PERCEPTUAL REFINEMENT FOR HIERARCHICAL RADIOSITY

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF SCIENCE

AT THE UNIVERSITY OF CAPE TOWN

IN FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Adrian Secchia

February 2000

Supervised by

E.H. Blake

Abstract

This dissertation explores the use of a simple model of the human visual system to yield a performance improvement with hierarchical radiosity. Hierarchical radiosity is a physically based rendering algorithm and hence makes no attempt to optimize computation for human perception. We used a model of the edge enhancement properties of the human visual system to produce a perceptually based refinement oracle for the hierarchical radiosity algorithm. Tests of the perceptual oracle shows that it allows the hierarchical radiosity algorithm to produce the same visual quality output in half the time and using half the memory compared to the same algorithm using the standard refinement oracle.

Acknowledgements

I would like to extend my gratitude to my supervisor Prof. Edwin Blake for his guidance and advice throughout my research. I would also like to thank Dr. Nicolas Holzschuch for his help as a research advisor and co-supervisor.

Furthermore I would like to thank my fellow students and friends whose support and often excellent suggestions were invaluable.

Contents

List of Figures

Chapter 1

Introduction

> For the past 25 years, researchers in the field of computer graphics have continuously striven for the production of realistic images of nonexistent environments. To attain this goal and its ultimate potential for design and aesthetic evaluations, it is necessary to accurately represent the appearance of objects and scenes as they look to us. This requires knowledge of how to simulate both the physical behavior of light and the perceptual behavior of the human visual system — *Donald P Greenberg*[6][1]

Over the past few years, radiosity methods have emerged as one of the primary realistic image synthesis tools in computer graphics — in particular hierarchical radiosity[16] has been developed as a very efficient radiosity method. Radiosity methods are examples of physically based rendering algorithms which attempt to solve the so-called global illumination problem by modeling physical laws of light and energy interaction using finite element methods. Much research has been invested into making hierarchical radiosity algorithms more efficient and accurate but, by far, the majority has been making it more physically accurate and little or no research has been aimed at incorporating the perceptual behavior of the human visual system (HVS).

The human visual system is a vastly complex and incompletely understood system that begins at the eye and proceeds through to the visual cortex. A lot of image

[1] Quoted from the foreword of [6]

processing is performed by the HVS including compression (from roughly 135 million photoreceptors in the eye to about 1 million optic nerve fibers), edge detection, filtering, adaption and shape recognition. Knowledge of the HVS has been very profitable in other areas of computer graphics such as image coding. The development of second generation image coding algorithms has lead to compression improvements of about an order of magnitude[28][21]. If a realistic image synthesis algorithm could predict which data would be discarded as irrelevant, which grouped and treated as a unit and which were visually important to the HVS, it could have a substantial benefit in terms of computation cost — without noticeably compromising image quality.

In this dissertation we marry a restricted model of the HVS with hierarchical radiosity — a powerful physically based realistic image synthesis algorithm — to yield significant improvement in not only rendering time, but also memory requirements. Our model of the HVS is based on the receptive field theory in neurophysiology[14] and incorporates such visual phenomena as edge detection and contour sensitivity (Mach banding). Our model of the HVS predicts that contours, highlights and shadow boundaries are visually more important features so we present a refinement oracle that increases refinement in these areas and suppresses refinement in visually less important areas such as those with linear radiosity. Our perceptual refinement oracle uses accurate computation of radiosity gradients[19][2] to estimate the visual importance in a region represented by a patch and thus drive the refinement. We produce a theoretical visual importance measure and an objective image quality measure based on perceptual criteria to compare output images.

1.1 Chapter Outline

The first three Chapters (Chapters 2, 3 and 4) consist of background information relevant to the discussion of a perceptually based refinement oracle and the rest of the dissertation (Chapters 5 through 8) is devoted to the development and discussion of the perceptual refinement oracle.

1. Chapter 1, is this Introduction.

2. Chapter 2, The Background Chapter, contains some relevant background information about global illumination and hierarchical radiosity.

3. Chapter 3, Human Visual Perception, is a discussion about some aspects of human perception relevant to computer image synthesis.

4. Chapter 4, Computing Radiosity Gradient, is a chapter concerning the computation of radiosity gradients using the formulations by James Arvo and Nicolas Holzschuch.

5. Chapter 5, Preliminary Exploration, contains a description of a preliminary experiment conducted using a simple heuristic refinement criterion on a restricted radiosity algorithm.

6. Chapter 6, Theory, describes the relevant theory pertaining to the development of the perceptual refinement oracle which also appears in this chapter.

7. Chapter 7, Results, sets out the tests made and the results obtained. These results are discussed in some detail.

8. Chapter 8 is the Conclusion.

Chapter 2

Background

2.1 Global Illumination

2.1.1 Introduction

In general realistic image synthesis is concerned with converting an internal representation of some environment into a picture with some degree of realism so that the user can better understand and interpret the original model. The environment is stored in the computer memory as a geometric model of objects with their surface properties — colour, reflective properties, transmission, etc. Very often this model doesn't even correspond to a "real-world" environment and is just some construct developed in the mind of the designer and memory of the computer. Some objects are flagged as being emitters — light sources — and therefore have an emitted colour attribute as well. Some realistic image synthesis algorithms don't distinguish between emitting objects and other objects in the model, they rather treat all objects as being potential emitters of light energy. Some of these objects could have a light emittance of zero. The model often includes a camera or eye position for viewing the scene.

The objective of realistic image synthesis is to render the model using all the information about light sources, shape and surface properties. This rendering step computes the colour of each point on the display as seen from the eye (or viewpoint). Some

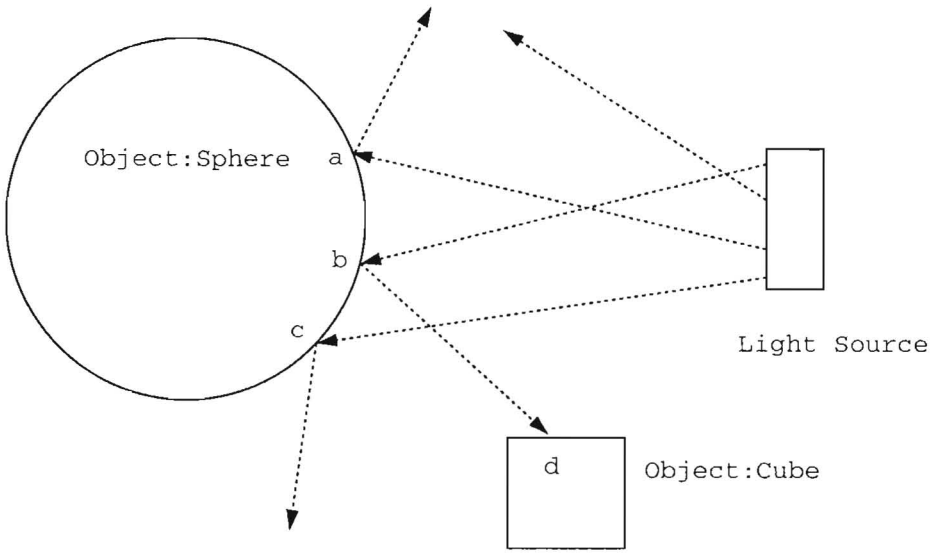


Figure 1: This figure shows a few of the rays from a light source causing illumination on the surface of a sphere.

rendering algorithms compute the visible colour for each point on all surfaces in the model. This effectively computes the colour for viewing from all possible viewpoints in the model.

Illumination effects can be separated into two categories; local and global. Local illumination effects depend on the surface and the incoming light from the light sources. These effects can be accurately rendered knowing the position and orientation of the surface and light sources and the surface properties of the receiving surface. For example in Figure 1 local illumination can be computed at surface point a of the sphere where light arrives from the source. This light arriving from the source depends only on the object receiving the light and the source itself — this is local illumination. Algorithms designed to approximate local illumination effects include Gouraud and Phong[27] shading. Local illumination effects also include the computation of shadows and visibility information.

Global illumination has been the subject of an immense amount of research since almost the first colour displays were developed. The problem of global illumination is that of realistically modeling visual phenomena including the complex interreflections of light between surfaces in the modeled environment. Global illumination

effects incorporate subtle phenomena such as colour bleeding and diffuse interreflection. Algorithms that render global illumination effects require information about all surfaces in the environment to correctly render any surface. Global illumination is essentially the tendency of a surface to behave like an area light source as a result of reflecting light back into the scene and affecting nearby surfaces. This behavior causes the fairly subtle effects like colour bleeding that are often taken for granted in the real world. Global illumination effects, while subtle, contribute substantially to the perceived realism of the output — without them rendered pictures appear very stark and have a discernible computer-generated feel. In Figure 1 the light received at point d on the surface of the cube from the light source via point b on the surface of the sphere causes global illumination effects. Say the light source emitted white light and the sphere was red then the light from b to d would cause a red tinge to whatever colour the cube would have been.

2.1.2 The Rendering Equation

There are many varied algorithms, such as radiosity and ray-tracing, designed to cope with specific visual phenomena. All of these algorithms solve some subset of the complete global illumination problem. One standard formulation of the global illumination problem that unifies a wide variety of illumination algorithms is the rendering equation due to Kajiya[20].

$$I(x,x') = g(x,x')\left[\epsilon(x,x') + \int_S \rho(x,x',x'')I(x',x'')dx''\right] \tag{1}$$

where:

$I(x,x')$	Intensity of light from point x' to x
$g(x,x')$	"geometry" term
$\epsilon(x,x')$	Intensity of emitted light from x' to x
$\rho(x,x',x'')$	Intensity of light scattered from x'' to x by point x

As is noted in the paper[20] this equation simply balances the energy flows from one surface to another.

The rendering equation (1) is a non-trivial integral equation that does not submit to tractable analytic solution except in extremely restrictive circumstances. It is interesting to note that most rendering algorithms can be viewed in terms of this unified equation — mostly using an approximation to increase the efficiency of evaluation. Ray-tracing solutions restrict this equation to perfect specular reflection and transmission and approximate all the other phenomenon. Radiosity uses finite element analysis to compute the complete solution for all diffuse interactions between patches (discrete elements of the surfaces present).

It is worth noting (as noted in Kajiya's paper) that as an approximation to Maxwell's equation for electromagnetics, the rendering equation makes some simplifying assumptions and ignores some – possibly important – physical phenomena. The ignored effects include polarization, participating media for transparent objects and diffraction. These effects can be included, however the increased complexity of the rendering equation may cause the rendering algorithm to become unacceptably complex and slow. The extra computation expense far outweighs the increased visual realism except in fairly specialized and restricted circumstances. Undoubtedly as hardware speed and capacity increases and more sophisticated algorithms are developed modeling of these phenomena will become more commonplace.

As mentioned above, the rendering equation doesn't submit to tractable analytic solution for almost all useful applications. It also turns out that most realistic image synthesis algorithms can be regarded as partial solutions and approximations to the rendering equation. Given certain simplifying assumptions, the rendering equation can be computed in reasonable time for environments having a useful level of complexity.

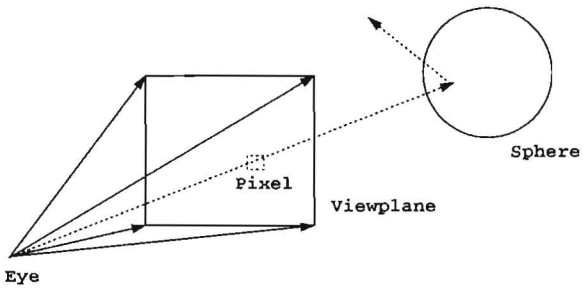


Figure 2: This figure shows the viewing frustum for a ray traced scene.

2.1.3 Ray Tracing

Ray Tracing operates by computing a ray within the viewing pyramid from the eye out into the scene — typically one ray per pixel (Figure 2). These eye rays are compared against the world model for intersections — the closest one to the eye being the most important. At this point of intersection all local illumination effects are computed by examining each light source, the relevant surface properties, etc. Light sources are traditionally modeled as point light sources and are distinct from the visible objects in the scene. Rays are then computed in the specular reflection and transmission direction to investigate the properties of incoming light on those directions. The process is recursively repeated along each of these specular rays till some user defined recursion limit is hit or the ray intersects no objects. In this way, ray tracing algorithms capture all local illumination effects and all perfect specular global illumination effects. Diffuse global illumination is usually approximated by adding some ambient light term[11]. The ray tracing algorithm has been extended to incorporate many different types of surfaces and objects[17] and hence is a widely used tool in realistic image synthesis.

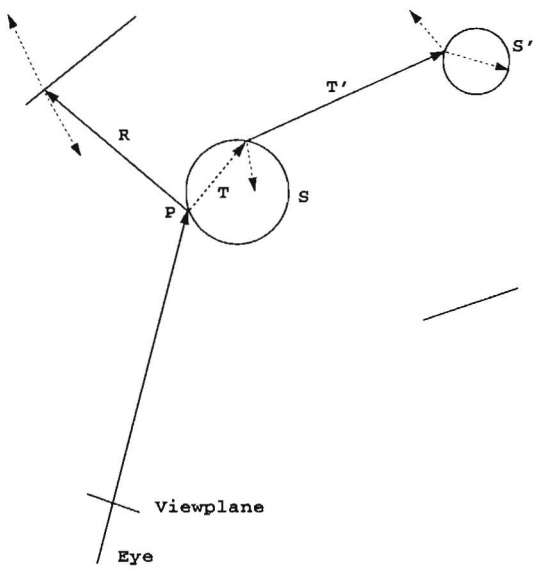


Figure 3: This figure illustrates the generation of a ray tree for a ray tracing algorithm. At each point of intersection, all of the local illumination is computed and the reflected and transmitted rays are computed recursively.

2.2 Radiosity Methods

2.2.1 Introduction

Radiosity is the name given to the luminous flux density per steradian leaving the patch [4] usually denoted by the letter B. It is also used to refer collectively to the algorithms that use radiosity to compute diffuse interreflections. Intuitively the radiosity on a patch is the rate at which energy leaves the surface[8].

Radiosity solves the rendering equation for diffuse surface interreflections by means of finite element analysis. This approach is based on emission and reflection of radiation in thermal-engineering models and was first introduced by Goral et. al.[13] and Nishita and Nakamae[25]. These methods are based on the conservation of light energy in a closed environment and operate by balancing the energy flows from one surface to another. Light energy interactions are considered between all surfaces and their radiosities computed.

Firstly all the surfaces are broken down into a finite number of simple polygonal patches - these patches are the finite elements used to evaluate the radiosity approximation to the rendering equation (Equation 1). The radiosity on any given patch is assumed to be constant. The integral equation is solved iteratively to convergence using numerical methods. The radiosity values for each patch are stored along with the polygonal description of the scene to yield a complete solution for all diffuse light interactions between the patches. The solution can then be examined using some (interactive) polygon simulation and is accurate (within certain bounds) and independent of viewpoint.

It is this viewpoint independence that makes radiosity one of the more convenient global illumination solutions. The entire solution can be precomputed and then viewed interactively without any need for viewtime recomputation. Even if some of the light sources are changed, most of the computation — specifically the formfactors and visibility — can be reused for the new solution. Form factors are defined in the next section.

2.2.2 The Radiosity Equation

Hanrahan et. al. [16] use the term "element" to refer to the leaves on the surface subdivision tree and the term patch to refer to any larger piece.

Using this terminology, the radiosity equation is

$$B_i = E_i + \rho_i \sum_{j \in S} F_{ij} B_j \tag{2}$$

where the i and j refer to two different elements, B_i is the radiosity, E_i is the emissivity, ρ_i is the diffuse reflectance, F_{ij} is the form factor and S is the set of elements in the scene.

Where the form factor between two finite elements i and j is defined as

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r_{ij}^2} dA_i dA_j \tag{3}$$

and represents the fraction of light leaving element i and arriving at element j. The form factor is a completely geometric term and is independent of any lighting effects. In this equation the angle θ_i relates the normal of element i to the vector joining the two differential elements on i and j. The area of element i is denoted as A_i.

This integral equation can be computed by a closed form analytic solution in certain special cases. The one used in this dissertation is the point (or differential area) to closed polygon, which can be evaluated by a contour sum around the edges of the polygon[6]. The subscripts for r are computed modulo n where n is the number of vertices on the emitter polygon.

$$F_{dA_i \to A_j} = \frac{1}{2\pi} \sum_{i=1}^{n} \beta_i N_i \cdot \| r_i \times r_{i+1} \| \tag{4}$$

We note that the radiosity equation (2) actually represents a set of simultaneous equations and can be represented as a matrix as follows:

$$\begin{pmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix} \tag{5}$$

Equation 5 can be solved using Gauss-Seidel iteration or other variations of Jacobi methods.

Generally, the form factor is sampled at various points over the patch and used to compute values for the radiosity at each of these points. The radiosities are then interpolated between these sample points. Usually the radiosities are computed at the vertices and/or center of each patch or element.

2.2.3 Hierarchical Radiosity Methods

The problem with standard radiosity is that the form factor matrix in equation 5 is $O(n^2)$ where n is the number of elements in the scene. Obviously this $O(n^2)$ is

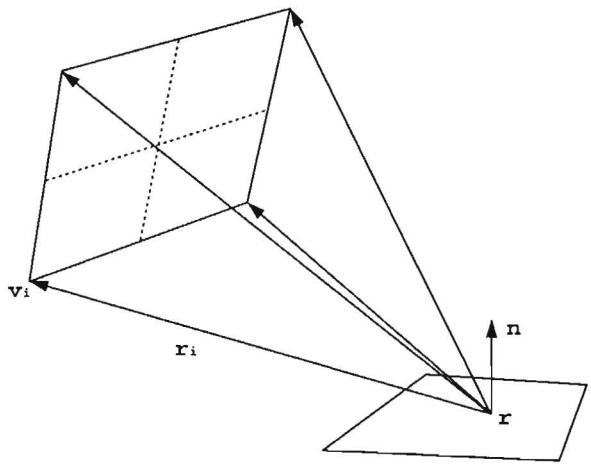


Figure 4: This figure demonstrates how the form factor is computed from a point r to an emitting patch.

a problem, both in space and time constraints. Hierarchical radiosity reduces this matrix, by blocking methods, to $O(n)$. This is equivalent to reducing the number of patch-to-patch interactions in the scene to $O(n)$ rather than $O(n^2)$.

Hierarchical radiosity algorithms achieve this optimization by maintaining a tree of recursive patch refinement and limiting the interactions between two patches to a level in the tree at which the error involved in the interaction is just below some user defined error threshold.

Typically radiosity programs approximate the patch-to-patch form factor equation (double integral equation 3) by a differential area to patch calculation. In these cases, we find that the error on an interaction is almost exclusively due to the form factor calculation not approximating the actual form factor when the patches are too close relative to their size. The two interacting patches need to be small enough relative to their separation to make the differential area-to-patch approximation sensible.

Each patch in the scene has an associated patch interaction hierarchy. The radiosity contributions are obtained from each interaction at the appropriate level of the tree. The radiosities propagated pack into the scene at each level are computed by a single depth first traversal of the hierarchy in what is known as a push-pull. Firstly all the radiosities are added (pushed) down the tree to their children. These values are then

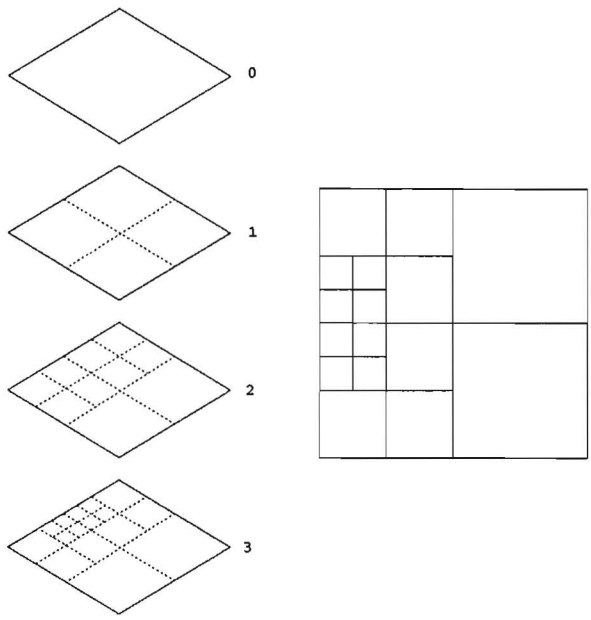


Figure 5: The adaptive sampling due to hierarchical patch subdivision (refinement).

pulled back up the tree with each parents radiosity being the area weighted average of its children's radiosities.

2.2.4 Refinement

Definition 2.1 *Refinement is the process of splitting a patch into smaller sub patches (as in Figure 5). The radiosity calculations are also forced down the tree to interact with smaller patches by this refinement process.*

Definition 2.2 *The refinement threshold (or refinement criterion) is the user specified maximum allowable error on an interaction.*

Definition 2.3 *The refinement oracle is a module of code used to evaluate the error on a patch-to-patch interaction. If the error is greater than the refinement threshold, then the interaction is refined.*

The refinement process is the method used by radiosity algorithms to increase the

sample density in an area. This process is driven by the refinement oracle and is effectively an adaptive subdivision algorithm (Figure 5). Standard progressive radiosity requires all radiosity interactions happen at the lowest level of refinement which results in most interactions occurring at a much lower error level than the user specified refinement threshold. While this has no negative impact on the output visual quality, it substantially increases the computation load. Hierarchical radiosity algorithms maintain the refinement subdivision in a quadtree as opposed to only keeping the leaf patches. This allows the radiosity interaction to occur at a higher level in the tree — effectively minimizing computation while still ensuring that the interaction error is below the user specified threshold.

The refinement oracle is generally a heuristic measuring the error on an interaction between two patches. It is generally designed to ensure that the error is never underestimated (hence often substantially overestimates the error). The major source of error in an interaction arises from the use of the point-to-patch form factor (Equation 4). The assumption is that one of the patches can be represented as a point or differential area and this is only accurate if the separation is large compared to the patches area.

The standard error-control refinement oracle (often called BF refinement[16]) makes use of the form factor size as an indication of the error on the interaction. The form factor value (F) is weighted by the amount of radiosity (B) on the interaction. When more than one sample point is used per patch, ΔF can be used ($\Delta F = F_{\text{max}} - F_{\text{min}}$). This form factor value is usually weighted by B the radiosity, A, the area of the patch and, ρ, the reflectance coefficient. If this computed error value is above the refinement threshold, then the interaction is refined. This process is repeated recursively till all interactions occur below the specified error threshold.

The formulation for the standard refinement threshold value[18] is then

$$R_{cs} = (F_{\text{max}} - F_{\text{min}})BA\rho \tag{6}$$

We will usually refer to the refinement criterion as R_c and leave the specification of

standard as understood by the context.

Discontinuity Meshing

Not all radiosity algorithms use quad tree structures to represent patches and refinement. Some radiosity systems implement a scheme known as discontinuity meshing.

The radiance function is smooth except along certain discontinuity lines or curves[23]. Radiance discontinuities are denoted $D0$ and occur along the intersection of surfaces in the scene. For instance, a table top touching a wall creates a $D0$ discontinuity. Discontinuities in the first and second derivatives ($D1$ and $D2$) occur when objects in the scene line up — for instance moving along a line on the receiver from a shadow into an illuminated area crosses a line where the emitter becomes partially visible (from being completely obscured). See Section 4.3.1 for example radiosity functions with discontinuity lines.

Discontinuity meshing strategies split the polygons along the lines of discontinuities - usually limiting the discontinuities considered to a small subset of those possible for efficiency reasons. For a more complete description of discontinuity meshing cases and an algorithm that performs discontinuity meshing, including discontinuity curves caused by "EEE" events (curves of discontinuity caused by three edges interacting), see Drettakis and Fiume[7].

2.3 Summary

In this section we have described the global illumination problem in terms of the rendering equation (1). We have briefly discussed ray-tracing and, in much more detail, radiosity as two partial solutions to the rendering equation. We have developed some of the introductory mathematics for the radiosity solution and described a fairly fundamental optimization called hierarchical radiosity — which we intend to make use of in this dissertation.

Chapter 3

Human Visual Perception

3.1 Introduction

In this chapter we motivate the use of perception theory for hierarchical radiosity and outline some properties of the human visual perception system.

We describe the human visual system (HVS) is some detail. The HVS is complex and interesting enough to merit much more space than we devote to it here so we concentrate on an overview and some specific properties relevant to computer graphics and image synthesis. These properties include Mach banding and contrast sensitivity.

Many of the relevant properties of the HVS can be described in terms of receptive fields of neurons in the visual pathways. These receptive fields are responsible for edge detection, shape resolution, Mach banding and many more properties. It is the edge detection property that is the most useful for exploitation by image synthesis algorithms and hence will be the primary property of the HVS considered in this dissertation. The description of the rest of the HVS is included to give a more complete overview of the processing performed on the path from the eye to the brain.

We outline some of the drawbacks of physically based rendering algorithms - with particular emphasis on the specific problems in radiosity. These include aliasing of the radiosity function and over refinement due to error estimates.

We also show how to take advantage of features built in to most radiosity algorithms — such as the bi-linear reconstruction of output polygons — to improve the radiosity refinement algorithm.

3.2 Motivation

Radiosity is an example of both a visualization system and a realistic image synthesis algorithm. The goal of a radiosity simulation is to render an environment to the screen so that an observer can understand it and interact with it. As with most visualization systems, the primary target is the Human Visual System (HVS) rather than the screen. Typically the screen is just the medium for communicating the data to the eye.

An example of the successful application of knowledge of the HVS to computer science is in image coding. Up until the 1980's, most image coding systems relied entirely on information theory to exploit redundancy in images[28]. These techniques were pixel-based and hence the primary goal was screen accuracy rather than visual perception accuracy. With the incorporation of the HVS knowledge, compression rates improved from around $2 - 10 : 1$ to $70+ : 1$ with acceptable image quality — ie the decoded image appears identical to the original.

There is considerable scope for the application of HVS knowledge to hierarchical radiosity algorithms. Most radiosity algorithms overestimate the error on an interaction and hence overcompute the interaction. Most image synthesis algorithms are concerned with computing exact values for all the pixels in the output image or producing the most physically accurate rendering of the data without considering the HVS. While this is useful for analysis purposes, it is usually a waste of effort for viewing. Due to the fact that the HVS processes the input data and codes it in much the same way as a second-generation image compression algorithm, much of the laboriously computed (pixel) data is ignored or discarded as irrelevant in the process of actually viewing the image. If the radiosity algorithm could predict which data would be discarded as irrelevant by the HVS, which grouped and treated as a unit and

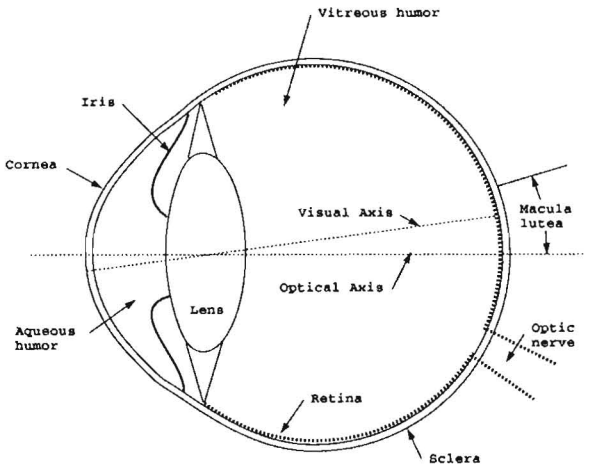


Figure 6: A schematic view of the eye after [12].

which were in some way more visually important, it could have a substantial benefit in terms of computation cost and data storage — without noticeably compromising the image quality.

3.3 Human Visual System

3.3.1 The Eye

The eye is a complex visual sensor and acts as the entry point for the human visual system. At the front of the eye Figure 6 is a fairly complex focusing mechanism consisting of the cornea, iris and lens. The cornea and lens act to focus the light onto the retina at the back of the eye and the iris acts as a diaphragm controlling the size of the pupil (aperture). The retina is a thin but complex neurosensorial layer covering about 200 deg on the back of the eye. It contains two different types of photoreceptors: cones and rods. In a normal eye, there are approximately $120 - 130$ million rods and $6 - 6.5$ million cones[21][12]. Rods are much smaller than cones and are sensitive to light intensity. Because more of them can be packed into the same space, they improve the spatial resolution. Rods are used in low light conditions (scotic vision) hence the black and white appearance of night vision. There are three

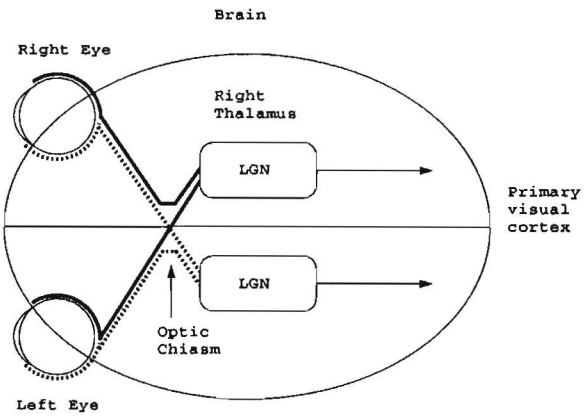


Figure 7: A schematic diagram of the visual pathways from the eyes to the visual cortex. (Diagram after [5])

types of cones corresponding to three different wavelengths of light. Cones need more light than rods (photopic vision) and are typically active in daylight. There is a small region in the center of the visual axis known as the fovea which contains only cones. Cones are used for resolving details and colour.

The rest of the retina is made up of other neurons: bipolar, horizontal, amacrin and ganglion cells. Photoreceptors synapse with bipolar cells which in turn connect to the ganglion cells. The axons of the ganglion cells make up the optical nerve and go all the way to the Lateral Geniculate Nucleus (LGN) in the thalamus (Figure 7). Horizontal and amacrin cells are responsible for parallel and feedback connections in the retina. The parallel and feedback connections are responsible for lateral inhibition of photoreceptors.

It is interesting to note that although there are about 135 million photoreceptors in the eye, there are only about a million fibers in the optical nerve[12]. These figures give a rough idea of how much visual processing is performed by the retinal cells.

The optic nerve crosses over in the optic chiasm. At this point the nerve fibers corresponding to the left field of view from both eyes proceed into the right side of the brain and the right field of view to the left side of the brain.

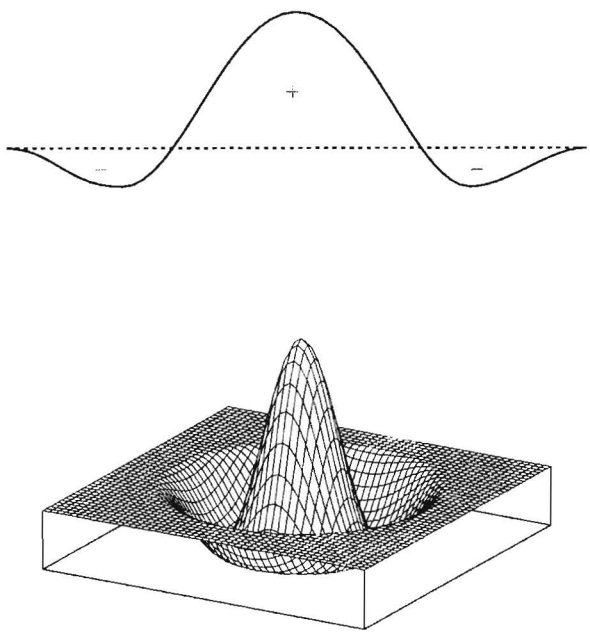


Figure 8: The response of the retinal ganglion cell center on surround off receptive field. Note that this response is very similar to an edge detection filter.

3.3.2 Receptive Fields

Definition 3.1 *The receptive field of a cell is simply the area of the retina which can influence the behavior of that cell[21].*

The excitory and inhibitory areas of the receptive field are known as "on" and "off" areas respectively. The ganglion cells in the retina have on-center off-surround or off-center on-surround receptive fields[1][21][14]. These receptive fields take the shape of a circular center area with a circular surrounding annulus. The receptive fields for individual ganglion cells overlap on the retina and have a response curve cross section like that of Figure 8. This response curve is exactly an edge detection filter. This is why accurate representation of edge information is essential to our perception of an image[28].

The area of the receptive fields generally increases as one moves out toward the edges

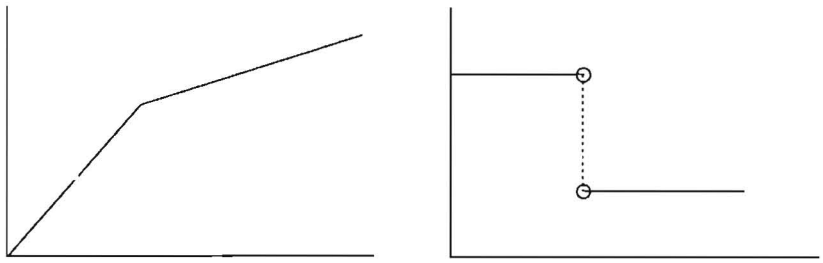


Figure 9: Mach bands can occur when the intensity gradient is discontinuous. This figure shows the intensity vs position on the left and the gradient of the intensity vs position on the right.

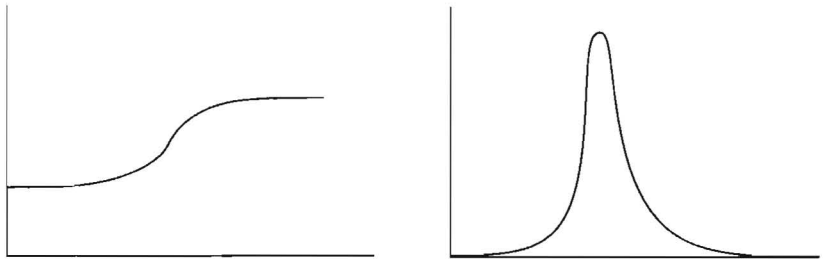


Figure 10: Mach banding can occur when the intensity gradient is continuous but sufficiently large. The intensity plot is on the left and its gradient is shown on the right.

of the retina. The retinal ganglion cells are responsible for edge detection in the visual system due to the filtering nature of their receptive fields.

3.3.3 Mach Bands

Mach bands are the illusion of edges or discontinuities in a region of a picture where the intensity is in fact varying continuously, i.e., no edge is actually present. Mach bands generally occur in regions where the intensity gradient is discontinuous as shown in Figure 9.

Mach bands are not limited to discontinuities in the intensity gradient, however. They can quite easily occur in regions with large but continuous gradient values such as in Figure 10.

The causes of Mach bands, while not fully understood, are usually explained in terms

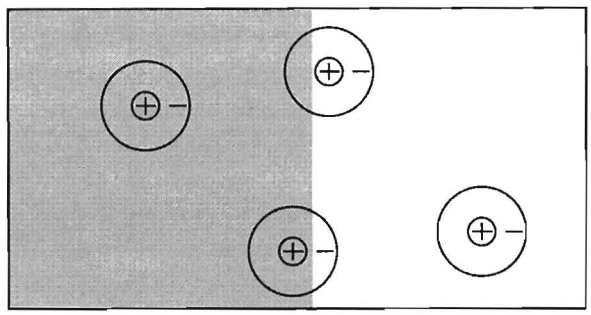


Figure 11: Several retinal ganglion cells superimposed on a region that is partitioned by two luminous intensities. This diagram shows how the receptive fields accentuate the bright/dark border causing the Mach banding effect.

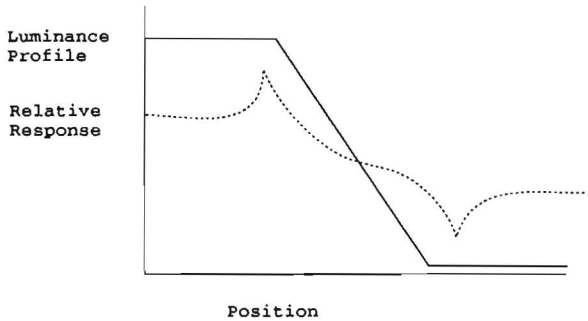


Figure 12: Contrasting the the relative response of the eye (dotted line) to the luminance profile (solid line).

of retinal ganglion cells (Figure 11), in particular the center-surround receptive field[1] These cells act as weighted integrators of the response. The ganglion cells sum the response of the photoreceptors over a small circular region and subtract over adjacent cells (center on surround off). This phenomenon is also partly the result of lateral inhibition [12] in the retinal cells.

This analysis suggests that Mach bands will occur in regions where the intensity gradient is sufficiently large or undefined - i.e. where the intensity is changing rapidly. Note how the weighted integration over this visual field increases the response in the region of the edge (intensity boundary).

It is interesting to note that Ernst Mach based his conclusions more on observation and insight than on neural ganglion response theory and he deduced the lateral

inhibition caused by the ganglion cells.

We note that the human visual perception system doesn't present the true luminance profile to the brain, rather it exaggerates the importance of certain areas in a method analogous to an edge detection filter (Figure 12). This implies that the brain is going to be more sensitive to edges and regions of rapid intensity change (large gradient) rather than on large regions of plain unchanging colour. It is precisely these areas than need to be captured in detail by any perceptually based image synthesis algorithm.

3.3.4 The LGN and Visual Cortex Cells

After the optic chiasm (Figure 7), the nerve fibers pass into the LGN in the thalamus. This region of the brain is organized into layers of cells that perform a similar function to the retinal ganglion cells — they have circular receptive fields. Information from each eye alternates across layers in the LGN. This is known as optical dominance.

From the LGN the signal passes into the visual cortex. The visual cortex is also arranged into layers, in this case layers of columns corresponding to directional dependence. The cells in the visual cortex are simple, complex, hypercomplex or higher order hypercomplex. The receptive field of simple cells is elliptic with a bar shaped center causing it to have a response with an orientation dependence. The cells in the visual cortex all have different responses and enable the detection of edges, lines, endpoints on a line, corners and discontinuities. The LGN and visual cortex also contribute significantly to edge and feature enhancement.

3.4 Contrast Sensitivity

Contrast sensitivity can be measured in a variety of ways. Some of the common ways include just noticeable differences (JND) and response to frequency gratings.

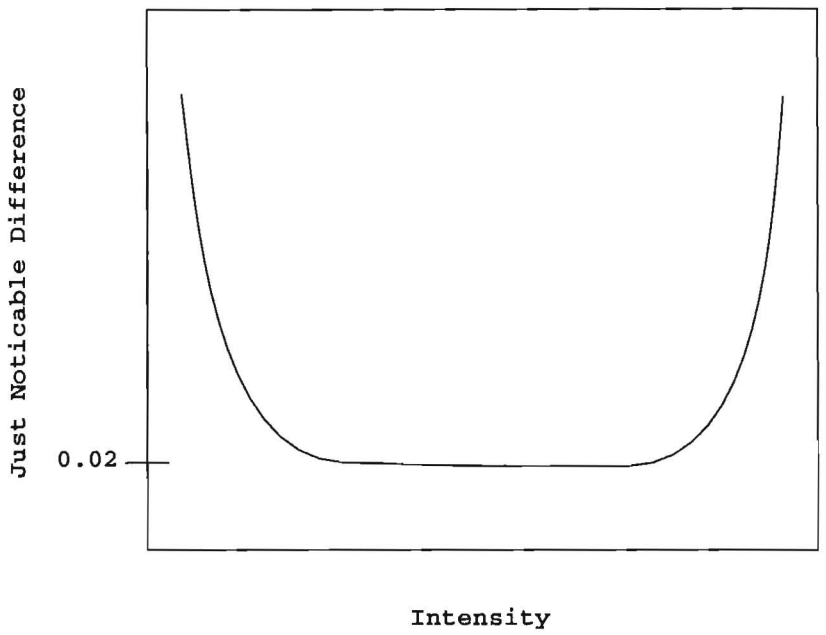


Figure 13: Ratio of just noticeable differences

3.4.1 Just Noticeable Difference

Suppose we have a visual field with two adjacent areas having an intensity of I. If the intensity of one of the areas is increased by ΔI till a difference is noticed, the ratio $\frac{\Delta I}{I}$ remains constant for a very wide range of I. The ratio $\frac{\Delta I}{I}$ is known as the Weber constant and is approximately equal to 0.02 (Figure 13).

This suggests that the HVS is sensitive to ratios of intensities rather than on the absolute intensity value. This ratio sensitivity is a major contributing factor accounting for the HVS being able to operate under a very wide range of absolute intensity conditions.

Gibson and Hubbold[10] show how to use the JND to optimize radiosity computation. In their paper they show that computing the colour change on each interaction in a perceptually uniform colour space (CIE LUV) allows reductions in computation by ignoring interactions that change less than the JND. Their results show a 93% saving in the number of shadow rays for radiosity solutions of a large number (500) of iterations.

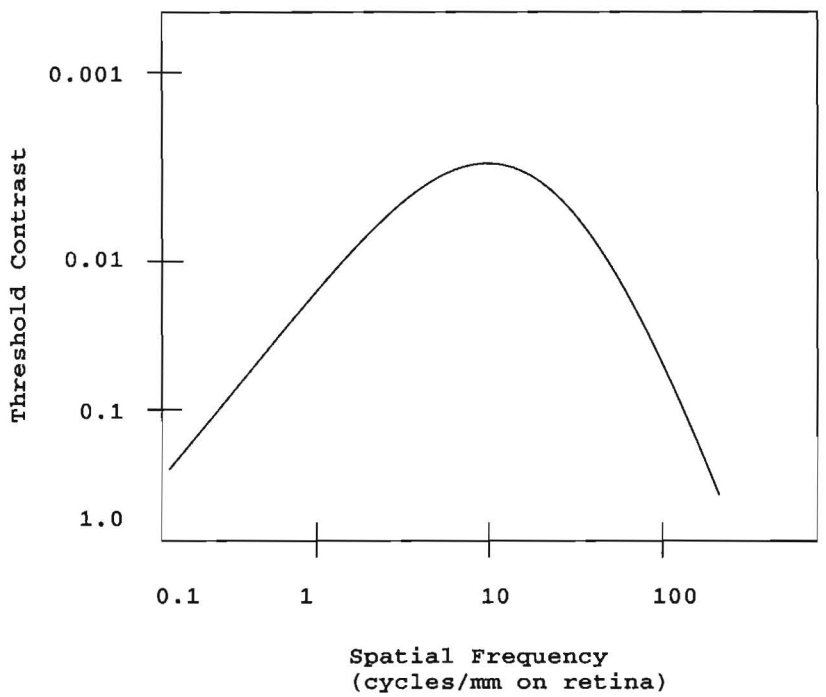


Figure 14: The contrast sensitivity function for sine waves for an adult human visual system. This figure is reproduced from Andrew Glassner, Principles of digital image synthesis, Figure 1.18[12].

3.4.2 Gratings

Contrast sensitivity can be measured with respect to gratings of different frequencies and contrasts. Figure 14 shows the contrast sensitivity function (CSF) of a human adult to sine wave gratings of different frequencies. The spatial frequency of the sine wave is measured by the number of cycles per mm on the retina. Above the line in the graph there is not enough contrast to resolve the frequency of the grating and it appears as a gray blurred field.

Contrast sensitivity is affected by many factors including age, adaption to a frequency, use of photopic or scotic vision and direction.

3.5 Drawbacks of Physically Based Rendering

Physically based rendering algorithms are those that are based on properties found in physics. For instance, radiosity algorithms based on variants of Maxwell's equations of electromagnetic energy propagation. These are contrasted to empirical or heuristic algorithms which have no consistent basis and perceptual rendering algorithms which are based (at least in part) on properties of the HVS.

The end point of most rendering algorithms is the HVS and not the screen as is the usual misconception. Consequently physically based rendering algorithms exclude a substantial part of the realistic image synthesis process — the human perception of the image produced. Without a doubt the HVS modifies the form of the input image in complex and interesting ways. If knowledge of the process of the HVS is not incorporated in the rendering algorithm, it is likely that the rendering process will make mistakes. For instance some areas may be computed in detail while the HVS would ignore that area causing a lot of computation time to be wasted. In other regions, not enough detail may be computed and give rise to image artifacts.

3.5.1 Excessive Refinement

In hierarchical radiosity, physically based rendering generally applies to the decision process devoted to calculating the refinement of an interaction between two patches. This process is often called the refinement oracle (See Definition 2.3). In a physically based radiosity program, the refinement oracle is based on measuring the error present in the computation of the energy flow across and interaction. This energy error is computed based on the approximations made in the computation of the energy flow (radiosity). The oracle often guesses at this error as it is fairly complex to compute and the oracle needs to be a fairly streamlined routine. The guess made is an overestimate of the error and hence often promotes excessive refinement of an interaction which adds to the total computation time.

This problem of excessive refinement is even more pronounced where the error on an interaction is large but perceptually the interaction is unimportant — ie the HVS will

ignore much of the information in this region. The physically based radiosity oracle will force a lot of excess refinement in this area. Due to the extra refinement, a lot of space is wasted storing information for areas that could be represented with much less refinement.

3.5.2 Aliasing

Definition 3.2 *Aliasing is the misreconstruction of a high frequency phenomenon as a lower frequency phenomenon caused by the sample density being too low.*

In radiosity algorithms this problem frequently manifests itself by the algorithm completely missing visual phenomena such as small highlights and shadows. These phenomena are typically contained within a patch — and hence miss all the sampling points which are usually at the vertices of the patch.

Physically based radiosity algorithms make no attempt to assign importance to visual phenomena such as shadows and highlights. The error values (which drive the refinement oracle) are computed in the same way as for regions which have less important visual phenomena. All radiosity algorithms are subject to this aliasing problem due to the discrete nature of the sample points and the impracticality of sampling on the pixel scale. Physically based algorithms, however, suffer from non-uniform picture artifacts. Picture artifacts of a given scale are more noticeable in areas which have visually important features than in other areas. Physically based algorithms compute the error on an interaction to the same level for every interaction regardless of the visually important features and weather that amount of error is going to be more noticeable than in other regions.

3.6 Reconstruction

There are two phases to radiosity algorithms. Firstly the radiosity surface is sampled at points determined by the radiosity algorithm. The sample points are then used to

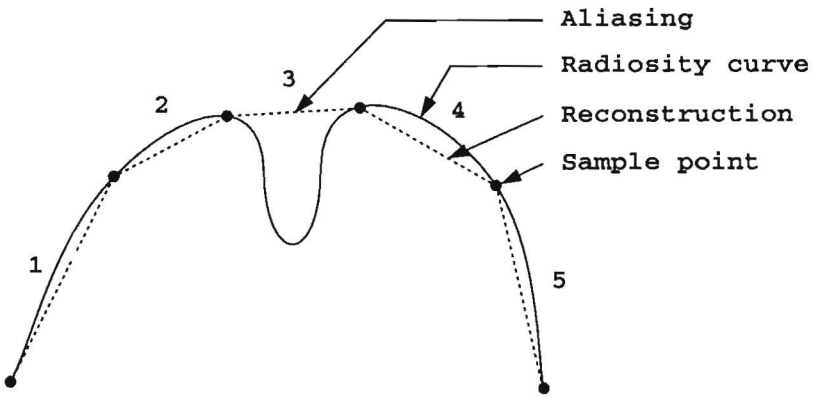


Figure 15: Reconstruction of a radiosity curve. Segment 3 shows aliasing occurring.

reconstruct the radiosity function on the screen for viewing by the user. Generally it is impractical to compute sample points on a pixel level.

Most radiosity systems compute the sample points in a regular adaptive grid. Some use a triangular mesh of samples. Almost all radiosity algorithms use bi-linear interpolation to reconstruct the radiosity function. Bi-linear interpolation assumes that the radiosity surface is linear between two sample points — this is not true in general and hence the interpolated shading can be in error. This is often seen as aliasing where small visual phenomena are missed or incorrectly reconstructed. This is generally a problem of too few sample points rather than a reconstruction error (See Figure 15).

If the radiosity refinement oracle has knowledge of the bi-linear nature of the reconstruction system, it can optimize the selection of refinement. When the radiosity function is linear over a region, the sample density can be reduced because the bi-linear interpolation is going to (coincidently) represent the linear nature very accurately. Care should be used when determining the linear nature of a region of the radiosity function, to avoid aliasing problems.

3.7 Summary

This chapter is about motivating the use of knowledge about the human visual system (HVS) in realistic image synthesis — specifically to hierarchical radiosity.

To this end we describe the HVS in some detail including descriptions of the eye and visual pathways in the brain. Many of the properties discovered about the HVS can be explained using the theory of receptive fields for neural cells in the visual pathways. Among other things these receptive fields account for edge and line detection and Mach banding. For this dissertation, however, we will only require knowledge of the edge detection filter performed by the HVS at various levels in the visual pathways.

We point out the shortcomings of classical or physically based rendering algorithms -- again specifically hierarchical radiosity. We pay particular attention to the reconstruction of the sample points provided by the radiosity algorithm. The sample points can cause aliasing of the radiosity function. The reconstruction of the sample points is usually accomplished by bi-linear interpolation and we point out that this is exploitable by the radiosity refinement algorithms.

Chapter 4

Computing Radiosity Gradient

4.1 Introduction

In Chapter 3 we determined that the HVS performs edge detection over the field of view. Because of this, we note that accurate reproduction of edges is essential to our perception of the image. To be able to more accurately represent edges, we need to be able to predict where they will occur in the radiosity simulation.

As the radiosity is closely related to the intensity (Chapter 2) we can predict where edges occur by examining the slope of the radiosity function over a surface. These intensity edges are indicated by rapidly changing or large radiosity gradient.

Most of the relevant work on computing radiosity gradients is contained in papers by James Arvo[2] and Nicolas Holzschuch[19]. Paulin and Caubet[26] also produce a closed form derivation of the radiosity gradient combining the work of Arvo and Holzschuch.

In any given scene description the light sources define a radiance function with the domain being all spatial positions and directions. This radiance function is uniquely defined and completely defines the radiant energy distribution[2]. When this function is restricted to the surface of an object, it can be used to compute the radiosity over the surface as a function of surface position. This function determines the intensity

at each point on that surface. Arvo's paper defines and computes the irradiance Jacobian which can be employed to compute the irradiance gradient as a special case and therefore can be used to determine the radiosity gradient. Holzschuch's paper provides a formulation that directly computes the radiosity gradient.

4.2 Irradiance Gradient

Most of the derivations in this section are due to James Arvo and are set out in more detail in his paper[2]. They are reproduced here for convenience and completeness.

4.2.1 Definitions

Definition 4.1 *Irradiance: The energy flux per unit area arriving on a surface.*

Definition 4.2 *Radiosity: The emissive flux per unit area, plus the irradiance times the diffuse surface reflectance.*

These definitions are due (in this form) to Ward and Heckbert[30].

Vector Irradiance and Light Field

A vector field is a function $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. The particular vector field used in these derivations is known as the light field. Assuming that the radiance at point r in direction ω is defined as $L(r, \omega)$ we define

$$\Phi(r) \equiv \int_{S^2} L(r, \omega)\omega d\sigma(\omega) \tag{7}$$

The vector irradiance is then defined as $\Phi(r)$ for each point r and is a vector quantity.

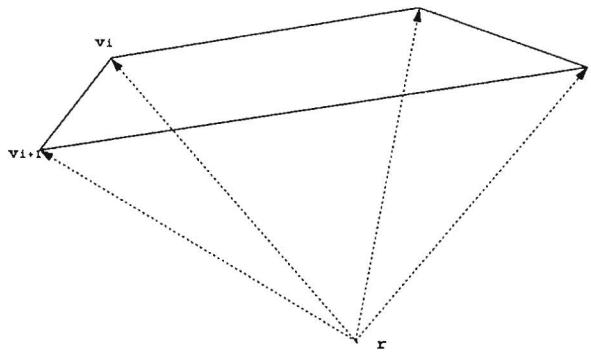


Figure 16: This figure shows the emitter and a receiving point r. The vertices are labeled as v_i.

Jacobian

A vector field derivative $D\Phi$ is a 3×3 Jacobian matrix defined as follows:

$$J_r(\Phi) = \begin{bmatrix} \frac{\partial \Phi_x(r)}{\partial x} & \frac{\partial \Phi_y(r)}{\partial x} & \frac{\partial \Phi_z(r)}{\partial x} \\ \frac{\partial \Phi_x(r)}{\partial y} & \frac{\partial \Phi_y(r)}{\partial y} & \frac{\partial \Phi_z(r)}{\partial y} \\ \frac{\partial \Phi_x(r)}{\partial z} & \frac{\partial \Phi_y(r)}{\partial z} & \frac{\partial \Phi_z(r)}{\partial z} \end{bmatrix} \tag{8}$$

Vertex Vectors

Vertex vectors are vectors from the receiving point r pointing in the direction of each of the vertices of the emitter as in Figure 16. In most cases the vertex vectors point to the vertices of the emitter after visibility information is computed — ie. the occluders are clipped from the emitter. These vertex vectors are used in many of the equations that follow. Setting the vertices of the emitter $v_i \in v_1..v_n$ we define the vertex vectors u_i as:

$$u_i = \frac{v_i - r}{\|v_i - r\|}$$

These u_i are normalized vectors from the receiving point pointing in the direction of the emitter vertex v_i.

4.2.2 Computing the Irradiance Jacobian

The irradiance at a point r on a surface is derived from Equation (7) giving

$$\phi(r) = -\Phi(r) \cdot \vec{n} \tag{9}$$

which defines the net flux at any point r in space - - equal to the irradiance at a point r on a surface. To derive the gradient of the irradiance over the surface of a patch in a radiosity simulation Equation 9 is differentiated as follows:

$$\phi = \vec{\Phi} \cdot \vec{n} \tag{10}$$

Deriving

$$\begin{aligned}
\nabla\phi &= \left(\frac{\partial B}{\partial x}\right)\vec{i} + \left(\frac{\partial B}{\partial y}\right)\vec{j} + \left(\frac{\partial B}{\partial z}\right)\vec{k} \\
&= \frac{\partial}{\partial x}(\vec{\Phi}\cdot\vec{n})\vec{i} + \frac{\partial}{\partial y}(\vec{\Phi}\cdot\vec{n})\vec{j} + \frac{\partial}{\partial z}(\vec{\Phi}\cdot\vec{n})\vec{k} \\
&= \frac{\partial\phi_x}{\partial x}n_x\vec{i} + \frac{\partial\phi_y}{\partial x}n_y\vec{i} + \frac{\partial\phi_z}{\partial x}n_z\vec{i} + \\
&\quad \frac{\partial\phi_x}{\partial y}n_x\vec{j} + \frac{\partial\phi_y}{\partial y}n_y\vec{j} + \frac{\partial\phi_z}{\partial y}n_z\vec{j} + \\
&\quad \frac{\partial\phi_x}{\partial z}n_x\vec{k} + \frac{\partial\phi_y}{\partial z}n_y\vec{k} + \frac{\partial\phi_z}{\partial z}n_z\vec{k} + \\
&= \vec{n}^T[J(\vec{\Phi})]
\end{aligned} \tag{11}$$

where J is the Jacobian matrix defined in Equation (8).

This assumes a constant $\vec{n}$ which is the case under the assumption that the object is planar. For nonplanar objects, the derivation is slightly more complex and results in an additional term in Equation 11. This additional term is dependent on how $\vec{n}(r)$ changes as a function of r and reduces to 0 for planar objects.

Arvo distinguishes between two types of emitter vertices — intrinsic and apparent (Figure 17). Intrinsic vertices are those that appear either as a vertex in the emitter or a vertex on one of the blockers i.e. they are actual vertices. An apparent vertex is not actually a vertex. It is the illusion of a vertex that appears due to the conjunction

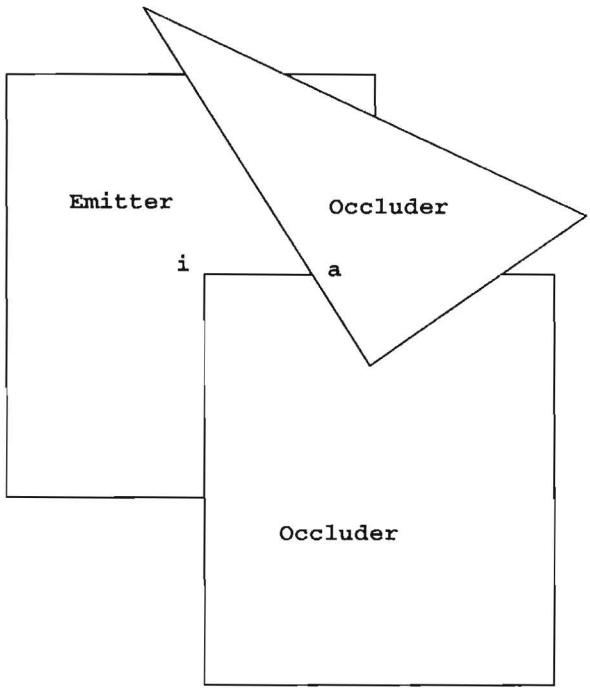


Figure 17: This illustrates a partially occluded emitter with an apparent vertex (a) and intrinsic vertex (i) labeled.

of an emitter edge and a blocker edge or two blocker edges as seen from a receiving point. Intrinsic vertices exist independent of the viewpoint, but apparent vertices may or may not appear depending on the viewpoint chosen. Two expressions are constructed to define the vertex Jacobian for both intrinsic and apparent vertices. These vertex Jacobians are then used to compute the irradiance Jacobian. According to Arvo the vertex Jacobian is the 3×3 matrix resulting from taking the derivative of the vertex vectors.

For apparent vertices we set $\vec{s}$ and $\vec{t}$ to be unit vectors parallel to the two lines (L_1 and L_2) defining the vertex as in Figure 18. Firstly we have

$$J(\vec{u})\vec{u} \;=\; 0$$

So $\vec{u}$ — which is any vertex vector — is an eigenvector of $J(\vec{u})$ with an eigenvalue of 0. This corresponds to the fact that the Jacobian doesn't change when the receiving point r is moved in the direction of $\vec{u}$.

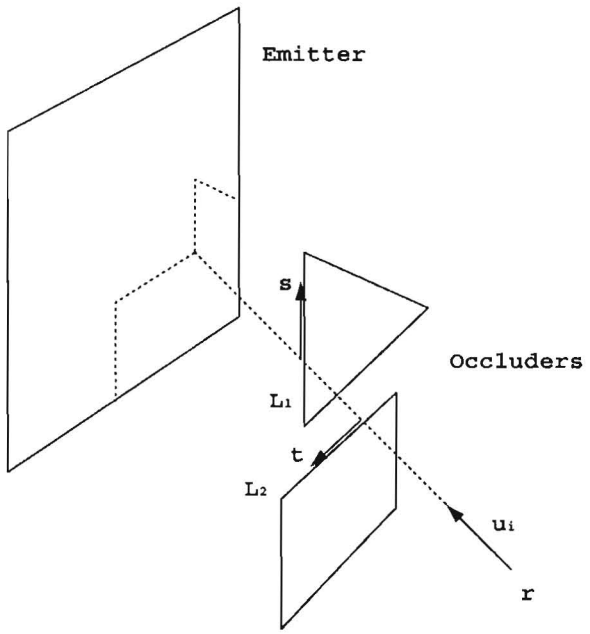


Figure 18: This is the same scene as in Figure 17 seen from a different angle. The vertex Jacobian is determined by the two vectors $\vec{s}$ and $\vec{t}$ and the vertex vector $\vec{u_i}$.

Then defining

$$\begin{aligned} \hat{s} &\equiv (I - \vec{u}\vec{u}^T)\vec{s} \\ \hat{t} &\equiv (I - \vec{u}\vec{u}^T)\vec{t} \end{aligned}$$

Set α_t to the distance to L_2 along $\vec{u}$ and similarly α_s the distance to L_1. Moving the receiving point in the direction of either $\hat{s}$ or $\hat{t}$ gives us the other two eigenvector and eigenvalue pairs.

$$J(\vec{u})\hat{s} = -\frac{\hat{s}}{\alpha_t} \tag{12}$$

$$J(\vec{u})\hat{t} = -\frac{\hat{t}}{\alpha_s} \tag{13}$$

Then using the three eigenvectors $\hat{s}$, $\hat{t}$ and $\vec{u}$ with their corresponding eigenvalues of $-\frac{1}{\alpha_t}$, $-\frac{1}{\alpha_s}$ and 0 we have

$$J(\vec{u}) = \begin{bmatrix} \hat{s} & \hat{t} & \vec{u} \end{bmatrix} \begin{bmatrix} -\frac{1}{\alpha_t} & 0 & 0 \\ 0 & -\frac{1}{\alpha_s} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{s} & \hat{t} & \vec{u} \end{bmatrix}^{-1} \tag{14}$$

Equation (14) can be found in both Arvo's[2] and Paulin's[26][1] papers and is an accurate method for computing the vertex Jacobian in all cases (provided the Jacobian exists).

The vertex Jacobians for the implicit vertices is a special case of Equation (14) with $\alpha_t = \alpha_s = \alpha$. Using the simplification, the vertex Jacobian for intrinsic vertices is as follows:

$$J(\vec{u}) = -\frac{1}{\alpha}(I - \vec{u}\vec{u}^T) \tag{15}$$

Setting

$$\begin{aligned} \vec{a} &= \vec{u_i} \\ \vec{b} &= \vec{u_{i+1}} \\ \vec{w} &= a \times b \end{aligned}$$

we define E as

$$E(\vec{a}, \vec{b}) = \left(\frac{1}{\vec{a}^T\vec{b}}\right)\frac{\vec{w}\vec{w}^T}{\vec{w}^T\vec{w}} + \frac{\cos^{-1}\vec{a}^T\vec{b}}{\|\vec{w}\|}\left(I - \frac{\vec{w}\vec{w}^T}{\vec{w}^T\vec{w}}\right)$$

where $E(\vec{a}, \vec{b})$ is the edge matrix defined by Arvo[2].

For notational convenience we set Q as

[1] We note that the version of Equation (14) in Paulin's paper [26] has a typographical error. They list part of it as $A = \begin{bmatrix} \hat{e}_1 & \hat{e}_1 & \vec{w} \end{bmatrix}$ and it should read $A = \begin{bmatrix} \hat{e}_2 & \hat{e}_1 & \vec{w} \end{bmatrix}$

$$
\begin{aligned}
Q(\vec{p})\vec{q} &= \vec{p} \times \vec{q} \\
Q(\vec{p}) &= \begin{pmatrix} 0 & p_z & -p_y \\ -p_z & 0 & p_x \\ p_y & -p_x & 0 \end{pmatrix} \\
\Rightarrow \vec{q} \times \vec{p} &= \vec{q}^T(Q(\vec{p}))
\end{aligned}
$$

The corner matrix C is defined as

$$C(a,b,c) \equiv E(a,b)Q(a) - E(b,c)Q(c) \tag{16}$$

Using the corner matrix and the equation for the Jacobian we have

$$J(\vec{\Phi}) = \frac{M}{2\pi} \sum_{i=1}^{m} C(\vec{u_{i-1}}, \vec{u_i}, \vec{u_{i+1}}) J(\vec{u_i})$$

From Equation (11) we have

$$\nabla\phi = \vec{n}^T[J(\vec{\Phi})]$$

Substituting gives

$$
\begin{aligned}
\nabla\phi &= \frac{M}{2\pi} \sum_{i=1}^{m} \vec{n}^T C(\vec{u_{i-1}}, \vec{u_i}, \vec{u_{i+1}}) J(\vec{u_i}) \\
&= \frac{M}{2\pi} \sum_{i=1}^{m} \left[\vec{n}^T E(\vec{u_{i-1}}, \vec{u_i}) Q(\vec{u_{i-1}}) - \vec{n}^T E(\vec{u_i}, \vec{u_{i+1}}) Q(\vec{u_{i+1}}) \right] J(\vec{u_i}) \\
&= \frac{M}{2\pi} \sum_{i=1}^{m} \left[(\vec{n}^T E(\vec{u_{i-1}}, \vec{u_i})) \times \vec{u_{i-1}} - (\vec{n}^T E(\vec{u_i}, \vec{u_{i+1}})) \times \vec{u_{i+1}} \right] J(\vec{u_i})
\end{aligned}
$$

As suggested by Arvo, a substantial optimization is achieved by multiplying out by the normal vector $\vec{n}$. This eliminates the need for matrix mathematics except for the

computation of the vertex Jacobians. As this results in a substantial computation saving we show the multiplication in full.

$$\begin{aligned}
\vec{n}^T E(\vec{a},\vec{b}) &= \vec{n}^T \left(\frac{1}{\vec{a}^T\vec{b}}\right)\frac{\vec{w}\vec{w}^T}{\vec{w}^T\vec{w}} + \vec{n}^T\frac{\cos^{-1}\vec{a}^T\vec{b}}{\|\vec{w}\|}\left(I - \frac{\vec{w}\vec{w}^T}{\vec{w}^T\vec{w}}\right) \\
&= \left(\frac{1}{\vec{a}^T\vec{b}}\right)\frac{\vec{n}^T\vec{w}\vec{w}^T}{\vec{w}^T\vec{w}} + \frac{\cos^{-1}\vec{a}^T\vec{b}}{\|\vec{w}\|}\left(\vec{n}^T - \frac{\vec{n}^T\vec{w}\vec{w}^T}{\vec{w}^T\vec{w}}\right) \\
&= \left(\frac{1}{\vec{a}\cdot\vec{b}}\right)\frac{(\vec{n}\cdot\vec{w})\vec{w}}{\vec{w}\cdot\vec{w}} + \frac{\cos^{-1}\vec{a}\cdot\vec{b}}{\|\vec{w}\|}\left(\vec{n} - \frac{(\vec{n}\cdot\vec{w})\vec{w}}{\vec{w}\cdot\vec{w}}\right) \qquad (17)
\end{aligned}$$

Pseudo Code

In the pseudo code that follows, all subscripts are computed with modulo arithmetic so that the vertex indices wrap around e.g. $\vec{u_m} \equiv \vec{u_0}$.

We point out that the computation figures listed in Paulin's paper are for the full computation of the corner matrix at each vertex without the optimization described above. This full computation requires a costly recomputation of the edge matrix — the corner matrix requires the edge matrix data for both contributing edges. We propose the fairly simple optimization of computing all the edge matrices (with the normal already multiplied out) once for each edge and storing the values. This optimization is implied by Arvo in his description of the algorithm. The corner matrices are then computed by looking up the edge matrix values resulting in a fairly substantial computation saving ($\approx 50\%$).

For all vertices i do {
    //Compute edge matrix from Equation 17
    $E_i \leftarrow \vec{n}^T E(\vec{u_i}, \vec{u_{i+1}})$
    //Compute vertex Jacobian
    $J_i \leftarrow J(\vec{u_i})$
}

$\vec{G} \leftarrow 0$
For all vertices i do {
    //Use corner matrix to compute gradients
    $\vec{G} += [E_i \times \vec{u_i} - E_{i+1} \times \vec{u_{i+2}}]\, J_{i+1}$
}
$\vec{G} \leftarrow \frac{M}{2\pi}\vec{G}$

Another fairly obvious optimization can be made by noting that in the computation of the gradient, we have computed most of the terms needed to compute the radiosity value. Due to this fact, if the radiosity is combined with the gradient calculations, we have the radiosity value almost for free — see Holzschuch[18] for details.

4.3 Calculating Visibility Information

Calculating the gradient requires the distance from the receiving point r to each of the vertices (apparent or intrinsic) appearing in the depth clipped emitter (Figure 18). Calculating this depth clipped polygon is equivalent to computing the visibility information between point r and the emitter with the provision that the 3D information is preserved.

One of the most time consuming tasks for a rendering algorithm is that of computing visibility information. It is still a research issue to efficiently compute visibility between two objects in a scene. The naive or brute force methods compare every

object in the scene with the two objects under consideration for the interaction and compute the extent of the occlusion that results. This algorithm is $O(n^3)$ in the number of objects in the scene. There are many more efficient algorithms to handle this problem. Most of these algorithms operate by maintaining lists of potential occluders for each pair of objects and culling obviously non-occluding objects from the list. This system is usually $O(n^2)$ in time and in storage. Several optimizations on this system allow for ignoring pairs of objects that cannot interact - - for example two objects in different rooms (clustering) or two objects with no occluders. Other optimizations provide more efficient and accurate culling algorithms for reducing the size of the potential occluder lists. One such algorithm is known as shaft culling [15] which culls patches in a radiosity simulation based on a shaft generated between the two interacting patches.

Once the rendering algorithm has determined that an object is occluding an interaction between two other objects, it needs to determine the amount of light that actually arrives at the receiving object. Some algorithms cast some rays between the two objects and use the ratio of unblocked to blocked rays to determine how much of the emitter's light arrives at the receiver [16]. Some systems project the occluder edges onto the receiver from the points of the emitter and compute the discontinuity mesh, splitting the receiver into portions that are fully visible, shadowed or partially visible [23].

In this dissertation we use the system of backprojection and depth clipping proposed by Arvo for his gradient computation. It turns out that precise computation of the irradiance Jacobian or radiosity gradient requires an accurate clipped contour of the visible portion of the emitter and the distance from the receiving point to each vertex on the clipped emitter. This requires a 3D clipping algorithm or a projection of the occluders to the emitter from the receiving point, 2D planar clipping algorithm and then a reverse projection back to 3D.

We use a variation of the Weiler/Atherton clipping algorithm due to Alexey Nikitin and Michael Leonov[24][2]. This algorithm is a modification of the Klammer Schutte

[2]http://woland.afti.nsu.ru/ leonov/clipdoc.html

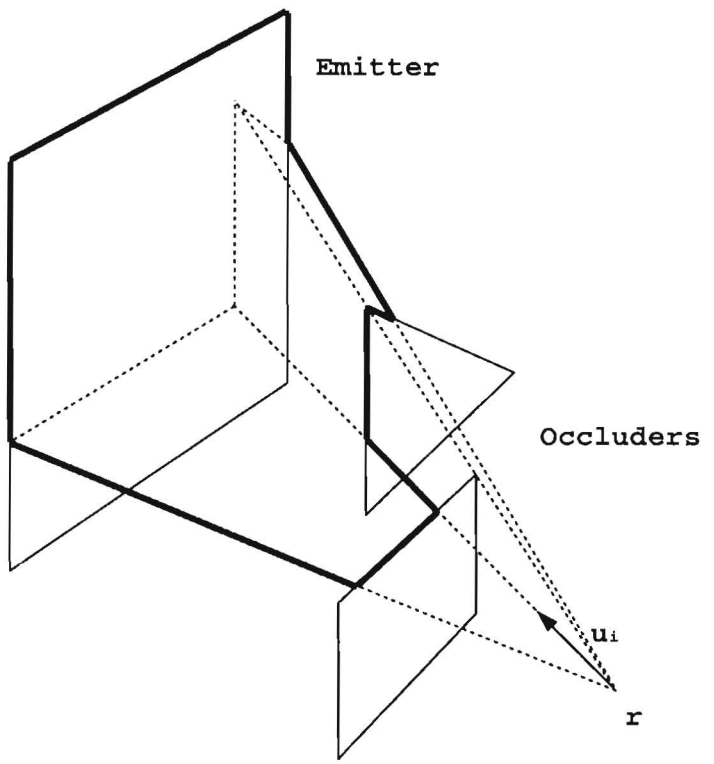


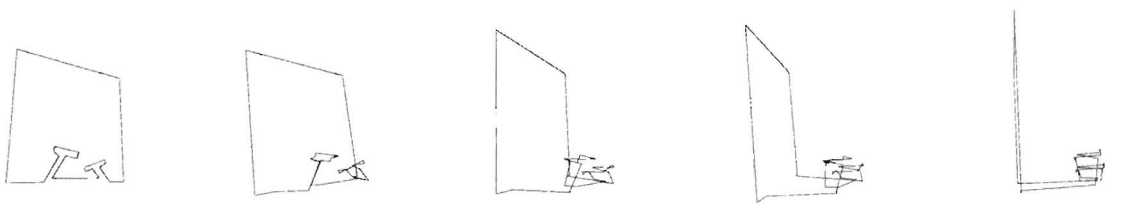
Figure 19: The bold contour illustrates the depth clipped polygon — clipping retaining the 3D information required for the gradient equations.

Clippoly program[3]. Our modified version of this algorithm clips projections of the occluders from the emitter in the emitter plane and maintains the 3D information for each vertex so that the vertices can be reverse projected back into 3D as shown in Figure 19.

Figure 20 illustrates five frames of a rotating model representing a depth-clipped polygon. The depth clipped polygon is that of the emitter (wall) as seen from a point on the floor. The occluder is a kneel chair on the floor obscuring some of the light transport from the wall to the receiving point on the floor.

The radiosity function is dependent on the depth clipped polygon of the resultant emitter and occluders. It can be seen in this figure that the radiosity can be a fairly

[3]Copyright of GNU



a b c d e

Figure 20: This figure shows a series from a rotation of a depth clipped emitter polygon from a kneel chair casting a shadow on a wall. The view in (a) is the shape of the emitter as seen from approximately the receiving point.

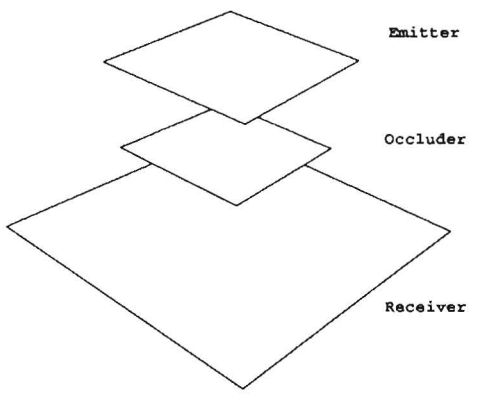


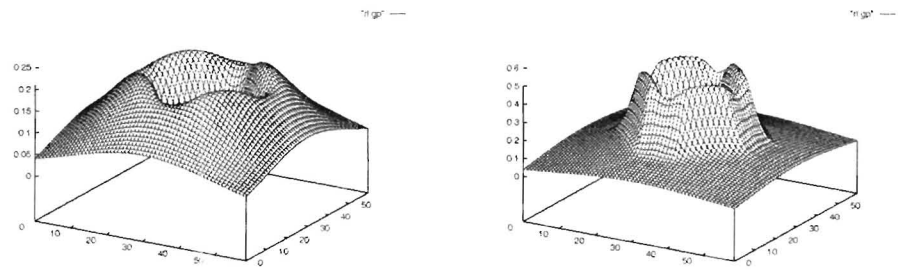
Figure 21: This is the scene used to generate the graphs that follow.

complex function. for even simple "real-world" models, exhibiting similarly complex behavior in the gradient.

4.3.1 Measured Gradient and Discontinuities

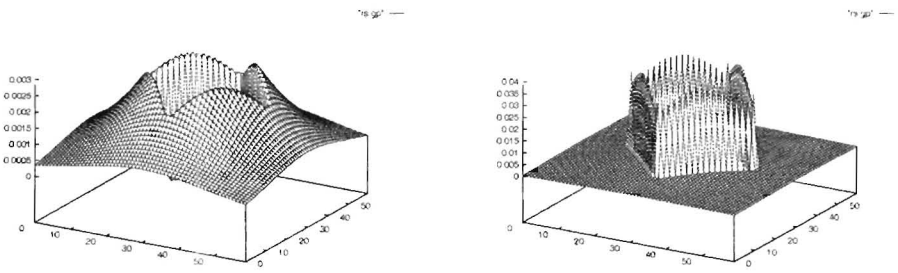
Graphs

In all the scenes the occluder, emitter and receiver are in parallel planes as shown in Figure 21. We tested two planar rotations of the occluder - - first with its edges parallel to the emitter as in Figure 21 and second with the edges at 45° to the edges of the emitter. For both of these cases we tested two different sizes of the emitter. Each of the four figures is presented with a part (a) which shows the graph of the computed form factor and a part (b) which shows the norm of the gradient.



a b

Figure 22: 45° occluder with a large emitter. Note that the shadow causes a depression in the form factor plot (left). Also note the vertical creases in the gradient plot (right) which indicate second derivative discontinuities.

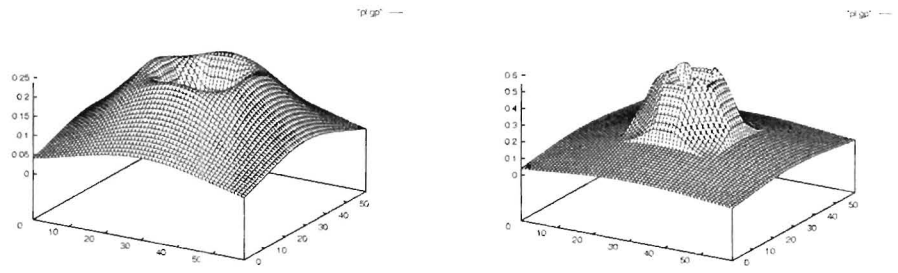


a b

Figure 23: 45° occluder with a small emitter. Note now sharp the edges of the shadow depression are compared to the previous set of graphs.

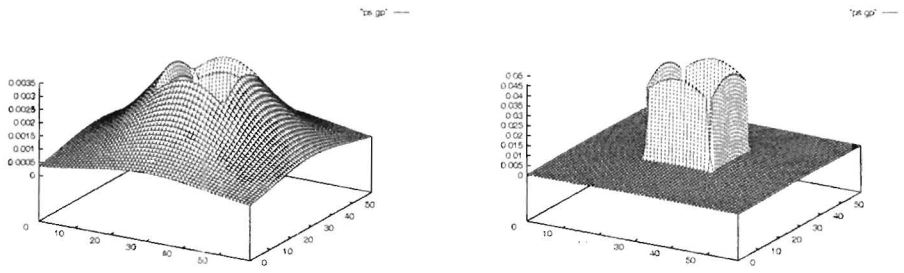
For the first scene, the occluder is rotated 45° to the emitter. In Figure 22a we see the plot of the form factors resulting from this scene. Its companion Figure 22b shows the norm of the computed gradients for the same set of points and scene. We note that while the form factor graph is fairly smooth, the gradient exhibits some discernible creases. These are lines of second order discontinuities in the resulting function.

The second scene illustrates the same configuration as the first — the emitter and occluder are rotated 45° to each other, however, in this scene the emitter is much smaller (and brighter) than in the first scene. This scene is shown in Figure 23. As can be seen by comparing the two figures, this scene has much more defined edges and a much sharper gradient. This is the result of having a much smaller penumbra



a b

Figure 24: Aligned emitter and occluder with a large emitter. Note that the alignment of the emitter and occluder causes the gradient (right) to exhibit discontinuities. These are seen as creases in the form factor plot (left).



a b

Figure 25: Aligned emitter and occluder with a small emitter. Note that the discontinuities in the gradient (right) are much more obvious than in the previous pair of graphs.

as the smaller emitter is a closer approximation to a point light source.

In the third scene the emitter and occluder are aligned (edges are parallel). In the Figure 24a the graph appears to be fairly smooth and intuitively correct. The Figure 24b, however we see that the gradient shows some fairly pronounced creases and even some discontinuity lines.

In the fourth scene the emitter and occluder are in the same configuration as the third, but again the emitter is substantially reduced in size. This results in very sharp gradient values with obvious discontinuity lines.

Discontinuities

Note that several discontinuities of varying order are visible in the Figures 22-25. These discontinuities are particularly noticeable in the gradient plots such as Figure 24b. The D1 discontinuities - discontinuities in the gradient plot or creases in the radiosity plot — are the result of an emitter edge aligning with the occluder edge. This causes a sudden change in the behavior of the radiosity and hence a gradient discontinuity. This is most visible in Figure 25b. D2 discontinuities arise due to the conjunction of an occluder edge with an emitter vertex or an occluder vertex and emitter edge. This results in a more subtle change in the radiosity giving rise to creases in the gradient plot. These are most visible in figure 22b and 24b. Note that 24b also has some fairly visible D1 discontinuities.

4.4 Summary

In this chapter, we show how to compute the irradiance and radiosity gradients required for the prediction of edges in a radiosity scene. The derivations for the computation of the gradient are primarily due to James Arvo and Nicolas Holzschuch. We also point out some fairly useful optimizations we used for these computations.

We show how the visibility information is computed using a general depth clipping algorithm. This depth clipping is essential to the computation of the gradients.

Finally we show some measured gradients for a simple class of environments. The gradient plots are contrasted to each other and to their corresponding form factor (radiosity) plot. We also point out how discontinuities arise in the gradient graphs.

Chapter 5

Preliminary Exploration

5.1 Introduction

This chapter describes the initial experimentation with perceptual refinement ideas. These experiments were in the nature of a feasibility study to determine if there was anything worth exploring and if so what directions were likely to be fruitful.

The experiment proposes a few heuristics for determining "areas of interest". These areas are those patches with enough detail to warrant extra refinement. The critical factor being that if a patch containing an area of interest is not refined, the visual quality of the image will suffer in a way that is much more noticeable than for a patch containing no areas of interest.

5.2 Description of the Algorithm

5.2.1 Motivation

Our refinement method is designed to increase the refinement process in areas that are visually important such as highlights, contours and shadow boundaries. Our hypothesis is that the eye is more sensitive to contours and boundaries than it is sensitive

to differences in lighting and colors. In order to detect such contours and boundaries, we suggest using differences in the radiosity derivatives, instead of differences in the radiosity values, as suggested by Lischinski [22].

We note that large but constant radiosity gradient values over a region indicate a linearly varying radiosity which is modeled very well by the standard bi-linear interpolation methods common in most radiosity programs.

On the other hand, a large difference in the radiosity gradient, whether a difference in orientation or a difference in norm indicates an area where the radiosity function is not modeled properly by the standard linear interpolation. Hence, it makes sense to refine such areas.

This idea is consistent with previous work on perceptual reproduction and images by Ashdown [3].

5.2.2 Refinement Method

In this section, we present our algorithm for computing the perceptual error. We single out an interaction between a given emitter and a given receiver, and we measure the visual effect of this interaction on the final output. If the visual error due to this interaction is above a certain threshold, then the interaction will be refined.

We start by computing the point-to-area form-factor values and its gradient at the vertices of the receiver, using the method presented by Arvo and Holzschuch in Chapter 4.

Projection of the Radiosity Gradient

The gradient computation gives a vector in 3D-space which needs to be projected to the plane of the receiver.

$$pr(\vec{G}) = \vec{G'} = \vec{G} - \vec{N}(\vec{G} \cdot \vec{N}) \tag{18}$$

Equation 18 projects $\vec{G}$, the gradient in 3-space, to a plane parallel to the receiver defined by its normal $\vec{N}$.

Perceptual Error Measurement

To compute the perceptual error for a given interaction, we simply compute the difference of the point-to-area form-factor gradients at the vertices of the receiver:

$$\Delta G = \max_{i \neq j} \left\{ \| pr(\vec{G}_i) - pr(\vec{G}_j) \| \right\} \tag{19}$$

where $\vec{G}_i$ is the form-factor gradient at the vertex i of the receiver. This Equation is simply the maximum norm of the radiosity gradients at each vertex of the patch. Each gradient must be projected to the plane of the patch under consideration.

We then use this gradient difference to define the refinement oracle using the radiosity of the emitter, $\vec{B}_E$, the reflectivity of the receiver, $\vec{\rho_R}$, and the area of the receiver, A_R:

$$\vec{r} = \Delta G \vec{B}_E A_R \vec{\rho_R} \tag{20}$$

We define the refinement criterion R_{cl} as the brightness of the resulting $\vec{r}$ vector. The refinement criterion is compared to the refinement threshold. If it is larger than the threshold then this interaction (either the receiver or emitter) requires refinement. This constitutes the refinement oracle.

The point-to-area form-factor gradients are computed for each interaction and are stored for the duration of the refinement process, so that we can reuse the values of the form-factor gradient for patches sharing common vertices. This optimization reduces the time spent computing the gradients.

5.2.3 Error and Decision

Once it has been determined that the interaction requires refinement, we need to determine which of the receiver or emitter requires the refinement. It is often the case that both the receiver and emitter require refinement, but only the one that

requires it the most is refined and the other is refined in a subsequent iteration of the refinement algorithm.

We first compute the change in the radiosity derivatives on the receiver due to changes in radiosity values on the emitter:

$$E_E = (B_{\max} - B_{\min})G_{\max} \tag{21}$$

where $G_{\max}$ is the maximum norm of the form-factor gradient on the vertices of the receiver, and $B_{\max}$ and $B_{\min}$ are the maximum and minimum radiosity on the emitter.

We then compute the changes in the radiosity derivatives on the receiver due to changes in the point-to-area form-factor gradient:

$$E_R = \Delta G B_{\max} \tag{22}$$

If $E_R \geq E_E$ then the receiver is refined, otherwise the emitter is refined.

5.3 Results

We ran the radiosity simulation contrasting the standard error-control refinement method described by Lischinski [22] with our perception-based refinement method.

We produced a reference image with our refinement method (Figure 26b). The refinement threshold was deliberately chosen to produce lower levels of refinement so that the difference between the pictures would be more obvious. We then ran the standard error-control refinement criterion with the threshold adjusted to produce an output mesh that is within 0.09% of the size of our reference image (Figure 26a). We also adjusted the threshold to produce a picture with the standard error-control refinement which took the same computation time as our reference image — to within 0.03%. In both cases we find that the image produced with our refinement method has a noticeably better visual quality.

It is important to note that if the standard error-control refinement criterion and our perception-based refinement criterion are used to produce similar visual quality

output, our refinement method would not only produce smaller output, but would also take less time.

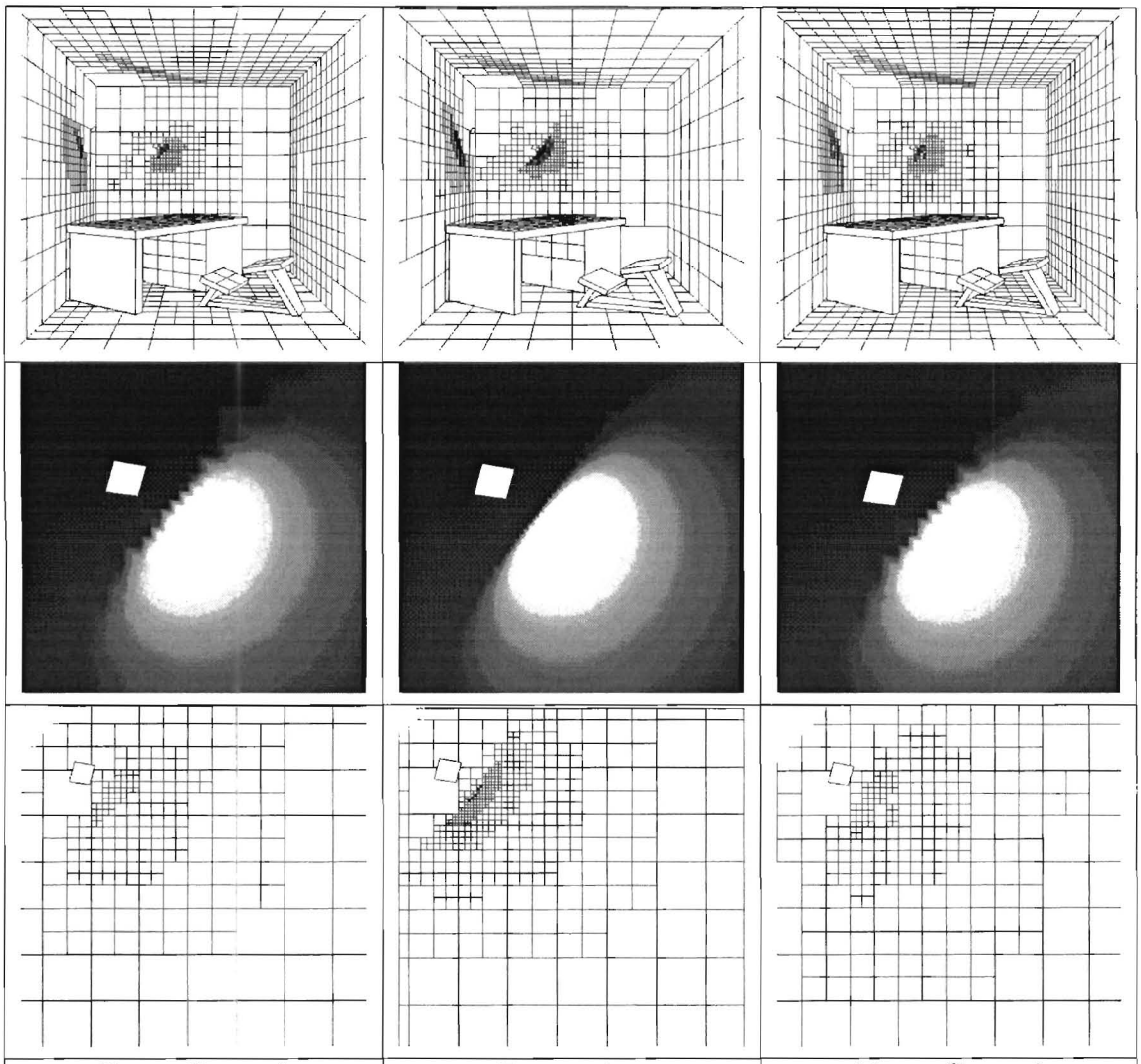
The sizes quoted in Figure 26 are the output mesh sizes (in KBytes) and are proportional to the number of patches generated by the relevant refinement method. The times quoted are average execution times for each scene on a 132MHz Indy with a MIPS R4600 CPU. The results are displayed for the first iteration of the hierarchical radiosity algorithm.

The refinement grid patterns shown in Figure 26 shows how our refinement method (26b) produces deeper levels of refinement for the chosen area. The chosen area being one which contains visually important effects — in this case a highlight on a wall. Our refinement method produces less refinement in areas with constant or linear radiosity as is evident by comparing the walls of the overall mesh images in the top row of Figure 26. In column (a) of Figure 26 we have the standard algorithm constrained to the same output mesh size as our refinement method (b) and in column (c) the standard algorithm is constrained to have the same rendering time compared to our method.

5.4 Conclusions and Implications

In this chapter we have presented a refinement oracle based on our intuitive notion that visually important regions are indicated by large gradient difference. This refinement oracle makes use of some simple heuristics involving the radiosity gradient to determine the areas of interest or areas that are visually important. The heuristics then concentrate more of the cpu time on visually important regions.

This initial experiment produced some modest savings in both time and number of output polygons. The most surprising result, however, was the improvement in the visual quality over standard refinement methods as shown in Figure 26. These results suggest that perceptual refinement methods can be constructed using the radiosity gradient that would perform in reasonable time/space constraints (similar to existing algorithms) but producing substantially better looking output.



Error-Control (equal size)	Perception	Error-Control (equal time)
500Kl	500Kb	634Kb
2.51s	3.04s	3.04s
a	b	c

Figure 26: Comparison of our perception-based refinement oracle with the standard error-control refinement oracle, showing the complete mesh (top), and a close-up on a highlight (middle) (with the mesh(bottom)), with the size of the output and the execution time.

While these results may sound impressive, the refinement oracle presented in this chapter is incomplete which makes it unsuitable for a final perception based algorithm. First and foremost is that no visibility computation is attempted. This means that the algorithm was not tested on shadow boundary conditions. Shadow boundaries contribute some very visually important information to the scene, so any serious perceptually based refinement criterion must be able to handle them efficiently.

Secondly the heuristics that the algorithm uses are chosen because they produce results, rather than being based on some theory of perception. The aim of the perception based refinement criterion is not to be accurate according to some theory on perception or physics, but to produce results with a better visual quality in a reasonable time. It would, however, be useful to base the heuristics on some model of human perception as the extra insight should promote even better visual importance prediction.

Another problem is that the heuristics for determining which of the emitter or receiver needs to be refined were found to be incorrect and actually produce unnecessary refinement on some patches (equations 21 and 22) .

Even considering this incompleteness, the positive results certainly suggest the potential for an effective perception based refinement oracle for radiosity algorithms. These results also strongly suggest that the use of radiosity gradients should be fundamental to any perception based refinement method as part of the prediction of visually important regions.

Chapter 6

Theory

6.1 Introduction

Firstly we introduce the visual importance function. This function is based on our HVS model and an intuitive notion of areas of importance.

Due to the abstract nature of the visual importance function, we present a heuristic approximation and develop the refinement oracle based on this. This perceptual refinement oracle uses the radiosity and radiosity gradient computed at each sample point on the patch to determine if the patch needs refinement. We also include a coefficient that optimizes refinement for use with a bilinear interpolation based rendering program.

6.2 Visual Importance Function

The visual importance function is a function that associates each point on a patch with a number representing the relative importance of rendering this point to the visual quality of the scene. The visual importance function is then a surface over the patch having large values associated with areas deemed important by the HVS (for instance edges or shadow boundaries), and correspondingly lower values in other

less important regions. The visual importance function then mimics the importance associated with each visual effect over a patch by the HVS. The use of the visual importance function is in increasing the *quality* of the resulting image. Thus a visual importance function has a corresponding image quality metric associated with it. In this section we define the visual importance function and derive an effective image quality metric.

6.2.1 Definition and Dependence

We define the *visual importance function* V as follows:

$$V(X) \quad : \quad P \rightarrow (0, 1] \tag{23}$$

Where the domain P is the set of all points on the surface of the patch in question.

This equation associates each point on the patch with a scalar quantity which we call *visual importance*. This quantity is proportional to the amount which the point X effects the final output visual quality. It is not possible for a point X to have a value of 0 for its visual importance as every point, even if completely in shadow, contributes to the visual quality of the scene.

Intuitively, the visual importance associated with a point X is dependent on the other points in the neighbourhood of X. The visual importance is dependent on the intensity function over the surface of the patch and hence on the radiosity at each point. Because of this, the visual importance function must be dependent on all the patches in the scene. We note that this global dependence of the visual importance function makes it useless for direct application to refinement oracles even if an exact formulation could be found (an exact formulation would require complete knowledge of the HVS and complete foreknowledge of the solution). Instead we use the visual importance function as a guide in developing a heuristic that estimates the visual importance in a region based on only a few sample points.

6.2.2 Areas of Interest

We define *areas of interest* or *visually important areas* as those areas on the surface of the patch containing a local maximum or an otherwise large value of the visual importance function V.

As noted in Section 3.3.2 edges are important to visual quality and hence our visual importance function should have maxima associated with shadow boundaries and other rapidly changing intensity phenomena such as highlights. Thus our approximate formulation of the visual importance function would be similar to passing the entire region's intensity image through an edge detection filter, however such an image is not available during the application of the refinement oracle since it has not yet been computed.

Larger values of the visual importance function in a region imply that more refinement in the region will have a larger positive impact on the final visual quality than in other areas. Similarly less refinement will have a proportionally large negative effect.

The amount of refinement in an area (sample point density) should be proportional to the visual importance to get the best visual quality for the smallest number of sample points and hence the smallest amount of computation.

6.2.3 Error Measure for Image Quality

Having determined that edges are of significant importance to image quality (Section 3.3.2), we wish to have a measure of image quality that incorporates the edge information as the most significant contributor.

Radiosity algorithms output view independent model and colour information. In order to compare radiosity refinement due to the two oracles we set all the other variables to be constant. We produce radiosity output for exactly the same model and then take shapshots of the view independent solution using exactly the same viewing parameters, i.e. the initial colours and object placement in the comparison images are the same and only the refinement level is allowed to vary. In this way we can isolate only the refinement information for measurement by the metric.

In addition to isolating the refinement information, we wish to measure how accurately the oracle represents edges for a given level of refinement. We reiterate that accurate representation of edges is essential to correct perception of an image (Section 3.3.2). The first step in computing the error measure for a picture is to pass it (and an ideal reference image) through an edge detection filter. All the pictures in the following discussion are filtered with an edge detection filter.

Given a picture P we define P_i as pixel i in P following the standard lexicographical ordering (left to right, top to bottom). As our metric requires a single value associated with each pixel, we used the brightness of the pixel to be the value of P_i.

We define the distance d_v between two pictures P and Q to be

$$d_v(P,Q) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - Q_i)^2} \tag{24}$$

where n is the number of pixels in a picture — note that the pictures' dimensions must be equal.

This distance measure is simply the root mean square of the difference between the two pictures taken pixel by pixel. We define $||P||_R$ to be $d_v(R,P)$ where R is the reference picture. The reference picture R is typically considered the ideal picture and is usually a radiosity scene computed to excessively high levels of refinement. For convenience we will refer to $||P||$ and leave the subscript as understood.

We use $||P||$ to represent the error in visual quality of the image. This value represents how far (in our metric) the picture P is from the ideal picture. Obviously as P gets closer to R, the value of $||P||$ gets closer to 0. This quantity, $||P||$, represents the error, or the difference from the ideal image. In this way it is similar in nature to the benefit heuristics used in level of detail computations[9].

The use of an edge detection filter on a picture often leaves the picture with low amplitude noise in areas that have no edges. This is mostly due to the edge detection filter picking up boundaries between intensity changes in otherwise uniform fields. To a certain extent this can be fixed by choosing an edge detection filter with a larger radius, but in practice the RMS calculation is sufficient to minimize the low amplitude noise.

As this metric requires snapshots of a computed radiosity scene, it is not very useful for radiosity computation. It does however give us a consistent means of comparing the visual quality of output scenes. Indeed, above a modest level of refinement, two pictures with the same error in visual quality ($||P|| = ||R||$) are visually indistinguishable.

6.3 Perceptual Refinement Oracle

The visual importance function (23) is very abstract and serves only to guide our construction of an appropriate refinement oracle. Note that (from Section 6.2.1) the visual importance function V is dependent on the knowledge of the complete intensity function over the patch. This implies that it would be impractical to apply the visual importance function directly in the computation of the refinement oracle — the oracle is used to guide the computation of radiosity sample points and it would be counterproductive to require foreknowledge of the result (intensity function). For this reason, we develop a heuristic for the visual importance function that estimates the level of visual importance over a patch based on the radiosity and gradients computed at a small number of sample points.

6.3.1 Visual Importance Heuristic

Each patch radiosity function is sampled at five points -- indicated in Figure 27. These points coincide with the vertices and the center of the patch. The radiosity function is a three dimensional surface over the domain of the patch area. At each sample point indicated in Figure 27 the radiosity and the (vector) gradient values are computed. The primary use for the refinement oracle is as an indicator of whether there are enough sample points to reconstruct the radiosity function with a reasonable level of accuracy. This level of accuracy is set by the user defined refinement threshold.

In Figure 29 we show how the refinement criterion is computed. A line is taken through the patch between two radiosity sampling points (shown in Figure 28 and

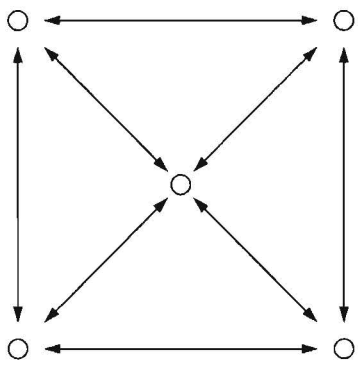


Figure 27: The small circles indicate the points where the patch radiosity function is sampled. The double headed arrows indicate the line segments along which the radiosity function is examined.

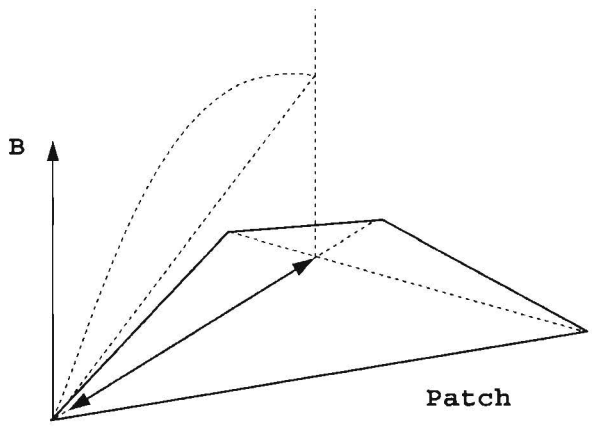


Figure 28: This figure shows the line through the patch and the associated radiosity graph.

27). At the two end points of the line (sample points) we have both the computed value of the radiosity and the gradient of the radiosity function. To find the gradient to the radiosity curve in the two dimensional graph in Figure 29 we need to project the vector gradient computed for that sample point to the plane through the line segment perpendicular to the patch.

The scalar gradient (G_i) at sample point i in the direction of sample point j is defined as

$$G_i = \left(\vec{\nabla\phi_i} - \left(\vec{N} \cdot \vec{\nabla\phi_i}\right)\vec{N}\right) \cdot \left(\frac{j-i}{\|j-i\|}\right) \tag{25}$$

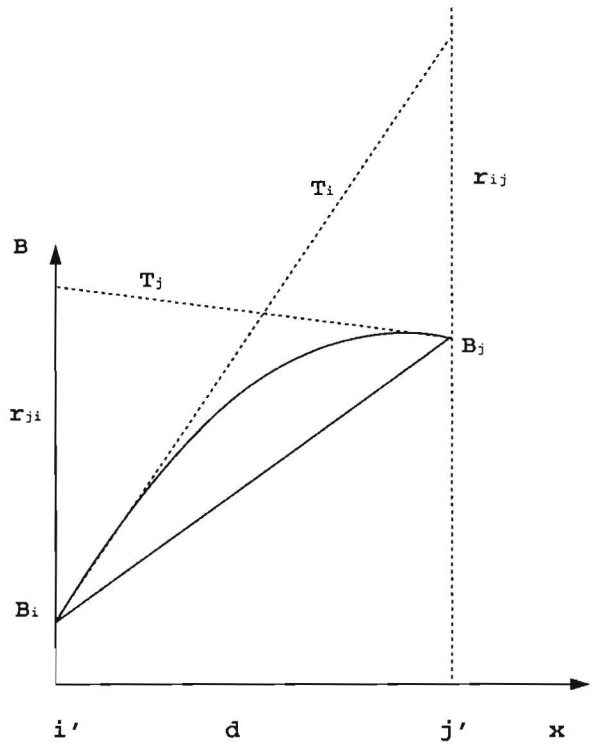


Figure 29: Computation of the refinement criterion. The x axis is the spatial dimension in the plane of the patch. The y axis in the size of the radiosity on the patch at each point along the line cut through the patch.

where $\nabla\vec{\phi}_i$ is the vector gradient at point i and $\vec{N}$ is the normal to the patch. This (Equation 25) is simply the projection of the vector gradient $\nabla\vec{\phi}_i$ to the relevant plane. See Chapter 4 for the computation of $\nabla\phi_i$ (Equation 11).

As the patch is going to be reconstructed using bilinear interpolation, this will be equivalent to a linear radiosity function on this line segment. The refinement criterion is a measure of how closely the linear radiosity matches the actual radiosity. The only information we have about the actual radiosity function behavior on this line is what we can deduce from the gradient at the two end points.

In Figure 29 the gradient computed at point i makes a tangent, labeled T_i, to the radiosity curve at point (i', B_i). This tangent T_i makes an intercept with the line $x = j'$ at a point $(j', B_j + r_{ij})$. The distance r_{ij} is an indication of the behavior of the radiosity curve near (i', B_i). The closer r_{ij} is to 0, the closer to linear the curve

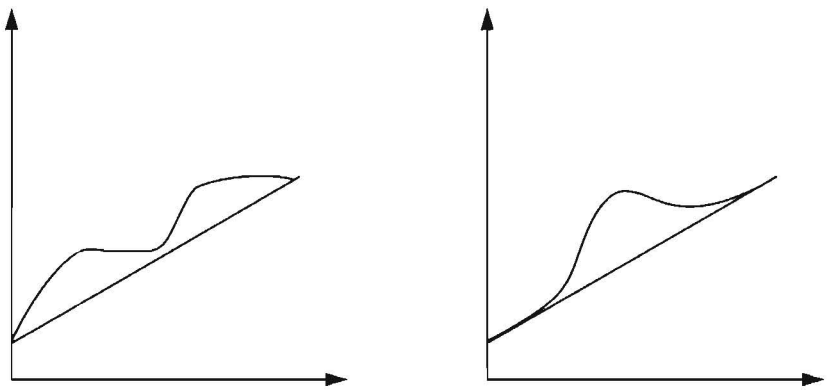


Figure 30: This figure shows two cases where the endpoint gradients don't predict the behavior of the radiosity. The left hand plot shows the effect of a small shadow and the right hand plot of a small highlight.

is at that point. Similarly the displacement r_{ji} is computed from the tangent line T_j and is an indication of the behavior of the radiosity graph near the point (j', B_j). These r values are also distance dependent which implies that we tolerate less error on larger patches. While this is a good indicator of the linear nature of the gradient over the area under consideration, it is by no means accurate. There are any number of cases — particularly involving shadows — where the gradient values at the end points of the line are not a very good indicator of the radiosity behavior over the entire area. For example, in Figure 30, the left graph is an example of a shadow affecting the radiosity on a scale smaller than the sample granularity. This effect is not detected by the gradients computed at the end points. Also the right hand image shows an apparently linear radiosity - - if the end point gradients are examined. The radiosity actually contains a highlight in the center, again, on a scale smaller than the distance between sample points. As these phenomena are not sampled at the sample point density used, aliasing occurs when the radiosity is reconstructed and these phenomena are not present.

If the gradient of the tangent T_i is G_i and similarly the gradient of T_j is G_j then

$$\begin{aligned} r_{ji} &= |-G_j d + B_j - B_i| \\ r_{ij} &= |G_i d + B_i - B_j| \end{aligned} \tag{26}$$

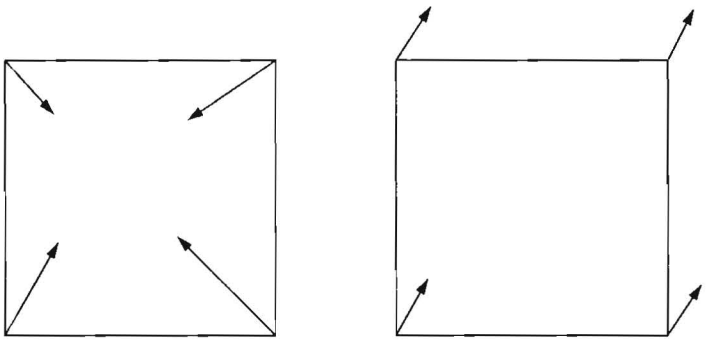


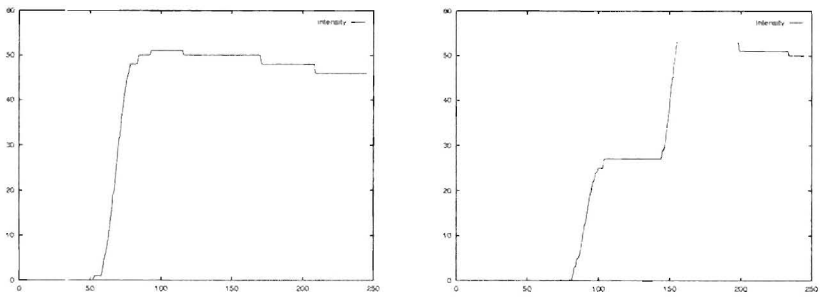
Figure 31: The left patch shows gradient vectors for a highlight (radiosity local maximum) contained within the patch. The right patch shows linear radiosity over the patch with the vector gradients pointing towards the maximum radiosity outside the patch.

6.3.2 Linearity Coefficient

In addition to the calculation of the tangents above, we also use a linearity coefficient, k, to roughly determine the overall linearity of the patch radiosity by using the dot product of the gradient vectors to determine how much they "line up" — i.e. the degree of collinearity of the vector gradients. The linearity coefficient is where we use knowledge of the nature of the reconstruction algorithm (See Section 3.6) in determining the refinement.

$$k = \frac{1}{2}(1 - \nabla\varphi_i \cdot \nabla\phi_j) \tag{27}$$

The linearity coefficient, Equation 27, simply scales the dot product of a pair of vector gradients ($\nabla\phi$) to the range $[0, 1]$ and reverses the order so that a k value of 0 implies the two vector gradients are identical, and a value of 1 implies vector gradients pointing in opposite directions. The vector gradients sampled over the patch are taken pairwise and the largest value (indicating the worst divergence from linear radiosity) is used as a coefficient in computing the the refinement criterion. Two example configurations of patch vertex vector gradients are shown in Figure 31.



a b

Figure 32: The intensity plot along two lines across a shadow boundary in a radiosity scene.

6.3.3 Perceptual Refinement Formulation

The complete refinement oracle is then governed by

$$R_{cp} = k \cdot \max\{r_{ij}, \forall i, j \text{ on patch } P\} \tag{28}$$

where the i and j are vertex endpoints of the line segments depicted in Figure 27, k is defined in Equation 27 and r_{ij} is defined by Equation 26. This R_{cp} is the perceptual refinement criterion and we will usually refer to it as R_c when it is clear from the context that we are dealing with the perceptual version. Note that $r_{ij} \neq r_{ji}$ in general and therefore r_{ij} and r_{ji} are included as distinct quantities in the formulation of R_c.

This R_c is then compared against the user supplied refinement threshold to determine if the patch needs refinement.

6.3.4 Behaviour of the Perceptual Refinement Oracle

Due to the nature of the geometry of the model, a patch can never cross an object boundary as the patches define the object. In this discussion we choose a sharp shadow boundary as a typical case of rapidly changing gradient in a visually important feature.

In Figure 32 we show the intensity plot for a line across a shadow boundary in a radiosity scene. The line in Figure 32a passes through a single shadow boundary

while that in Figure 32b passes through two. The intensities are plotted on a per pixel basis. The steep vertical region corresponds to the penumbral shadow.

As can be seen immediately by applying the method of Figure 29 to Figure 32, if one of the sample points falls on the slope of the penumbra, it's corresponding value of r_{ij} or r_{ji} can be very large compared with the values at any other point as the gradient (G) at that point is very large.

There are four main cases for a patch interacting with a shadow boundary:

a) It is fairly clear that if a patch doesn't cross the penumbra boundaries, all the r_{ij} for that patch will be very close to zero — the tangents are horizontal ($G \approx 0$) and the point radiosities (intensities) are approximately equal ($B_i \approx B_j$). This is the case for a region of low visual importance. Substituting these values into Equation 26 gives $r_{ij} = 0$. This gives low values for R_c which implies low refinement levels.

b) For a patch that crosses the entire penumbra boundary, but has no sample points (vertices) within the penumbra region, we have the tangents horizontal ($G \approx 0$) but we have the radiosities different ($B_i \neq B_j$). In this case we have the values for r_{ij} being the magnitude of the difference between B_i and B_j which (by Equation 28) implies that $R_c \approx B_{\max} - B_{\min}$. This is almost identical to the standard refinement oracle in Section 2.2.4 which implies similar refinement levels to the Standard algorithm.

c) If a patch crosses the penumbra boundary with at least one sample point within the penumbra region, we have some of the r_{ij} very large relative to the other cases which indicates a visually important feature. This is due to the fact that the tangent at the sample point within the penumbra region is approaching vertical (G very large). In this case we find that $R_c >> B_{\max} - B_{\min}$ due to at least one large G value. This implies that the R_c for the perceptual oracle is much larger in this case than that of the standard oracle and consequently that the perceptual oracle is much more likely to cause refinement. In patches

of this nature we have the gradient value changing greatly over the extent of the patch which is a good indicator for edges.

d) In the final case, all sample points of the patch lie within the penumbra, we find that although all the tangents are non-horizontal, they are all roughly equal. Also the radiosity over this area is close to linear, so the r_{ij} are going to be quite small. Due to the fact that all the gradients are approximately equal (in size and direction), the linearity coefficient k becomes closer to 0 (Equation 27). These factors contribute to make R_c fairly small, inhibiting refinement in this case.

As can be seen from the above cases, our perceptual refinement oracle reduces to the standard oracle in cases **a** and **b**. In case **c** we find that our perceptual refinement oracle produces much more refinement than the standard oracle. These are the patches containing large gradient variation. In case **d** the radiosity is linear over the patch and our refinement oracle suppresses refinement in this region.

6.4 Summary

Firstly we define the visual importance function V. We point out some of the properties and dependences of V - particularly its dependence on the perceptual model of the HVS. We define *areas of interest* as maxima in the visual importance function. We also produce an objective image quality measure based on our edge detection model of the HVS.

Next we point out that the major drawback of the visual importance function is the required foreknowledge of the solution which we are trying to compute. For this reason a heuristic is developed that can operate on knowledge of the radiosity and gradient at a few sample points over the patch. With the use of a visual importance heuristic, computation of the perceptual refinement oracle becomes practical.

We describe in detail how the heuristic is constructed. The heuristic involves the correct use of a projection for the vector gradients and a method of computing the

effective divergence from linear of the actual radiosity function based on these gradients. We also point out that using the gradients can lead to the aliasing of the radiosity function when it contains effects on a small enough scale.

When the radiosity solution is reconstructed using bilinear interpolation rendering (e.g. Gouraud shading), an optimization can be applied directly to the refinement oracle. When the radiosity is close to linear it is very effective to leave the reconstruction of the radiosity up to the bilinear interpolation shading, so we develop a linearity coefficient which can be incorporated into the refinement oracle.

Lastly we present the complete computation of the perceptual refinement oracle as a simple equation based on our heuristic visual importance function and the linearity coefficient. We discuss the behavior of this oracle. We note that under certain conditions R_c reduces to the standard refinement oracle. When the patch crosses an edge the refinement is enhanced and when the radiosity is linear refinement is inhibited by the perceptual refinement oracle.

Chapter 7

Results and Analysis

7.1 Introduction

This chapter details the results obtained for a comparative experiment between the perceptual refinement oracle developed in the previous chapter and the standard refinement oracle (Section 2.2.4).

Firstly we provide some sample output pictures to demonstrate global illumination and other effects modeled by radiosity. These included soft shadows and colour bleeding.

Next we present the performance analysis, which shows the linear relation between the output mesh size and the computation time. This analysis is performed for two scenes with very different occlusion properties and the results contrasted.

Due to the fact that the performance data doesn't incorporate the visual quality, we show contrasting pictures produced by the two radiosity refinement oracles. These pictures are constrained to have the same output polygon mesh size — the only variable is the sample density distribution which is determined by the refinement oracle. We then demonstrate that the perceptual oracle better captures the information in visually important areas with no visible degradation in less important areas.

Finally we use our objective visual quality measure (Section 6.2.3) to measure output

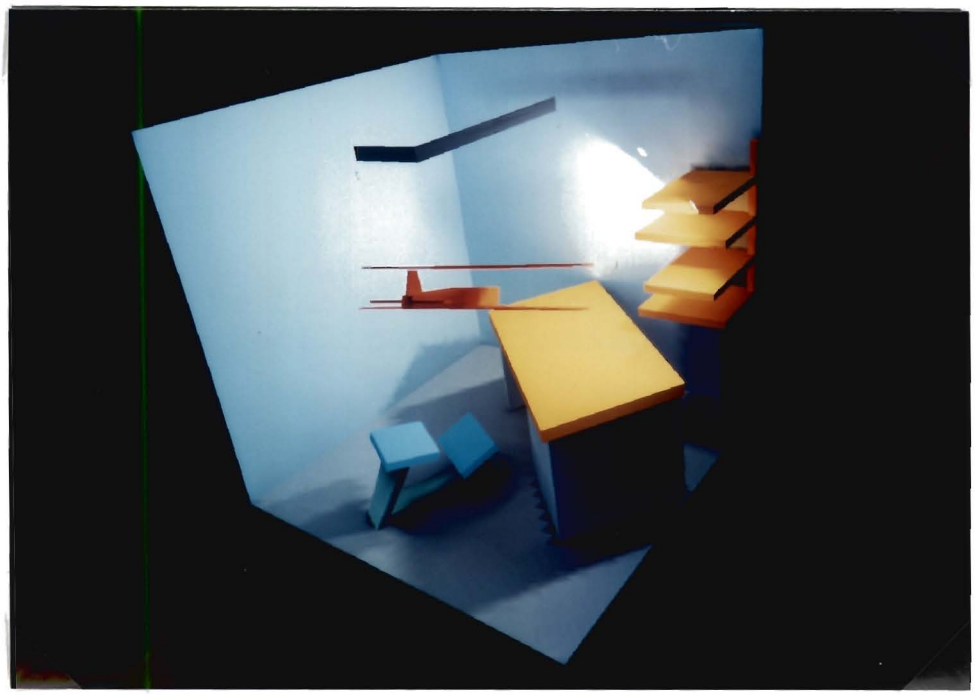


Figure 33: This scene shows the Full Office rendered for three iterations. Colour bleeding can be noticed along the far wall.

quality. We perform a filter analysis modeled on the center-surround receptive fields presented in Section 3.3.3. This experiment is divided into two parts. Firstly two pictures are computed with the output mesh sizes equal one image for each refinement oracle. These images are passed through an edge detection filter and then compared with a reference image to see which oracle has better edge reproduction. Secondly we compare two pictures with the same visual quality and show that the perceptual oracle achieves a substantial reduction in output mesh size and computation time.

7.2 Sample Output

We produced some rendered scenes to demonstrate the global illumination solution due to hierarchical radiosity. These scenes demonstrate the multiple iterations, soft shadows and colour bleeding, etc.

The Figures 33-36 all represent various differing scenes rendered at different levels of

Figure 34: This scene is a stripped down version of the Full Office. It is rendered with a higher refinement level than the previous scene. Again colour bleeding is noticeable along the walls.

refinement. They are included as a demonstration of both our testbed program and the perceptual refinement oracle.

The Full Office, Figure 33, is the representation of a simple office with a table, chair, shelves, etc. This scene has a relatively high degree of occlusion and is rendered with three iterations.

Figure 34 is a stripped down version of the above Figure 33. It is rendered at at a fairly high level of refinement and for three iterations of the radiosity algorithm. The colour bleeding from the table top onto the adjacent wall is fairly clear. This scene has a higher level of refinement than the previous one.

The Dining Room — Figure 35 is a large room with a single light source. It was mostly used to test the for the perceptual rendering oracle in areas with complex shadows. This scene was also rendered for three iterations. The scene is large compared to the size and brightness of the light source so the extremities of the room are fairly dim

Figure 35: The Dining Room is very large compared to the size of the light source. This causes the table top to be brightly lit and the corners of the room to be very dim.

compared to the brightness of the table surface.

The Cube (Figure 36) is the interior of a cube with different colour walls. Inside the cube is a single light source and a smaller cube which casts a shadow on the floor. This scene has a very low level of occlusion — indeed a low number of overall patches. This scene was rendered for three iterations and clearly demonstrates colour bleeding from the coloured walls onto surrounding surfaces.

All of these scenes were rendered for three iterations of the radiosity algorithm. We found increasing the number of iterations did little more than increase the overall brightness of the images.

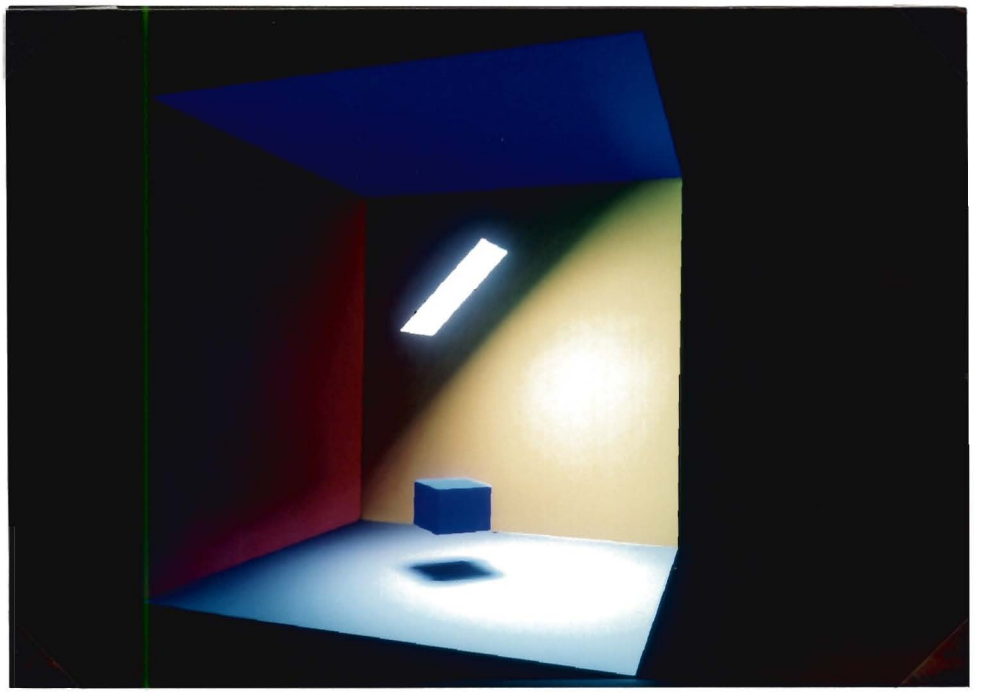


Figure 36: The Cube is a very small scene designed to have a small amount of internal polygons. This reduces the amount of visibility computation.

7.3 Performance

7.3.1 Graphing Assumptions

The input control for radiosity simulations is the refinement threshold or refinement criterion. We denote this quantity as R_c (Equations 6 and 28. As noted in Section 2.2.4, R_c is a user specified maximum allowable error tolerance on an interaction.

Both the output scene size and the time taken to render are dependent on R_c which is justification for plotting size vs time in the graphs that follow. We note that the size is not actually proportional to R_c, instead it follows the relation shown in Figure 37. In this figure we show that R_c is related to the size in a way similar to a hyperbola.

Setting s as the output data size we have determined that the variables are related approximately by:

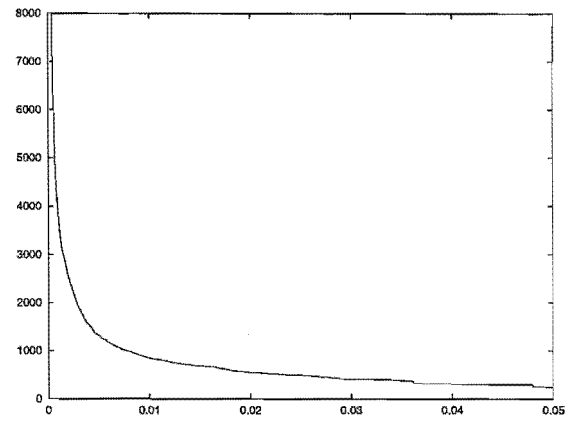


Figure 37: This is a plot of R_c (domain) against the output size.

$$s \approx 10.0/R_c \tag{29}$$

It is important to note that Equation 29 was determined experimentally and is not an accurate relation, rather it was chosen as being simple and one that fit the data fairly well. Equation 29 is used to get an approximately constant spacing between the data in the size domain of the graphs that follow — linearly varying the input control (R_c) gives a graph with all the data points concentrated in the region near the origin with very few points describing the rest of the curve. What is required is a linear variation of the size parameter which requires varying R_c according to the relation 29.

Factors such as a minimum polygon area limit in the hierarchical radiosity simulation and the fact that the size increases according to 4^n with respect to the quadtree depth all have an impact on Equation 29. Even so, we still find it useful and have retained it's use in generating the graphs shown.

7.3.2 Occluded Scene

All the timing statistics were generated using a 200MHz MIPS R4400 CPU in a Silicon Graphics Onyx equipped with 128Mb RAM and a Reality Engine II. The results are

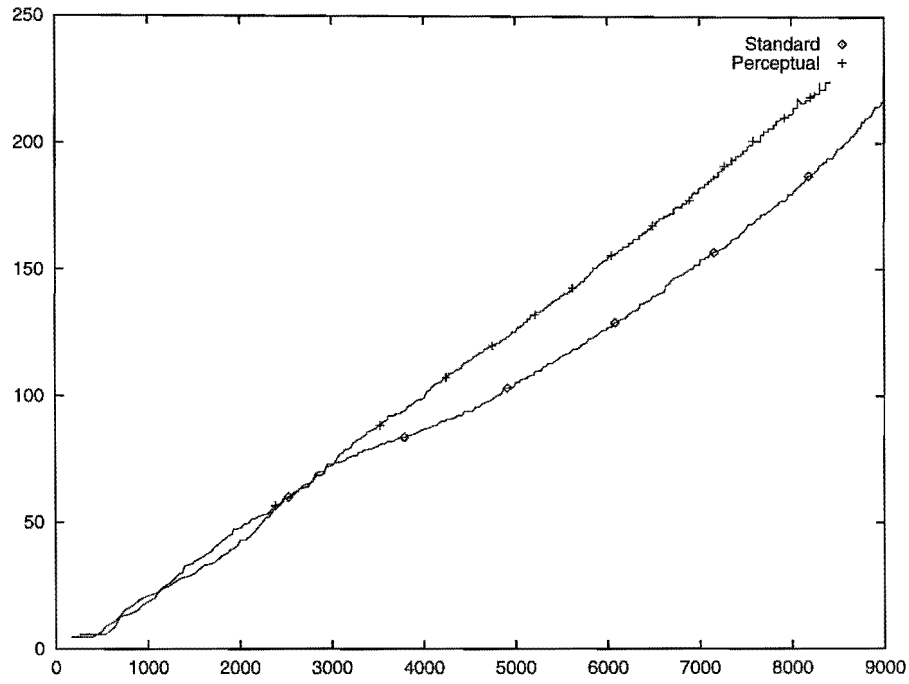


Figure 38: Shows the our method vs the standard BF refinement method. The horizontal axis is number of output polygons and the vertical axis is time to render in seconds. Our method is shown above the standard method.

displayed using an Inventor viewer.

Figure 38 shows the comparison between our perceptual refinement oracle and the standard refinement oracle. The domain is the number of leaf polygonal patches generated and the range is the time taken to render a scene at that size. The scene is rendered for one iteration and the size and time measured for different values of R_c. The scene under consideration is the Office scene shown in Figure 34. This scene has a fair amount of occlusion so visibility calculations have a substantial impact on the computation time. As expected, the use of our perceptual oracle causes a slight increase in rendering time for the same output mesh size. It is important to note that no attempt is made to factor in the visual quality of the output scene in the above graph, so the graph is a naive comparison of the two refinement oracles' computation time for the same output size. As we shall note later, our perceptual refinement oracle substantially improves the visual quality of the scene given the same output size constraints, i.e., the same quality is reached for much smaller output size.

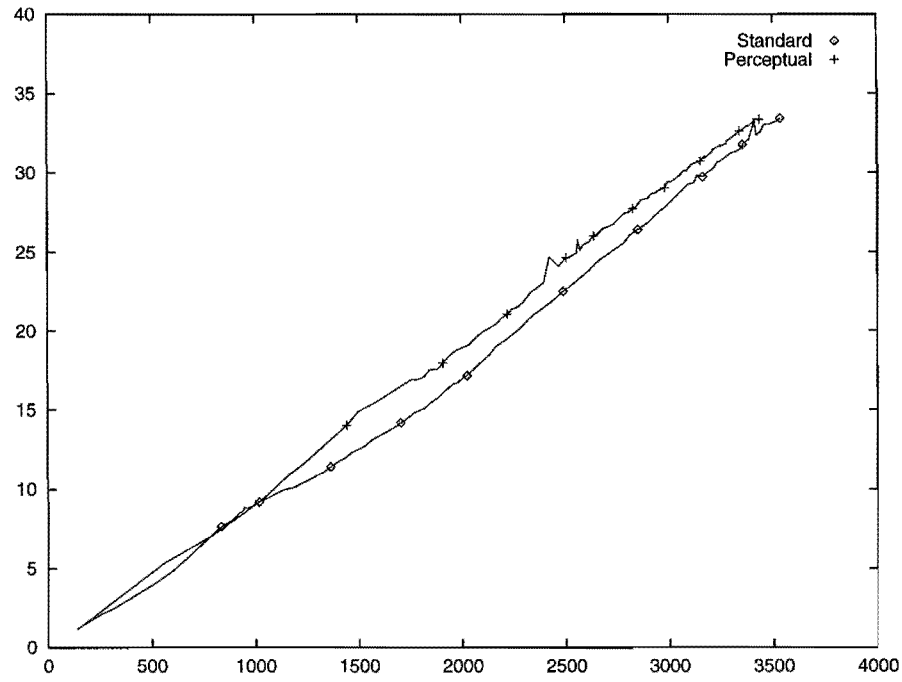


Figure 39: Timing statistics for the Cube scene. Has a lower polygon count and much less occlusion.

7.3.3 Unoccluded Scene

The graph in Figure 39 shows the size vs. time relation for the relatively unoccluded scene in Figure 36. This scene has very few interior patches and hence visibility calculations are much less of a factor in the computation time. Nicolas Holzschuch[18] has shown that visibility computation in a hierarchical radiosity scene contribute about 71-80% of the entire scene computation time. Again, the graph of our perceptual oracle lies above that of the standard refinement oracle. This relatively unoccluded scene is also relatively simple compared to the previous scene (Figure 34) containing many fewer initial (top level) patches.

7.3.4 Linear Regression

In order to predict the extra per polygon overhead incurred by the perceptual refinement oracle we need to ascertain the correlation between the size and time variables.

Also we need to compute the best fit straight line to each of the graphs. Using the best fit lines, we can predict the computation time for any given output mesh size.

The method we use to fit a line to the data shown in Figures 38 and 39 is the least squares linear regression[29][1].

Given that

$$y = a + bx$$

is the linear best fit to a set of data, we have determined that for the data sets presented in Figures 38 and 39 we have:

	Occluded (Figure 38)		Unoccluded (Figure 39)	
	Standard	Perceptual	Standard	Perceptual
n	1000	689	100	100
a	3.918	-10.17	-1.34	-0.606
b	0.02247	0.02758	0.00973	0.0101
r	0.994	0.999	0.997	0.994

where n is the sample size and r is the correlation coefficient.

As an example we note that at 10000 leaf patches, our perceptual refinement oracle adds about 16% onto the total rendering time for the occluded scene (Figure 38) and about 4% for the relatively unoccluded scene (Figure 39) given the same output size constraints. These percentages are computed by substitution into the regression line formulae.

The linear regression analysis shows that our perceptual refinement oracle increases the rendering time in scenes with higher occlusion levels. As the oracle is only invoked when we need to determine if an interaction needs refinement, this increase can only be attributed to our oracle increasing the amount of visibility computation by the choice of which interactions to refine. While this may sound counter-productive, we note

[1] Or see any standard undergraduate statistics textbook

Figure 40: This picture shows the Office scene rendered at a fairly low level of refinement with the standard BF refinement oracle to highlight where the rendering artifacts are noticeable.

that we defined areas of interest to include shadow boundaries (Section 6.2), so our perceptual refinement oracle increases refinement in areas with shadow boundaries. This actually increases the proportion of partially occluded patches that need to be evaluated. This proportional increase is what causes our refinement oracle to appear to perform worse in the case of higher occlusion — causing the increased use of the visibility computation algorithm required for partial occlusion. We note, however, that it is this proportional increase that causes scenes rendered with our perceptual refinement oracle to have a higher visual quality compared to scenes rendered with standard oracles.

Figure 41: This picture shows the Office scene rendered with the perceptual refinement oracle with the level of refinement computed to yield the same number of output polygons as the standard oracle above.

7.4 Subjective Image Quality Evaluation

In this section we allow the reader to participate in a subjective visual quality evaluation. In Section 7.5.3 we present our objective measure.

Candidate areas for high visual importance are those containing a highlight, shadow boundary or edge. The areas of interest in Figures 40-43 are the shadow boundaries along the back and side walls and the shadow on the floor underneath the table. These pictures show the refinement for the first iteration only. The refinement levels have been computed to result in comparison scenes that have the same number of output polygons. This means that the same number of sample points is used for both methods and thus the same amount of information is used to encode each scene.

When comparing the shaded scenes (Figures 40 and 41) it can be seen that the perceptual refinement oracle produces a better representation of the shadow boundaries

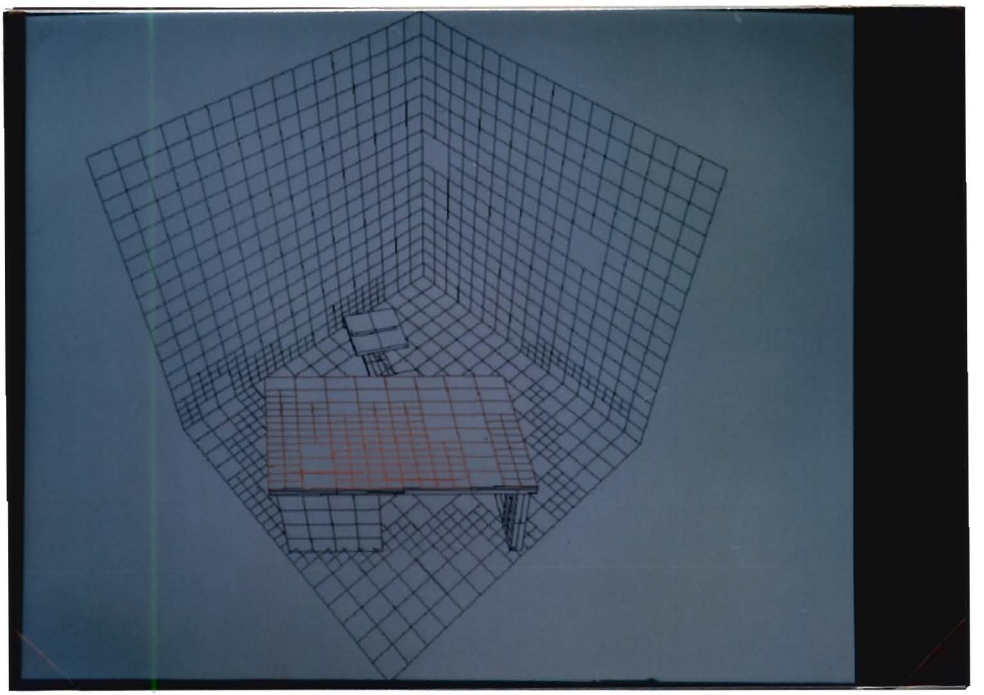


Figure 42: The mesh version of the standard BF refinement oracle figure.

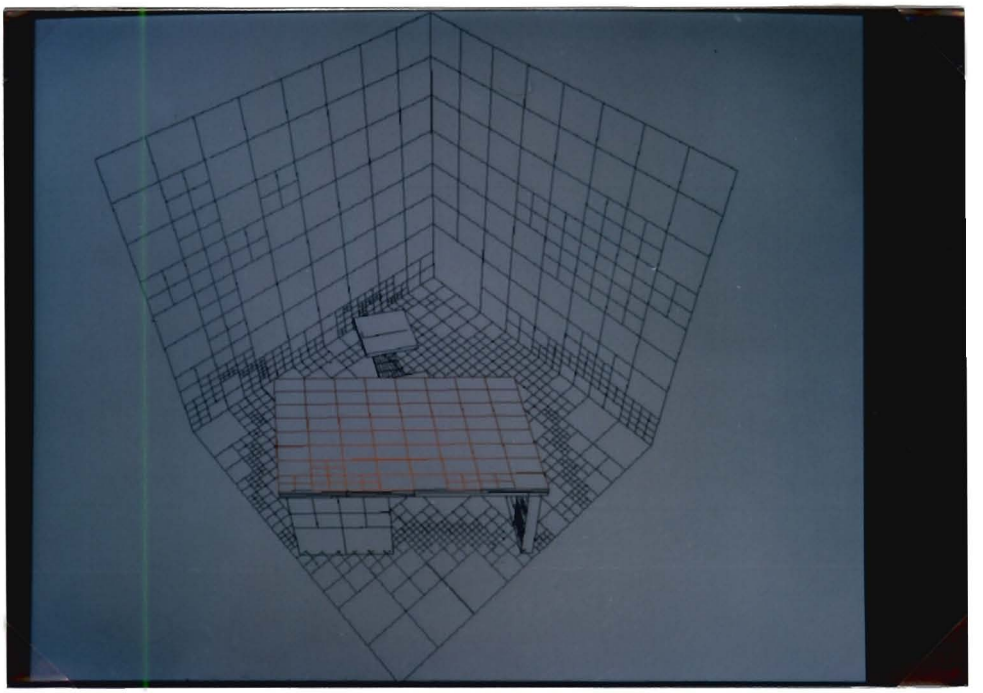


Figure 43: The mesh version of our perceptual refinement oracle.

along the back and side walls. This effect is particularly noticeable on the shadow boundary under the table. While the jagged edges are still seen in the perceptual result, it is obvious that the shadow boundaries are rendered with more detail than the same regions in the standard result. Also we note that the areas of lower visual importance — the table top and the back and side wall areas not containing shadow boundaries — show no discernible difference when compared.

When comparing the mesh diagrams for the same scenes (Figures 42 and 43) it can be seen immediately that the perceptual refinement oracle has substantially reduced the number of patches used to represent the table top and the back and side walls — the areas of lower visual importance. We reiterate that this reduction is obtained at no sacrifice to the (perceived) visual quality in these regions. Also it is fairly clear how the improved visual quality in the areas of interest is achieved. The additional refinement allowed by the reduction of polygons in the regions of lower visual importance is fairly easy to see in Figure 43 and results in a marked improvement in the overall visual quality of the scene.

It is fairly obvious from the mesh diagram in Figure 42 that the standard oracle produces more polygons in the regions of low visual importance than our perceptual refinement oracle. This implies that the standard oracle causes much more data to be devoted to encoding this region. The shaded images show that there is little or no difference in the visual quality of the output between the standard and our refinement methods in these regions. This implies that the standard method includes wasted data in these areas.

7.5 Filter

7.5.1 Center-Surround Receptive Field

As shown in Figure 8 the on-center off-surround filter performed by the retinal ganglion cells takes the form of an additive center with a subtractive surrounding annulus.

Figure 44: This is the reference image for the center-surround image filter experiment. This picture is rendered at a much higher refinement level than the two comparison images.

The filter performed by the eye is a weighted sum of the stimuli from the photoreceptors and is very similar in nature to an edge detection filter. It is generally deduced that edge information is essential to human perception of a scene (Section 3.3.2).

We use a very highly refined reference scene — Figure 44 and Figure 45 — to represent the ideal image of a radiosity simulation for a given set of viewing parameters. The aim of the experiment is to show that after processing by the HVS, the perceptual refinement oracle (Section 6.3.3) produces substantially better looking pictures in less time and using less memory than the standard refinement oracle.

As the level of refinement used in the reference image is orders of magnitude more than that used in the test images, it shouldn't make a difference which refinement oracle was used to produce it. However, in case the choice of refinement method for the reference image does introduce a bias into the results, we chose the standard refinement oracle to produce the reference image. If a bias is introduced, it will tend

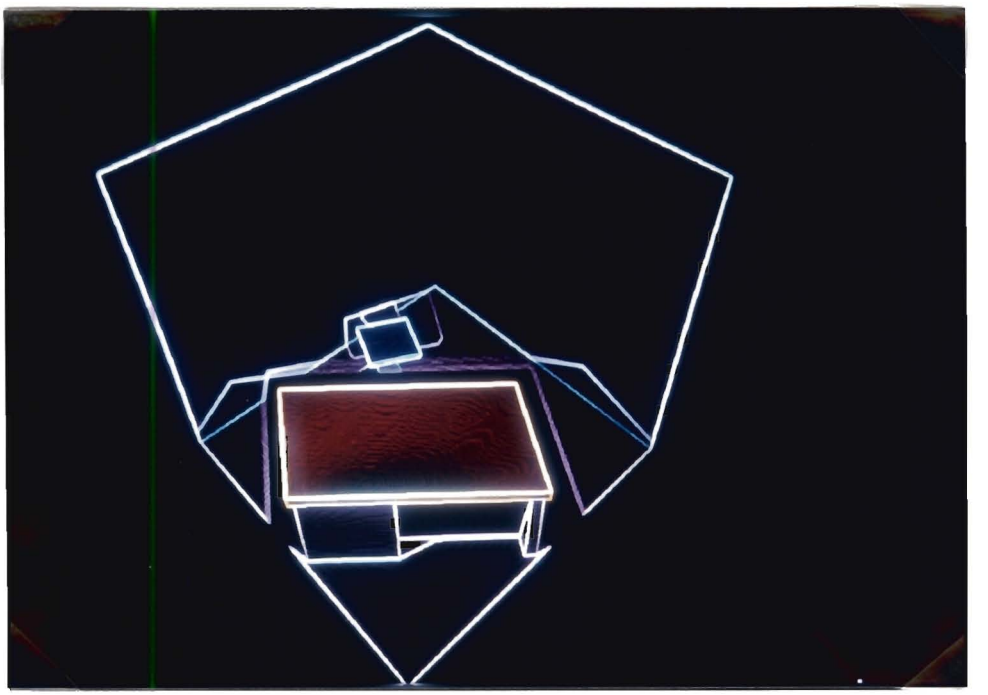


Figure 45: Filtered reference image

to favour the standard refinement oracle, as all the test images are compared against the reference image.

We split the experiment into two parts. Firstly we constrain the rendering size to be the same and compare the two produced radiosity simulations. Constraining the size forces the two simulations to be rendered using exactly the same amount of information (sample points). The refinement level is chosen low enough so that artifacts are visible and the results are compared to see which oracle produces the more noticeable artifacts. The scenes are passed through an edge detection filter before comparison to mimic the process of the center-surround receptive fields in the HVS.

The second part of the experiment also uses edge filtered pictures for comparison, but this time the output size is allowed to vary. Instead for each picture we compute the error in visual quality for the image using our metric defined in Section 6.2.3. This visual quality measure allows us to compare the size and time requirements of the two oracles for pictures of similar visual quality.

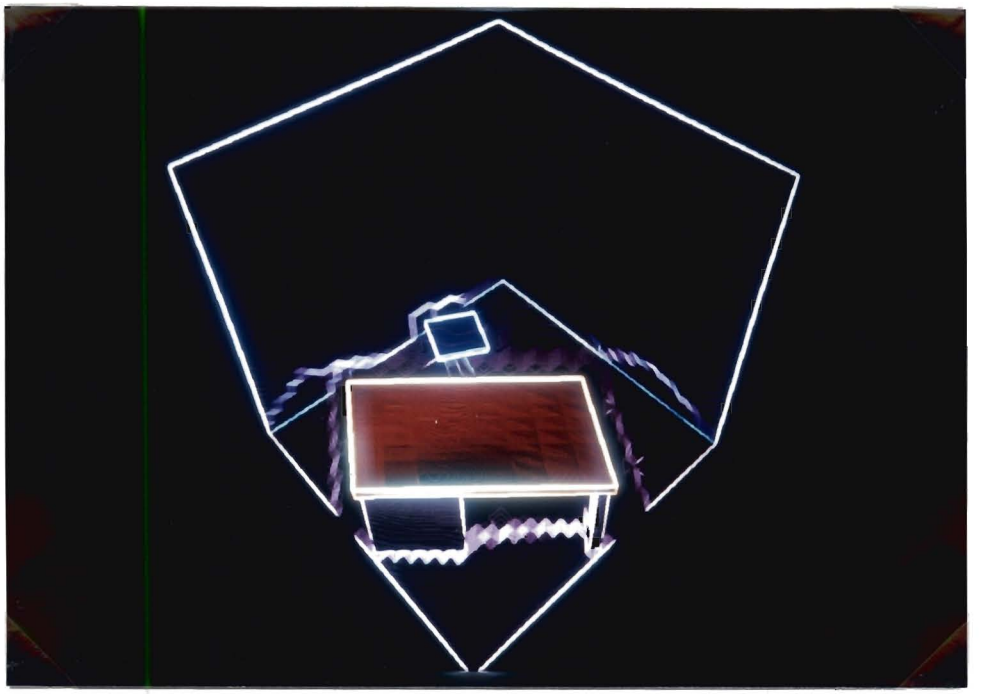


Figure 46: Filtered standard refinement test image.

7.5.2 Constrained Size Experiment

We reuse the data from the previous experiment (Section 7.4) to compute the images used in this experiment — Figures 46 and 47. These edge filtered pictures are computed from Figures 40 and 41 respectively. This previous experiment also constrained the output mesh sizes to be the same for both oracles, so the size constraint required for this experiment is met.

We note that, by examining the filtered reference image (Figure 45), shadow boundaries produce very strong and pronounced edges. These shadow edges are certainly the most significant features of this filtered image, save only the object boundaries themselves.

Note that we consider mainly the shadow edges, as the object edges are effectively identical independent of the refinement oracle used.

The standard refinement oracle produces an image in which the shadow edges can be seen to be very jagged and broad (Figure 46). A certain amount of jaggedness

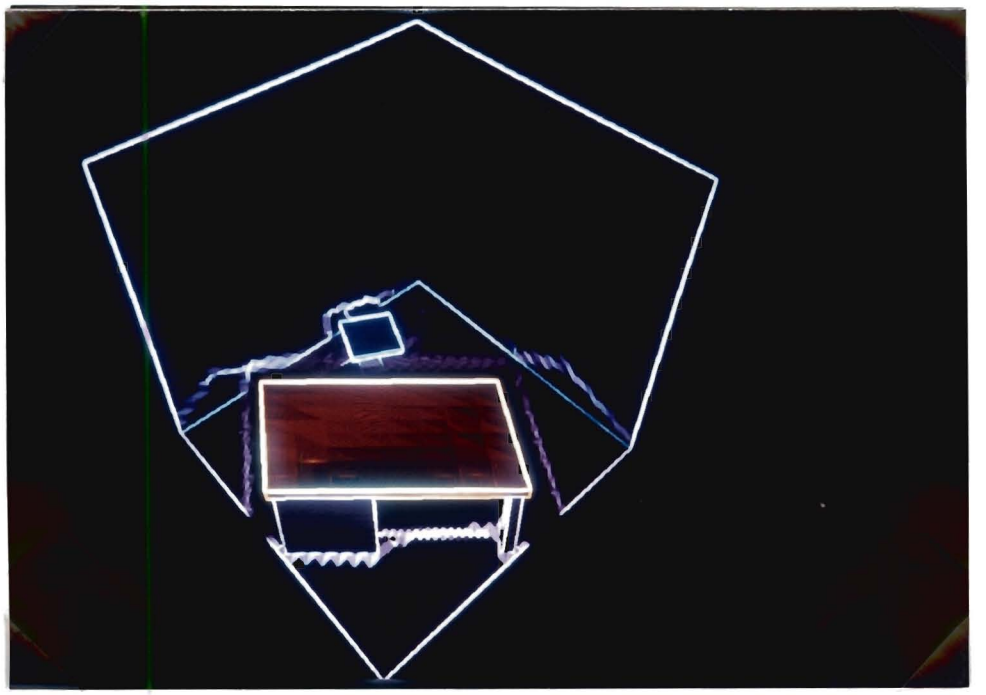


Figure 47: Filtered perceptual refinement test image.

and indistinctness is to be expected compared to the reference image as the standard oracle was restricted to a much lower level of refinement.

The perceptual refinement oracle produces the image in Figure 47. Bearing in mind that the two oracles were constrained to use the same amount of information to represent the radiosity simulation, we can immediately see from Figure 47 that the shadow edges produced by the perceptual refinement oracle show a marked improvement in distinction and definition compared to those produced by the standard oracle. The shadow edges still exhibit a jagged nature, but this is generally confined to a smaller area.

7.5.3 Visual Quality Comparison

In this part of the experiment we are concerned with comparing images produced by the standard and perceptual oracles that have the same visual quality. We use the definition of visual quality developed in Section 6.2.3. This definition depends on a

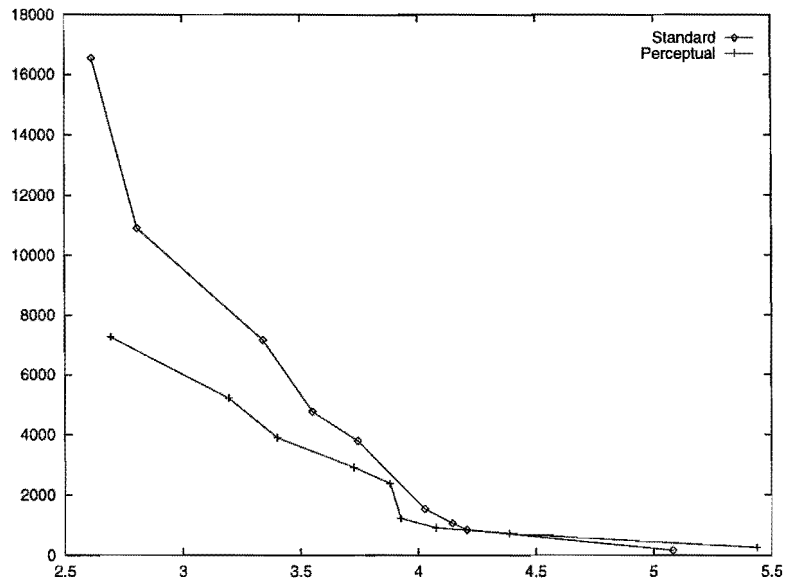


Figure 48: Plot of the error metric for various refinement levels against the output polygon size. Note that picture quality and refinement increases (error decreases) going *left* (towards the origin).

simple metric on the edge filtered images. The reference image used for the metric is Figure 45. The measure of visual quality is based on the distance between the test image and the reference image using the defined metric. The metric depends on the RMS difference between the two images.

We computed images for several radiosity simulations (of the same model) for both the standard and perceptual oracles, varying the refinement threshold. For each radiosity simulation we capture a picture P with the same viewing parameters as in Figure 41. We measured the error in visual quality $||P||$ and mesh size for each of the radiosity simulations. These values are plotted in Figure 48. The error ($||P||$) is the domain and the size is the range. As the error tends to 0 along the horizontal axis, the corresponding radiosity simulation in both cases gets closer to the ideal reference image.

The graph shows that, for the same visual quality, the perceptual refinement oracle produces substantially smaller mesh files. In fact as the refinement levels increase (error drops) the ratio increases to better than 2 : 1 so the perceptual refinement

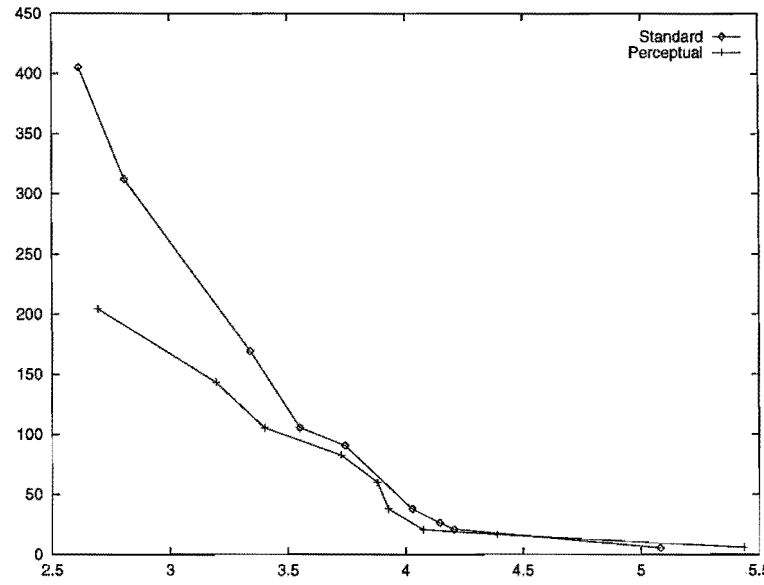


Figure 49: Plot of the error metric for various refinement levels against the rendering time in seconds. Note that picture quality and refinement increases (error decreases) going *left* (towards the origin).

oracle uses approximately half the number of mesh elements as used by the standard oracle. As was shown previously (Section 7.3.4 and Figures 38 and 39) the radiosity solution computation time is linearly related to its output mesh size. This implies our similar 2 : 1 improvement in computation time. We also point out that the perceptual refinement oracle performs better relative to the standard oracle as the refinement levels increase (rendering becomes more detailed). This is a very useful property, as the more highly refined the scene, the more the optimization is required. We note that below a certain level of refinement, less than $1800 - 2000$ polygons in this case, none of the edge information is captured at all (aliasing) so both methods are equally bad. In Figure 48 this is shown for error levels above (to the right of) ± 4.3.

As expected the visual quality vs time plot in Figure 49 is very similar to the quality vs size plot in Figure 48. This is due to the linear relation of output size to computation time (Section 7.3.4). Again note that the visual quality increases to the *left*.

7.6 Summary

In this chapter, we set out to compare our perceptual refinement oracle against the standard refinement oracle.

Firstly we presented some introductory images to demonstrate that the code produces correct output at high levels of refinement for more than one iteration (typically 3). We found that more than four iterations added very little to the output.

Next we showed that, for a naive comparison of output mesh size, our perceptual refinement oracle adds a small computation overhead compared to the standard oracle — $\sim$ 4% to $\sim$ 16%. This increase is due to the additional computation of vertex radiosity gradients and the increased ratio of partially occluded patches relative to completely occluded or visible patches. We note that although the computation of vertex gradients in addition to the vertex radiosity is substantially more expensive than simply computing the vertex radiosities required for the standard oracle, we don't substantially increase the per patch computation time. This is due to the fact that the oracle is a very small part of the load involved in the computation of a radiosity solution — the majority being the visibility computation. More significantly we show that for a high level of correlation, the output mesh size is linearly related to the computation time which agrees with our understanding of the complexity of hierarchical radiosity.

We show two comparison images with the same output size to demonstrate that the perceptual oracle produces images that have noticeably better visual quality. This improvement in visual quality is clearly seen in the comparison of the mesh diagrams where we noted that our perceptual oracle "moves" refinement from visually less important regions to areas of interest.

We also showed that our notion of visually important areas is accurate by using an edge detection filter that matches some of the visual processing performed in the HVS. This processing is performed by the center-surround fields of the retinal gangleon cells and cells in the LGN and is analogous to edge detection. We filter two comparison images and show that our perceptual refinement oracle produces results that, after

filtering, are a closer match to a very highly refined scene passed through the same filter. This implies that the signal produced by the perceptual image along the optical nerve to the brain is a closer match to that produced by the reference image that the standard oracle.

Finally we use our previously developed visual quality metric to measure the visual quality of radiosity images. When comparing images of equal visual quality we have shown that the perceptual refinement oracle produces its results in half the time and using half the memory compared to the standard refinement oracle.

Chapter 8

Conclusion

The purpose of this dissertation was to explore the application of knowledge of the human visual system (HVS) to a powerful physically based rendering algorithm such as hierarchical radiosity. Inspired by the impressive achievements of perceptually based algorithms in other fields of computer graphics and on our preliminary experiment, we set out to produce a perceptually based hierarchical radiosity algorithm. We found it effective to create a perceptually based refinement oracle and leave the rest of the hierarchical radiosity algorithm physically based — the refinement oracle guides the refinement process and hence the sample point density distribution. Effectively the oracle decides how much computation to devote to each region of the scene.

To build a perceptual refinement oracle, we needed a model of the HVS. The model we adopted is based on neuron receptive field theory in neurophysiology. This effectively means that much of the information presented to the brain is an edge detection filtered version of the field of view. The choice of this fairly simplistic model for the HVS is supported in part by claims that edge information is essential to our perception of an image[28] and in part by the results we obtained in Chapter 7. Our model includes properties of the HVS such as sensitivity to contours and edges.

After examining the visual importance function (Section 6.2), we deduced that the perceptual oracle would have to be a heuristic that estimated the visual importance in a region based on a small number of computed sample points. In order to predict

the occurance of edges or other visually important features, we required accurate computation of the radiosity gradient (Chapter 4). We produced an oracle based on the HVS model using the gradient to estimate the visual importance and hence predict the presence of edges and other visually important features. This oracle produces high refinement in areas with a predicted high visual importance (such as shadow boundaries) and low levels of refinement in regions with low visual importance (such as areas with linear radiosity which are accurately reconstructed by the bilinear shading algorithms). Our refinement oracle reduces to the standard refinement oracle in other cases (See Section 6.3.4).

After verifying that the algorithm worked (Figures 33 to 36) we set out to measure its performance relative to the standard refinement oracle. Our first (naive) approach was to measure size v.s. time. The size measured is the number of polygons in the output mesh. The graphs show that our oracle takes slightly longer to render for the same output size (Figures 38 and 39). These results appear dissapointing till one realizes that two equal size renderings with the two oracles do not have the same appearance (visual quality). We did (accidently) verify the unique (among radiosity methods) $O(n)$[16] complexity of hierarchical radiosity in output size v.s. time to a very high degree of correlation.

To demonstrate this visual quality difference, we photographed a scene rendered with each oracle with the constraint that they produced the same number of output polygons. From these pictures we easily verified that the perceptual refinement oracle produced images of a higher visual quality. In addition to a better representation of shadow boundaries and other visually important features, we noted that there was no noticeable visual degradation in regions with lower visual importance (hence fewer sample points with our oracle) which implies that these regions are being overrefined by the standard oracle — compared to the overall visual quality.

Considering our edge detecting model of the HVS, we performed an edge detection filter of the two images discussed above. Comparing these to a filtered image of a highly refined reference scene, we were able to determine that our perceptual refinement oracle more faithfully reproduced the edge information. This is an approximation to the

objective visual quality measure we developed and it compares very well with subjective experience in spite of its simplicity. This is likely due to the variable restriction required in Section 6.2.3.

In order to compare the two oracles at the same level of visual quality, we needed a measure of visual quality. We proposed a visual quality measure based on the RMS difference between an edge filtered test image and the edge filtered reference image. This measure coincides with our model of the HVS in that accurate edge reproduction is equated with a good visual quality. Also for a wide range of refinement levels we verified that images with very similar visual quality measures were visually indistinguishable. Using our visual quality measure, we were able to compare the two oracles at the same level of visual quality output (See Figure 48). We found that for the same visual quality our perceptual refinement oracle requires half the memory (half the output polygons) and takes about half the time to render. Furthermore, as the level of refinement increases, so our perceptual refinement oracle improves relative to the standard refinement oracle.

8.1 Future Work

Having verified that using the HVS in hierarchical radiosity yields substantial performance improvements for even the restricted HVS model we used, we propose the use of a more complete model of the HVS. A more complete model would possibly include features like the more complex behavior of receptive fields in the visual cortex, contrast sensitivity and just noticeable differences. The use of a more complete model for the HVS would require a more complex perceptual refinement oracle to take into account these added features. Also it would probably invalidate our visual quality measure as it was based on our simplistic model of the HVS. More complex HVS models would require the use of a perceptual experiment using the observations of a number of test subjects to ascertain relative visual quality rather than relying solely on a mathematical measure.

Bibliography

[1] Class notes: Psy280f. `http://server.esc.cquest.utoronto.ca/psych/psy280f/ch3/mb/mb.html`, 1998. University of Toronto, CQUEST Psychology Notes.

[2] J. Arvo. The Irradiance Jacobian for Partially Occluded Polyhedral Sources. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 343–350, 1994.

[3] I. Ashdown and P. J. Franck. Luminance gradients: Photometric analysis and perceptual reproduction. In *1995 IESNA Conference*, May 1995.

[4] L. Aupperle. *Hierarchical Algorithms for Illumination.* Ph.D. thesis, Technical Report, Princeton, NJ, 1993.

[5] J. A. Bednar. Anatomy and physiology of the visual system. `http://www.cs.utexas.edu/users/jbednar/papers/bednar.thesis/node6.html`, September 1997. Department of Computer Sciences, The University of Texas at Austin.

[6] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press Professional, Boston, MA, 1993.

[7] G. Drettakis and E. Fiume. A Fast Shadow Algorithm for Area Light Sources Using Backprojection. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 223–230, 1994.

[8] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics, Principles and Practice, Second Edition.* Addison-Wesley, Reading, Massachusetts, 1990.

[9] T. A. Funkhouser and C. H. Séquin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. Computer Graphics Proceedings, Annual Conference Series, pages 247–254. ACM, 1993.

[10] S. Gibson and R. J. Hubbold. Perceptually driven radiosity. *Computer Graphics Forum*, 1997. To appear.

[11] A. S. Glassner. *An Introduction To Ray Tracing.* Academic Press, 1989.

[12] A. S. Glassner. *Principles of digital image synthesis*, volume 1. Morgan Kaufmann Publishers, 1995.

[13] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modelling the Interaction of Light Between Diffuse Surfaces. In *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, pages 212–222, July 1984.

[14] I. E. Gordon. *Theories of Visual Perception.* John Wiley and Sons, 1989.

[15] E. A. Haines and J. R. Wallace. Shaft Culling for Efficient Ray-Traced Radiosity. In P. Brunet and F. W. Jansen, editors, *Photorealistic Rendering in Computer Graphics (Proceedings of the Second Eurographics Workshop on Rendering)*, New York, NY, 1994. Springer-Verlag.

[16] P. Hanrahan, D. Salzman, and L. Aupperle. A Rapid Hierarchical Radiosity Algorithm. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.

[17] P. S. Heckbert. Ray tracing jell-o brand gelatin. *Communications of the ACM*, 31(2):131–134, February 1988.

[18] N. Holzschuch. *Le Controle de l'Erreur dans la Methode de Radiosite Hierarchique (Error Control in Hierarchical Radiosity).* Ph.D. thesis, Université Joseph Fourier, Grenoble, France, March 5th, 1996.

Available (in French) as `ftp.imag.fr/pub/Mediatheque.IMAG/theses/96-Holzschuch.Nicholas/these.ps.gz`.

[19] N. Holzschuch and F. Sillion. Accurate Computation of the Radiosity Gradient for Constant and Linear Emitters. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 186–195, New York, NY, 1995. Springer-Verlag.

[20] J. T. Kajiya. The Rendering Equation. In *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.

[21] M. Kunt, A. Ikonomopoulos, and M. Kocher. Second-generation image-coding techniques. *Proceedings of the IEEE*, 73(4):549–574, April 1985. Invited Paper.

[22] D. Lischinski, B. Smits, and D. P. Greenberg. Bounds and Error Estimates for Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 67–74, 1994.

[23] D. Lischinski, F. Tampieri, and D. P. Greenberg. Combining Hierarchical Radiosity and Discontinuity Meshing. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 199–208, 1993.

[24] A. Nikitin and M. Leonov. A polygon clipping algorithm. `http://woland.afti.nsu.ru/ leonov/clipdoc.html`, 1997.

[25] T. Nishita, I. Okamura, and E. Nakamae. Shading Models for Point and Linear Light Sources. *ACM Transactions on Graphics*, 4(2):124–146, April 1985.

[26] M. Paulin and R. Cabet. Computing the radiosity gradient for partially occluded polygons. In H. P. Santo, editor, *Proceedings of Fifth International Conference on Computational Graphics and Visualization Techniques (Compugraphics '96)*, P. O. Box 4076, Masama 2745, Queluz, Portugal, December 1996. GRASP - Graphic Science Promtions and Publications.

[27] B.-T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.

[28] M. M. Reid, R. J. Millar, and N. D. Black. Second-generation image coding: An overview. *ACM Computing Surveys*, 29(1):3–29, March 1997.

[29] L. Underhill. *Introstat.* Juta and Co. Ltd, 1987.

[30] G. J. Ward and P. Heckbert. Irradiance Gradients. In *Third Eurographics Workshop on Rendering*, pages 85–98, Bristol, UK, May 1992.