A Perceptually Based Physical Error Metric for Realistic Image Synthesis

Mahesh Ramasubramanian Sumanta N. Pattanaik Donald P. Greenberg

Program of Computer Graphics *
Cornell University

Abstract

We introduce a new concept for accelerating realistic image synthesis algorithms. At the core of this procedure is a novel *physical* error metric that correctly predicts the *perceptual* threshold for detecting artifacts in scene features. Built into this metric is a computational model of the human visual system's loss of sensitivity at high background illumination levels, high spatial frequencies, and high contrast levels (visual masking). An important feature of our model is that it handles the luminance-dependent processing and spatially-dependent processing independently. This allows us to *precompute* the expensive spatially-dependent component, making our model extremely efficient.

We illustrate the utility of our procedure with global illumination algorithms used for realistic image synthesis. The expense of global illumination computations is many orders of magnitude higher than the expense of direct illumination computations and can greatly benefit by applying our perceptually based technique. Results show our method preserves visual quality while achieving significant computational gains in areas of images with high frequency texture patterns, geometric details, and lighting variations.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism.

Keywords: Realistic Image Synthesis, Global Illumination, Adaptive Sampling, Perception, Visual Masking, Error Metric, Visual Threshold.

1 INTRODUCTION

Realistic Image Synthesis is an important area of research in computer graphics and has been widely studied in the last few decades. It aims at producing synthetic images that are visually indistinguishable from the actual scene it seeks to reproduce. Earlier work attempted to simulate this realism by modeling only the direct illumination of the scene by light sources, but this approach failed to capture many important visual cues: indirect illumination, soft shadows, color bleeding, and caustics. *Global Illumination* algorithms developed in recent years have been able to accurately render these effects in addition to the direct illumination [12]. These algorithms physically simulate accurate light reflection and the complex light interactions between surfaces in the environment. Unfortunately, these simulations make global illumination algorithms computationally very expensive, with execution times many orders of magnitude slower than simple direct illumination algorithms. The research described in this paper is focused on improving the efficiency of computing global illumination.

Perceptually based techniques promise dramatic performance gains. As the final result of a global illumination algorithm is an image interpreted by the human eye, it is sufficient to aim for perceptual accuracy. Due to the limitations of the human visual system, the degree of effort required to attain acceptable perceptual accuracy varies over the image. Perceptually based rendering algorithms take advantage of this phenomenon and attempt to expend only "just-necessary" effort over the image to meet this perceptual accuracy criteria. To accomplish this a computational model of the visual system [7, 17, 21] is necessary. Such vision models typically predict the visual sensitivity variations with background illumination levels, spatial frequency, and scene contrast features. These predictions can then be used to make image quality judgements. The vision model is applied as the image is being generated and additional efforts are then expended only in areas with detectable visual artifacts, thereby reducing the overall computation time.

Vision models are normally expensive to evaluate as some of their components perform multiscale spatial processing. Moreover, in a progressive image synthesis algorithm, image quality judgements are required at every iteration and the vision model is invoked many times. Hence, although perceptually based algorithms show significant potential for acceleration, they also involve considerable amount of additional overhead due to the repeated vision model evaluation.

In this paper we introduce a new framework for perceptually based rendering that drastically reduces the overhead of incorporating a perceptual basis. *We develop a threshold model which defines a physical error metric that correctly predicts the perceptual threshold for detecting artifacts in scene features. This allows image quality judgements to be made in the physical domain using the perceptually based physical error metric.* This framework opens new avenues for using a perceptual basis in speeding up global illumination computations.

A key ingredient for realism in synthesized images is *complexity* [5]. Realistic images tend to be richly detailed. This richness and detail take many forms, such as texture patterns, geometric details, and lighting variations, and contain high spatial frequency content. But since the visual system's threshold and suprathreshold sensitivity to high spatial frequency content is poor [10], more artifacts can be tolerated in these areas. Our framework takes advantage of these phenomena to reduce global illumination computation time. The texture patterns and geometric detail are inherent in the scene specification and can be captured during direct illumination computation. Direct illumination algorithms also capture most of the high spatial frequency of the image due to lighting variations. From this solution, we "precompute" the spatially-dependent component of our threshold model before the indirect illumination computation

* 580 Rhodes Hall, Ithaca, NY 14853, USA.
WWW: http://www.graphics.cornell.edu/
E-mail: {mahesh,sumant,dpg}@graphics.cornell.edu

(a) reference global illumination solution (b) sample density pattern (c) adaptive global illumination solution

Figure 1: An illustration of our framework applied to an adaptive global illumination algorithm. Image (b) is the sample density pattern used for the indirect illumination computation (darker areas indicate fewer samples), and image (c) is the final solution resulting from the adaptive global illumination computation. For comparison, image(a) is a reference global illumination solution generated using uniformly high sampling density. While it may seem counterintuitive, areas with higher spatial frequency content require less computational effort.

stage and thus avoid recomputing this complex component over subsequent iterations. As computing this component of the threshold model involves the expensive multiscale spatial processing, we benefit enormously by this precomputation. Figure 1 illustrates the sampling density pattern when our threshold model is used to direct the sampling in an adaptive global illumination algorithm. Notice that the sampling density reflects the psychophysical observation that our visual system is less sensitive in areas with higher spatial frequency content.

The remainder of this paper is organized as follows: Section 2 reviews the previous approaches to perceptually based rendering. In Section 3 we introduce our new framework. At the core of this framework is the *threshold model* which defines a perceptually based physical error metric. The perceptual basis and implementation of this model are described in Section 4. In Section 5 we illustrate the utility of our framework by applying it to an adaptive global illumination algorithm. We conclude the paper with a summary of our framework and the future directions for this research.

2 PREVIOUS WORK

Many researchers have attempted to develop perceptually based rendering algorithms. Bolin and Meyer [3] present an excellent survey of the early algorithms. Our discussion concentrates on the algorithms most relevant to our work.

All of these algorithms attempt to exploit the limitations of the human visual system to speed up rendering computations without sacrificing visual quality. They differ in the extent to which they model the visual system and the way they apply this vision model to the rendering algorithms. Mitchell [19] defined an adaptive sampling strategy for his ray tracing algorithm by taking advantage of the poor sensitivity of the visual system to high spatial frequency, to absolute physical error (threshold sensitivity), and to the high and low wavelength content of the scene. Meyer and Liu [18] took into account the human visual system's poor color spatial acuity in developing an adaptive image synthesis algorithm. Bolin and Meyer [2] developed a frequency based ray tracer using a simple vision model which incorporated the visual system's spatial processing behavior and sensitivity change as a function of luminance. Myszkowski [20] and Bolin and Meyer [3] applied sophisticated vision models to guide Monte Carlo based ray tracing algorithms. The models they used incorporated the visual system's threshold sensitivity, spatial frequency sensitivity, and contrast masking behavior. Gibson and Hubbold [11] and Hedley *et al.* [14] have applied the threshold sensitivity of the visual system to speed up radiosity computations.

Of all the approaches described above, the recent work by Myszkowski [20] and Bolin and Meyer [3] needs special mention for two reasons. First, they used sophisticated vision models which incorporate the most recent advances in the understanding of the human visual system [7, 17]. Thus in principle their algorithms can take maximum advantage of the limitations of the visual system. Second, they introduced a perceptual error metric into their rendering algorithms. Thus their algorithms were able to adaptively allocate additional computational effort to areas where errors remained above perceivable thresholds and stop computation elsewhere.

Both approaches were conceptually similar and used a *visual difference predictor* [7, 17] to define a perceptual error metric. A visual difference predictor takes a pair of images and transforms them to multidimensional visual representations by applying a vision model. It then computes the "distance" between this pair of visual representations in a multidimensional space, producing the form of a local visual difference map. This is compared against a perceptual threshold value to ascertain the "perceivability" of the difference. Figure 2 illustrates the functioning of such a predictor.

When one of the two input images to the predictor is the final converged image and the other is the image at any intermediate stage of computation, then the visual difference map becomes an error estimate for that stage and the visual difference predictor functions as an estimator of the perceived error. Myszkowski, and Bolin and Meyer used such an estimator during their image computation and used this information to direct subsequent computational effort. Unfortunately, during the image synthesis process one does not have the luxury of accessing the final converged image at an intermediate stage. Myszkowski assumed that two intermediate images obtained at consecutive time steps of computation could be used as input to the visual difference predictor to get a functional error estimate. Bolin and Meyer computed the upper and lower bound images from the computation results at intermediate stages and applied the predictor to get the error estimate for that stage. Their approach thus estimates the error bounds.

These algorithms achieve the ability to focus computational efforts in areas with perceivable errors, but only at considerable cost. They use the perceptual error metric at every stage of image computation which requires repeated evaluation of the embedded vision model. The vision model is very expensive to compute as some of its components require multiscale spatial processing, and this over-

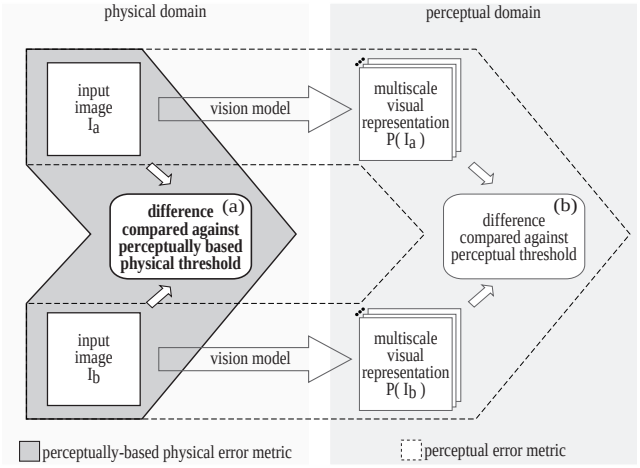


Figure 2: Conceptual difference between a perceptually based physical error metric and a perceptual error metric. A perceptual metric operates in the perceptual domain. Images to be compared are first transformed into their multi-scale visual representation and the perceptual metric is applied to the difference of the visual representation (b). In contrast, a perceptually based physical error metric operates in the physical domain. The metric is applied to the physical luminance difference between the images (a). Our metric is non-uniform over the physical space of the image.

head offsets some of the advantages gained by using the perceptual error metric to speed up the rendering algorithm.

3 NEW FRAMEWORK

We propose a new framework for perceptually based rendering which drastically reduces the overhead of introducing the perceptual basis while still gaining maximum advantage from the limitations of the human visual system. To achieve this we first develop a *threshold model* which incorporates the human visual system's threshold sensitivity, spatial frequency sensitivity, and contrast sensitivity (masking) to predict the perceptual threshold for detecting artifacts in scene features. The threshold model T operates on an image I to generate a *threshold map* $T(I)$ which is an index to the maximum physical luminance error that can be tolerated at any location on the image, while preserving visual quality.

We call our framework a *perceptually based physical error metric* to emphasize the fact that once the threshold map is computed, the pair of images can be compared directly in the physical luminance domain, while still accounting for the limitations of the visual system. Figure 2 illustrates the conceptual difference between the perceptual error metric and our perceptually based physical error metric. A major advantage of our approach is that we can precompute the expensive components of our threshold model at an earlier rendering computation stage and thus avoiding the recomputation that has hindered earlier approaches.

4 THRESHOLD MODEL

In this section we develop a model for computing a threshold map for any given image. The threshold map predicts the maximum luminance error that can be tolerated at every location over the image. This model makes use of three main characteristics of the visual system, namely: threshold sensitivity, contrast sensitivity, and contrast masking. An important feature of this model is that it handles the luminance-dependent processing and the spatially-dependent processing independently. The luminance-dependent processing computes a starting threshold map ΔL_{tvi} for the luminance distribution using the *threshold-vs-intensity* (TVI) function. The spatially-dependent processing computes a map containing elevation factors $F_{spatial}$ for the spatial pattern using the *contrast sensitivity function* (CSF) and *masking function*. From these two we derive the final threshold map $\Delta L_T(x, y)$ as:

$$\boxed{\Delta L_T(x,y) = \Delta L_{tvi}(x,y) \times F_{spatial}(x,y)} \quad (1)$$

The separate handling of luminance distributions and spatial patterns allows us to precompute the expensive spatially-dependent component of the threshold model, making our model extremely efficient when used in perceptually-based rendering algorithms. Figure 3 provides an overview of the model.

4.1 Model Description

Threshold Sensitivity The *threshold-vs-intensity* (TVI) function describes the threshold sensitivity of the visual system as a function of background luminance. The threshold, as defined by this function, is the minimum amount of incremental luminance, ΔL, by which a test spot should differ from a uniform background of luminance L to be detectable. Figure 3(b) plots thresholds computed from this function at different background luminance values. The two curves in the figure represent the thresholds of the rod and cone systems. The linear part of each curve follows Weber's law, which means that the threshold increases linearly with luminance. The threshold from this TVI function, ΔL_{tvi}, provides the luminance-dependent starting values from which we build our final threshold map.

Contrast Sensitivity The threshold given by the TVI function predicts sensitivity in uniform visual fields. However, the luminance distribution in any complex image is far from uniform. The *contrast sensitivity function* (CSF) [21] provides us with a better understanding of the visual sensitivity in such situations. The sensitivity is highest at frequencies in the range of 2 to 4 cycles per degree (cpd) of visual angle and drops off significantly at higher and lower spatial frequencies. The peak sensitivity is normally predicted by the TVI function. What the TVI function does not predict is the loss of sensitivity as the frequency deviates from this range. The relation between contrast sensitivity, S_{csf}, and visual threshold, ΔL_{csf}, at any frequency f is derived as:

$$S_{csf}(f) = \frac{1}{\Delta C_{csf}(f)} = \frac{1}{(\Delta L_{csf}(f)/L)} \quad (2)$$

where ΔC_{csf} is the threshold contrast, and L is the background luminance. From this we get:

$$\Delta L_{csf}(f) = \frac{L}{S_{csf}(f)} \quad (3)$$

Thus, the CSF function gives us the threshold $\Delta L_{csf}(f)$ for detecting a sinusoidal grating pattern of any given frequency from a background luminance L. The threshold predicted by CSF for a grating is conceptually different from the threshold from TVI function. The difference lies in the fact that the threshold itself is a pattern of the same frequency with a peak value of $\Delta L_{csf}(f)$ and defined around a mean value of zero[1].

[1]This difference derives from the fact that in psychophysics two types of contrast definitions are used: Weber contrast is used in experiments with aperiodic signals (spot on background tests), which is $\frac{\Delta L}{L_{background}}$ and Michaelson contrast is used in experiments with periodic signals (tests with sinusoidal gratings) which is $\frac{L_{max}-L_{min}}{L_{max}+L_{min}} = \frac{\Delta L_{peak}}{L_{mean}} \approx \frac{\Delta L_{peak}}{L_{background}}$.

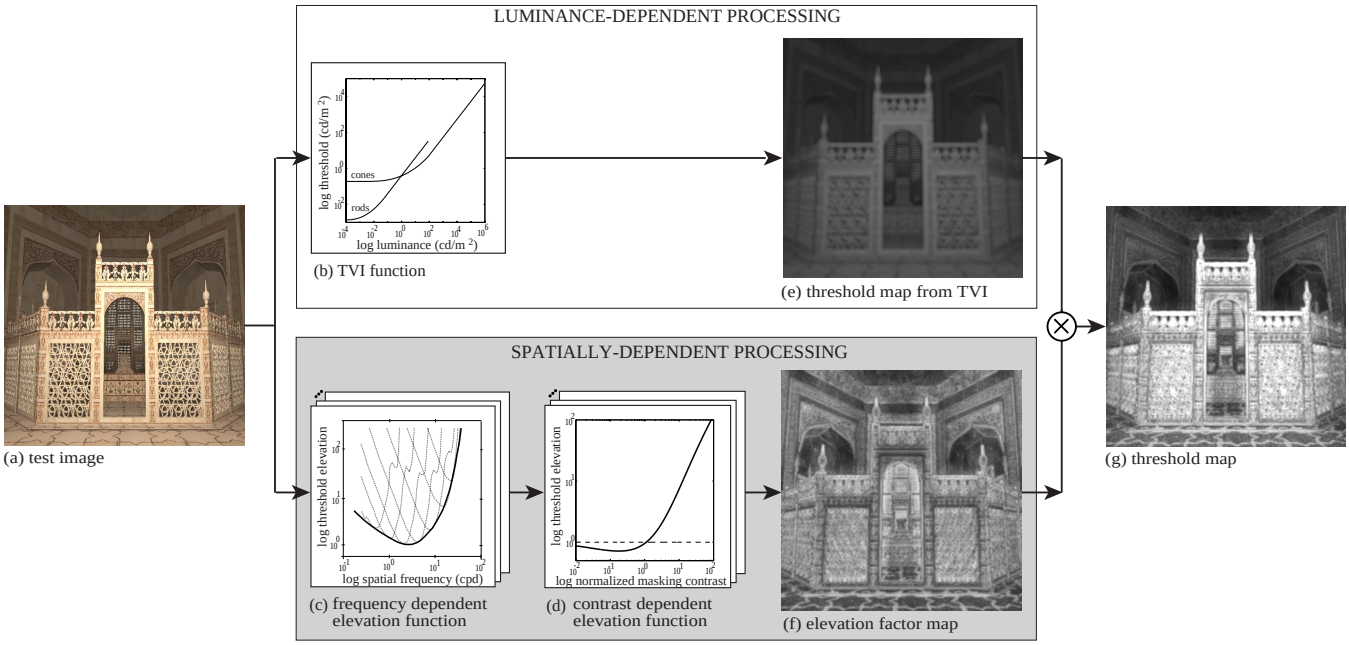


Figure 3: Flow chart outlining the computational steps of our threshold model.

As the sensitivity decreases for frequencies outside the range of 2 to 4 cpd, this ΔL_{csf} increases. We write this increase in threshold for any frequency f, as compared to the threshold at peak of the CSF function as:

$$F_{csf}(f) = \frac{\Delta L_{csf}(f)}{\Delta L_{csf}^{peak}} \tag{4}$$

We refer to this relative increase as the *threshold elevation factor* due to contrast sensitivity, $F_{csf}(f)$. Figure 3(c) plots this elevation factor as a solid line. The peak contrast sensitivity is normally predicted by the TVI function i.e. $\Delta L_{csf}^{peak} = \Delta L_{tvi}$. Thus from the TVI and F_{csf} functions we can compute the threshold for patterns at any frequency f as:

$$\Delta L_{csf}(f) = \Delta L_{tvi} \times F_{csf}(f) \tag{5}$$

where ΔL_{tvi} is the threshold for the background luminance L of the frequency pattern.

Multi-scale Spatial Processing The CSF behavior of the visual system is believed to be the result of the spatial processing of the frequency patterns by multiple bandpass mechanisms. Each mechanism processes only a small band of spatial frequencies from the range over which the visual system is sensitive. The inverse of the response curves of these bandpass mechanisms normalized with respect to the peak CSF value are shown by the curves drawn in broken lines in Figure 3(c). As can be inferred from the figure, the peak sensitivity of each mechanism is equal to the CSF sensitivity at their peak frequencies.

Most of the frequencies in the range over which the visual system is sensitive are processed by multiple bandpass mechanisms. We can describe the contribution of each mechanism to the threshold elevation factor for a grating of frequency f as:

$$F_{csf}^{i}(f) = F_{csf}(f_{peak}^{i}) \times fraction^{i}(f) \tag{6}$$

$$fraction^{i}(f) = \frac{C^{i}(f)}{\sum_i C^{i}(f)} \tag{7}$$

where f_{peak}^{i} is the peak frequency of the i^{th} bandpass mechanism, and C^{i} is the band-limited contrast of the grating pattern at the i^{th} bandpass mechanism.

The elevation factor for the grating of frequency f due to all the bands is then given by:

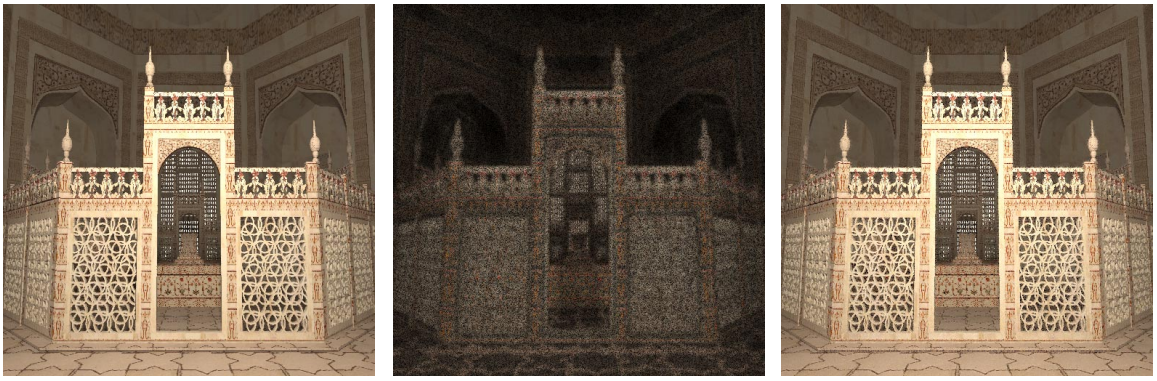
$$\begin{aligned} F_{csf}(f) &= \sum_i F_{csf}^{i}(f) \\ &= \sum_i \left(F_{csf}(f_{peak}^{i}) \times fraction^{i}(f) \right) \end{aligned} \tag{8}$$

Similar summation techniques are used to compute the distance in multi-dimensional perceptual space [7, 17]. Equation 4 and Equation 8 are two different representations of the elevation function for a sinusoidal grating of frequency f. Equation 8 is more useful for deriving elevation from complex patterns. Any complex pattern can be represented as a sum of sinusoidal grating patterns of various wavelength, amplitude, orientation and phase. We can use the same summation technique given in the above equation to compute the elevation factor map for complex patterns. However, to account for the complexity of the patterns we redefine Equation 7 as:

$$fraction^{i}(x,y) = \frac{C^{i}(x,y)}{\sum_i C^{i}(x,y)} \tag{9}$$

where $C^{i}(x,y)$ is the band-limited Weber contrast of the complex pattern at the i^{th} bandpass mechanism at every point (x,y) of the pattern. (Computation of this band-limited Weber contrast is described in the next section.) Consequently, the elevation factor in Equation 8 for complex patterns becomes an elevation factor map $F_{csf}(x,y)$ which is given by:

$$\begin{aligned} F_{csf}(x,y) &= \sum_i (F_{csf}^{i}(x,y)) \\ &= \sum_i F_{csf}(f_{peak}^{i}) \times fraction^{i}(x,y) \end{aligned} \tag{10}$$



(a) test image (b) noise map (c) test image + noise map

Figure 4: Testing the threshold model. Our threshold model computes a threshold map, shown in Figure 3(g) for the test image shown in (a). The threshold map is used to create a noise map. The absolute luminance value at every pixel in the noise map is below the threshold value given for that pixel in the threshold map. Image (b) shows the "absolute values" of this noise map. Image (c) is obtained by adding this noise to the test image. This image, though now containing noise, is visually indistinguishable from the original test image.

Contrast Masking The multiple bandpass mechanisms of the visual system are known to have non-linear response to pattern contrast. This compressive non-linearity results in further elevation of threshold with increases in the contrast of the pattern. Such behavior of the visual system is known as visual masking [10]. The elevation of threshold as a function of contrast is shown in Figure 3(d). We combine this elevation due to masking with the elevation due to CSF to compute a cumulative elevation factor map as:

$$F_{spatial}(x,y) = \sum_i \big(F_{csf}(f^i_{peak}) \times F^i_{masking}(x,y) \times fraction^i(x,y)\big) \quad (11)$$

where $F_{csf}(f^i_{peak})$ is the elevation factor due to contrast sensitivity, and $F^i_{masking}(x,y)$ is the elevation factor due to masking which is computed for the band-limited contrast at location (x,y) for the i^{th} band.

From the elevation factor derived in Equation 11 and the ΔL_{tvi} derived from the TVI function, our model computes a threshold map for any complex image patterns as given by Equation 1.

4.2 Implementation

In this section we describe the specific computational procedures that were used to implement each of the components of the model. Input to the model is a luminance image and output is the threshold map containing the threshold luminance values in cd/m^2. We use the scene luminance image as input to the threshold model with the assumption that whatever tone reproduction operator is used to display the final image will preserve its appearance [22]. To find the threshold map we need to evaluate Equation 1 over the image.

The first step of the model is to find the luminance-dependent threshold (ΔL_{tvi}) from the TVI function. We employ Ward's [16] piecewise approximation of the TVI curves given by Ferwerda *et al.* [9]. Computation of $\Delta L_{tvi}(x,y)$ at a pixel using this function requires the adaptation luminance at that pixel. Following the procedure adopted by Ward *et al.* [16] we computed the adaptation luminance by averaging the luminance over a 1° diameter solid angle centered around the pixel.

The next step is to evaluate the cumulative elevation factor $F_{spatial}$ given in Equation 11. The terms in this equation require spatial decomposition of the image to "band-limited contrast responses". We use Lubin's approach [17] for this spatial decomposition. First, the image is decomposed into a Laplacian pyramid (Burt and Adelson [4]), resulting in six bandpass levels with peak frequencies at 1, 2, 4, 8, 16, and 32 cycles/degree (cpd). Then, a *contrast pyramid* is created by dividing the Laplacian value at each point in each level by the corresponding point upsampled from the Gaussian pyramid level two levels down in resolution. The resulting contrast measure in the bands is equivalent to the *band-limited Weber contrast* [17] referred to in Equation 9.

The first term in Equation 11 is the band-limited peak elevation factor F_{csf}. This factor is derived from Barten's CSF formula [1]:

$$S_{csf}(f,L) = af\exp(-bf)\sqrt{1+0.06\exp(bf)} \quad (12)$$

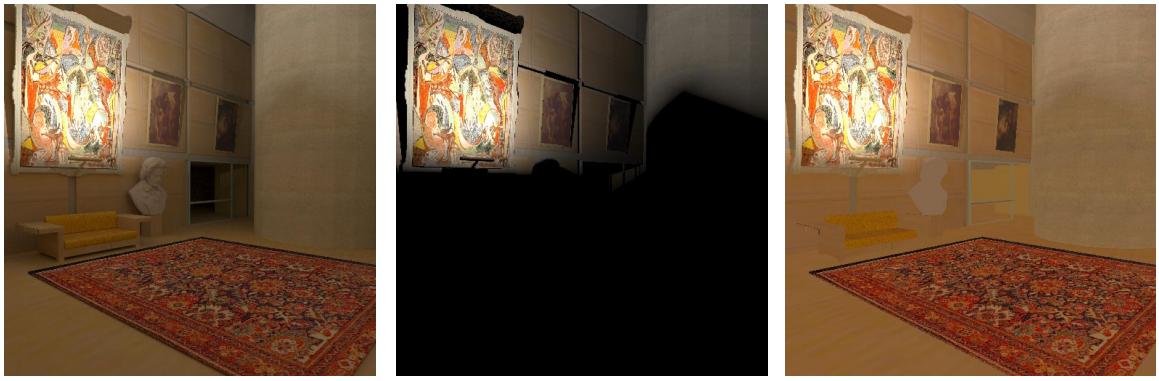
$$\begin{aligned} where \quad S_{csf} &= \text{contrast sensitivity} \\ a &= 440(1+0.7/L)^{-0.2} \\ b &= 0.3(1+100/L)^{0.15} \\ L &= \text{display luminance in } cd/m^2 \\ f &= \text{spatial frequency in cycles per degree (cpd)} \end{aligned}$$

We use the normalized CSF curve at 100 cd/m^2. Measurements by van Nes [25] show that the CSF is relatively independent of luminance level for levels above 100 cd/m^2, so the shape of the CSF curve at 100 cd/m^2 is a good match for higher luminance levels. At lower levels of illumination there is a proportionate decrease in sensitivity. However, the relative falloff in sensitivity at low spatial frequencies, as normally observed in a CSF curve, reduces with lowering of illumination level. To avoid any overestimation of threshold at lower frequencies we set the normalized CSF sensitivity factor below 4 cpd to be one. The reciprocal of the normalized CSF sensitivity values gives us the threshold elevation factors at various frequencies. The elevation factors at the discrete frequencies from 1 through 32 cpd are:

f^i_{peak} (cpd)	1	2	4	8	16	32
$F_{csf}(f^i_{peak})$	1.00	1.00	1.02	1.57	4.20	31.32

(13)

Next we need to evaluate the elevation factor due to masking in the bands, $F^i_{masking}(x,y)$. This elevation factor is determined using a masking function. These functions are usually given as compressive transducers [17] and can be converted to a threshold elevation function using numerical inversion methods. In the current



(a) test scene (b) direct illumination solution (c) direct illumination solution + ambient term

Figure 5: The direct illumination solution plus an approximate ambient term capture most of the high spatial frequency and contrast content in the scene.

implementation we have used the simpler analytic function given by Daly [7]:

$$F_{masking}(C_n) = (1 + (0.0153(392.498 \cdot C_n)^{0.7})^4)^{1/4} \quad (14)$$

$$where \qquad F_{masking} = \text{threshold elevation factor due to masking}$$
$$C_n = \text{normalized masking contrast}$$

Before using this function the contrasts in the bands are first normalized by the CSF function evaluated at luminance values from the low pass Gaussian pyramid.

Finally, the term $fraction^i(x, y)$ is evaluated using Equation 9. In this equation, $C^i(x, y)$ is the band-limited Weber contrast at point (x, y) in the i^{th} band.

The map in each band is spatially pooled by a disc-shaped kernel of diameter 5 [17] before applying Equation 11. This is to account for the influence of the number of cycles in the various frequencies present in the image on the elevation functions, as suggested by Lubin [17].

Figure 4 shows the threshold map for a test image and verifies our claim that the threshold map is an index to the maximum physical luminance error that can be tolerated at any location on the image.

5 APPLICATION TO GLOBAL ILLUMINATION ALGORITHMS

The threshold model described in the previous section operates on an input image to generate a threshold map which predicts the maximum luminance error that can be tolerated at every location over the input image, while preserving perceived visual quality. We can use this threshold map to predict the visible differences of another image relative to the input image; the areas in which the luminance difference between these images is below the threshold map are visually indistinguishable. In a progressive global illumination algorithm, we can use the threshold model to compare intermediate rendered images at two consecutive time steps to locate areas where the global illumination solution has not perceptually converged and concentrate computational effort in those areas. Computation can be stopped in areas where the luminance differences are below threshold. This perceptually based error metric could potentially lead to a significant savings in computation time, but as we saw in the previous section, the threshold model includes components which perform multiscale processing and are quite expensive to evaluate at each intermediate stage of a progressive algorithm. This adds considerable additional overhead to the global illumination algorithm. However, as we shall see in the next subsection, we exploit the representation of our threshold model and information from an earlier stage of the global illumination, to apply the threshold model in a global illumination framework and drastically reduce this overhead.

5.1 Precomputing the expensive components of the threshold model

The most expensive component in the threshold model is the processing of the input image with band-limited multi-scale visual filters. As shown in Figure 3, this operation is required for computing the frequency-dependent elevation function and the contrast-dependent elevation function. These functions predict the loss of sensitivity to scene features with high spatial frequencies and high contrast regions. If we can capture these scene features at an early stage of global illumination computation, these two functions could be evaluated once and reused at later stages.

Our target application provides a structure in which we can evaluate these functions once and re-use them to avoid repetitive model evaluations. Global illumination computation has two major components: direct illumination computation and indirect illumination computation. Indirect illumination computation involves simulating complex light interactions between the surfaces in the scene and is many orders of magnitude more expensive than direct illumination computation. Fortunately, indirect illumination generally varies only gradually over the surfaces and accounts for more subtle effects. Direct illumination computation is comparatively less expensive, but captures most of the higher spatial frequency and contrast content in the scene, such as texture patterns, geometric details, and shadow patterns. These two features make the direct illumination solution a perfect candidate for use in the precomputation stage. In order to ensure capturing the high spatial frequency and contrast present in shadowed portions of the scene, we add an approximate ambient term. This ambient term is computed in much the same way as the ambient term in radiosity algorithms [11, 6]. As shown in Figure 5, the direct illumination solution plus an approximate ambient term capture most of the high spatial frequency and contrast content even in scenes with large portions in shadow. This ambient term is not included while computing global illumination and does not affect the physical accuracy of the global illumination solution.

The scene rendered by direct illumination plus an approximate ambient term is used to evaluate the elevation factor map (the

shaded parts of Figure 3) in a precomputation stage prior to the expensive indirect illumination computation. This serves two purposes: the expensive components of the threshold model are evaluated only once and can be reused, and the noise patterns introduced during the indirect illumination computation do not influence the evaluation of the elevation factor map. The indirect illumination solution is generally "soft" and causes only gradual variation in lighting patterns. The components we precompute only predict the elevation factor due to high frequency content in the scene and are not affected much by the variations in the low frequency content. These components need not be recomputed during the indirect illumination computation. However, the indirect illumination solution does add significantly to the luminance distribution and hence we need to recompute the luminance-dependent threshold during the indirect illumination computation. Fortunately, evaluation of this component of the threshold model is cheap.

5.2 An adaptive global illumination algorithm

We applied our framework to speed up a path tracing algorithm [15]. Path tracing is a type of stochastic ray tracing that traces random paths through the scene to compute the illumination value for each pixel on the image plane. The variance for computing indirect illumination is generally much higher than for computing direct illumination, so a large number of samples have to be taken over the image plane to obtain an acceptable estimate for the indirect illumination component. The algorithm we implemented attempts to reduce the number of samples required for the indirect illumination computation by adaptively refining this component using our threshold model.

The algorithm proceeds through a few basic steps as illustrated in the flowchart in Figure 6. First, the direct illumination solution is computed and an approximate ambient term is added. This is used as an input to the threshold model to generate the elevation factor map which involves precomputing the spatially-dependent functions. This completes the precomputation stage. Next, the computationally expensive indirect illumination solution is progressively computed. At every iteration, the computed indirect illumination solution is added to the direct illumination solution to get an intermediate global illumination solution. The current solution is used to compute the luminance-dependent threshold by evaluating only the TVI function which is not spatially-dependent and is much simpler to compute. The precomputed elevation factor map is then used to scale this luminance-dependent threshold to generate the threshold map which guides the refinement. The luminance difference between the global illumination solutions at the i^{th} iteration and the $(i-1)^{th}$ iteration is compared against the threshold map evaluated at the $(i-1)^{th}$ stage to locate the regions where the solution has perceptually converged. In the next iteration, the regions where the difference remains above threshold are refined. The iteration is continued until the difference over the entire image plane is below the threshold map.

During each iteration the refinement can be carried out by uniformly distributing samples, but it is more advantageous to vary the number of samples in a region based on its "perceptual importance". Higher ratios between the luminance difference map and the threshold map reflect higher perceivability of error. Alternatively, the threshold map at the current stage can be treated as a predictor of the perceivability of error on areas of the image plane, where lower thresholds imply higher perceivability and indicate greater need to sample. In our implementation we used the latter approach to determine the distribution of samples over the regions which need further refining.

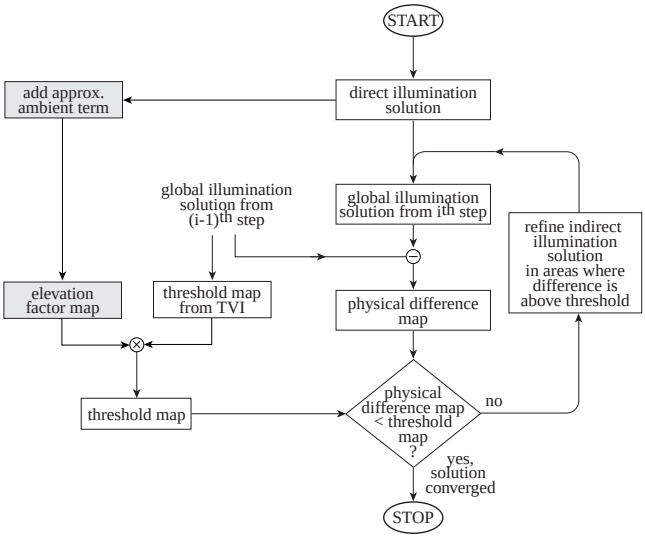


Figure 6: Flow chart of the adaptive global illumination algorithm.

5.3 Results

The adaptive path tracing algorithm described above was applied to several test scenes.

In Figure 7 we show some of the results on the test scene shown in Figure 7(a). The elevation factor map is computed from Figure 7(b). This is used to evaluate the threshold map at every iteration to guide the refinement of the indirect illumination solution. Figure 7(c) is the threshold map at an intermediate iteration. Notice that it has correctly predicted larger thresholds in areas with high spatial frequency and contrast content, indicating the poor sensitivity of the eye in such image features. Figures 7(d-f) show the results from the final iteration of the algorithm. The computed adaptive indirect illumination solution is shown in Figure 7(e) and the sample density pattern it traced is shown in Figure 7(d). Figure 7(f) is the final adaptive global illumination solution. Notice that because our adaptive sampling technique uses smaller number of samples on the areas with high frequency and contrast, the indirect illumination solution for the wall painting and floor carpet shown in Figure 7(e) is noisy. But this noise is completely masked in the final global illumination solution in Figure 7(f). This demonstrates that our threshold model correctly predicted the loss of sensitivity in these textured areas and that we did not have to compute a very accurate solution in these areas. The number of samples taken over the entire image plane required for this solution was approximately 6% of those of the reference solution (Figure 7(a)) computed using uniform sample density, where the number of samples for each pixel is the maximum of all the pixels in the corresponding sampling density map. Notice that these two solutions are visually indistinguishable. (Subtle differences might be noticeable as the threshold model was calibrated to our display device, and the perceivability of differences depends on the image reproduction method and ratio of physical image size to observer viewing distance.)

The two test scenes in Figure 8 were selected to illustrate the computational savings in areas of the image plane which contain texture patterns, geometric detail, and shadow patterns with high spatial frequency and high contrast. The two images on the right, image (c) and image (f), are global illumination solutions obtained using sample density patterns shown in image (b) and image (e) respectively. In the sampling pattern shown here, lighter areas indicate more samples and darker area indicate fewer samples. The sample density patterns result from applying the threshold model

(a) reference global illumination solution (b) direct illumination solution + ambient term (c) threshold map

(d) sample density pattern (e) adaptive indirect illumination solution (f) direct illumination solution + adaptive indirect illumination solution

Figure 7: Applying the threshold model in an adaptive global illumination algorithm.

to our adaptive path tracing algorithm. In image (b), observe that fewer samples were taken on the texture patterns and geometric detail in the scene. Image(e) shows that fewer samples were taken on the shadow pattern on the floor. The two images on the left, image (a) and image (d), are the solutions computed using uniform sample density, where the number of samples for each pixel is the maximum value of all the pixels in the corresponding sampling density map. Notice that the image pairs (a), (c) and (d), (f) are indistinguishable even though the number of samples required by our algorithm was approximately 5% of those of the reference solution in the scene on the top left and approximately 10% of the reference solution in the scene on the bottom left.

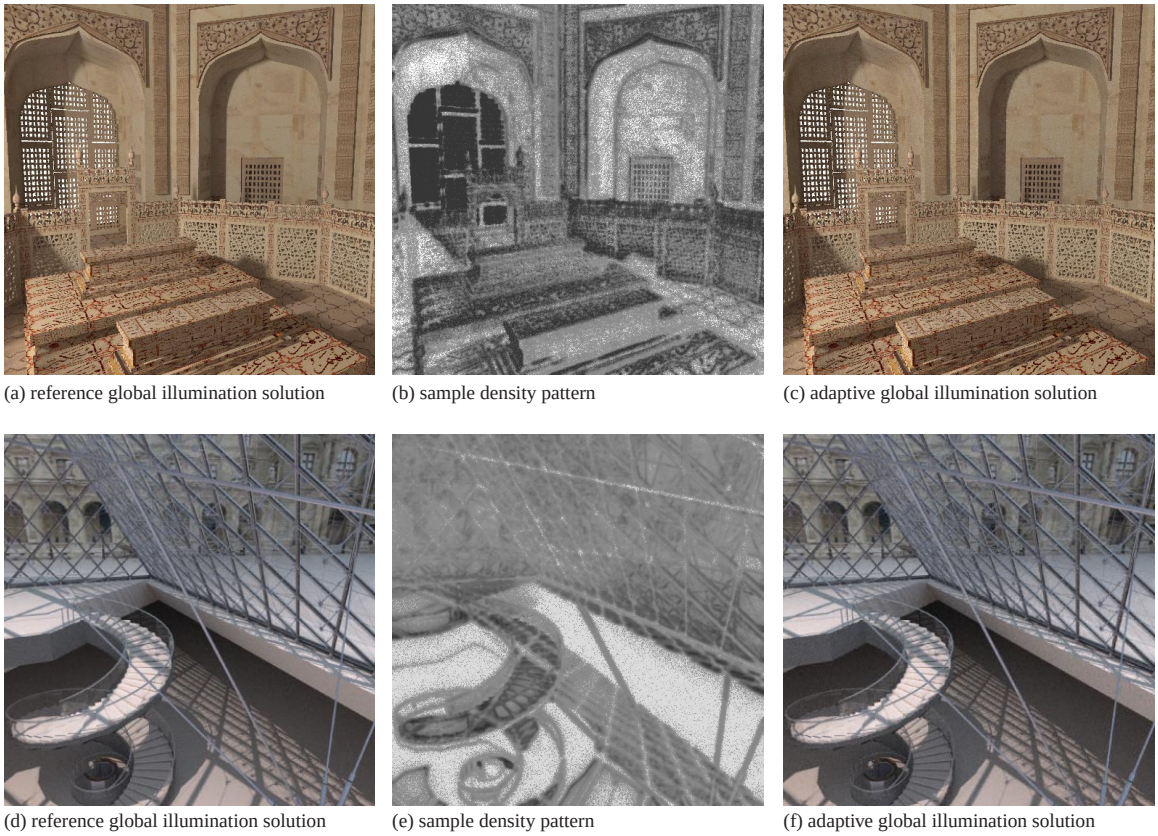
We have tested the algorithm on a number of test scenes and all results show that we can correctly exploit the limitations of the visual system at high frequency and contrast to reduce the expensive global illumination computations. Timing tests reveal that it has given us great benefit at very little extra cost. This is because the expensive components of the threshold model were evaluated only once at a precomputation stage and reused during the rendering iterations. The cost of computing the spatially-dependent component on an image of resolution 512 by 512 is 12 seconds (or, 0.05 ms per pixel) on a 400 MHz processor. In comparison, the luminance-dependent component takes only 0.1 seconds (or, 0.4 μs per pixel) for the same image resolution. These figures are independent of the specific global illumination algorithm used to generate the direct and indirect illumination solutions. Comparisons with uniform sampling methods and adaptive approaches with purely physical error metrics showed that our approach took many fewer samples for computing images of similar visual quality.

5.4 Discussion

The adaptive technique we described above makes very few assumptions about the underlying global illumination computation algorithm. The illumination at each sample on the image plane could be computed using most image-space global illumination algorithms. We only require that the direct illumination solution be computed first, before the indirect illumination solution. There are many methods already developed which make direct illumination computation very efficient [23, 8, 26, 13] and we concentrate on speeding up the relatively expensive indirect illumination computation.

Further research is necessary to better capture details in shadowed areas. Using an approximate ambient term has certain drawbacks. If the ambient term is overestimated then it affects the contrast in the scene and the contrast-dependent elevation function is no longer conservative. One possible approach is to compute the elevation factor map in shadowed areas using only the frequency-dependent elevation function. Another approach is to use a very small ambient term which is sufficiently conservative. The ambient term also fails to capture the high spatial frequencies caused purely by geometric detail in the areas under shadow. For example, in Figure 5(c) the ambient term captured the texture patterns in shadowed areas (the carpet) but overlooked high frequencies caused purely by geometric detail (features of the statue).

In scenes with significant specular-to-diffuse light transfers, high frequency patterns may result at later stages of global illumination (e.g. mirror reflections and caustics). In such cases the spatially-dependent component of the threshold model can be recomputed after these effects become apparent.



(a) reference global illumination solution (b) sample density pattern (c) adaptive global illumination solution

(d) reference global illumination solution (e) sample density pattern (f) adaptive global illumination solution

Figure 8: Sample density patterns and adaptive global illumination solutions for two test scenes.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have described a new framework for perceptually based image synthesis. The objective of this framework was twofold:

- To speed up realistic image synthesis using a perceptual basis which exploits the limitations of the human visual system, and
- To reduce the overhead (in terms of both memory and time) of incorporating such a perceptual basis in the image synthesis algorithm.

To achieve these objectives, we modeled the visual system as a number of components that affect the visual threshold for detecting artifacts depending on the image features. These components together form the threshold model which was used in our framework. Tests on an adaptive global illumination algorithm showed that our threshold model exploits texture patterns, geometric details, and lighting variations in the image to enormously reduce computation time, while preserving image fidelity. By precomputing the expensive spatial components of our threshold model before the more expensive indirect illumination computations, we nearly eliminated all visual processing during the later iterations and also minimized memory requirements. Incorporating the threshold model added an overall insignificant overhead over our standard global illumination algorithm.

In summary, we have vastly improved the computation times for view dependent global illumination solutions using a perceptually based physical error metric.

Through this framework, we have introduced three fundamentally new concepts which have been independently tested and together hold promise for making realistic image synthesis more efficient. These concepts are:

- Predicting the maximum physical luminance error that can be tolerated at any location in an image while preserving perceptual quality.
- Guiding image synthesis algorithms with a perceptually based physical error metric.
- Precomputing expensive components of the vision model essential to perceptually based image synthesis algorithms.

A major goal while designing the framework presented in this paper was to keep it sufficiently general for application to most view dependent realistic image synthesis algorithms.

There is still much work to be done. Our threshold model does not include color, orientation, or temporal processing. Temporal extension to the model is particularly important and would be very useful for dynamic image sequences such as animations or architectural walkthroughs.

Our framework is also especially suited for architectures which switch between model-based and image-based rendering, such as Talisman [24]. These systems render and transform objects as image layers (image-based rendering) instead of re-rendering their geometry (model-based rendering). We could precompute our threshold model from the scene and use it as a perceptual guide for establishing distortion criteria. This could improve performance as it would correctly predict locally higher acceptable distortions due to loss of visual sensitivity.

ACKNOWLEDGEMENTS

Special thanks to James Ferwerda and Jonathan Corson-Rikert for help in preparing this paper.

This work was supported by the NSF Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219) and by NSF grant ASC-9523483, and performed on workstations generously donated by the Intel Corporation and by the Hewlett-Packard Corporation.

References

[1] Peter G. J. Barten. The Square Root Integral (SQRI): A New Metric to Describe the Effect of Various Display Parameters on Perceived Image Quality. In *Human Vision, Visual Processing, and Digital Display*, volume 1077, pages 73–82. Proc. of SPIE, 1989.

[2] Mark R. Bolin and Gary W. Meyer. A Frequency Based Ray Tracer. In *SIGGRAPH 95 Conference Proceedings*, pages 409–418, Los Angeles, California, August 1995.

[3] Mark R. Bolin and Gary W. Meyer. A Perceptually Based Adaptive Sampling Algorithm. In *SIGGRAPH 98 Conference Proceedings*, pages 299–310, Orlando, Florida, July 1998.

[4] Peter J. Burt and Edward H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, 31(4):532–540, April 1983.

[5] Kenneth Chiu and Peter Shirley. Rendering, Complexity and Perception. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 19–33, Darmstadt, Germany, June 1994.

[6] Michael Cohen, Donald P. Greenberg, Dave S. Immel, and Philip J. Brock. An Efficient Radiosity Approach for Realistic Image Synthesis. *IEEE Computer Graphics and Applications*, 6(3):26–35, March 1986.

[7] Scott Daly. The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity. In A. B. Watson, editor, *Digital Images and Human Vision*, pages 179–206. MIT Press, 1993.

[8] George Drettakis and Eugene Fiume. A Fast Shadow Algorithm for Area Light Sources Using Backprojection. In *SIGGRAPH 94 Conference Proceedings*, pages 223–30, Orlando, Florida, July 1994.

[9] James A. Ferwerda, Sumanta N. Pattanaik, Peter Shirley, and Donald P. Greenberg. A Model of Visual Adaptation for Realistic Image Synthesis. In *SIGGRAPH 96 Conference Proceedings*, pages 249–258, New Orleans, Louisiana, August 1996.

[10] James A. Ferwerda, Sumanta N. Pattanaik, Peter Shirley, and Donald P. Greenberg. A Model of Visual Masking for Computer Graphics. In *SIGGRAPH 97 Conference Proceedings*, pages 143–152, Los Angeles, California, August 1997.

[11] S. Gibson and R. J. Hubbold. Perceptually-Driven Radiosity. *Computer Graphics Forum*, 16(2):129–141, 1997.

[12] Donald P. Greenberg, Kenneth E. Torrance, Peter Shirley, James Arvo, James A. Ferwerda, Sumanta N. Pattanaik, Eric P. F. Lafortune, Bruce Walter, Sing-Choong Foo, and Ben Trumbore. A Framework for Realistic Image Synthesis. In *SIGGRAPH 97 Conference Proceedings*, pages 477–494, Los Angeles, California, August 1997.

[13] David Hart, Philip Dutré, and Donald Greenberg. Direct Illumination with Lazy Visibility Evaluation. In *SIGGRAPH 99 Conference Proceedings*, Los Angeles, California, August 1999.

[14] David Hedley, Adam Worrall, and Derek Paddon. Selective Culling of Discontinuity Lines. In *Proceedings of the Eighth Eurographics Workshop on Rendering*, pages 69–80, St.Etienne, France, June 1997.

[15] James T. Kajiya. The Rendering Equation. In *Computer Graphics (SIGGRAPH 86 Proceedings)*, volume 20, pages 143–150, Dallas, Texas, August 1986.

[16] Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, October 1997.

[17] Jeffrey Lubin. A Visual Discrimination Model for Imaging System Design and Evaluation. In E. Peli, editor, *Vision Models for Target Detection and Recognition*, pages 245–283. World Scientific, 1995.

[18] Gary W. Meyer and Aihua Liu. Color Spatial Acuity Control of a Screen Subdivision Image Synthesis Algorithm. In Bernice E. Rogowitz, editor, *Human Vision, Visual Processing, and Digital Display III*, volume 1666, pages 387–399. Proc. SPIE, 1992.

[19] Don P. Mitchell. Generating Antialiased Images at Low Sampling Densities. In *Computer Graphics (SIGGRAPH 87 Proceedings)*, volume 21, pages 65–72, Anaheim, California, July 1987.

[20] Karol Myszkowski. The Visible Differences Predictor: Applications to Global Illumination Problems. In *Proceedings of the Ninth Eurographics Workshop on Rendering*, pages 223–236, Vienna, Austria, June 1998.

[21] Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild, and Donald P. Greenberg. A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display. In *SIGGRAPH 98 Conference Proceedings*, pages 287–298, Orlando, Florida, July 1998.

[22] Holly Rushmeier, Greg Ward, C. Piatko, P. Sanders, and B. Rust. Comparing Real and Synthetic Images: Some Ideas About Metrics. In *Proceedings of the Sixth Eurographics Workshop on Rendering*, pages 213–222, Dublin, Ireland, June 1995.

[23] Peter Shirley, Chang Yaw Wang, and Kurt Zimmerman. Monte Carlo Techniques for Direct Lighting Calculations. *ACM Transactions on Graphics*, 15(1):1–36, January 1996.

[24] Jay Torborg and Jim Kajiya. Talisman: Commodity Real-time 3D Graphics for the PC. In *SIGGRAPH 96 Conference Proceedings*, pages 353–364, New Orleans, Louisiana, August 1996.

[25] F. L. van Nes and M. A. Bouman. Spatial Modulation Transfer in the Human Eye. *J. Opt. Soc. Am.*, 57:401–406, 1967.

[26] Andrew Woo, Pierre Poulin, and Alain Fournier. A Survey of Shadow Algorithms. *IEEE Computer Graphics and Applications*, 10(6):13–32, November 1990.