point. This information is valuable for calculating the probability of points being shadowed by surrounding elevated or recessed areas.

Based on this, the paper proposes an algorithm for generating ambient occlusion maps using the information stored in normal maps. The algorithm relies on the principle of the central differencing method. It takes the normal map as input and, leveraging the central differencing method, samples the normal information around each point to calculate the degree of occlusion.

The algorithm is designed with the following steps:

- (1) **Sampling Normal Map:** Sample the input normal map in both horizontal and vertical directions based on the differential step size.

- (2) Gradient Calculation: Compute the gradients in the horizontal and vertical directions using Eq.2. Then, calculate the total normal gradient vector at the point using Eq.4.
- (3) Transformation to Tangent Space: Convert the normal information to tangent space. Utilizing the obtained tangent space vectors, compute the ambient occlusion values for the four sampled points based on Eq.5.
- (4) Average AO Calculation: Using Eq.6, calculate the final average ambient occlusion value.

$$normal = normalize(cross(dux, dvy)) \quad (4)$$

$$aoValues = float4(
\begin{aligned}
& saturate(dot(normal, normal1.xyz)), \\
& saturate(dot(normal, normal2.xyz)), \\
& saturate(dot(normal, normal3.xyz)), \\
& saturate(dot(normal, normal4.xyz))
\end{aligned}
) \quad (5)$$

In Eq.5, "normal" represents the normal information in tangent space obtained from the gradient information, while "normal1", "normal2", "normal3", and "normal4" represent the normal information sampled from the other four points based on the differential step size.

$$AO = (aoValues.x + aoValues.y + aoValues.z + aoValues.w) / 4 \quad (6)$$

In Eq.6, AO represents the final ambient occlusion value for that point, which is calculated based on the environmental light obstruction.

The pseudocode for the algorithm is as shown in the following table:

**Table 1.** AO Map Generation Algorithm Based on Normal Map

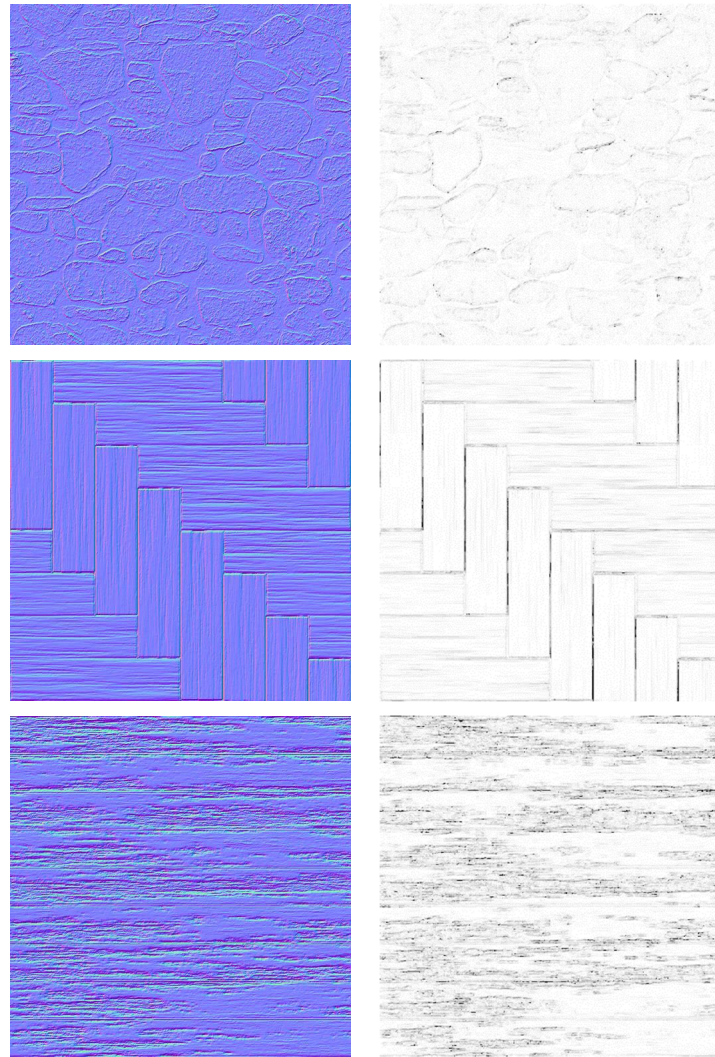
---

AO Map Generation Algorithm Based on Normal Map
Input: Normal map, Differential step size
Output: Ambient occlusion value
float CalculateAO(sampler2D normalMap,float step)
{
float2 textureSize=float2(1024,1024);
// Calculate the offset values in the uv space based on the differential step size.
float2 delta = float2(step*1.0 / textureSize.x, step*1.0 / textureSize.y);
float3 normal1 = UnpackNormal(tex2D(normalMap, uv+float2(-delta.x,0)));
float3 normal2 = UnpackNormal(tex2D(normalMap, uv+float2(delta.x,0)));
float3 normal3= UnpackNormal(tex2D(normalMap, uv+float2(0,delta.y)));
float3 normal4= UnpackNormal(tex2D(normalMap, uv+float2(0,-delta.y)));
float du=(normal2-normal1)/2;
float dv=(normal4-normal3)/2;
// Compute the gradient information
float3 dux=float3(delta.x,0,du);
float3 dvy=float3(0,delta.y,dv);
float3 normal = normalize(cross(dux, dvy));
// Central Difference Algorithm, simultaneously calculating occlusion in multiple directions
float4 aoValues = float4(
saturate(dot(normal, normal1.xyz)),
saturate(dot(normal, normal2.xyz)),
saturate(dot(normal, normal3.xyz)),
saturate(dot(normal, normal4.xyz))
);
// Calculate the average value
float ao = (aoValues.x + aoValues.y + aoValues.z + aoValues.w) * 0.25;
return ao;
}

---

### 2.3. Results

According to the above algorithm, the experimental results of generating AO maps from the input normal maps are as follows:



(a) Input Texture

(b)AO Texture

**Figure 2.** Experimental Results of AO Texture

### 3. CONCLUSION

The principle of calculating ambient occlusion (AO) map using normal maps involves storing surface normal information for each pixel in the normal map. In this paper, the characteristics of storing lighting information in the normal map are utilized, and, in combination with the central differencing method, the normal gradient is calculated to estimate the rate of normal change at each pixel. This approach allows for inferring the degree to which the surface normal points toward a concave direction, thereby estimating the occlusion of ambient light at that point. The ambient occlusion map represents the light occlusion situation in the environment around a specific point, with the algorithm indicating that the more the normal points toward a concave direction, the more pronounced the occlusion. The effectiveness of this algorithm is validated through experiments.

### REFERENCES

- [1] Landis H. Production-ready global illumination[J]. Siggraph course notes, 2002, 16(2002): 11.
- [2] Bredow R, Imageworks S P. Renderman on film[J]. SIGGRAPH 2002 Course 16 notes, RenderMan in Production, 2002.

- [3] Bavoil L, Sainz M, Dimitrov R. Image-space horizon-based ambient occlusion[M]//ACM SIGGRAPH 2008 talks. 2008: 1-1.
- [4] Bavoil L, Sainz M. Multi-layer dual-resolution screen-space ambient occlusion[M]//SIGGRAPH 2009: Talks. 2009: 1-1.
- [5] ZUCKER SW, LANGER MS. SHAPE-FROM-SHADING ON A CLOUDY DAY[J].467-478,1994,11(2):467-478.
- [6] Prados E, Jindal N, Soatto S. A non-local approach to shape from ambient shading[C]//International Conference on Scale Space and Variational Methods in Computer Vision. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 696-708.
- [7] Park S, Baek N. A Corner-Highlighting Method for Ambient Occlusion[J]. Applied Sciences-Basel, 2021, 11(7): 3276.
- [8] Engel D, Ropinski T. Deep Volumetric Ambient Occlusion[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(2): 1268-1278.