Irradiance Gradients

Gregory J. Ward*
LESO-PB
Ecole Polytechnique Federale de Lausanne
CH-1015 Lausanne

Paul S. Heckbert**
Department of Technical Mathematics & Informatics
Delft University of Technology
Julianalaan 132
2628 BL Delft
Netherlands

ABSTRACT

A new method for improving the accuracy of a diffuse interreflection calculation is introduced in a ray tracing context. The information from a hemispherical sampling of the luminous environment is interpreted in a new way to predict the change in irradiance as a function of position and surface orientation. The additional computation involved is modest and the benefit is substantial. An improved interpolation of irradiance resulting from the gradient calculation produces smoother, more accurate renderings. This result is achieved through better utilization of ray samples rather than additional samples or alternate sampling strategies. Thus, the technique is applicable to a variety of global illumination algorithms that use hemicubes or Monte Carlo sampling techniques.

1. Introduction

Global illumination can be simulated using both ray tracing and radiosity algorithms. Both approaches typically rely on calculations of patch irradiances which are used to revise other patch irradiances iteratively or to render a final image†. In most radiosity algorithms, patch radiosities are considered

*First author's current address:
Lawrence Berkeley Laboratory
1 Cyclotron Rd., 90-3111
Berkeley, CA 94720
USA
+1-510-486-4757
+1-510-486-4089 fax
e-mail: gjward@lbl.gov

**Second author's current address:
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh PA 15213-3890
USA
+1-412-268-7897
e-mail: ph@cs.cmu.edu

†Irradiance is the energy flux per unit area arriving on a surface. Radiosity is the emissive flux per unit area, plus the irradiance times the diffuse surface reflectance.

constant during the solution stage, and bilinear interpolation (Gouraud shading) is used to compute pixel values during rendering. It has been shown that these piecewise-constant approximations are quite inaccurate, and that much more accurate approximations can be simulated using linear, quadratic, and higher order approximation [Heckbert91b]. Linear approximations have recently been implemented for 3-D radiosity [Max92] and 2-D radiosity [Heckbert91a] [Lischinski91].

Higher order approximations require more information about the irradiance function in order to be worthwhile, however; one must increase either the number of samples or the information content of each sample. If higher order interpolation were used without additional information, the resulting shading would look smoother, but it would be objectively no more accurate than standard bilinear interpolation.

Rather than increase the number of samples, as more brute-force algorithms have done, our method increases the information content of each sample to include estimates of the first derivative, or gradient, of the irradiance.

In this paper, we will show how the irradiance gradient at a point can be computed during a standard Monte Carlo evaluation of irradiance at almost no additional cost. The key to this innovation is the wealth of information contained in a sampling of the hemisphere. During the sampling process, the distances, brightnesses, and directions of the visible surfaces are all known, and from this knowledge it is possible to deduce with reasonable accuracy how the irradiance will change with respect to position and orientation of the test surface element. The gradient approximation given here is based on minimal, intuitive assumptions of geometric continuity.

Knowing the irradiance gradient along with the irradiance value at a point allows us to justify a bicubic or like-order interpolation method, and produces not only smoother but significantly more accurate results. The irradiance gradient method will be demonstrated in the context of a meshless irradiance caching scheme [Ward88b], though the technique may also be applied in mesh-dependent radiosity algorithms.

2. The RADIANCE Simulation

Radiance is a physically-based lighting simulation system developed over the past seven years at the Lawrence Berkeley Laboratory in California and the Ecole Polytechnique Federale de Lausanne in Switzerland. The software is free and publicly available from anonymous ftp sites at both locations. Since the algorithm described in this paper has been implemented in the context of the *Radiance* program, it is necessary to briefly explain the workings of this simulation before delving into the irradiance gradient calculation itself.

Radiance is basically a light-backwards ray tracing program [Whitted80] that uses irradiance caching to efficiently account for diffuse interreflection between surfaces. The basic algorithms used in *Radiance* are described in [Ward88a] and a general overview of the software is provided in [Ward90]. Basically, *Radiance* uses ray tracing in a recursive evaluation of the radiance equation†:

$$L_r(\theta_r,\phi_r) = \int_0^{2\pi}\int_0^{\pi/2} L_i(\theta_i,\phi_i)\, f(\theta_i,\phi_i;\theta_r,\phi_r)\, \cos\theta_i\, \sin\theta_i\, d\theta_i\, d\phi_i \tag{1}$$

where:

θ is the polar angle measured from the surface normal

ϕ is the azimuthal angle measured about the surface normal

$L_r(\theta_r,\phi_r)$ is the reflected radiance (watts/steradian/meter^2 in SI units)

$L_i(\theta_i,\phi_i)$ is the incident radiance

$f(\theta_i,\phi_i;\theta_r,\phi_r)$ is the bidirectional reflectance-transmittance distribution function (steradian^{-1})

†The radiance equation is essentially Kajiya's rendering equation [Kajiya86] with the notion of energy transfer between

To reduce the variance between samples and speed convergence of Monte Carlo integration, light sources are accounted for separately using an adaptive sampling scheme [Ward91]. As in most ray tracing algorithms, specular contributions are computed with separate rays in the appropriate directions. Once the direct and specular contributions have been removed from the integral, the indirect diffuse contribution is computed using a Monte Carlo sampling of the hemisphere. Since this "indirect irradiance" value is view-independent, and it changes slowly over surfaces in most scenes, it is more efficient to perform the calculation only occasionally, caching the computed values for local interpolation. This caching of irradiance samples is a significant optimization of more brute-force Monte Carlo ray tracing algorithms such as Kajiya's [Kajiya86].

In the meshless caching scheme described in [Ward88b], the location of the computed indirect irradiance values is determined by the proximity and curvature of the surfaces, and does not fall on a regular grid, so a weighted sum is used in place of a more standard bilinear interpolation. Furthermore, since values are only computed as needed by the algorithm, extrapolation may occur in a region where no previous values existed, until the need for a new value is strong enough to trigger another Monte Carlo calculation. This process can result in some rather disturbing artifacts in a single pass scanline rendering, as shown in Figure 1a. The commonly applied solution to this problem involves a low-resolution overture calculation to fill the desired view with indirect irradiance values prior to the final high-resolution pass. Although this prepass requires only a modest additional expense, it is rather annoying that it should be required by an otherwise elegant rendering algorithm.

The benefit of calculating the irradiance gradient is two-fold for the caching scheme used by *Radiance*. First, we are able to produce more accurate interpolated values because we can use the gradient information effectively in a higher order interpolating function. Second, we are able to produce more accurate *extrapolated* values and thus greatly reduce caching artifacts. Figure 1b shows the same single-pass calculation, this time using estimates of the irradiance gradient to more accurately extrapolate values in unsampled regions of the image. We emphasize that the second image took approximately the same time to produce as the first, and used the same hemisphere samples to compute the irradiances. The only difference is that the second image extracted additional information from the hemisphere samples to deduce the irradiance gradient at each sample point, and these gradients were used to better interpolate and extrapolate irradiance values for the image. (Since the test environment contains no specular surfaces and no direct illumination sources, changes in the diffuse interreflection calculation are more evident here than in most scenes.)

Figure 1c plots interpolated irradiance values for a vertical line passing just under the right side of the sphere. The difference between the actual irradiance and the cubic interpolation is too small to see in this plot, but the poor match of the linearly interpolated irradiance is clearly evident. The relative errors shown in Figure 1d amply demonstrate that a cubic interpolation of irradiance based on gradient information is much more accurate than a standard linear interpolation.

2.1. Indirect Irradiance Calculation

The indirect irradiance is calculated in *Radiance* using a fixed number of samples in a uniformly weighted, stratified Monte Carlo sampling:

$$E = \frac{\pi}{M \cdot N} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} L_{j,k} \qquad (2)$$

where:

$L_{j,k}$ is the indirect radiance in the direction $(\theta_j, \phi_k) = \left(\sin^{-1}\sqrt{\frac{j+X_j}{M}},\ 2\pi\frac{k+Y_k}{N} \right)$

X_j, Y_k are uniformly distributed random variables in the range [0,1)

two points replaced by energy passing through a point in a specific direction.

$M \cdot N$ is the total number of samples and $N \approx \pi M$

Note that sample rays that intersect light sources must be excluded from the above summation because direct illumination is accounted for in a separate step. The resulting sum is the indirect contribution to irradiance at a specific point on a surface.

Irradiance samples, consisting of a point, a normal vector, and an irradiance value, are stored in an octree for later interpolation. This irradiance octree is separate from the octree used to optimize ray-surface intersection. In the original implementation, interpolation was a simple weighted sum over usable irradiance values. The weight of a sample decreased as the interpolation point or normal vector deviated away from the sample's point and normal. A sample was considered usable if the estimated error of its contribution to the approximation was less than a user-specified accuracy tolerance. The error was estimated from the local geometry using the "split sphere model" to compute an upper bound on the magnitude of the irradiance gradient.

The split sphere model is only a crude estimate of the gradient magnitude. It may be sufficient to decide the spacing of irradiance calculations, but to improve our interpolation we need to know the actual irradiance gradient, not just a directionless upper bound. Fortunately, the information we need is already contained in the hemisphere sampling.

3. The Gradient Calculation

Since the irradiance in a scene is a function of five variables, three for the position and two for the direction, the irradiance gradient should be a five-dimensional vector. For computational convenience, we will compute instead two separate three-dimensional vectors. One will correspond to the expected direction and magnitude of the **rotational gradient** and the other to the direction and magnitude of the **translational gradient**. Both gradient vectors will lie in the base plane of the sample hemisphere, which is the tangent plane of the sample. Thus, each vector will in fact represent only two degrees of freedom. This representation of the gradient is used because we only interpolate across a surface. Furthermore, the irradiance above and below most surfaces is discontinuous, and the gradient with respect to displacement in the polar direction is therefore undefined.

Our calculations of the rotational and translational irradiance gradients are based on very simple observations about the sampled environment. The sampling of rays over a hemisphere tells us much more than the total light falling on the surface. It tells us the distance, direction, and brightness of each contribution.

The directions and brightnesses tell us how irradiance changes as the sample hemisphere is rotated because they indicate how the cosine projection of those contributions affects the overall sum. To take a simple example, Figure 2a shows a single contributing surface. The background is assumed to be darker than the surface. If we rotate our sample hemisphere to face this surface, its contribution becomes proportionally larger than other contributions. If we rotate away from the surface, its overall contribution is diminished. By summing over all such potential changes, we can compute the total rotational gradient for the hemisphere.

For the translational gradient, the distances to the contributing surfaces must be considered because occlusion plays an important role. In Figure 2b, a darker surface occludes a brighter surface in the background. As the sample hemisphere is moved to the right in the diagram, the influence of the brighter background surface becomes stronger, and therefore the translational gradient is positive in this direction. By summing over at all such changes, we can compute the overall gradient with respect to translation.

As an example of the kind of information available during the hemisphere sampling, see Figure 3. Figure 3a shows a projected hemisphere as seen from a point on the floor of a conference room. Figure 3b is a false color image showing the distances to the surfaces as determined by a ray tracing calculation. If we were to move towards the chair in the upper left of the image, we would note a decrease in the overall irradiance as the chair's dark underside covered more of our view of the ceiling. Figure 3c shows a uniformly weighted stratification of about 2000 samples as computed by the *Radiance* interreflection calculation. Notice that the light sources appear dark, as they must be excluded from the indirect contributions. Notice also that this image appears very crude as a rendering, yet it contains many more samples than are typically used to calculate the indirect irradiance.

3.1. The Rotational Gradient

The rotational gradient formula simply sums the differential of the cosine for each contribution sample. For the hemisphere sampling given in Equation (2), the formula is:

$$\vec{\nabla}_r E = \frac{\pi}{M \cdot N} \sum_{k=0}^{N-1} \left\{ \hat{v}_k \sum_{j=0}^{M-1} -\tan\theta_j \cdot L_{j,k} \right\} \qquad (3)$$

where:

$\hat{v}_k$ is the base plane unit vector in the $\phi_k + \frac{\pi}{2}$ direction

The tangent function appears in the summation because the differential of the cosine is negative sine and our sampling contains the cosine weighting implicitly, thus it is necessary to multiply the sample values by the tangent (sine over cosine) to get back a sine weighting.

3.2. The Translational Gradient

To compute the translational irradiance gradient, we consider how the projected solid angle of each of the *MN* hemisphere cells is affected by a translation of the hemisphere center. The change in projected solid angle, when multiplied by the radiance of a cell, will give the change in that cell's contribution. This rate of change is determined in part by the distance to the contributing surface. The closer the surface, the higher the rate of change with respect to a displacement perpendicular to the boundary between neighboring cells. In fact, it is always the distance to the *closer* surface that determines the rate of change in occlusion, since the relative motion of a foreground surface is greater than that of a background surface.

Figure 4 shows the projection of a stratified hemisphere sampling onto the tangent plane of the surface. Each cell has an equal projected solid angle (ie. $\frac{\pi}{MN}$ steradians), thus each cell should cover the same area in this diagram. To determine how the irradiance changes with translation in this tangent plane, we sum the marginal changes for each cell. For cell (j,k) shown in the diagram, we consider two approximately perpendicular directions. (Note that we have shown the sample direction at the center of the cell, but in fact it lies at some random location in the cell.) One direction is polar, the other is azimuthal. Computing the marginal change in irradiance for this cell reduces to computing the marginal change in the two highlighted cell walls with respect to translation.

The change in irradiance with respect to translation for each cell wall is simply the length of the cell wall multiplied by the rate of motion of the wall with respect to motion in a specific direction. For the wall separating the two adjacent cells with the same θ, the length of the cell wall is given by the integral of cos(θ) from θ_{j_-} to θ_{j_+}. This is simply $(\sin\theta_{j_+} - \sin\theta_{j_-})$. For motion perpendicular to this wall (ie. in the direction $\hat{v}_{k_-}$ defined below), it is simple to show that motion of the cell wall is proportional to $1/Min(r_{j,k}, r_{j,k-1})$, where $r_{j,k}$ is the intersection distance in cell (j,k). Thus, the change in irradiance with respect to motion along $\hat{v}_{k_-}$ for cell (i,j) is:

$$\frac{\sin\theta_{j_+} - \sin\theta_{j_-}}{Min(r_{j,k}, r_{j,k-1})} \cdot (L_{j,k} - L_{j,k-1})$$

Since we have computed the change in the location of the cell wall, we must use the difference in the adjacent radiance samples to determine how this will affect the overall irradiance sum.

For the cell wall separating the two adjacent cells in the polar direction, the length of the cell wall is $\frac{2\pi}{N} \cdot \sin\theta_{j-}$. The motion of the wall with respect to the vector perpendicular to it (ie. $\hat{u}_k$ defined below)

Figure 4. Cells of an example hemisphere sampling.

is equal to $\frac{\cos^2\theta_{j_-}}{Min(r_{j,k},r_{j-1,k})}$. One cosine comes from the projection of the hemisphere's tangent into the plane, and the other cosine comes from the reduced change in θ as a function of angle. (A rigorous proof of this formula is left as an exercise for the reader.) Combining these terms, we arrive at the following formula for the change in irradiance with respect to motion along $\hat{u}_k$ for the cell (i,j):

$$\frac{2\pi}{N}\cdot\frac{\sin\theta_{j_-}\cdot\cos^2\theta_{j_-}}{Min(r_{j,k},r_{j-1,k})}\cdot(L_{j,k}-L_{j-1,k})$$

Combining these terms into a sum over all cells, we arrive at the following formula for the translational irradiance gradient:

$$\vec{\nabla}_t E = \sum_{k=0}^{N-1}\left\{\hat{u}_k\frac{2\pi}{N}\sum_{j=1}^{M-1}\frac{\sin\theta_{j_-}\cdot\cos^2\theta_{j_-}}{Min(r_{j,k},r_{j-1,k})}\cdot(L_{j,k}-L_{j-1,k}) \;+\; \hat{v}_{k_-}\sum_{j=0}^{M-1}\frac{\sin\theta_{j_+}-\sin\theta_{j_-}}{Min(r_{j,k},r_{j,k-1})}\cdot(L_{j,k}-L_{j,k-1})\right\} \tag{4}$$

where:

$\hat{u}_k$ is the unit vector in the ϕ_k direction

$\hat{v}_{k_-}$ is the unit vector in the $\phi_{k_-}+\frac{\pi}{2}$ direction

θ_{j_-} is the polar angle at the previous boundary, $\sin^{-1}\sqrt{\frac{j}{M}}$

θ_{j_+} is the polar angle at the next boundary, $\sin^{-1}\sqrt{\frac{j+1}{M}}$

$\phi_{k\_}$ is the azimuthal angle at the previous boundary, $2\pi \frac{k}{N}$

$r_{j,k}$ is the intersection distance for cell (j,k)

This summation has been regrouped to use the *differences* in radiance between neighboring samples, summing over the boundaries rather than the cells themselves. This yields a much simpler formula.

3.3. Applying Gradients to Interpolation

If the irradiance were being interpolated on a mesh, it would be straightforward to apply the rotational and translational gradients in a bicubic interpolation. However, the meshless interpolation used by *Radiance* demands a slightly different approach. The method we have chosen uses the same weighted average and even the same weights as the original method, but with the irradiance values adjusted by their corresponding rotational and translational gradients. This in effect increases the order of the interpolation. The interpolation does not have a polynomial basis, however, so it is difficult to compare to more conventional forms.

The irradiance interpolation using the gradient vectors calculated by Equations (3) and (4) is as follows:

$$E(\vec{P}) = \frac{\sum_{\mathbf{S}} w_i(\vec{P}) \left[E_i \; + \; (\hat{n}_i \times \hat{n}) \cdot \vec{\nabla}_r E_i \; + \; (\vec{P} - \vec{P}_i) \cdot \vec{\nabla}_t E_i \right]}{\sum_{\mathbf{S}} w_i(\vec{P})} \tag{5}$$

where:

$$w_i(\vec{P}) = \frac{1}{\frac{||\vec{P} - \vec{P}_i||}{R_i} + \sqrt{1 - \hat{n} \cdot \hat{n}_i}} \; = \; \text{weight of sample } i$$

$\vec{P}$ is the test point position

$\hat{n}$ is the surface normal at the test point

E_i is the irradiance value at sample i

$\vec{P}_i$ is the position of sample i

$\hat{n}_i$ is the surface normal at sample i

R_i is the harmonic mean distance to objects visible from $\vec{P}_i$

S is the set of valid irradiance values, $\{ i : w_i(\vec{P}) > 1/a \}$

a is a user-selected accuracy goal

Note that our calculation of the irradiance gradient is not used to determine the spacing of samples. Such an approach would tend to bias the calculation, since areas that just happened to have small irradiance gradients would be sampled at low density, even though there could still be sudden changes in the irradiance value due to nearby surfaces. It is better to use the split sphere model to respond to the largest expected gradient in determining the sample density, so that we do not miss anything important. However, we may still want to use the more accurate formula (4) to increase the sample density where the split sphere model underestimates the gradient.

The interpretation of the irradiance gradient allows one to perform the calculation separately for each sample wavelength, or to combine into a spectral average. The latter was chosen for our implementation to save on storage costs, although this may be an unwarranted compromise in some contexts.

4. Conclusion

The irradiance gradient can be calculated from the same hemisphere samples used to compute the point irradiance. The extra computational cost is negligible compared to the cost of computing the samples, and the benefit is greatly improved interpolation accuracy. This may sound like we are getting something for nothing, but in fact we are only better utilizing the information already available from a hemisphere sampling.

Figure 5a shows a room with daylight entering through a doorway. This image was computed without irradiance gradients in a two-pass calculation using interpolated values. Figure 5b shows the same scene rendered using irradiance gradients to effectively increase the order of interpolation. Notice that the second image is smoother around the doorway, where the indirect component is most influential. Both images took roughly the same time to compute.

The gradient calculation as described in this paper is appropriate to any global illumination method where surface brightnesses are being sampled over a hemisphere, such as a gathering radiosity or Monte Carlo algorithm. Z-buffer methods such as the hemicube also yield the information necessary to compute gradients, so the approach is not limited to ray tracing algorithms.

It may also be possible to use hemisphere sampling information to speed convergence of a progressive radiosity or shooting Monte Carlo algorithm by noting the arrangement of visible surfaces and subdividing shooting patches where a shadow boundary is indicated. Since the final radiance of samples is not known in such a technique, subdivision would have to be based mainly on geometric considerations. Nevertheless, the information contained in a hemisphere sampling is considerable, and it seems wasteful to ignore it.

5. Acknowledgements

The first author was supported by the Swiss LUMEN Project, and the Assistant Secretary for Conservation and Renewable Energy, Office of Building Technologies, Buildings Equipment Division of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098. The second author was supported by a fellowship from the Delft University of Technology. Jean-Louis Scartezzini of the University of Geneva and Raphael Compagnon of EPFL in Switzerland provided invaluable support and assistance. Raphael and his students created the model shown in Figure 5. Anat Grynberg created the conference room model shown in Figure 3.

6. Software Availability

The *Radiance* software discussed in this paper is available from anonymous ftp at the following sites:

hobbes.lbl.gov	128.3.12.38	Berkeley, California
dasun2.epfl.ch	128.178.62.2	Lausanne, Switzerland

7. References

[Heckbert91a]
Paul Heckbert, *Simulating Global Illumination Using Adaptive Meshing*, PhD Thesis, Tech. Report UCB/CSD 91/636, Computer Science Division, University of California at Berkeley, June 1991.

[Heckbert91b]
Paul Heckbert and Jim Winget, ‘‘Finite Element Methods for Global Illumination,’’ Tech. Report UCB/CSD 91/643, Computer Science Division, University of California at Berkeley, July 1991.

[Kajiya86]
James T. Kajiya, ‘‘The Rendering Equation,’’ *Computer Graphics,* Vol. 20, No. 4, August 1986.

[Lischinski91]
Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg, *Improving Sampling and*

Reconstruction Techniques for Radiosity, Computer Science Dept., Cornell University, Tech. Report 91-1202, Aug. 1991.

[Max92]
Nelson Max and Michael Allison, ‘‘Linear Radiosity Approximation using Vertex-to-Vertex Form Factors,’’ *Graphics Gems III*, edited by David Kirk, Academic Press, 1992 (to appear).

[Ward88a]
Gregory Ward and Francis Rubinstein, ‘‘A New Technique for Computer Simulation of Illuminated Spaces,’’ *Journal of the Illuminating Engineering Society*, Vol. 17, No. 1, Winter 1988.

[Ward88b]
Gregory Ward, Francis Rubinstein, and Robert Clear, ‘‘A Ray Tracing Solution for Diffuse Interreflection,’’ *Computer Graphics*, Vol. 22, No. 4, August 1988.

[Ward90]
Gregory Ward, ‘‘Visualization,’’ *Lighting Design and Application*, Vol. 20, No. 6, June 1990.

[Ward91]
Gregory Ward, ‘‘Adaptive Shadow Testing for Ray Tracing,’’ *Second Eurographics Workshop on Rendering,* Barcelona, Spain, April 1991.

[Whitted80]
Turner Whitted, ‘‘An Improved Illumination Model for Shaded Display,’’ *Communications of the ACM,* Vol. 23, No. 6, June 1980, pp. 343-349.

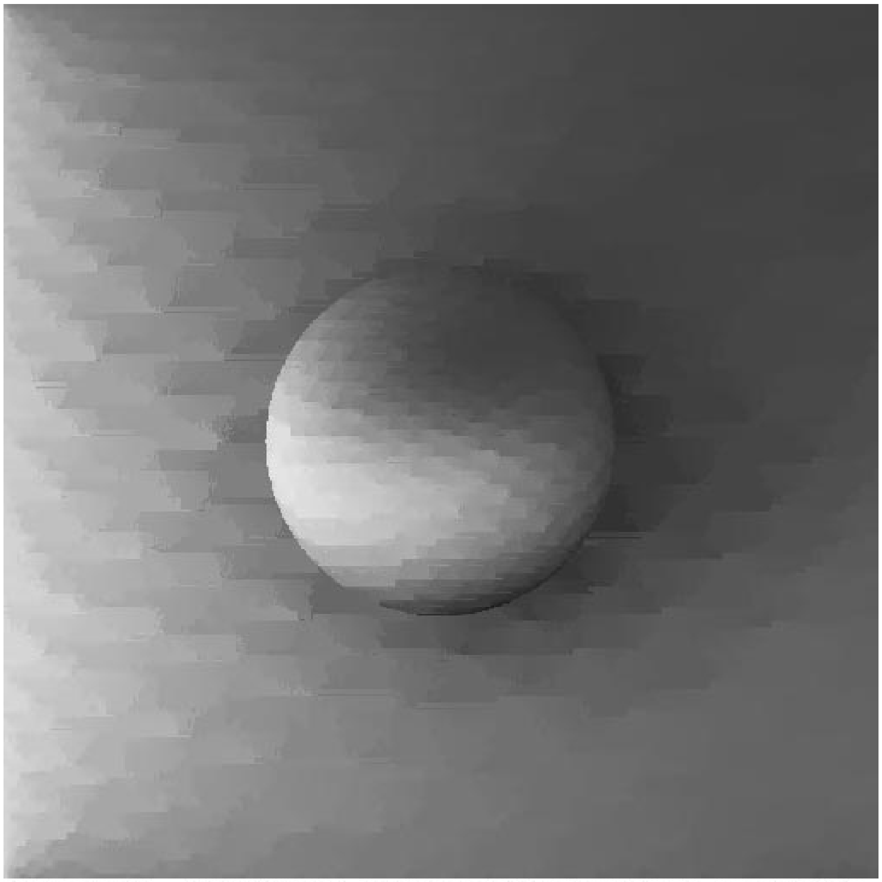


Figure 1a. Irradiance extrapolation without gradients.

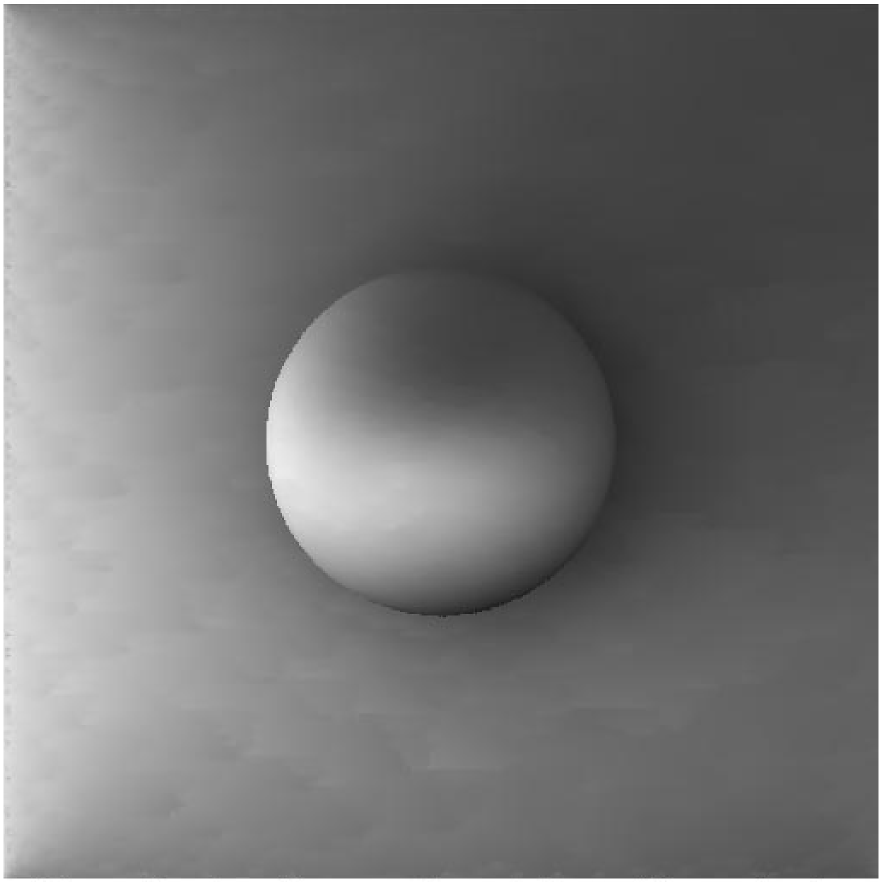
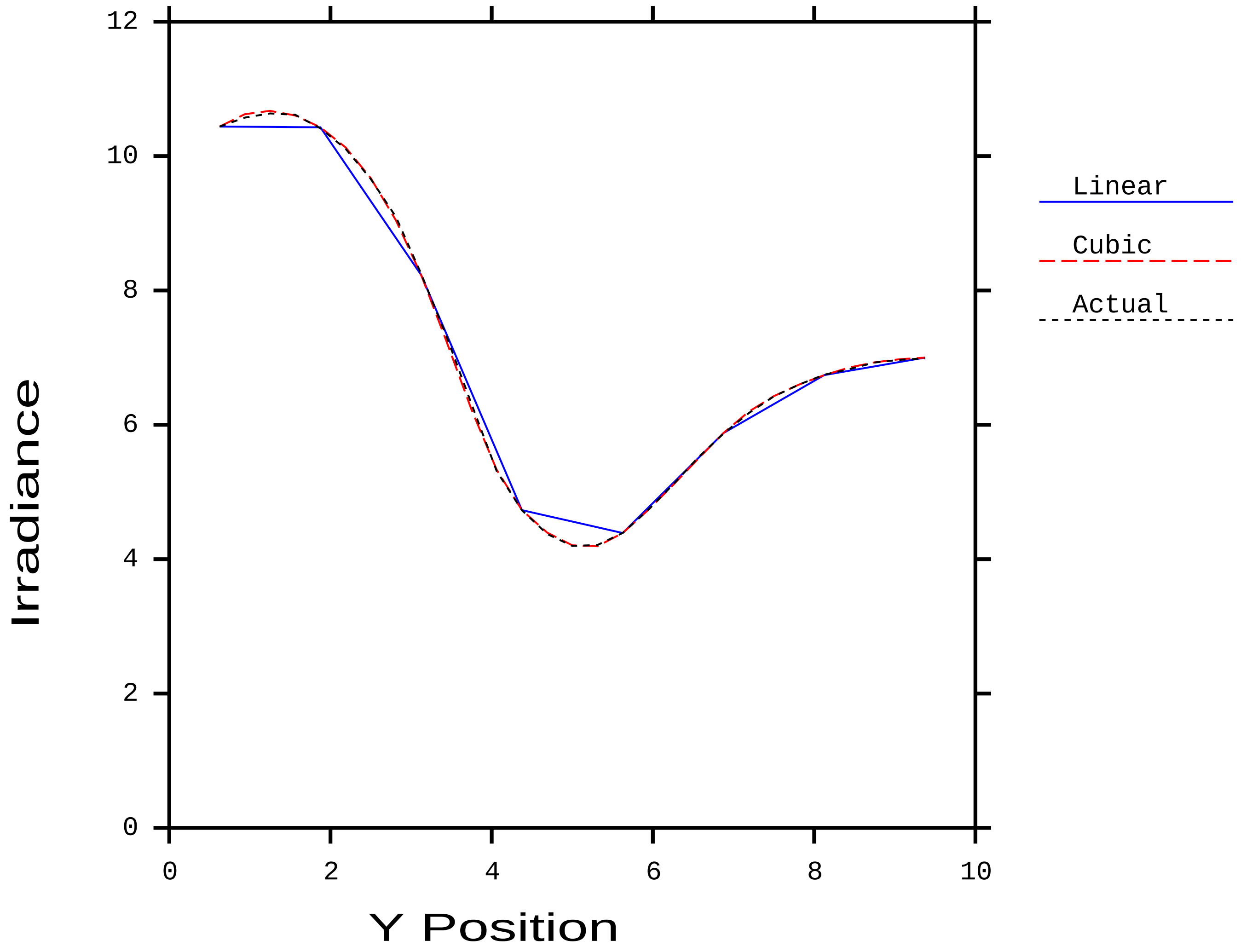


Figure 1b. Irradiance extrapolation with gradients.

Irradiance Interpolation

x=6.875



Irradiance Interpolation Error

x=6.875

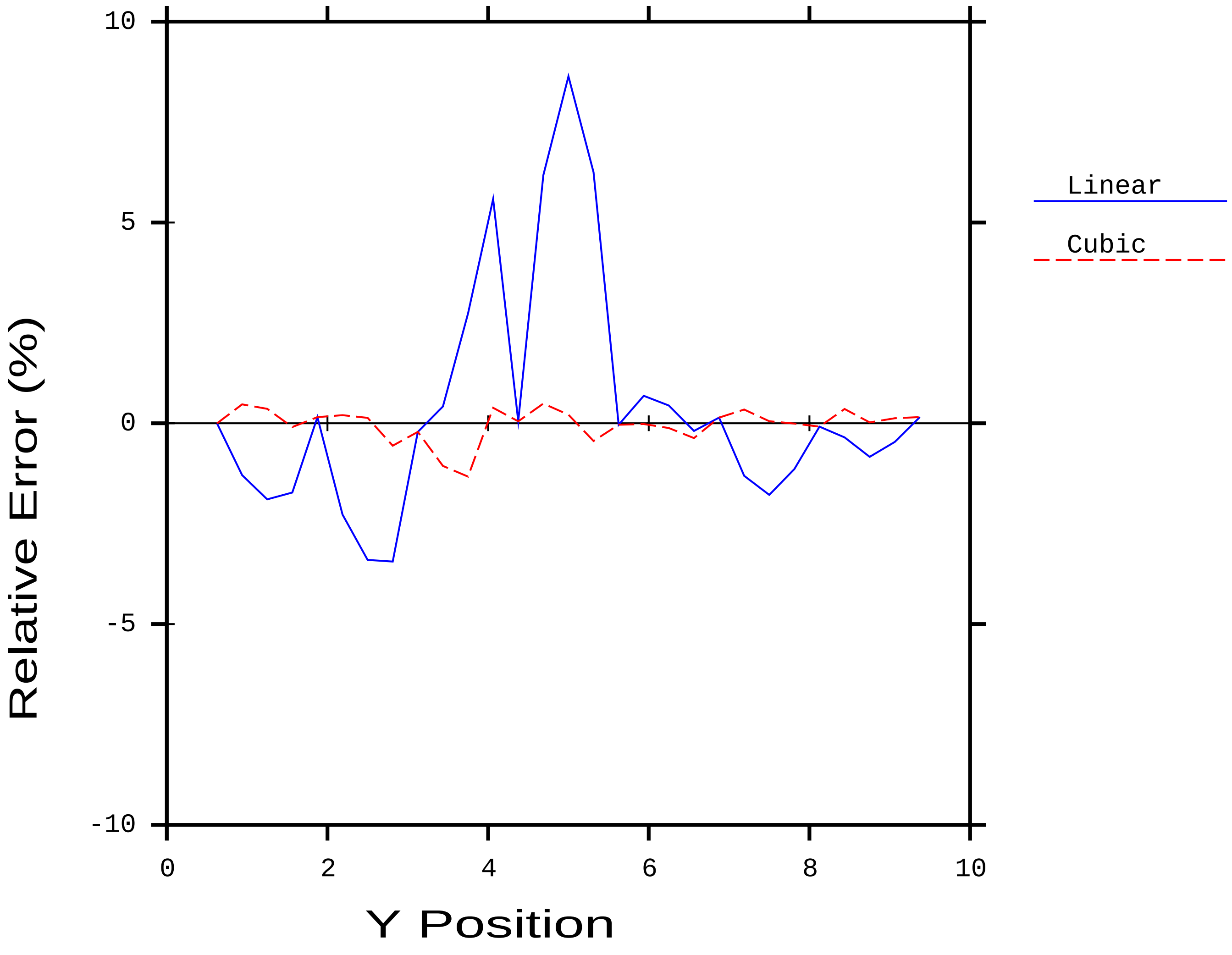


Figure 1c. Comparison between actual irradiance and interpolated values under diffuse sphere.

Figure 1d. Relative error due to interpolation under diffuse sphere.

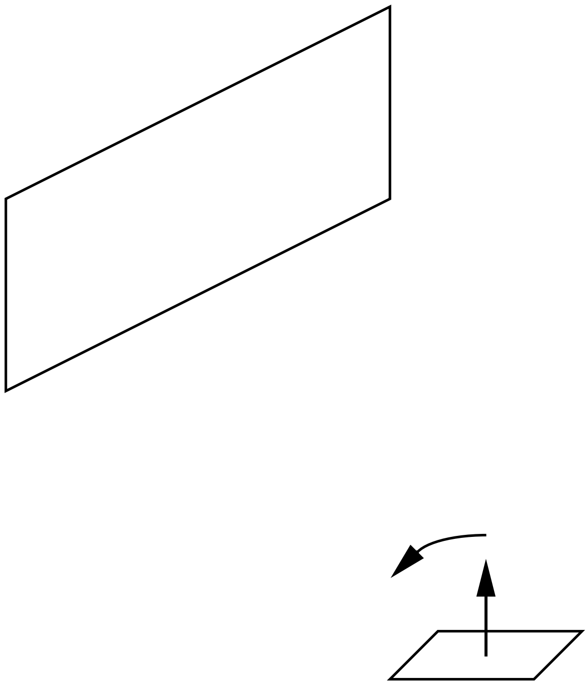


Figure 2a. As our point is rotated counter-clockwise, the surface's contribution increases.

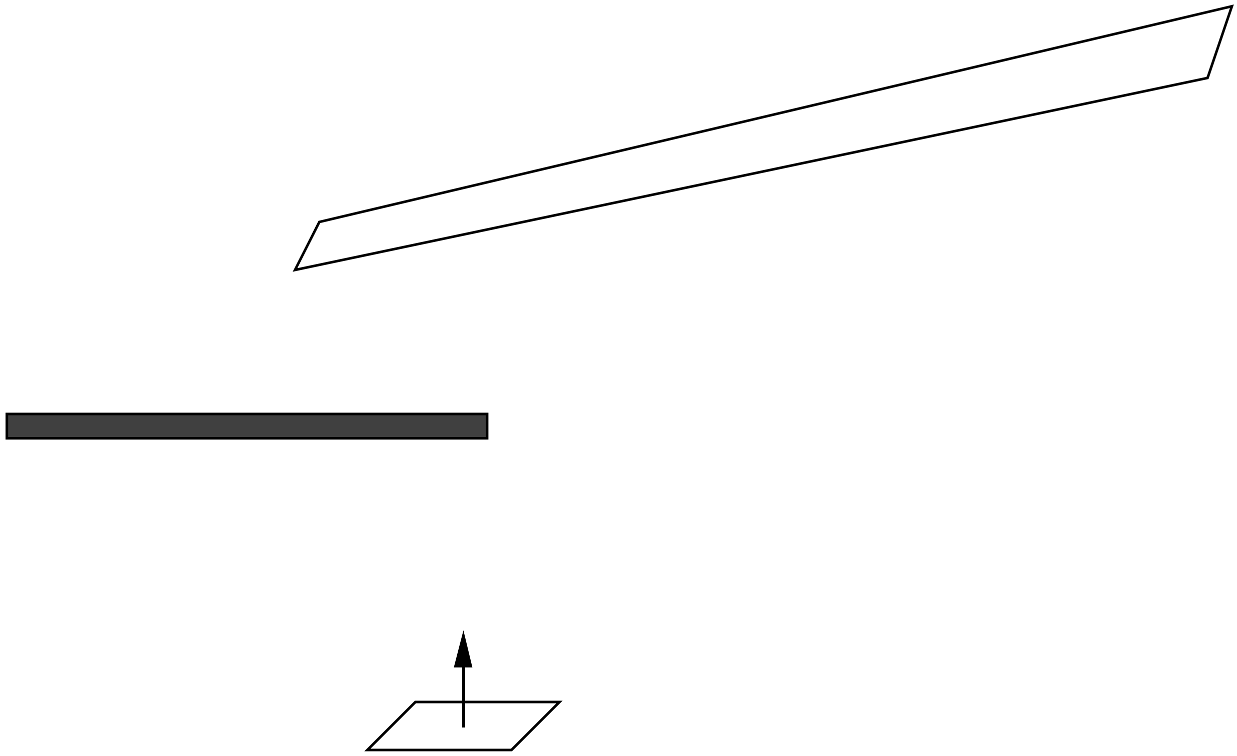


Figure 2b. Translational Gradient. As our point moves to the right, irradiance increases.

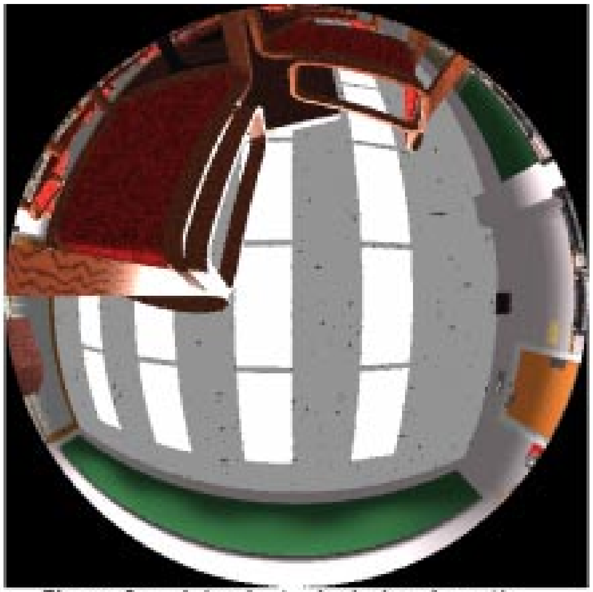


Figure 3a. A hemispherical view from the the floor of a conference room.

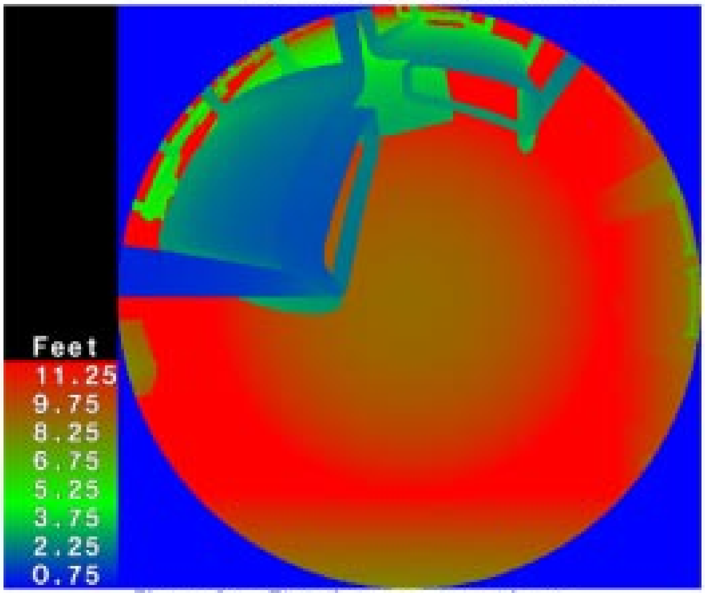


Figure 3b. The distances to surfaces visible in Figure 3a.

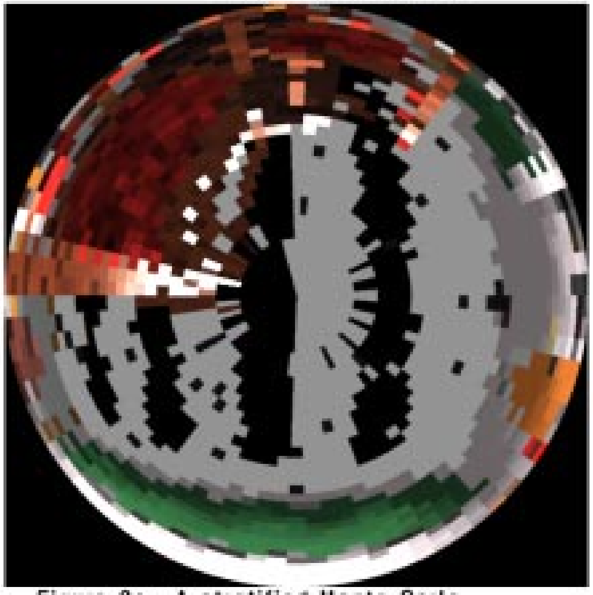


Figure 3c. A stratified Monte Carlo sampling of the same hemisphere.

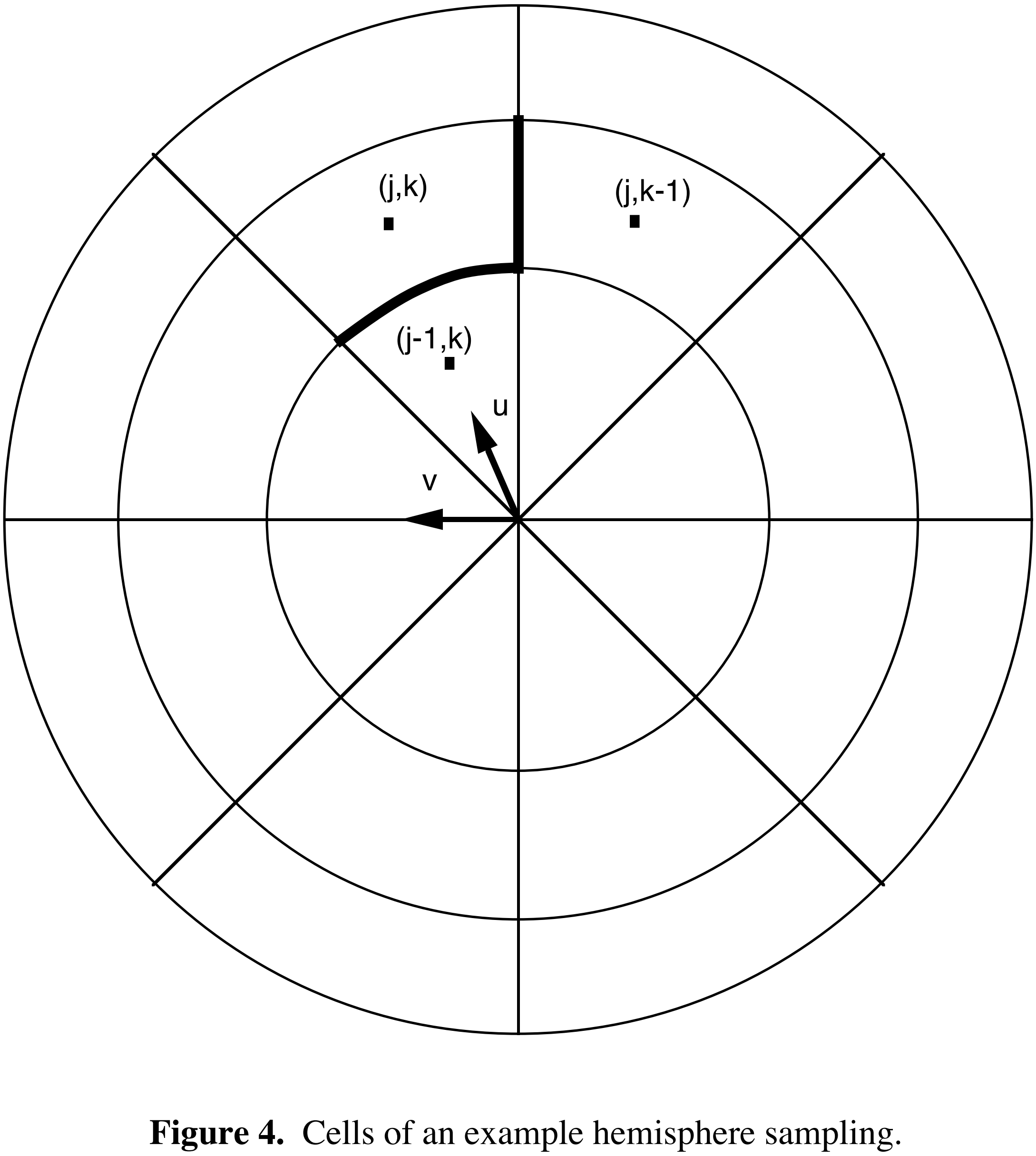


Figure 4. Cells of an example hemisphere sampling.

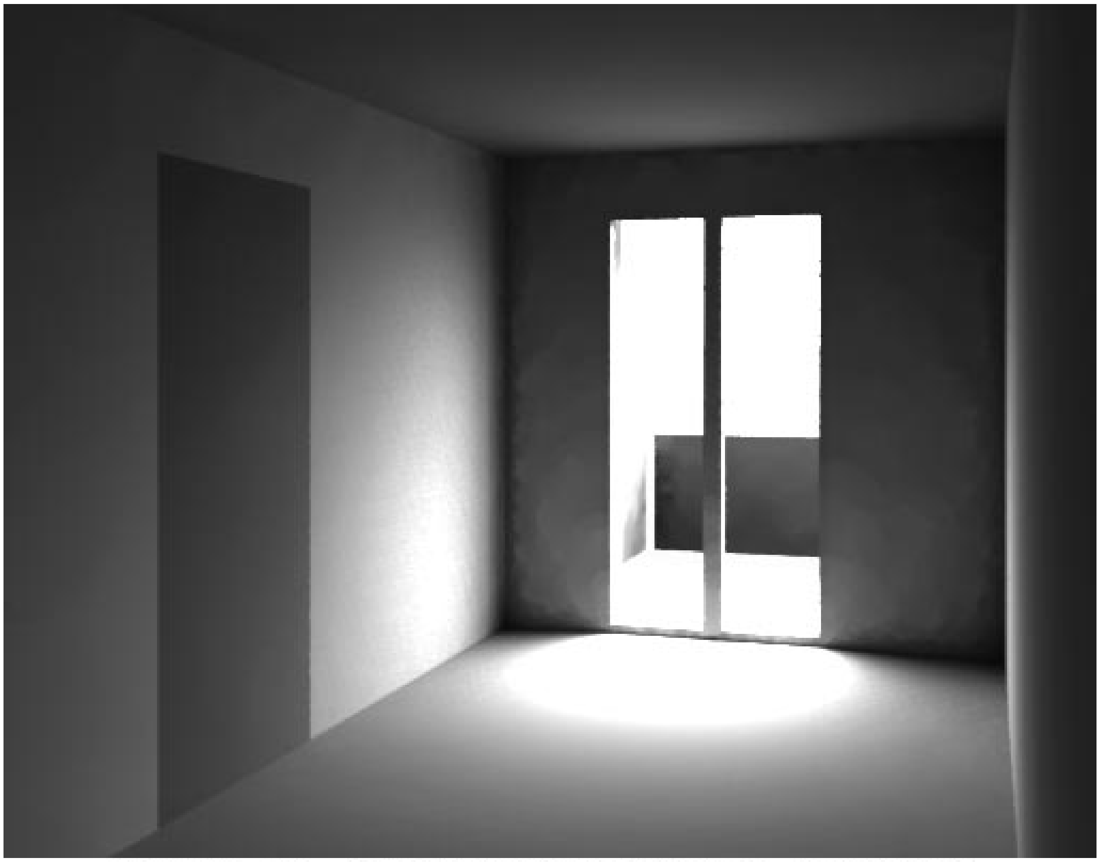


Figure 5a. Interpolation without gradients.

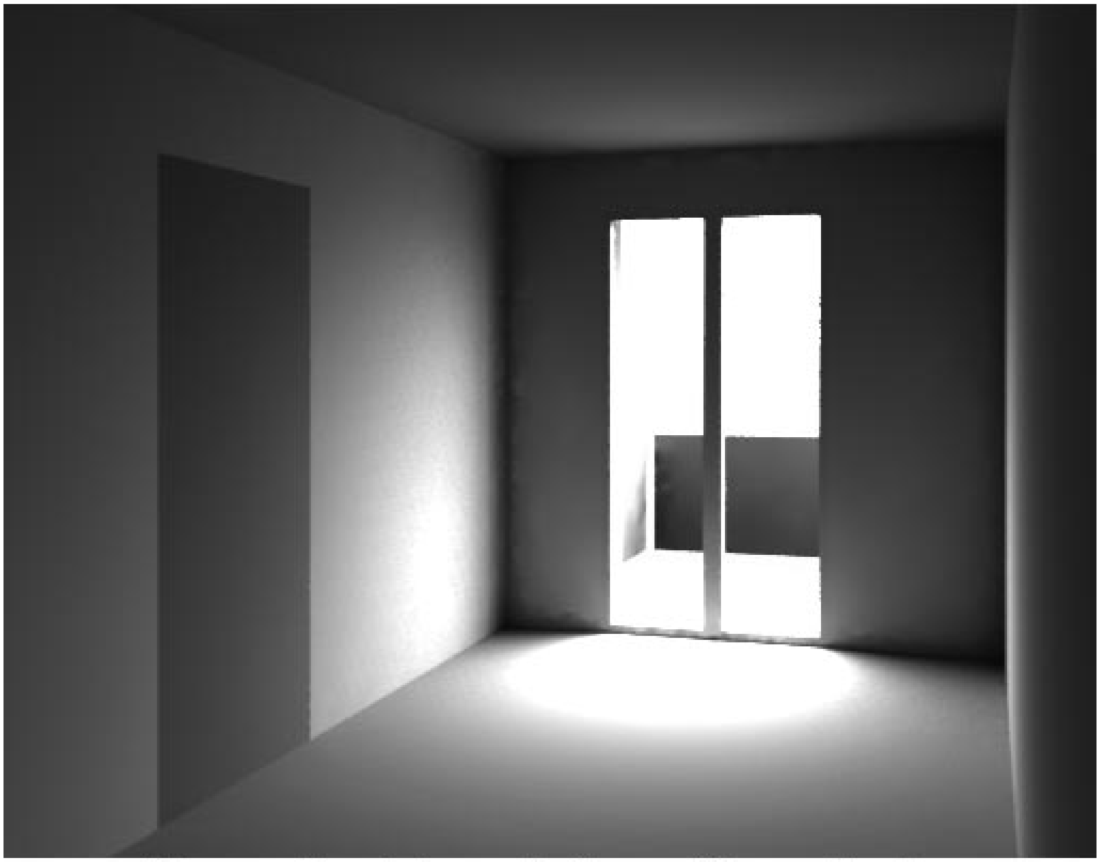


Figure 5b. Interpolation with gradients.