

Image-Based Diffuse Lighting Using Visibility Maps

Ivan Neulander*
Rhythm and Hues Studios

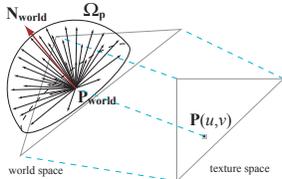
1 Introduction

We present an efficient technique for systematically constructing texture maps that approximate the diffuse lighting of an object, based on illumination from an environment map. Image-based diffuse lighting involves integrating an irradiance function over the full visibility hemisphere at each point of the object’s surface. This requires expensive occlusion detection, typically using a ray tracer. Our technique factors out the ray tracing into an intermediate *visibility map* (VM). Once computed, the VM is rapidly combined with an arbitrary environment map to produce an approximate diffuse shaded texture, which is applied in the final render.

The advantages of our method are: 1) fast initial computation of the diffuse lighting texture and 2) near real-time subsequent computation given a new environment map and any rotation of the object with respect to the map. Compared to the recent work by [Sloan et al. 2002], our approach is limited to diffuse shading but is simpler and requires less precomputation and storage. We have yet to implement our method in graphics hardware, which we expect will improve interactivity.

2 Generating a Visibility Map

Our technique is similar in spirit to the estimation of ambient occlusion [Christensen 2002], which uses a ray tracer to compute at each surface point the proportion of the visibility hemisphere that is occluded. We require the target object to be parametrized so that each point on its surface maps to a unique texture location. We assume that the object consists of triangles, with vertices containing texture and position coordinates, and surface normals. We rasterize each triangle in texture space and bilinearly interpolate the position and normal vectors.



At each interior pixel, we cast a set of rays from the interpolated position covering the hemisphere defined by the interpolated normal. The ray directions are distributed according to a jittered two-dimensional Halton sequence. At each pixel, we record into the VM the average ray visibility value, and the average direction of all unoccluded rays, weighted by each ray’s Lambertian contribution (dot product with normal). This data neatly fits into 4 IEEE floats per pixel. No antialiasing or hidden surface removal is needed during rasterization.

3 Combining with an Environment Map

We first apply a cosine filter to the environment map to produce a *cosine-filtered environment map* (CFEM), as described in [Miller and Hoffman 1984]. Each pixel of the CFEM represents the diffuse lighting contribution of the entire hemisphere of the EM centered about that pixel. We map the average unoccluded direction vector

found at each pixel of the VM to the appropriate pixel of the CFEM, scaling the CFEM color by the ambient visibility value from the VM. Our software implementation converts a 1024×1024 VM into a shaded texture in 2 seconds on a *PIII-1GHz*. We expect a vertex-shaded GPU implementation to be considerably faster.

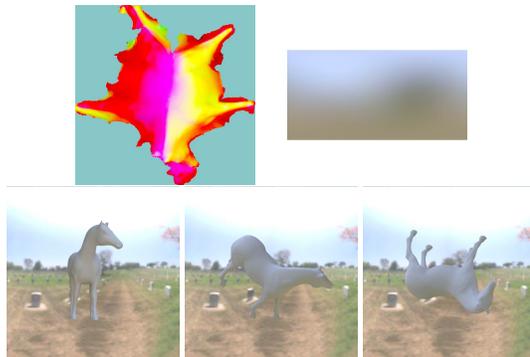


Figure 1: The VM and CFEM (above) are used to render multiple frames (below) in under 5 seconds each. The VM took 5 minutes to generate on a *PIII-1GHz* for a 40K triangle model (courtesy of Alla Sheffer and John C. Hart).

4 Limitations and Conclusions

The accuracy of our VM lighting model is diminished by complex occlusion and high frequencies in the environment map. Nevertheless, the results are generally convincing because, though physically inaccurate, they are noise-free and consistent in animation.

A VM is reusable over many frames as long as the target geometry is rigid, though minor deformations are forgivable. Our technique is best suited to shading a single object isolated from others. While we can capture object-to-object shadowing, the VM will not be reusable if the objects move relative to one another.

Despite these limitations, VM lighting is useful in many scenarios: interactively previewing environment-lit objects; optimizing the image-based lighting of complex surfaces such as hair; and efficient image-based lighting of large static scenes, such as buildings and landscapes.

References

- CHRISTENSEN, P. H. 2002. Note #35: Ambient occlusion, image-based illumination, and global illumination. *PhotoRealistic RenderMan Application Notes*.
- MILLER, G. S., AND HOFFMAN, C. R. 1984. Illumination and reflection maps: Simulated objects in simulated and real environments. In *SIGGRAPH '84 Advanced Computer Graphics Animation seminar notes*, ACM SIGGRAPH.
- SLOAN, P.-P., ET AL. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH '02 Proceedings*, ACM SIGGRAPH.

*e-mail: ivan@rhythm.com