

# **Eigenvector Radiosity**

by

Ian Ashdown, P. Eng., FIES

B. App. Sc., The University of British Columbia, 1973

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

**Master of Science**

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

We accept this thesis as conforming  
to the required standard

---

---

**The University of British Columbia**

April 2001

© Ian Ashdown, 2001

# Abstract

Radiative flux transfer between Lambertian surfaces can be described in terms of linear resistive networks with voltage sources. This thesis examines how these “radiative transfer networks” provide a physical interpretation for the eigenvalues and eigenvectors of form factor matrices. This leads to a novel approach to photorealistic image synthesis and radiative flux transfer analysis called *eigenvector radiosity*.

# Table of Contents

Abstract .....	ii
Table of Contents .....	iii
List of Figures .....	iv
Acknowledgments .....	vii
Dedication .....	ix
Preface .....	x
1. Introduction .....	1
2. Solving the Radiosity Equation .....	16
3. Eigenvector Radiosity – Theory .....	40
4. Eigenvector Radiosity – Practice .....	64
5. Numerical Experiments .....	73
6. Conclusions .....	92
Bibliography .....	95
Appendix A – MATLAB Experiments .....	102
Appendix B – Eigenspectra .....	120
Appendix C – Radiometric and Photometric Units .....	124

# List of Figures

1.1	Architectural lighting design example .....	3
1.2	Modeling the flow of light in a rectangular room .....	5
1.3	A rectangular room with surfaces divided into patches .....	6
1.4	Patch $E_j$ receiving flux $\Phi_{ij}$ from patch $E_i$ .....	7
1.5	Form factor geometry between two differential elements .....	8
1.6	Determining the form factor $F_{dE_i-dE_j}$ by area integration over $E_j$ .....	10
2.1	Gathering flux from the environment .....	27
2.2	Shooting flux into the environment .....	29
2.3	Progressive refinement radiosity algorithm .....	32
2.4	Progressive refinement radiosity algorithm with positive overshooting .....	35
3.1	Example radiative transfer network .....	46
3.2	Example rectangular room .....	49
3.3	Loop conductance .....	51
4.1	Subdividing a surface into a hierarchy of patches and elements .....	66
5.1	Warehouse (wireframe) .....	74
5.2	Warehouse (progressive radiosity) .....	74
5.3	Conductance array eigenvalues for Warehouse environment .....	76
5.4	Warehouse (40 eigenvectors) .....	77
5.5	Tatami Room (wireframe) .....	78
5.6	Tatami Room (progressive radiosity) .....	78
5.7	Conductance array eigenvalues for Tatami Room environment .....	80
5.8	Tatami Room (35 eigenvectors) .....	81

5.9	Tatami Room (78 eigenvectors) .....	81
5.10	Modified Tatami Room (wireframe) .....	82
5.11	Modified Tatami Room (progressive radiosity) .....	83
5.12	Conductance array eigenvalues for modified Tatami Room environment .....	84
5.13	Modified Tatami Room (20 eigenvectors) .....	84
5.14	Modified Tatami Room (70 eigenvectors) .....	85
5.15	Modified Tatami Room (progressive radiosity – daylight) .....	86
5.16	Modified Tatami Room (20 eigenvectors – daylight) .....	87
5.17	Ashdown Residence (wireframe) .....	88
5.18	Ashdown Residence (progressive radiosity) .....	88
5.19	Conductance array eigenvalues for Ashdown Residence environment .....	89
A.1	Patch-to-patch form factor geometry .....	103
A.2	Perpendicular rectangles with common edge .....	104
A.3	Parallel rectangles .....	104
A.4	Empty rectangular room .....	105
C.1a	Irradiance .....	127
C.1b	Radiant exitance .....	127
C.2	A ray intersecting a surface .....	129
C.3a	Radiance (arriving) .....	129
C.3b	Radiance (leaving) .....	129
C.4	Inverse square law for point sources .....	131
C.5	Radiance of a point source .....	131
C.6	CIE photometric curve .....	133

C.7	Reflection from a Lambertian surface .....	137
C.8	A Lambertian emitter illuminating the interior of a sphere .....	138

# Acknowledgments

This thesis would not have been possible without the assistance and encouragement of many people over the past nine years. I must express my gratitude to Diane Cerra of John Wiley & Sons (New York, NY), for having faith that I could write a book on radiosity; to Dr. Alain Fournier of the University of British Columbia, who accepted my first computer graphics paper; to Peter Murphy of Ledalite Architectural Products (Langley, BC), who provided financial and moral support; to Dr. Holly Rushmeier of IBM Watson Research Center, for encouraging me to present my first SIGGRAPH course; to Greg Ward Larson, who forgave me for encouraging him to write a book on **Radiance**; to David DiLaura of the University of Colorado at Boulder and his student Peter Franck, who provided unbounded inspiration through their paper; to Bruce Froehlich of FHT Incorporated (Columbia, MD), who provided an irresistible incentive; to Dr. Alain Fournier again, who accepted me as his student after an academic hiatus of 25 years; and to Todd Saemisch and Dave Speer of Lighting Analysts Inc. (Littleton, CO), who entrusted their corporate future to me for two harrowing years.

More recently, I would like to thank Dr. Murray Hodgson of the University of British Columbia for encouraging me not to give up, and to awaken my interest in acoustic radiosity, and Dr. Wolfgang Heidrich of the University of British Columbia for his somewhat bemused acceptance of a graduate student a generation older than himself.

I am also indebted beyond measure to Drs. Chris Jackson and Adam Harris-Harsanyi of Maya Heat Transfer Technologies (Montreal, PQ). It was their off-topic comment during a conversation on thermal analysis techniques that helped me answer the question

that this thesis addresses: What is the physical meaning of the eigensystem of a form factor matrix?

Writing a thesis is like preparing a conference paper – 90% of the writing and editing gets done within a few weeks of the deadline. With this in mind, I must thank my thesis advisor Dr. Wolfgang Heidrich and my reader Dr. Paul Lalonde of the University of British Columbia for taking time from their busy schedules to assist me on irrationally short notice. Their much-appreciated comments and suggestions appear throughout this thesis.

On a more personal note, I also thank my wife and partner Jan Ashdown and my parents Frank and Marjorie Ashdown for their continued support, understanding, and sympathy. It was at times desperately needed and very much appreciated.

Finally, I would like to thank my friends in the computer graphics and illumination engineering research communities – all six hundred or so of you. We may have exchanged no more than an e-mail message, but your work and advice contributed to the foundation on which this thesis has been built.

Thank you all, each and every one.

Ian Ashdown

April 2001

West Vancouver, BC



# Dedication

In memory of Dr. Alain Fournier.

# Preface

## Nine Years

This thesis is the culmination of a nine-year odyssey to answer a single question: What is the physical meaning of the eigensystem of a form factor matrix?

In 1992, I presented my first academic paper (“Near-Field Photometry: A New Approach”) at the annual conference of the Illuminating Engineering Society of North America. Another conference paper that caught my attention at that time was “On Setting Up and Solving Large Radiative Transfer Systems” by David DiLaura and Peter Franck.

In this paper, the authors demonstrated that the radiosity equation – a finite element method for determining the transfer of radiant flux between perfectly diffusing surfaces – could be reformulated in terms of the eigensystem of the form factor matrix associated with the equation. Unfortunately, it applied only to radiative transfer systems where all surfaces had the same area and there were no occlusions – the proverbial empty rectangular room. It was at best an interesting theoretical result with little practical application.

Being more comfortable with physical concepts than I was at that time with abstract mathematics, I asked David DiLaura whether there was a physical interpretation of their technique. He replied that the physical meaning of eigenvectors in terms of radiative transfer systems was unknown.

Their meaning was unknown ... this question became a quiet obsession for me. It was on my mind while I was writing my book, “Radiosity: A Programmer’s Perspective” [ashd94], to the extent that I wrote an (unpublished) appendix devoted to what I called

“eigenvector radiosity.” This was mostly a rewrite of the original paper; all I had to contribute then was the name.

In 1998, I decided to devote some serious thought to the question by enrolling as a computer science graduate student at my alma mater, the University of British Columbia. It was my intention to write a master’s thesis on eigenvector radiosity.

After two years of research and numerous journeys down blind alleys (including principal components analysis, the Karhunen-Loeve transform, singular value decomposition, eigenspectra of graphs, probability theory, and neural networks), I still did not have an answer. I did know however that the question had been addressed – albeit without much success – by a number of other researchers in illumination engineering, computer vision, and computer graphics. Sixty years of investigation and the question was still open.

This all changed in October 2000 when I was reminded during a discussion on thermal analysis that a radiative transfer system could be represented as a lumped electrical network.

For some unknown reason, I decided to review the original paper on the topic. (Judging by the layer of dust on the library copy, I was the first person to do so in forty years.) I casually noted that the electrical network analogy required a slightly different formulation of the radiosity equation. Apart from this, there appeared to be nothing of interest. Another blind alley.

It was on the evening of December 25<sup>th</sup>, 2000 that everything suddenly fell into place. While doing little more than doodling with the equations, I realized that there was indeed

a fundamental relationship between the electrical network analogy and the eigensystem of a form factor matrix. This was a much-appreciated Christmas present!

In retrospect, the answer is almost embarrassingly simple: 1) multiplying the form factor matrix by the diagonal matrix of the surface areas produces the conductance matrix for the network representation of the radiative transfer system; and 2) each eigenvector of the conductance matrix represents a physically realizable network (if you allow for “negative” light). The spectral decomposition theorem states that the parallel node-by-node summation of these networks is equivalent to the original network.

This revelation solved a second and more fundamental problem: reformulation of DiLaura and Franck’s equations in terms of the conductance matrix rather than the form factor matrix generalized their technique to all physically realizable radiative transfer systems.

As with any research project, answering one question raises many more. In this case, it appears that eigenvector radiosity may offer a practical technique for a number of engineering disciplines. This includes *any physical system* that can be modeled as a lumped electrical network, including gas distribution networks, structural frameworks, communication networks, nuclear reactors, neural networks, transportation systems, and more.

Due to time and space constraints, this thesis addresses eigenvector radiosity only in the related contexts of radiative transfer analysis and photorealistic image synthesis. Investigating its potential applications in other disciplines is for future consideration.

... and so it begins. Nine years is a long time to ponder any question. Regardless, it was one well worth answering.

## Thesis Outline

This thesis consists of six chapters and three appendices.

Chapter One (“Introduction”) introduces the topic of radiosity as an engineering tool, followed by a discussion of form factors and the radiosity equation. Much of the material is adapted from my book on the topic [ashd94].

Chapter Two (“Solving the Radiosity Equation”) discusses three iterative techniques: Jacobi, Gauss-Seidel, and the lesser-known Southwell iteration (better known in computer graphics as *progressive radiosity*). As discussed in Chapter Four, it is this latter technique that provides the foundation for a practical implementation of eigenvector radiosity. Again, much of the material is adapted from [ashd94].

Chapter Three (“Eigenvector Radiosity – Theory”) presents the mathematical basis of eigenvector radiosity in terms of equivalent electrical networks and their conductance matrices.

Chapter Four (“Eigenvector Radiosity – Practice”) presents an extension of eigenvector radiosity that does not require knowledge of the full form factor matrix. This allows the technique to be used in conjunction with progressive radiosity and related global illumination methods.

Chapter Five (“Numerical Experiments”) presents the application of eigenvector radiosity to several exemplars of architectural lighting design. These illustrate both the advantages and limitations of eigenvector radiosity within the context of typical illumination engineering problems. (The experiments were conducted using a modified version of the author’s commercial **Helios Radiosity Renderer**<sup>®</sup>.)

Chapter Six (“Conclusions”) discusses the results and considers future research directions for eigenvector radiosity.

Appendix A (“MATLAB Experiments”) presents a **MATLAB**<sup>®</sup> implementation of eigenvector radiosity using analytic form factors and a discussion of the results for an empty rectangular room.

Appendix B (“Eigenspectra”) reviews various theorems concerning eigenspectra and their bounds as they relate to eigenvector radiosity.

Finally, Appendix C (“Radiometric and Photometric Units”) reviews the not-always-intuitive measurements units of radiometry and photometry. The material is adapted from [ashd94].

Will eigenvector radiosity prove to be a useful computer graphics rendering technique? This remains to be seen, as there is much work to be done between the proof of concept presented in this thesis and a commercial software product.

Will eigenvector radiosity prove to be useful in other engineering disciplines? This also remains to be seen. Chapter Five demonstrates that the technique is not a panacea for all illumination engineering problems; the same issues will likely occur in other fields. Nonetheless, there are potential applications waiting to be addressed, particularly in acoustics research.

I can see that this will be an ongoing project ...

---

<sup>®</sup> **Helios Radiosity Renderer** is a registered trademark of byHeart Consultants Limited.

<sup>®</sup> **MATLAB** is a registered trademark of The MathWorks, Inc.

# Chapter 1

## Introduction

### 1.1 Radiosity – An Engineering Tool

Radiosity is a finite element method that simulates the transfer of radiant flux<sup>1</sup> between Lambertian surfaces. Originally proposed in 1926 as a calculation tool for architectural lighting design [yama26], radiosity methods have since been successfully applied to photorealistic image synthesis in computer graphics [e.g., cohe93, sill94].

While a phenomenal amount of research effort has gone into developing various radiosity methods as robust and practical rendering tools for computer graphics applications, they have also been developed into calculation tools for a variety of engineering disciplines. These include illumination engineering (architectural and roadway lighting design), thermal engineering (furnace and radiator design), aerospace engineering (spacecraft and satellite design), military hardware engineering (battlefield simulations for infrared vision and imaging radar systems), remote sensing applications (light transport in forest canopies and crops), and astronomy (stellar atmosphere modeling).

The principles of radiant flux transfer between surfaces have also been applied to the acoustic domain, where *acoustic radiosity* methods [e.g., dale96, tsin98] allow architects and engineers to analyze the impulse responses of concert halls and other large architectural spaces.

---

<sup>1</sup> Radiometric and photometric terminology is reviewed in Appendix C.

Somewhat surprisingly, relatively little research effort has been devoted to developing radiosity methods as engineering design tools. In thermal engineering, for example, many commercial applications still rely on algorithms developed in the 1950's.

The computer graphics community has focused its research efforts mostly on photorealism – the computer synthesis of images that are difficult to distinguish from photographs of physical spaces. While this makes sense in terms of computer graphics, engineers are more interested in physically correct simulations and accurate radiometric and photometric predictions.

The requirements of illumination engineering provide a good example. Given a three-dimensional CAD model of a virtual architectural space, the lighting designer must choose and position light fixtures (referred to in the industry as *luminaires*) to provide artificial illumination for the space.

Radiosity methods allow the lighting designer to produce architectural visualizations that accurately represent how light behaves in a physical environment (e.g., Figure 1.1). Because these methods are based on physical principles, they also enable the designer to prepare point-by-point illuminance distribution (isolux) plots, luminance distribution summaries, and statistical reports that accurately characterize the luminous environment. Radiosity methods therefore provide the basis for both visual communication with the client and reliable engineering calculations.

Other engineering disciplines have similar requirements. While it is often useful to visualize a design or problem, it is more important to be able to derive quantitative radiometric or photometric predictions.



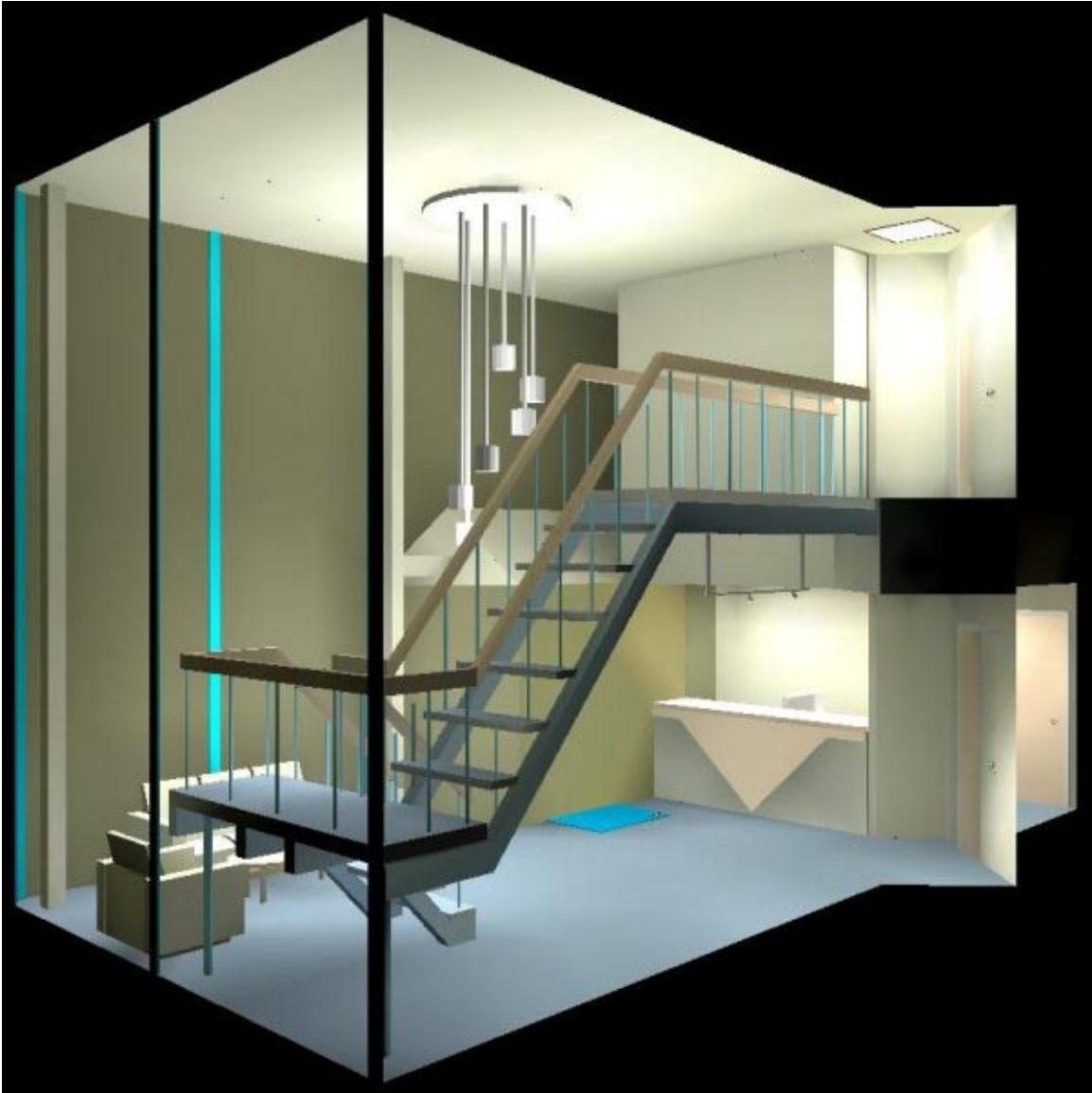


Figure 1.1 – Architectural lighting design example. (Image courtesy Matthew Walters, Thorlux Lighting.)

Engineers may not require photorealistic image synthesis, but they will demand calculation tools that are as fast as possible. Unfortunately, the radiosity method is compute-intensive.

An architectural lighting design visualization (e.g., Figure 1.1) may require several minutes of calculation time on a desktop personal computer. While this is in itself

reasonable, it is only a small portion of the design process. On viewing the associated isolux plots, the lighting designer may choose to change or reposition the luminaires. Similarly, the architect or client may request changes to the wall finishes and floor coverings.

It is therefore desirable to have the ability to quickly recalculate the luminous flux distribution within an architectural environment when such design changes are made. Ideally the calculations should take seconds rather than minutes – this would provide lighting designer and architects with a truly interactive engineering design tool.

The benefits of such a design tool would not be limited to lighting designers and illumination engineering. As another example, aerospace engineers need to analyze the solar heating characteristics of satellites. Because the distribution of solar radiant flux changes as the satellite rotates, radiosity calculations must be performed for dozens to hundreds of different orientations. Any improvement in the calculation times would be welcome.

Acoustical engineers would also benefit from an interactive acoustic radiosity design tool. Ideally, this tool would allow for real-time playback of arbitrary sound sources where both the source and the listener are moving through the environment.

The goal of this thesis is to develop a radiosity method that will meet these requirements.

## **1.2 Radiosity Explained**

The underlying principles of radiosity are most easily explained with an example. Consider the rectangular room shown in Figure 1.2. It has a luminaire mounted on the

ceiling and a table sitting on the floor. Radiosity simulates the physical *flow of light* within this environment.

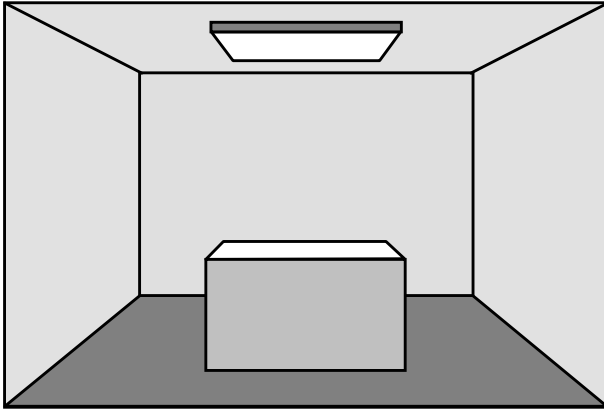


Figure 1.2 – Modeling the flow of light in a rectangular room.

In reality, this flow of light consists of photons that travel through space and optically homogeneous media (air, water, glass, and so forth) in straight lines. By modeling these paths as geometrical rays, we can (in principle) determine the distribution of radiant flux<sup>2</sup> within the room using ray-tracing techniques.

Radiosity methods take a fundamentally different approach. Each surface is divided into a mesh of elements called *patches* (Figure 1.3) – a prerequisite for any finite element method. Also, most radiosity methods make two simplifying assumptions: each surface is an ideal diffuse (Lambertian) reflector, and each light source is a Lambertian emitter. (Fortunately, most architectural finishes exhibit approximately Lambertian reflectance characteristics.)

The assumption that all surfaces are Lambertian is important – they have a constant radiance that is independent of the viewing direction. For a Lambertian reflector, the distribution of reflected radiant flux is independent of the angle of the incident flux. From

---

<sup>2</sup> For the sake of convenience, radiosity will henceforth be discussed in radiometric terms. The conversion of radiometric units to their photometric equivalents is immaterial to the discussion.

the point of view of a single patch, it does not matter where the light is coming from; if we know its irradiance and reflectance, we can easily calculate its radiant exitance and radiance.

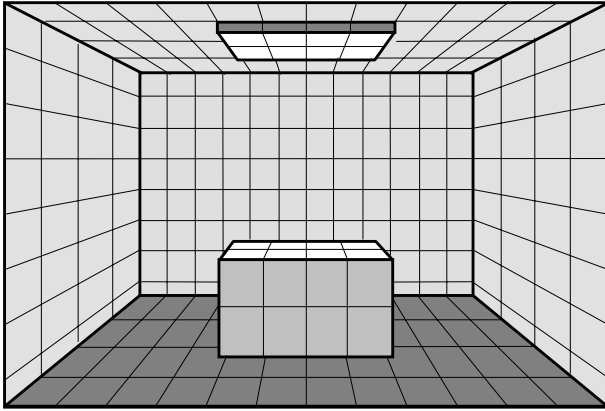


Figure 1.3 – A rectangular room with surfaces divided into patches.

We know that the distribution of flux leaving a Lambertian surface (or emitter) is given by Lambert's Cosine Law (Equation C.15). We can therefore calculate the flux emitted in any given direction by the light source (i.e., luminaire) patches. Simple geometry allows us to determine which patches are visible from each light source patch; this allows us to determine their irradiances. Each irradiated patch in turn reflects some of its incident flux back into the room. Again using Lambert's Cosine Law, we can determine the irradiances of all the patches visible to it.

This process is clearly iterative, and proceeds until all of the reflected flux is finally absorbed. If we keep a record of how much flux each patch reflects and/or emits, we end up knowing its radiant exitance  $M$ . Since the patch is Lambertian, we can divide  $M$  by  $\pi$  to determine its radiance  $L$  (from Equation C.17).

Finally, we know the geometry of each patch in the room. If we know its radiance (and consequently its luminance), we can use a 3-D graphics package to directly render a

photorealistic image of the room as a collection of shaded 3-D polygons from any viewpoint.

### 1.3 Form Factors

The most difficult aspect of developing a radiosity rendering program has nothing whatsoever to do with light *per se*. The claim in Section 1.2 that “simple geometry allows us to determine which patches are visible from each patch” is true, but only in an intuitive sense. Solving this problem analytically is anything but simple!

Stated in more formal terms, the problem is this: knowing the radiant exitance of one Lambertian patch, what portion of its flux will be received by a second patch in an environment?

Figure 1.4 shows this problem in its simplest form. The relative position and orientation of the two patches  $E_i$  and  $E_j$  is entirely arbitrary. Patch  $E_i$  is a Lambertian emitter that is emitting some quantity of flux  $\Phi_i$ , while patch  $E_j$  is receiving a portion of its emitted flux,  $\Phi_{ij}$ . The dimensionless fraction  $\Phi_{ij} / \Phi_i$  is called the *form factor* from  $E_i$  to  $E_j$ , and is denoted as either  $F_{E_i-E_j}$  or, more compactly,  $F_{ij}$ .

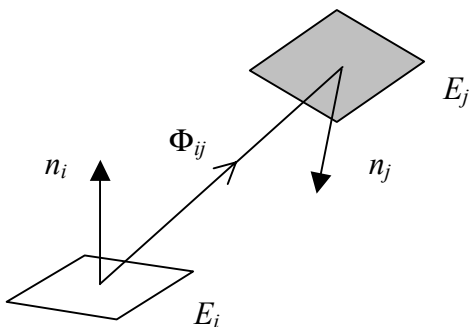


Figure 1.4 – Patch  $E_j$  receiving flux  $\Phi_{ij}$  from patch  $E_i$ .

The problem is deceptively simple. The total flux emitted by patch  $E_i$  is  $\Phi_i = M_i A_i$ , where  $M_i$  is its radiant exitance and  $A_i$  is its area. The flux received by  $E_j$  is  $\Phi_{ij} = F_{ij} \Phi_i$ . Unfortunately, calculating  $F_{ij}$  can be an extremely difficult problem in analytic geometry.

### 1.3.1 Form Factor Geometry

Consider the two differential area (that is, infinitesimally small) patches  $dE_i$  and  $dE_j$  shown in Figure 1.5, where  $dE_i$  is a Lambertian emitter. The fraction of flux emitted by  $dE_i$  that is received by  $dE_j$  is the *differential form factor* from  $dE_i$  to  $dE_j$ , denoted as  $dF_{dE_i-dE_j}$ .

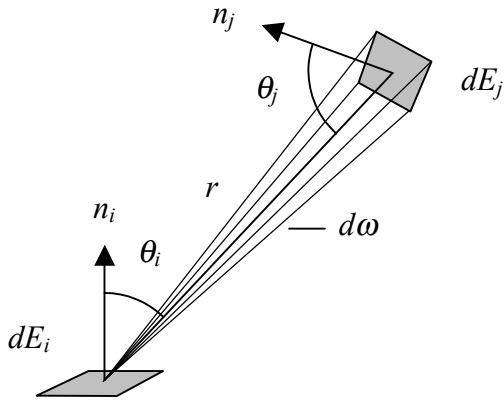


Figure 1.5 – Form factor geometry between two differential patches.

The solid angle  $d\omega$  subtended by  $dE_j$  as seen from  $dE_i$  is:

$$d\omega = \cos\theta_j dA_j / r^2 \quad (1.1)$$

where  $dA_j$  is the differential area of  $dE_j$ . From Equation C.8, the differential flux  $\Phi(\theta_i)$

leaving  $dE_i$  in the direction  $\theta_i$  is:

$$\Phi(\theta_i) = L(\theta_i) \cos\theta_i dA_i d\omega = \Phi_{ij} \quad (1.2)$$

where  $L(\theta_i)$  is the radiance of  $dE_i$  in the direction  $\theta_i$ . Since  $dE_i$  is a Lambertian emitter,  $L(\theta_i) = L_i$  (a constant) for all directions  $\theta_i$ . Substituting this and Equation 1.1 for  $d\omega$  gives:

$$\Phi_{ij} = L_i \cos\theta_i \cos\theta_j dA_i dA_j / r^2 \quad (1.3)$$

Since  $dE_i$  is a Lambertian emitter, the *total* emitted flux  $\Phi_i$  is given by Equation C.17, or:

$$\Phi_i = M_i dA_i = \pi L_i dA_i \quad (1.4)$$

The form factor  $dF_{dE_i-dE_j}$  for two differential area patches is thus:

$$dF_{dE_i-dE_j} = \frac{L_i \cos\theta_i \cos\theta_j dA_i dA_j}{\pi L_i dA_i r^2} = \cos\theta_i \cos\theta_j dA_j / \pi r^2 \quad (1.5)$$

Now, suppose that  $dE_j$  is the Lambertian emitter and  $dE_i$  is receiving its flux, namely  $\Phi_{ji}$ . We can determine the *reciprocal* differential form factor  $dF_{dE_j-dE_i}$  by simply reversing the patch subscripts in Equation 1.5. Doing so illustrates the *reciprocity relation* for form factors between any two differential areas  $dE_i$  and  $dE_j$ :

$$dA_i dF_{dE_i-dE_j} = dA_j dF_{dE_j-dE_i} \quad (1.6)$$

We can compute the form factor  $F_{dE_i-E_j}$  from a differential Lambertian emitter  $dE_i$  to a finite area  $E_j$  by integrating over the area of  $E_j$ :

$$F_{dE_i-E_j} = \int_{A_j} dF_{dE_i-dE_j} = \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_j \quad (1.7)$$

Next, we need to determine the form factor  $F_{E_i-dE_j}$  from a finite area Lambertian emitter  $E_i$  with a uniform radiance distribution across its surface to a differential area patch  $dE_j$ . We note that the total flux  $\Phi_i$  emitted by  $E_i$  is:

$$\Phi_i = M_i A_i \quad (1.8)$$

while the flux  $\Phi_{ij}$  received by  $dE_j$  is:

$$\Phi_{ij} = M_i \int_{A_i} dF_{dE_i-dE_j} dA_i \quad (1.9)$$

(Note that we are now integrating over the area of  $E_i$  rather than  $E_j$ .)

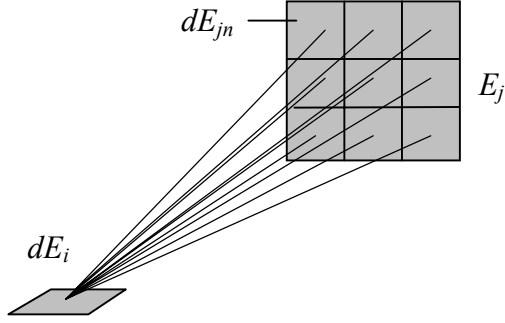


Figure 1.6 – Determining the form factor  $F_{dE_i-E_j}$  by area integration over  $E_j$ .

From our definition of a form factor, we then have:

$$F_{E_i-dE_j} = \frac{\Phi_{ij}}{\Phi_i} = \frac{M_i \int_{A_i} dF_{dE_i-dE_j} dA_i}{M_i A_i} = \frac{1}{A_i} \int_{A_i} dF_{dE_i-dE_j} dA_i \quad (1.10)$$

which yields:

$$F_{E_i-dE_j} = \frac{dA_j}{A_i} \int_{A_i} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_i \quad (1.11)$$

Of course, our interest is in patch-to-patch form factors, or the form factor from a finite area  $E_i$  to another finite area  $E_j$ . For this, we need to integrate over the areas of  $E_i$  and  $E_j$ . (In physical terms, we need to consider the contribution of each differential area of  $E_i$  to the illuminance of  $E_j$ ). The flux received by  $E_j$  is then:

$$\Phi_{ij} = M_i \int_{A_i} F_{dE_i-E_j} dA_i \quad (1.12)$$



so that the form factor  $F_{E_i-E_j}$  is:

$$F_{E_i-E_j} = \frac{M_i \int_{A_i} F_{dE_i-E_j} dA_i}{M_i A_i} = \frac{1}{A_i} \int_{A_i} F_{dE_i-E_j} dA_i \quad (1.13)$$

From Equation 1.7, this yields the double area integral equation:

$$F_{E_i-E_j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} dA_j dA_i \quad (1.14)$$

The reciprocal form factor  $F_{E_j-E_i}$  is obtained by reversing the patch subscripts. This demonstrates that the reciprocity relation (Equation 1.6) also holds true for finite area patches. In other words:

$$A_i F_{ij} = A_j F_{ji} \quad (1.15)$$

The importance of the reciprocity relation cannot be overstated. It says that if we can somehow calculate the form factor  $F_{ij}$  from an patch  $E_i$  to another patch  $E_j$ , then we can trivially calculate the reciprocal form factor  $F_{ji}$ . This is a key concept in radiosity theory.

The above equations implicitly assume that the two patches  $E_i$  and  $E_j$  are fully visible to each other. In a complex environment, two patches may be partially hidden by one or more occluding objects. If so, then a suitable term must be added to account for the occlusions, such as:

$$F_{E_i-E_j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} HID_{ij} dA_j dA_i \quad (1.16)$$

where the term  $HID_{ij}$  accounts for the possible occlusion of each point of patch  $E_j$  as seen from each point of patch  $E_i$ .

There are unfortunately no practical analytical methods for solving this equation; it must typically be solved using numerical quadrature or Monte Carlo ray tracing methods [e.g., cohe93].

### 1.3.2 Form Factor Properties

A form factor is a dimensionless constant representing the fraction of flux emitted by one surface patch that is received by another. It takes into account the shape and relative orientation of both surfaces and the presence of any obstructions, but is otherwise independent of any surface properties.

Form factors were first developed for use in thermal and illumination engineering, where they have been variously called *shape*, *configuration*, *and angle* and *view* factors. The thermal engineering literature is filled with discussions of *form factor algebra*, of which the reciprocity relation is only one example. Most of these discussions relate to a time when form factors were calculated by hand. Some properties, however, are still useful. For example, the *summation relation* states that:

$$\sum_{j=1}^n F_{ij} = 1 \quad (1.17)$$

for any patch  $E_i$  in a *closed* environment with  $n$  patches. (A closed environment is one where all of the flux emitted by any one patch must be received by one or more patches in the environment. In other words, none of it can escape into free space.) This summation includes the form factor  $F_{ii}$ , which is defined as the fraction of flux emitted by  $E_i$  that is also directly received by  $E_i$ . Clearly,  $F_{ii}$  can only be nonzero if  $E_i$  is concave. Thus:

$$F_{ii} = 0 \quad \text{if } E_i \text{ is planar (i.e., flat) or convex, and}$$

$F_{ii} \neq 0$  if  $E_i$  is concave

Most radiosity methods model surface patches as planar polygons, so that  $F_{ii}$  is always zero.

#### 1.4 The Radiosity Equation

If patches  $E_i$  and  $E_j$  are both Lambertian surfaces, the form factor  $F_{ij}$  indicates the fraction of flux emitted by  $E_i$  that is received by  $E_j$ . Similarly, the reciprocal form factor  $F_{ji}$  indicates the fraction of flux emitted by  $E_j$  that is received by  $E_i$ . However, form factors in themselves do not consider the flux that is subsequently reflected from these receiving patches.

Remember that we are trying to determine the radiant exitance  $M_i$  of each patch  $E_i$  in an  $n$ -patch environment. This exitance is clearly due to the flux initially emitted by the patch *plus* that reflected by it. The reflected flux comes from all of the other patches  $E_j$  visible to  $E_i$  in the environment.

Consider any patch  $E_j$  that is fully visible to  $E_i$ . The flux leaving patch  $E_j$  is  $\Phi_j = M_j A_j$ . The fraction of this flux received by patch  $E_i$  is  $\Phi_{ji} = M_j A_j F_{ji}$ . Of this, the flux subsequently reflected by  $E_i$  is  $\rho_i M_j A_j F_{ji}$ , where  $\rho_i$  is the reflectance of  $E_i$ . This gives us:

$$M_{ij} = \rho_i M_j A_j F_{ji} / A_i \quad (1.18)$$

where  $M_{ij}$  is defined as the exitance of  $E_i$  due to the flux received from  $E_j$ . Using the reciprocity relation, we can rewrite this as:

$$M_{ij} = \rho_i M_j F_{ij} \quad (1.19)$$

To calculate the final exitance  $M_i$  of patch  $E_i$ , we must consider the flux received by  $E_i$  from all other patches  $E_j$ . Thus:

$$M_i = M_{oi} + \rho_i \sum_{j=1}^n M_j F_{ij} \quad (1.20)$$

where  $M_{oi}$  is the initial exitance of patch  $E_i$  due to its emitted flux only. Rearranging terms results in:

$$M_{oi} = M_i - \rho_i \sum_{j=1}^n M_j F_{ij} \quad (1.21)$$

We can express this equation for all the patches  $E_1$  through  $E_n$  as a set of  $n$  simultaneous linear equations:

$$\begin{aligned} M_{o1} &= M_1 - (\rho_1 M_1 F_{11} + \rho_1 M_2 F_{12} + \dots + \rho_n M_n F_{1n}) \\ M_{o2} &= M_2 - (\rho_2 M_1 F_{21} + \rho_2 M_2 F_{22} + \dots + \rho_n M_n F_{2n}) \\ &\dots \\ M_{on} &= M_n - (\rho_n M_1 F_{n1} + \rho_n M_2 F_{n2} + \dots + \rho_n M_n F_{nn}) \end{aligned} \quad (1.22)$$

which we can write in matrix form as:

$$\begin{bmatrix} M_{o1} \\ M_{o2} \\ \dots \\ M_{on} \end{bmatrix} = \begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \dots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \dots & -\rho_2 F_{2n} \\ \dots & \dots & \dots & \dots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \dots & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \dots \\ M_n \end{bmatrix} \quad (1.23)$$

In matrix notation, this can be succinctly expressed as:

$$\mathbf{M}_o = (\mathbf{I} - \mathbf{T})\mathbf{M} \quad (1.24)$$

where  $\mathbf{I}$  is the  $n \times n$  identity matrix,  $\mathbf{M}$  is the final  $n \times 1$  exitance vector,  $\mathbf{M}_o$  is the initial  $n \times 1$  exitance vector, and  $\mathbf{T}$  is an  $n \times n$  matrix whose  $(i,j)^{\text{th}}$  element is  $\rho_i F_{ij}$ . The matrix  $(\mathbf{I} - \mathbf{T})$  is called the *radiosity matrix*.

This is the elegantly simple *radiosity equation*: a set of simultaneous linear equations involving only surface reflectances, patch form factors, and patch exitances. Solving

these equations provides us with the radiant exitance, radiance, and ultimately luminance of every patch in the environment it describes<sup>3</sup>.

Given a set of initial patch exitances  $M_{oi}$ , we need to solve the radiosity equation for the final patch exitances  $M_i$ . The matrix order is typically too large for direct methods such as Gaussian elimination. However, the radiosity equation is ideally suited for iterative techniques such as the Jacobi and Gauss-Seidel methods [e.g., golu96]. These methods are guaranteed to converge to a solution, because the matrix is always strictly diagonally dominant for flat and convex patches. That is,  $\rho_i F_{ij}$  is always less than one, while  $F_{ii}$  is always zero. Furthermore, they converge very quickly, typically in six to eight iterations [cohe86]. We will examine these methods and a more powerful and useful variation called Southwell iteration (also known as “progressive radiosity”) in the next chapter.

---

<sup>3</sup> Solving the radiosity equation for an environment is equivalent to determining its “energy balance.” The amount of radiant flux reflected and absorbed by a patch must equal the amount of flux incident on its surface. Flux is energy per unit time. If this balance is not maintained, the patch will steadily accumulate or lose energy over time. The final solution to the radiosity equation therefore ensures that the flow of energy is balanced for all patches in the environment.

# Chapter 2

## Solving the Radiosity Equation

### 2.1 Full Radiosity

We saw in Chapter One that the radiosity equation is a system of  $n$  linear equations of the form:

$$\begin{bmatrix} M_{o1} \\ M_{o2} \\ \dots \\ M_{on} \end{bmatrix} = \begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \dots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \dots & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \dots \\ M_n \end{bmatrix} \quad (2.1)$$

where  $n$  is the number of surface patches in the environment. We know the initial exitance vector; its entries  $M_{oi}$  will be mostly zeroes. The only non-zero entries are for those patches representing Lambertian light sources. We also know the reflectivity  $\rho_i$  of each patch, and we can estimate the form factor  $F_{ij}$  between any two patches  $i$  and  $j$ . All we have to do to obtain the final exitances  $M_i$  is to solve these equations.

Most environments result in linear systems that are far too large to solve using direct methods such as Gaussian elimination. The classic alternative is to use iterative techniques such as the Gauss-Seidel method [e.g. golu96]. This was the original approach taken by Goral et al. [gora84]. Baum et al. [baum89] referred to it as the *full radiosity* algorithm.

Unfortunately, this gives us a radiosity algorithm with  $\mathbf{O}(n^2)$  time and space complexity. Even a moderately complex environment with 50,000 patches can easily

consume one to ten gigabytes of memory for its form factors and take days of CPU time to compute a single image. We clearly need a better approach.

What we really want is an algorithm that consumes a minimal amount of memory and that generates a reasonable approximation of the final image almost immediately. More generally, we need to maintain a careful balance between the requirement for photorealistic images and the demands of interactive computing.

In a perfect world, our algorithm would generate a reasonable first approximation (that can be displayed as a bitmap image) and then progressively and gracefully refine the solution until it reaches its final form. This essentially describes how iterative techniques work, except that we need a much more effective algorithm than the Gauss-Seidel method.

The great surprise is that such an algorithm actually exists. Before examining it, however, we should review the basic principles of iterative techniques.

## 2.2 Iterative Techniques

Referring to Equation 1.24, we can express Equation 2.1 more succinctly in matrix notation as:

$$\mathbf{M}_o = (\mathbf{I} - \mathbf{T})\mathbf{M} \quad (2.2)$$

where  $\mathbf{I}$  is the identity matrix and  $\mathbf{T}$  is:

$$\mathbf{T} = \begin{bmatrix} \rho_1 F_{11} & \rho_1 F_{12} & \cdots & \rho_1 F_{1n} \\ \rho_2 F_{21} & \rho_2 F_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \rho_n F_{n1} & \rho_n F_{n2} & \cdots & \rho_n F_{nn} \end{bmatrix} \quad (2.3)$$

$$= \begin{bmatrix} \rho_1 & 0 & \cdots & 0 \\ 0 & \rho_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \rho_n \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1n} \\ F_{2n} & F_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ F_{n1} & F_{n2} & \cdots & F_{nn} \end{bmatrix} = \mathbf{RF}$$

where  $\mathbf{R}$  is the (diagonal) reflectance matrix and  $\mathbf{F}$  is the form factor matrix.

If we consider the radiosity matrix ( $\mathbf{I}-\mathbf{T}$ ) as an  $n \times n$  matrix – call it  $\mathbf{K}$  for convenience – we have a linear system of the form:

$$\mathbf{M}_o = \mathbf{KM} \quad (2.4)$$

which can be solved using any one of several iterative techniques.

A quick review of iterative techniques for solving linear systems may be in order. Suppose we are given a system of linear equations such as:

$$\mathbf{b} = \mathbf{Ax}$$

where  $\mathbf{x}$  is the unknown  $n \times 1$  vector,  $\mathbf{A}$  is a square  $n \times n$  matrix and  $\mathbf{b}$  is a known  $n \times 1$  vector. Most iterative techniques convert this system into an equivalent system with the form:

$$\mathbf{x} = \mathbf{Qx} + \mathbf{c}$$

where the  $n \times n$  matrix  $\mathbf{Q}$  and the  $n \times 1$  vector  $\mathbf{c}$  are derived from  $\mathbf{A}$  and  $\mathbf{b}$ . The details of the derivation depend on the particular iterative technique.

To solve for  $\mathbf{x}$ , we start with an initial  $n \times 1$  vector  $\mathbf{x}^{(0)}$  that hopefully approximates the final solution. At worst, it can have entirely random values for its elements. With it, we can generate a sequence of vectors  $\mathbf{x}^{(k)}$  by repeatedly computing:

$$\mathbf{x}^{(k)} = \mathbf{Qx}^{(k-1)} + \mathbf{c}, \quad k = 1, \dots$$



This is the iterative component of the technique. The sequence of vectors  $\mathbf{x}^{(k)}$  will be such that the elements of the vector either *converge* to those of the unknown vector  $\mathbf{x}$ , or else *diverge* into some random vector, as  $k$  increases.

While it is unlikely that  $\mathbf{x}^{(k)}$  will exactly equal  $\mathbf{x}$  for any finite value of  $k$ , the error between them will tend to grow progressively smaller as  $k$  increases (and if the sequence converges). This means that we can stop when:

$$\frac{|\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}|}{|\mathbf{x}_i^{(k-1)}|} \leq \text{threshold}, \quad i = 1, \dots, n$$

for some “threshold” value. At this point, the approximate solution vector  $\mathbf{x}^{(k)}$  is such that the fractional error between it and the unknown vector  $\mathbf{x}$  is guaranteed to be equal to or less than this value for each of its elements. The iterative method is then said to have *converged* to an acceptable solution.

Of critical importance to the user is the *convergence rate*. That is, what value of  $k$  is needed in order to attain an acceptable solution? This is determined by the characteristics of the chosen iterative method, the choice of  $\mathbf{x}^{(0)}$ , and the particular problem being solved.

There are two issues of concern here. First, there are linear systems where the solution vector diverges rather than converges to a solution. Fortunately, the radiosity equation is guaranteed to converge to a solution using either the Jacobi or Gauss-Seidel iterative methods. (The sum of any row of form factors is equal to or less than unity by virtue of the summation relation (Equation 1.17), and each form factor is multiplied by a reflectance value  $\rho$  that is less than unity. Also, the main diagonal term of  $\mathbf{K}$  in Equation 2.4 is always unity, since  $F_{ii} = 0$  for all planar or convex patches. Thus,  $\mathbf{K}$  is strictly diagonally dominant, which guarantees convergence for any choice of  $\mathbf{M}^{(0)}$ .)

Second, we need to consider what our choice of  $\mathbf{M}^{(0)}$  should be. The closer it is to the unknown final exitance vector  $\mathbf{M}$ , the more quickly our chosen iterative method will converge. Of course, the only *a priori* information we have concerns the initial exitances of the patches representing Lambertian light sources<sup>1</sup>. In other words, our best choice is to assign the initial exitance vector  $\mathbf{M}_o$  to  $\mathbf{M}^{(0)}$ . Interestingly enough, this choice has some physical significance.

### 2.2.1 Follow the Bouncing ... Light

Returning to Equation 2.2, suppose we rearrange it slightly to solve for  $\mathbf{M}$ . We then have:

$$\mathbf{M} = (\mathbf{I} - \mathbf{T})^{-1} \mathbf{M}_o \quad (2.5)$$

Again, we cannot solve this equation directly, since calculating the inverse of a matrix is rarely an easy task. However, we can approximate it with a MacLaurin power series expansion. It can be shown that:

$$\frac{1}{(1-x)} = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + \dots \quad (2.6)$$

which converges for  $-1 < x < 1$ . There is a similar series expansion for matrices [e.g., golu96]:

$$(\mathbf{I} - \mathbf{T})^{-1} = \mathbf{I} + \mathbf{T} + \mathbf{T}^2 + \mathbf{T}^3 + \dots \quad (2.7)$$

which gives us:

$$\mathbf{M} = \mathbf{M}_o + \mathbf{T}\mathbf{M}_o + \mathbf{T}^2\mathbf{M}_o + \mathbf{T}^3\mathbf{M}_o + \dots \quad (2.8)$$

---

<sup>1</sup> If the light sources have non-Lambertian radiant flux distributions, it is necessary to first calculate the direct irradiance and consequent initial radiant exitance of every patch in the environment due to these light sources. Each such irradiated patch then becomes a “secondary” Lambertian light source.

that converges if the spectral radius of  $\mathbf{T}$  (i.e., the absolute value of its largest eigenvalue) is less than one. Fortunately, this condition is true for any physically possible radiosity equation [e.g., heck91].

There is an important physical significance to Equation 2.8 [e.g., kaji86]. Each successive term  $\mathbf{T}^k \mathbf{M}$  represents the  $k^{\text{th}}$  bounce of the initially emitted light. The term  $\mathbf{M}_o$  represents the initial flux (i.e., the direct illumination),  $\mathbf{T}\mathbf{M}_o$  represents the first bounce component,  $\mathbf{T}^2\mathbf{M}_o$  the second bounce and so on. We can intuitively see this by observing that the patch reflectances  $\rho$  are multiplied with each successive bounce. This represents the accumulating light losses due to absorption.

We can express Equation 2.8 in its iterative form as:

$$\mathbf{M}^{(k)} = \mathbf{T}\mathbf{M}^{(k-1)} + \mathbf{M}_o, \quad k > 0 \quad (2.9)$$

In other words, the behavior of light flowing through an environment is itself an iterative method. Moreover, the initial exitance vector  $\mathbf{M}_o$  serves as its initial estimate to the final exitance vector  $\mathbf{M}$ .

Comparing Equation 2.9 to iterative techniques for solving linear systems, it becomes clear why the radiosity equation always converges to a solution when we apply these techniques. To do otherwise – that is, for the approximate solution vector  $\mathbf{M}^{(k)}$  to diverge with increasing values of  $k$  – would require the total quantity of light in an environment to *increase* with each successive bounce. This would in turn contravene the energy balance discussed in Section 1.4.

There is in fact only one iterative technique that faithfully models the physical reality of light's behavior as expressed by Equation 2.9. It is the Jacobi iterative method, the simplest iterative technique for solving systems of linear equations. While it may not be

necessary for our development of a practical algorithm for solving the radiosity equation, we should ask how the Jacobi method works for two reasons.. First, it will provide us with a better understanding of how and why iterative techniques work. More important, however, the Jacobi method offers a fascinating and instructive insight into the physical reality of the radiosity equation.

### 2.2.2 Jacobi Iteration

The Jacobi method splits (or *decomposes*) an  $n \times n$  matrix  $\mathbf{A}$  into a *diagonal* matrix  $\mathbf{D}$ , a strictly *lower diagonal* matrix  $-\mathbf{L}$  and a strictly *upper diagonal* matrix  $-\mathbf{U}$ . Written in matrix form, this becomes:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \mathbf{D} - \mathbf{L} - \mathbf{U} = \quad (2.10)$$

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ -a_{n1} & -a_{n2} & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & 0 & \cdots & -a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

From this, we get:

$$\mathbf{Ax} = (\mathbf{D} - \mathbf{L} - \mathbf{U})\mathbf{x} = \mathbf{b} \quad (2.11)$$

which becomes:

$$\mathbf{Dx} = (\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{b} \quad (2.12)$$

and so:

$$\mathbf{x} = \frac{(\mathbf{L} + \mathbf{U})}{\mathbf{D}}\mathbf{x} + \frac{\mathbf{b}}{\mathbf{D}} \quad (2.13)$$

The Jacobi iterative method is thus:

$$\mathbf{x}^{(k)} = \frac{(\mathbf{L} + \mathbf{U})}{\mathbf{D}} \mathbf{x}^{(k-1)} + \frac{\mathbf{b}}{\mathbf{D}} \quad (2.14)$$

or, expressed in its more familiar form:

$$x_i^{(k)} = \frac{\sum_{j=1, j \neq i}^n (-a_{ij} x_j^{(k-1)}) + b_i}{a_{ii}}, \quad i = 1, \dots, n \quad (2.15)$$

In plain English, this equation states that we can solve each element  $x_i^{(k)}$  of our approximate solution vector  $\mathbf{x}^{(k)}$  by using the values of all the other elements  $x_j^{(k-1)}$  of our previously calculated solution vector.

### 2.2.3 Modeling Light

The Jacobi iterative method models the flow of light in an environment. We can confirm this by deriving Equation 2.9 in terms of the Jacobi iteration. Following the development of the Jacobi method above, we start with Equation 2.2 and decompose  $\mathbf{T}$  into a diagonal matrix  $\mathbf{T}_D$ , a strictly lower diagonal matrix  $-\mathbf{T}_L$  and a strictly upper diagonal matrix  $-\mathbf{T}_U$  to get:

$$(\mathbf{I} - \mathbf{T}) = \mathbf{I} - \mathbf{T}_D + \mathbf{T}_L + \mathbf{T}_U \quad (2.16)$$

and thus:

$$\mathbf{M}_0 = (\mathbf{I} - \mathbf{T}_D + \mathbf{T}_L + \mathbf{T}_U) \mathbf{M} \quad (2.17)$$

This becomes:

$$(\mathbf{I} - \mathbf{T}_D) \mathbf{M} = -(\mathbf{T}_L + \mathbf{T}_U) \mathbf{M} + \mathbf{M}_0 \quad (2.18)$$

and:

$$\mathbf{M} = \frac{-(\mathbf{T}_L + \mathbf{T}_U)}{(\mathbf{I} - \mathbf{T}_D)} \mathbf{M} + \frac{\mathbf{M}_0}{(\mathbf{I} - \mathbf{T}_D)} \quad (2.19)$$

This is equivalent to the Jacobi iterative method presented in Equation 2.14. However, the form factor  $F_{ii}$  for planar or convex patches is always zero, which means each diagonal element of  $\mathbf{T}$  equals zero and so  $(\mathbf{I} - \mathbf{T}_D) = \mathbf{I}$ . Also,  $\mathbf{T} = -(\mathbf{T}_L + \mathbf{T}_U)$ . Thus:

$$\mathbf{M} = -(\mathbf{T}_L + \mathbf{T}_U)\mathbf{M} + \mathbf{M}_o = \mathbf{T}\mathbf{M} + \mathbf{M}_o \quad (2.20)$$

which results in the Jacobi iterative method:

$$\mathbf{M}^{(k)} = \mathbf{T}\mathbf{M}^{(k-1)} + \mathbf{M}_o, \quad k > 0 \quad (2.21)$$

for solving the radiosity equation. This is identical to Equation 2.9. Referring to Equation 2.3, this becomes:

$$\mathbf{M}^{(k)} = \mathbf{M}_o + \mathbf{R}\mathbf{F}\mathbf{M}^{(k-1)} \quad (2.22)$$

which, expressed in the form of Equation 2.15, is:

$$M_i^{(k)} = M_{oi} + \rho_i \sum_{j=1}^n F_{ij} M_j^{(k-1)}, \quad i = 1, \dots, n \quad (2.23)$$

This is the radiosity equation that we saw in Chapter One (Equation 1.20), expressed as an iterative method.

#### 2.2.4 Gauss-Seidel Iteration

The problem with Jacobi iteration is that it is often slow to converge to a solution. The Gauss-Seidel iterative method takes a simple but effective approach to improving this situation. We saw in Equation 2.15 that the Jacobi method calculates the value of each element  $x_i^{(k)}$  in sequence by using the values of the other elements from  $\mathbf{x}^{(k-1)}$ . Since the elements  $x_j^{(k)}$  (where  $j < i$ ) have already been calculated and are presumably closer approximations to the final solution vector elements than their  $x_j^{(k-1)}$  counterparts, why not use these values instead when calculating  $x_i^{(k)}$ ?

This is exactly what the Gauss-Seidel method does. Its iterative equation is:

$$\mathbf{x}^{(k)} = \frac{\mathbf{U}}{(\mathbf{D}-\mathbf{L})} \mathbf{x}^{(k-1)} + \frac{\mathbf{b}}{(\mathbf{D}-\mathbf{L})}, \quad k > 0 \quad (2.24)$$

or, expressed in its more familiar form:

$$x_i^{(k)} = \frac{\sum_{j=1}^{i-1} -a_{ij}x_j^{(k)} + \sum_{j=i}^n -a_{ij}x_j^{(k-1)} + b_i}{a_{ii}}, \quad i = 1, \dots, n \quad (2.25)$$

A derivation of Equation 2.24 can be found in most elementary linear algebra and numerical analysis texts [e.g., burd85].

The Jacobi method can be seen in terms of modeling light bouncing from surface to surface in an environment. This is not the case for the Gauss-Seidel method. In a sense, it tries to anticipate the light each surface will receive from the next iteration of reflections. There is no physical analogue to this process, but it does work in that the Gauss-Seidel method usually converges more quickly than the Jacobi method does. Cohen and Greenberg [cohe85] found that the Gauss-Seidel method solved the radiosity equation for typical environments in six to eight iterations.

### 2.2.5 Full Radiosity Disadvantages

When it was first introduced to the computer graphics research community by Goral et al. [gora84] and Nishita and Nakamae [nish85], radiosity rendering was for the most part viewed as an interesting mathematical curiosity. The Jacobi and Gauss-Seidel methods have a time complexity of  $\mathbf{O}(n^2)$  for each iteration. Given the available computer technology at the time, this made the full radiosity algorithm an impractical rendering technique for all but the simplest of environments.

Another disadvantage of full radiosity is that it requires storage for  $n^2 / 2$  form factors [e.g., Ashd94]. This means that the memory space complexity of the full radiosity

algorithm is  $\mathbf{O}(n^2)$  as well. We could possibly avoid this requirement by recomputing form factors “on the fly” for each patch during each iteration. However, the high cost of form factor determination (typically 90 percent of the run-time cost) means that we would have to wait much longer between each iteration. This is exactly what we are trying to avoid. We need to obtain an initial image as quickly as possible.

We can gain some relief by substructuring the environment into a two-level hierarchy of patches and elements [cohe86]. This brings both the time and space complexities down to  $\mathbf{O}(mn)$  for  $n$  patches and  $m$  elements. Substructuring is a useful technique, but we can do better.

### 2.3 Shooting Versus Gathering

There is an interesting and instructive physical interpretation of the Jacobi and Gauss-Seidel methods. We can think of each execution of Equation 2.15 (Jacobi) or 2.25 (Gauss-Seidel) as being one *step*; it takes  $n$  steps to complete one iteration of the method. At each step, we are updating the estimated exitance of one patch by processing one row of the radiosity equation. For the Jacobi method, this is Equation 2.23, repeated here as:

$$M_i^{(k)} = M_{oi} + \rho_i \sum_{j=1}^n F_{ij} M_j^{(k-1)}, \quad i = 1, \dots, n \quad (2.26)$$

We can show this diagrammatically as:

$$\begin{bmatrix} \cdot \\ x \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ x \\ \cdot \\ \cdot \end{bmatrix} + \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ x & x & x & x \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix} \quad (2.27)$$

The physical interpretation of this process is straightforward: we are simply summing the contribution of flux from all the other patches in the environment to the exitance of the current patch. Looking at Figure 2.1 and referring to Equation 2.26, each patch  $E_j$  has an



exitance  $M_j$  and an area  $A_j$ . Referring to Equation 2.26, the portion of the flux  $\Phi_j$  emitted by  $E_j$  that is received by  $E_i$  is:

$$\Phi_{ij} = M_j A_j F_{ji} \quad (2.28)$$

The amount of exitance  $\Delta M_i$  of  $E_i$  that is due to this flux subsequently being reflected by  $E_i$  is thus:

$$\Delta M_i = \rho_i \Phi_{ji} / A_i = \rho_i M_j A_j F_{ji} / A_i \quad (2.29)$$

However, we can apply the reciprocity relation  $A_i F_{ij} = A_j F_{ji}$  (Section 1.5.1) to obtain:

$$\Delta M_i = \rho_i M_j F_{ij} \quad (2.30)$$

More colloquially, this can be seen as the current patch  $E_i$  *gathering* exitance from all other patches  $E_j$  in the environment in order to determine its exitance due to these patches. The term  $M_{oi}$  in Equation 2.26 simply accounts for any initial exitance of  $E_i$ . This will be non-zero only if  $E_i$  is a Lambertian light source.

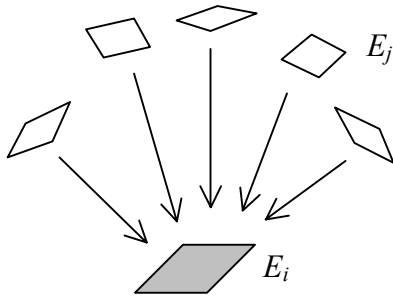


Figure 2.1 – Gathering flux from the environment.

It may be somewhat difficult to visualize exitance being transferred between patches. It becomes clearer when we multiply both sides of Equation 2.30 by  $A_i$  to obtain:

$$\Delta \Phi_i = \Delta M_i A_i = \rho_i M_j F_{ij} A_i \quad (2.31)$$

Again applying the reciprocity relation, we get:

$$\Delta \Phi_i = \rho_i M_j F_{ji} A_j = \rho_i F_{ji} \Phi_j \quad (2.32)$$

which shows that we are in fact gathering and subsequently reflecting radiant flux. Equation 2.30 is more useful in terms of Equation 2.26, however, and so we “gather” exitance to  $E_i$ . The difference is solely semantic.

A number of authors have loosely referred to this process as gathering “energy.” However, the physical quantity being discussed is radiant exitance (i.e., watts per unit area) times area. This is *power*, or radiant flux. Energy is “gathered” only in the sense that solving the radiosity equation balances the *flow* of energy (which is power) between elements in the environment.

The problem with this approach is that it can be excruciatingly slow. Consider a complex environment with perhaps 50,000 patches. Using the Jacobi or Gauss-Seidel method, we must perform one complete iteration before we have an image of the first bounce of light from the environment. That means we must execute Equation 2.26 50,000 times! This clearly does not satisfy our requirement for an “immediate but approximate” image.

This is where the physical interpretation becomes useful. If we think for a moment about how light flows in an environment, it becomes evident that we should be interested in those elements that emit or reflect the most light. It logically does not matter in what order we consider the distribution of light from patch to patch, as long as we eventually account for it being completely absorbed.

This leads to an entirely different paradigm. Given an environment with one or more light sources, we can think of them *shooting* flux to the other patches (Figure 2.2). These patches then become in effect secondary light sources, shooting some of the flux they receive back into the environment. By always selecting the patch that has the greatest

amount of flux to “shoot,” we will drastically improve our convergence rate. Again, it makes intuitive sense that the more quickly the light is absorbed, the more quickly our as-yet-unspecified iterative method will converge to a solution.

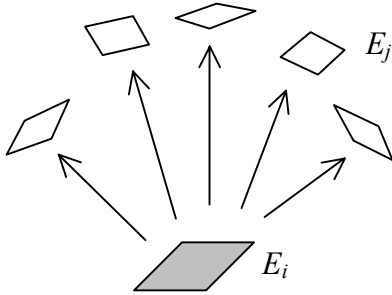


Figure 2.2 – Shooting flux into the environment.

It also becomes evident that this idea answers our need for both an immediate image and progressive convergence to the final solution. By shooting flux from one patch to all other patches in the environment, we immediately obtain an initial estimate for all patch exitances. This occurs in one step rather than a complete iteration. In fact, the concept of iteration no longer applies, for we may end up choosing one patch several times before we cycle through the entire set. It all depends on which patch currently has the greatest amount of flux to shoot.

Of course, we also obtain improved estimates for all the patch exitances at each step. This means that the rendered image will continuously and gracefully converge to the final photorealistic image.

Now, all we have to do is to express this idea in the form of a practical algorithm.

#### 2.4 Progressive Radiosity (Southwell Iteration)

What we are looking for is the *progressive radiosity* algorithm [cohe88]. Based on the concept of shooting flux, it offers not only an immediate image with continuous and graceful convergence, but also  $\mathbf{O}(n)$  space complexity. Given an environment with  $n$

patches, it requires memory space for only  $n$  form factors. Even better, it can generate an initial image almost immediately, and can generate if necessary updated images after each step (as opposed to each iteration).

So how does it work? To shoot flux or exitance back into the environment, we simply reverse the subscripts of Equation 2.30. For exitance, this becomes:

$$\Delta M_j = \rho_j M_i F_{ij} \frac{A_i}{A_j} = \rho_j M_i F_{ji} \quad (2.33)$$

Multiplying both sides of this equation by the area of patch  $E_j$  gives us the equation for shooting flux.

Unlike the full radiosity algorithm (i.e., Equation 2.26), this equation acts on one column of the radiosity equation at a time. Shown diagrammatically, this is:

$$\begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix} = \begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix} + \begin{bmatrix} . \\ x \\ . \\ . \end{bmatrix} \begin{bmatrix} . & x & . & . \\ . & x & . & . \\ . & x & . & . \\ . & x & . & . \end{bmatrix} \quad \text{for all patches } E_j \quad (2.34)$$

This means we can now display an image of the environment whenever one column of the radiosity equation has been processed. This has a time complexity of  $\mathbf{O}(n)$  as opposed to  $\mathbf{O}(n^2)$  for the full radiosity algorithm.

The progressive radiosity algorithm proceeds as follows. First, we assign an “unsent exitance” value  $\Delta M_i^{unsent}$  to each patch in the environment. This is in addition to its final exitance  $M_i$ , which we are trying to determine. The amount of flux each patch has to shoot is  $\Delta M_i^{unsent}$  times its area,  $A_i$ . Initially, only the patches representing Lambertian light sources will have non-zero values of flux, and so  $\Delta M_i^{unsent}$  is initialized to  $M_{oi}$ . The final exitance values  $M_i$  are also initialized to  $M_{oi}$ .

Choosing the patch  $E_i$  with the greatest amount of flux (not exitance) to shoot, we execute Equation 2.33 for every other patch  $E_j$  in the environment. Each of these patches “receives” a delta exitance  $\Delta M_j$ ; this value is added to both its unspent exitance  $\Delta M_j^{unspent}$  and its final exitance  $M_j$ .

After the flux has been shot to every patch  $E_j$ ,  $\Delta M_i^{unspent}$  is reset to zero. This patch can only shoot again after receiving more flux from other patches during subsequent steps.

This process continues until the total amount of flux remaining in the environment is less than some predetermined fraction  $\epsilon$  of the original amount, or:

$$\sum_{i=1}^n \Delta M_i^{unspent} A_i \leq \epsilon \quad (2.35)$$

At this point, the algorithm is considered to have converged to a solution.

Expressing this in pseudocode, we have:

```

FOR each patch  $i$ 
   $M_i = \Delta M_i^{unspent} = M_{oi}$ 
ENDFOR
WHILE  $\sum_{i=1}^n \Delta M_i^{unspent} A_i > \epsilon$ 
  Select patch  $i$  with greatest unspent flux  $\Delta M_i^{unspent} A_i$ 
  Calculate all form factors  $F_{ij}$ 
  FOR each patch  $j$ 
     $\Delta M = \rho_j F_{ij} \Delta M_i^{unspent} \frac{A_i}{A_j}$ 
     $\Delta M_j^{unspent} = \Delta M_j^{unspent} + \Delta M$ 
     $M_j = M_j + \Delta M$ 
  ENDFOR
   $\Delta M_i^{unspent} = 0$ 

```

## ENDWHILE

Figure 2.3 – Progressive radiosity algorithm

Progressive radiosity does not – repeat, *does not* – require any less time to *completely* solve the radiosity equation to some vanishingly small margin of error. It is an iterative approach that, like full radiosity, progressively refines the patch exitances as it converges to a solution. However, its overwhelming advantage is that usable images can be displayed almost immediately, and that each succeeding image takes much less time to calculate.

We still have the form factors to contend with. However, we only need to calculate the  $n$  form factors  $F_{ij}$  from the current patch  $E_i$  to all other patches  $E_j$  between displaying images. We have to recompute these form factors on the fly for each step of the progressive radiosity algorithm. However, the convergence rate is *much* faster than it is for full radiosity. Cohen et al. [cohe88] compared progressive refinement and full radiosity algorithms using an environment consisting of 500 patches and 7,000 elements. The progressive radiosity implementation converged to a visually acceptable image after approximately 100 steps. At this point, the full radiosity implementation was only 20 percent of its way through its first iteration.

Gortler and Cohen [gort93] established that the progressive radiosity algorithm is a variant of the *Southwell* iteration method [e.g., gast70]. Like the Jacobi and Gauss-Seidel methods, Southwell iteration will always converge to a solution for any physically possible radiosity equation.

### 2.5 Convergence Behaviour

Shao and Badler [shao93] presented a detailed and informative discussion of the convergence behavior of the progressive refinement radiosity algorithm. They observed

that while the algorithm may quickly converge to a visually appealing image, many more steps are often required to capture the nuances of color bleeding and soft shadows. They demonstrated that it took 2,000 or more steps to achieve full convergence in a complex environment of some 1,000 patches.

Much of the problem lies in how progressive refinement works. By always selecting the patch with the most flux to shoot, it concentrates first on the light sources. Most of their flux will be shot to what Shao and Badler called *global* patches – those patches which are relatively large and can be seen from much of the environment. For an architectural interior, these are typically the walls, floor and ceiling of a room. Their patches receive most of the flux from the light sources and consequently shoot it to the other global patches.

The *local* patches are those patches which are small and are usually hidden from much of the environment. Their flux will not be shot until that of the global patches has been exhausted. This is undesirable for two reasons. First, their small areas means that they will receive relatively little flux in comparison to the global patches. It may take several hundred steps before they shoot for the first time.

The second reason is that when these local patches do shoot, much of their flux often goes no further than their immediate neighbors. While this does not affect the global environment to any great extent, it does account for color bleeding and soft shadow effects. In this sense, a better error metric is the worst-case difference between the estimated and converged element exitances. In their experiments, Shao and Badler observed that it took twice as many iterations as there were patches in the environment.

One strategy to overcome this problem involves de-emphasizing the contributions due to the global patches, ensuring that all patches shoot their flux in a reasonable number of steps. This requires a modification of the progressive refinement radiosity algorithm that is described next.

## 2.6 Positive Overshooting

Convergence of the Gauss-Seidel algorithm can often be accelerated by using one of several techniques known as *successive overrelaxation* [e.g., nobl69]. Applied to the radiosity equation, these techniques can be interpreted as “overshooting” the amount of flux from a patch into the environment. That is, the amount of flux shot from the patch is more than the amount of unshot flux the patch actually has. The flux shot in subsequent steps by the receiving patches will tend to cancel this overshooting. In the meantime, the total amount of unshot flux in the environment is shot and absorbed more quickly. This tends to result in faster convergence rates.

Shao and Badler [shao93] presented a modified version of the progressive refinement radiosity algorithm that incorporates *positive overshooting* to accelerate the convergence rate by a factor of two or more. At the same time, it tends to prioritize the ordering of patches being shot such that the local patches are shot sooner, thereby enhancing the rendering of subtle lighting effects such as color bleeding and soft shadows.

The modification to the radiosity algorithm (Figure. 2.3), based on an earlier proposal by Feda and Purgathofer [feda92], is:

```
...
Select patch  $i$  with greatest positive unshot
    flux  $\Delta M_i^{unshot} A_i$ 
Estimate overshooting parameter  $\Delta M_i^{overshoot}$ 
Calculate all form factors  $F_{ij}$ 
FOR each patch  $j$ 
```



$$\Delta M = \rho_j F_{ij} \left( \Delta M_i^{unsent} + \Delta M_i^{overshoot} \right) \frac{A_i}{A_j}$$

$$\Delta M_j^{unsent} = \Delta M_j^{unsent} + \Delta M$$

$$M_j = M_j + \Delta M$$

**ENDFOR**

$$\Delta M_i^{unsent} = -\Delta M_i^{overshoot}$$

...

Figure 2.4 – Progressive refinement radiosity algorithm with positive overshooting

Shao and Badler [shao93] based their calculation of the overshooting parameter

$\Delta M_i^{overshoot}$  on the following heuristic:

$$\Delta M_i^{overshoot} = \rho_i \sum_{j=1}^n \Delta M_j^{unsent} F_{ij} \quad (2.36)$$

where:

$$\Delta M_j^{unsent} = \begin{cases} \Delta M_j^{unsent} & \text{if } \Delta M_j^{unsent} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.37)$$

This essentially sums the amount of unsend flux the patch will later receive from the elements in the environment and multiplies it by the reflectance of the patch. The patch effectively *gathers* the unsend flux it would otherwise receive in later steps and shoots it along with its own unsend flux.

Equation 2.37 ensures that the patch will never receive a negative amount of flux from any element. Thus, only positive overshooting can occur. On the other hand, the patch may shoot a negative amount of flux; this serves to cancel the overshoot flux in later steps.

Since we can now have both positive and negative unsend flux, we need to modify our convergence criterion. Equation 2.35 becomes:

$$\sum_{i=1}^n |\Delta M_i^{unsent} A_i| \leq \epsilon \quad (2.38)$$

Experiments performed by Shao and Badler on two complex environments demonstrated that the convergence rate with positive overshooting can be accelerated by a factor of two or more over that of conventional progressive radiosity. There was also strong evidence that the appearance of subtle color bleeding and soft shadow effects may appear as much as three to five times more quickly. Positive overshooting is clearly a useful addition to the basic progressive radiosity algorithm.

Other overrelaxation techniques for solving the radiosity equation are described by Gortler and Cohen [gort93] and Greiner et al. [grei93].

## 2.7 Environment Changes

Progressive radiosity is typically used to calculate the radiant flux distribution of static environments. If the initial distribution is changed (by changing or repositioning the light sources) or the surface reflectances are modified, the radiosity solution for the environment has to be recalculated.

### 2.7.1 Changes in Lighting

Suppose we want to change or reposition the light sources. The form factor matrix will remain unchanged, but this is little consolation where progressive radiosity is concerned. In general, we have to solve the radiosity equation, form factors and all, whenever the initial exitance due to a light source is changed.

If the changes are relatively minor, we can use the final exitance vector of the previous solution as the initial exitance vector  $\mathbf{M}_o$ . In most cases, progressive radiosity will quickly converge to the new solution. (At worst,  $\mathbf{M}_o$  will be no different than a random estimate, and progressive radiosity will eventually converge.)

Of course, if any of the light sources are dimmed, this will imply a negative quantity of unspent exitance [chen90]. Several minor changes to the progressive radiosity equation algorithm presented in Figure 2.3 will be needed to accommodate this physically impossible but eminently useful possibility.

If there are many lighting changes to be modeled – theatre lighting, for example – it may be useful to calculate separate solutions for each individual group of light sources [aire90]. These solutions are independent of one another, and can be scaled and summed to represent any possible combination of light sources and their final exitances.

Another approach is to precalculate a set of solutions for a range of initial radiant flux distributions and decompose them into a series of basis functions [nime94, doba97]. This is useful mostly for image synthesis of outdoor environments and daylighting analysis, where numerous solar positions must be accounted for. Approximate solutions for intermediate solar positions that have not been previously calculated can be obtained through blending of the basis functions.

The difficulty with this approach is that it is difficult to determine *a priori* how many solar positions need to be calculated to obtain a reasonable set of basis functions. In particular, it does not take into account the geometry of the environment – a small change in solar position may produce drastically different final radiant exitance distributions.

### **2.7.2 Changes in Surface Reflectances**

A second challenge comes when the surface reflectances are changed. One typical example is in architectural design where the designer may want to compare the visual appearance and effect of different wall or floor finishes. Again, the form factor matrix

remains unchanged. However, the solution may change drastically if the surface area is large and its reflectance is changed by any significant amount.

Chen [chen90] noted that the previous solution often provides a satisfactory initial exitance vector  $\mathbf{M}_o$ , particularly when the number of surfaces whose reflectances have been changed is small in number. From Equation 1.20, we know that the exitance of a surface patch is given by:

$$M_i = M_{oi} + \rho_i \sum_{j=1}^n M_j F_{ij} \quad (2.39)$$

If we define  $M'_{oi}$  as the new initial exitance and  $\rho'_i$  as the new surface reflectance, then the incremental change in final exitance is given by:

$$\Delta M_i = M'_{oi} - M_{oi} + (\rho'_i - \rho_i) \sum_{j=1}^n M_j F_{ij} \quad (2.40)$$

Substituting Equation 2.36 into this equation gives us:

$$\Delta M_i = M'_{oi} - M_{oi} + \frac{(\rho'_i - \rho_i)(M_i - M_{oi})}{\rho_i} \quad (2.41)$$

where the previous surface reflectance  $\rho_i$  is assumed to be greater than zero. We can add this value (which may be negative) to the current calculated patch exitance  $M_i$  and also its remaining unspent exitance (if any). From this, we can shoot the exitance until the radiosity algorithm converges to a new solution.

This approach does not however help when there are significant changes to the surface reflectances. As with changes to the light sources, major changes often result in completely different final exitance distributions. Calculating these distributions often takes as much effort when using the previous solution as when using the initial radiant flux distribution  $\mathbf{M}_{oi}$ .

## **2.8 A New Approach**

While progressive radiosity is a useful technique for producing architectural visualizations and generating photometric predictions for static environments, it requires too much computational effort to determine the effect of significant changes in the lighting or surface reflectances. The next chapter will introduce a new approach to solving the radiosity equation that addresses this problem.

# Chapter 3

## Eigenvector Radiosity – Theory

### 3.1 Eigenanalysis

Eigenanalysis is a well-known technique with many scientific and engineering applications, including structural vibration analysis, electrical power system stability, pattern recognition, radar and acoustic signal processing, quantum mechanics, and much more. As an analysis tool, it identifies the principal vibrational modes (or their abstract analogues) of a physical system. As a computational tool, it offers time and space savings by providing approximate solutions using only the principal modes.

Applying eigenanalysis techniques to a problem domain generally requires that the physical meaning of the resultant eigensystem be known. In the example of structural vibration analysis, eigenanalysis of the stiffness matrix associated with a finite element model identifies the natural vibration modes and frequencies of the structure. However, the physical meaning of eigenanalysis in terms of the radiosity equation has been an open question for the past sixty years.

#### 3.1.1 The Radiosity Kernel

Moon [moon40] appears to have been the first researcher to consider this question. He noted that the continuous form of the radiosity equation could be expressed as a Fredholm integral equation of the second kind:

$$M(\mathbf{x}) = M_o(\mathbf{x}) + \lambda \int_S M(\mathbf{y})K(\mathbf{x}, \mathbf{y})d\mathbf{y} \quad (3.1)$$

where  $M_o(\mathbf{x})$  is the initial exitance at point  $\mathbf{x}$  on a surface in the environment  $S$ ,  $M(\mathbf{x})$  is the final exitance at point  $\mathbf{x}$ ,  $M(\mathbf{y})$  is the final exitance at point  $\mathbf{y}$ ,  $\lambda$  is a constant, and the radiosity kernel  $K(\mathbf{x},\mathbf{y})$  is integrated over all surfaces.

Moon also noted that the associated homogeneous equation:

$$M'(\mathbf{x}) = \lambda \int_S M(\mathbf{y})K(\mathbf{x},\mathbf{y})d\mathbf{y} \quad (3.2)$$

has solutions  $M' = M'_1, M'_2, \dots$  that are its eigenfunctions, with corresponding eigenvalues  $\lambda = \lambda_1, \lambda_2, \dots$ . Once these functions are found for a given radiosity kernel, the solution for Equation 3.1 can be determined.

Moon was able to develop closed-form analytic solutions for a number of simple environments, including the sphere, infinite cylinder, and parallel and perpendicular plates. He later expanded on this work with his co-author Spencer in their book, “The Photic Field” [moon81]. However, each kernel required its own analysis, and so the eigenanalysis of arbitrary environments was still an open problem.

Koenderink and van Doorn [koen83] examined eigenfunctions of the radiosity kernel from the perspective of computer vision research. Given a complex geometric object such as a marble bust with Lambertian surfaces, the radiosity equation models the multiple interreflections of flux between these surfaces. Koenderink and van Doorn demonstrated that the eigenfunctions of the radiosity kernel describe “geometrical modes” that are dependent on the object geometry and invariant with respect to the initial irradiance. These modes were interpreted as pseudofacets on an equivalent convex object.

Langer [lang98] extended the work of Koenderink and van Doorn by examining the relationship between interreflections and shadows. He noted that the geometrical mode

corresponding to the eigenfunction of the radiosity kernel with the largest magnitude eigenvalue is a non-negative function that is physically realizable. Langer called this the “principal mode” of the environment. Further, given an environment with constant surface reflectance throughout, the principal mode is constant over  $\mathbf{x}$ , and may be interpreted as the “ambient light” term used in computer graphics. (This was also noted by Moon in [moon40].)

### **3.1.2 The Radiosity Matrix**

Baranoski et al. [bara97] and Baranoski [bara98] attempted to elucidate the physical significance of the radiosity matrix eigensystem by interpreting and visualizing the eigenvectors with the largest and smallest eigenvalues as patch exitance vectors. (The exitance vector elements were interpreted as patch luminances, which after appropriate shifting and scaling allowed images of the environments to be displayed.)

These authors also investigated the dominant eigenvectors of the “symmetric” radiosity matrix formed by multiplying the radiosity matrix by a diagonal matrix whose elements consisted of the quotient of the patch areas and their reflectances.

Unfortunately, qualitative analysis of the images proved inconclusive. The authors inferred some general properties relating to the flow of light within simple environments, but were unable to offer any quantitative physical interpretation of their results.

Ramamoorthi [rama99] performed similar experiments on the symmetric radiosity matrix, but for more complex environments with 2,600 and 4,500 patches. In addition to visualizing the individual eigenvectors (which he called “eigenmodes”), he generated visualizations using only the dominant eigenvectors.



Ramamoorthi noted that only a small fraction of the eigenvectors had significant eigenvalues. For the larger environment (a room with a table and four chairs), fewer than 20 of the 4,500 eigenvalues exceeded 10 percent of the spectral radius. A reasonably accurate (in the sense of 2-norm relative error) global approximation could be obtained with as few as 10 of the 4,500 eigenvectors. However, it required 150 to 200 eigenvectors to provide patch exitances that were locally accurate. (That is, where both the 2-norm and  $\infty$ -norm relative errors for the patch exitance vector were small.)

### 3.1.3 The Form Factor Matrix

Finally, DiLaura and Franck [dila93] demonstrated that the iterative formulation of the radiosity equation (Equation 2.22) could be reformulated in terms of the eigensystem of the form factor matrix. Unfortunately, their technique applied only to symmetric form factor (not radiosity) matrices, which generally occur only in unoccluded environments where all surface patches are the same size.

This leaves open the fundamental question: What is the physical significance of the eigensystem of a form factor (or radiosity) matrix? Curiously, the answer involves a physical analogy of radiative transfer systems that has been completely ignored for the past forty or so years.

## 3.2 Radiative Transfer Networks

Suppose we are given a system of  $n$  discrete surface patches  $A_i$  with Lambertian reflectance  $\rho_i$ , irradiance  $E_i$ , and possibly zero initial radiant exitance  $M_{oi}$  for each patch  $i$ .

The radiosity equation:

$$M_i = M_{oi} + \rho_i E_i = M_{oi} + \rho_i \sum_{j=1}^n F_{ij} M_j \quad (3.3)$$

describes the final radiant exitance  $M_i$  of each patch  $i$ , where  $F_{ij}$  is the form factor from patch  $i$  to patch  $j$ , and where  $E$  and  $M$  are assumed to be constant over each patch. Expressed in matrix form, this becomes:

$$\mathbf{M} = \mathbf{M}_o + \mathbf{RE} = \mathbf{M}_o + \mathbf{RFM} \quad (3.4)$$

where  $\mathbf{M}$  is the final radiant exitance vector,  $\mathbf{M}_o$  is the initial radiant exitance vector,  $\mathbf{E}$  is the irradiance vector,  $\mathbf{R}$  is the diagonal reflectance matrix (where  $R_{ii} = \rho_i$  and  $R_{ij} = 0$ ,  $i \neq j$ ), and  $\mathbf{F}$  is the form factor matrix.

Rearranged slightly, this becomes:

$$\mathbf{M}_o = (\mathbf{I} - \mathbf{RF})\mathbf{M} \quad (3.5)$$

which is the familiar matrix form of the radiosity equation.

As shown by Oppenheim [oppe54, oppe56] and O'Brien [obri55], the radiative flux transfer between these patches can be described in terms of a linear resistive network with one or more voltage sources. For a given patch  $A_i$ , the net flux transfer away from the patch is given by:

$$\Phi_i^{net} = A_i(M_i - E_i) \quad (3.6)$$

which is analogous to current in an electrical circuit.

Substituting  $E_i$  from Equation 3.3 into Equation 3.6 gives:

$$\Phi_i^{net} = \frac{A_i(1 - \rho_i)}{\rho_i} \left[ \frac{M_{oi}}{(1 - \rho_i)} - M_i \right] \quad (3.7)$$

where from Ohm's Law [e.g., thom94], the term  $\left[ \frac{M_{oi}}{(1 - \rho_i)} - M_i \right]$  is analogous to voltage,

and the term  $\frac{A_i(1 - \rho_i)}{\rho_i}$  is analogous to conductance (i.e., the inverse of resistance).

Now to conserve energy in the system, the summation relation must hold for each patch  $A_i$ :

$$\sum_{j=1}^n F_{ij} = 1 \quad (3.8)$$

This means that if the system is not fully enclosed, it must be enclosed in a box whose interior surfaces have zero reflectance and zero initial radiant exitance. This box will act as an energy sink to maintain the energy balance.

Therefore, by again substituting  $E_i$  from Equation 3.3 into Equation 3.6, we also have:

$$\Phi_i^{net} = \sum_{j=1}^n A_i F_{ij} (M_i - M_j) \quad (3.9)$$

which gives:

$$\sum_{j=1}^n A_i F_{ij} (M_i - M_j) = \frac{A_i (1 - \rho_i)}{\rho_i} \left[ \frac{M_{oi}}{(1 - \rho_i)} - M_i \right] \quad (3.10)$$

This is analogous to Kirchoff's Current Law [e.g., thom94], which states that the algebraic sum of the currents entering a node in an electrical circuit is always zero.

This completes the analogy: each patch represents a node in an electrical circuit, with radiative flux (current) flowing between these patches (nodes). The amount of flux is determined by the analogies of voltage and conductance.

Figure 3.1 shows as an example the RT network for an enclosed system defined by three surface patches with finite width and infinite length. The conductances  $g_{o1}$ ,  $g_{o2}$ , and  $g_{o3}$  represent the initial radiant exitances, and have the form:

$$g_{oi} = \frac{(1 - \rho_i) A_i}{\rho_i} \quad (3.11)$$

while the other conductances represent the interreflections of flux between the patches, and have the form (from the reciprocity relationship  $A_i F_{ij} = A_j F_{ji}$ ):

$$g_{ij} = A_i F_{ij} = A_j F_{ji} = g_{ji} \quad (3.12)$$

for any two patches  $A_i$  and  $A_j$ .

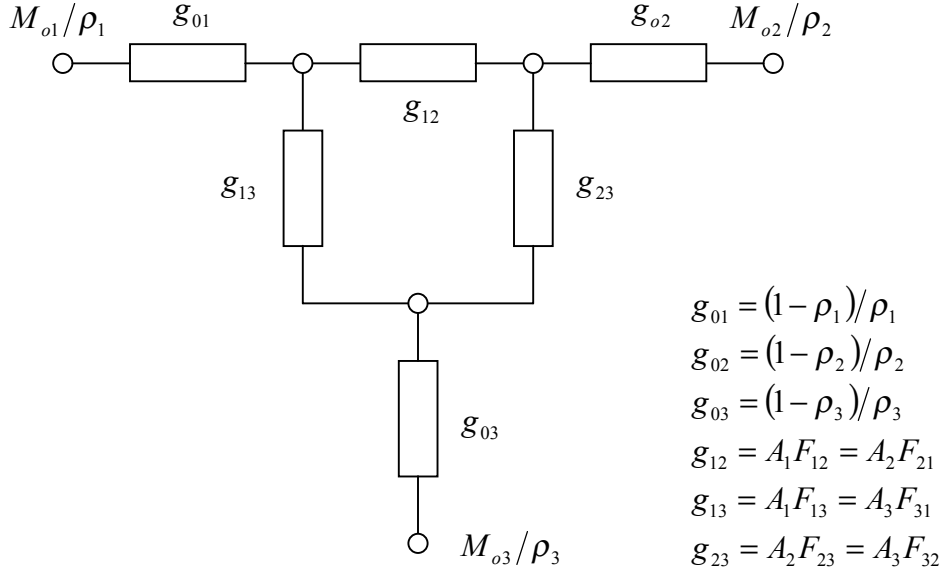


Figure 3.1 – Example radiative transfer network.

The patch exitances  $M_i$  are analogous to the circuit node voltages, while the initial patch exitances  $M_{oi}$  are analogous to the node voltages  $M_{oi} / \rho_i$ . If  $M_{oi}$  is zero, the node is connected to ground; otherwise it is connected to a voltage source.

All other RT networks are topologically similar to Figure 3.1. That is, each patch  $A_i$  will have associated with it a resistor with conductance  $g_{oi}$  that has one terminal connected to either a fixed voltage source if it has an initial radiant exitance  $M_{oi}$  or ground. The other terminal will be connected through resistors with conductances  $g_{ij}$  to every other patch  $A_j$ . Where  $F_{ij} = 0$ , these resistors will have zero conductance – in other words, an open circuit.

The RT network analogy represents an environment solely in terms of radiative flux transfer between its surface patches. If we are interested in simplifying an environment to reduce the computational effort needed to solve its radiosity equation, then it is reasonable to begin with its RT network.

### 3.3 Symmetric Matrices

From Equation 3.10 we can derive these simultaneous linear equations [obri55]:

$$\begin{aligned}
 \frac{A_1 M_{o1}}{\rho_1} &= \frac{A_1 M_1}{\rho_1} - F_{12} A_1 M_2 \cdots - F_{1n} A_1 M_n \\
 \frac{A_2 M_{o2}}{\rho_2} &= -F_{21} A_2 M_1 + \frac{A_2 M_2}{\rho_2} \cdots - F_{2n} A_2 M_n \\
 &\dots \\
 \frac{A_n M_{on}}{\rho_n} &= -F_{N1} A_N M_1 - F_{N2} A_N M_2 \cdots + \frac{A_N M_n}{\rho_n}
 \end{aligned} \tag{3.13}$$

Expressed in matrix form, these become:

$$\mathbf{AR}^{-1}\mathbf{M}_o = \mathbf{AR}^{-1}(\mathbf{I} - \mathbf{RF})\mathbf{M} \tag{3.14}$$

which is equivalent to Equation 3.5. As noted by Nievergelt [niev97], multiplying both sides of Equation 3.5 by  $\mathbf{AR}^{-1}$  in Equation 3.14 replaces exitances with equivalent radiant fluxes that produce the same exitances from the system's Lambertian surface patches.

Neumann [neum94] and Nievergelt [niev97] both noted that this operation produces a version of the radiosity matrix  $\mathbf{AR}^{-1}(\mathbf{I} - \mathbf{RF})$ , which is symmetric positive definite.

Trivially rearranging terms in Equation 3.4, we also have:

$$\mathbf{M} = \mathbf{M}_o + \mathbf{RA}^{-1}(\mathbf{AF})\mathbf{M} = \mathbf{M}_o + \mathbf{SGM} \tag{3.15}$$

where with reference to radiative transfer networks, the matrix  $\mathbf{G}$  is the symmetric *conductance matrix* for the interconnected conductances  $g_{ij}$ .

### 3.4 Spectral Decomposition

Given any square  $n \times n$  matrix  $\mathbf{B}$ , the spectral decomposition theorem [e.g., deif91] states that:

$$\mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^* = \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{v}_i^* \quad (3.16)$$

where  $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$  is a matrix of the orthonormalized eigenvectors of  $\mathbf{B}$ ,  $\mathbf{D} = \text{diag}(\lambda_1, \dots, \lambda_n)$  is a diagonal matrix of their associated eigenvalues, and  $\mathbf{v}_i \mathbf{v}_i^*$  is the  $i^{\text{th}}$  eigenprojection of  $\mathbf{B}$  corresponding to the eigenvalue  $\lambda_i$ .

In many cases, the matrix  $\mathbf{B}$  can be approximated by a subset of the eigenvectors with the largest magnitude eigenvalues. That is, if the eigenvectors are ordered such that  $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$ , then:

$$\mathbf{B} \approx \mathbf{B}' = \sum_{i=1}^p \lambda_i \mathbf{v}_i \mathbf{v}_i^*, \quad p \ll n \quad (3.17)$$

where  $\mathbf{B}'$  is rank deficient. (For a symmetric matrix, the complex conjugate vector  $\mathbf{v}_i^*$  becomes the transpose vector  $\mathbf{v}_i^T$ .)

### 3.5 Approximate Conductance Networks

Applying the spectral theorem to the symmetric radiative transfer conductance matrix  $\mathbf{G}$ , we have:

$$\mathbf{G} = \mathbf{V}\mathbf{D}\mathbf{V}^T = \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{v}_i^T \quad (3.18)$$

The question is whether this spectral decomposition of  $\mathbf{G}$  has any physical meaning. If it does, it may offer a useful approach to solving the radiosity equation.

As an example, consider the empty rectangular room shown in Figure 3.2 with measurements:

Length	Width	Height
5.0 m	3.0 m	2.5 m

and surface reflectances:

Ceiling	Walls	Floor
80 %	70 %	20 %

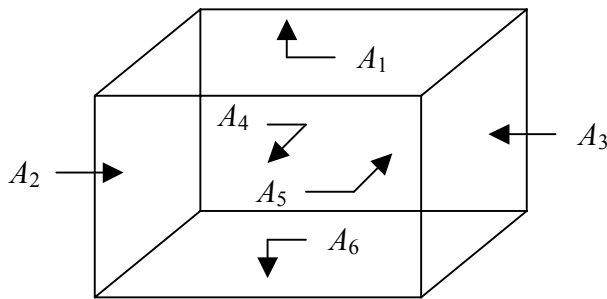


Figure 3.2 – Empty rectangular room.

From the analytic form factors equations for parallel and perpendicular rectangles (see Appendix A) and symmetry considerations, we have:

$$F_{11} = F_{22} = F_{33} = F_{44} = F_{55} = F_{66} = 0.0000$$

$$F_{12} = F_{13} = F_{62} = F_{63} = 0.1249$$

$$F_{14} = F_{15} = F_{64} = F_{65} = 0.2145$$

$$F_{16} = F_{61} = 0.3213$$

$$F_{21} = F_{26} = F_{31} = F_{36} = 0.2498$$

$$F_{23} = F_{32} = 0.0800$$

$$F_{24} = F_{25} = F_{34} = F_{35} = 0.2102$$

$$F_{41} = F_{46} = F_{51} = F_{56} = 0.2573$$

$$F_{42} = F_{43} = F_{52} = F_{53} = 0.1261$$

$$F_{45} = F_{54} = 0.2331$$

This gives the symmetric conductance matrix:

$$\mathbf{G} = \begin{bmatrix} 0.000 & 1.873 & 1.873 & 3.216 & 3.216 & 4.819 \\ 1.873 & 0.000 & 0.600 & 1.576 & 1.576 & 1.873 \\ 1.873 & 0.600 & 0.000 & 1.576 & 1.576 & 1.873 \\ 3.216 & 1.576 & 1.576 & 0.000 & 2.913 & 3.216 \\ 3.216 & 1.576 & 1.576 & 2.913 & 0.000 & 3.216 \\ 4.819 & 1.873 & 1.873 & 3.216 & 3.216 & 0.000 \end{bmatrix}$$

which has the eigenvalues:

$$12.401, \quad -4.819, \quad -2.913, \quad -2.639, \quad -1.429, \quad -0.600$$

and their associated eigenvectors:

$$\mathbf{V} = \begin{bmatrix} -0.495 \\ -0.271 \\ -0.271 \\ -0.425 \\ -0.425 \\ -0.495 \end{bmatrix}, \begin{bmatrix} +0.707 \\ +0.000 \\ +0.000 \\ +0.000 \\ +0.000 \\ -0.707 \end{bmatrix}, \begin{bmatrix} +0.000 \\ +0.000 \\ +0.000 \\ -0.707 \\ +0.707 \\ +0.000 \end{bmatrix}, \begin{bmatrix} -0.470 \\ +0.030 \\ +0.030 \\ +0.527 \\ +0.527 \\ -0.470 \end{bmatrix}, \begin{bmatrix} -0.183 \\ +0.652 \\ +0.652 \\ -0.201 \\ -0.201 \\ -0.183 \end{bmatrix}, \begin{bmatrix} +0.000 \\ -0.707 \\ +0.707 \\ +0.000 \\ +0.000 \\ +0.000 \end{bmatrix}$$

Using the five eigenvectors with the largest magnitude eigenvalues to reconstruct  $\mathbf{G}'$

according to Equation 3.17, we have:

$$\mathbf{G}' = \begin{bmatrix} 0.000 & 1.873 & 1.873 & 3.216 & 3.216 & 4.819 \\ 1.873 & \mathbf{0.300} & \mathbf{0.300} & 1.576 & 1.576 & 1.873 \\ 1.873 & \mathbf{0.300} & \mathbf{0.300} & 1.576 & 1.576 & 1.873 \\ 3.216 & 1.576 & 1.576 & 0.000 & 2.913 & 3.216 \\ 3.216 & 1.576 & 1.576 & 2.913 & 0.000 & 3.216 \\ 4.819 & 1.873 & 1.873 & 3.216 & 3.216 & 0.000 \end{bmatrix}$$

where the differences between  $\mathbf{G}$  and  $\mathbf{G}'$  are highlighted.

We may interpret each eigenvector  $\mathbf{v}_i$  and its associated eigenvalue  $\lambda_i$  of matrix  $\mathbf{G}$  as describing a radiative transfer network with the same topology as that described by  $\mathbf{G}$ ,



but with different conductances between its nodes (patches). From Equation 3.18, its conductance matrix is seen to be  $\mathbf{G}(\mathbf{v}_i) = \lambda_i \mathbf{v}_i \mathbf{v}_i^T$ .

Knowing that the conductance of  $n$  resistors connected in parallel is:

$$g_{total} = \sum_{i=1}^n g_i \quad (3.19)$$

it is therefore evident that Equation 3.18 describes the superposition of the networks represented by the eigenprojections of  $\mathbf{G}$ . Moreover, those eigenvectors with the largest magnitude conductances describe the *dominant paths* (i.e., largest conductances) for radiative flux transfer between the patches.

There are several points to note here. First, these paths do not indicate the net flow of radiant flux between patches – this is dependent on the initial patch exitances. The eigenprojections describe the network itself rather than any network state.

Second, the diagonal elements  $g'_{22}$  and  $g'_{33}$  have non-zero values. These can possibly be interpreted as loop conductances connecting the nodes to themselves, as shown in Figure 3.3.

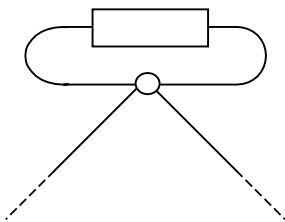


Figure 3.3 – Loop conductance.

In an electrical network, a loop conductance has no effect – the voltage is the same at both terminals, and so there is no net flow of current through the conductance.

The same argument applies to radiative transfer networks – there can be no net flow of flux through a loop conductance. This is evident from Equation 3.8, where the contribution of the form factor  $F_{ii}$  is cancelled by the null term  $(M_i - M_i)$ .

The argument against this interpretation is that if the loop conductance do not affect the net flow of flux within the network, then their values should be arbitrary. This is not the case – changing the diagonal elements by arbitrary amounts changes the conductance matrix eigensystem. The interpretation of the diagonal matrix elements as loop conductances must therefore be considered tentative and open to further investigation.

One more point regarding the approximate conductance matrix  $\mathbf{G}'$  requires a different example. The following conductance matrix  $\mathbf{G}$  was generated randomly:

$$\mathbf{G} = \begin{bmatrix} 0.000 & 0.451 & 0.043 & 0.027 & 0.312 & 0.012 \\ 0.451 & 0.000 & 0.384 & 0.683 & 0.092 & 0.035 \\ 0.043 & 0.384 & 0.000 & 0.612 & 0.608 & 0.015 \\ 0.027 & 0.683 & 0.612 & 0.000 & 0.016 & 0.190 \\ 0.312 & 0.092 & 0.608 & 0.016 & 0.000 & 0.586 \\ 0.012 & 0.035 & 0.015 & 0.190 & 0.586 & 0.000 \end{bmatrix}$$

and, using the fact that the sum of each column  $i$  represents the associated area  $a_i$  to obtain:

$$\mathbf{A} = \text{diag}[0.848, 1.646, 1.664, 1.529, 1.617, 0.841]$$

the associated form factor matrix  $\mathbf{F} = \mathbf{A}^{-1}\mathbf{G}$  is:

$$\mathbf{F} = \begin{bmatrix} 0.000 & 0.532 & 0.051 & 0.032 & 0.368 & 0.015 \\ 0.274 & 0.000 & 0.233 & 0.414 & 0.056 & 0.021 \\ 0.026 & 0.230 & 0.000 & 0.367 & 0.365 & 0.009 \\ 0.017 & 0.446 & 0.400 & 0.000 & 0.010 & 0.124 \\ 0.193 & 0.057 & 0.376 & 0.010 & 0.000 & 0.362 \\ 0.015 & 0.042 & 0.018 & 0.226 & 0.697 & 0.000 \end{bmatrix}$$

Using the four eigenvectors with the largest magnitude eigenvalues to reconstruct  $\mathbf{G}'$  according to Equation 3.17 gives:

$$\mathbf{G}' = \begin{bmatrix} -\mathbf{0.084} & +0.424 & +0.083 & +0.057 & +0.310 & +0.017 \\ +0.424 & +0.004 & +0.367 & +0.706 & +0.077 & +0.069 \\ +0.083 & +0.367 & +0.044 & +0.568 & +0.641 & -\mathbf{0.057} \\ +0.057 & +0.706 & +0.568 & +0.003 & +0.002 & +0.221 \\ +0.310 & +0.077 & +0.641 & +0.002 & +0.016 & +0.551 \\ +0.017 & +0.069 & -\mathbf{0.057} & 0.221 & +0.551 & +0.078 \end{bmatrix}$$

where the highlighted elements represent negative conductances.

In electrical networks, the linear amplifier has the transfer characteristic  $i = mv$ , where  $v$  is the input current,  $i$  is the output voltage, and  $m$  is a constant. From Ohm's law, the device conductance is  $m$ . If  $m$  is negative, the amplifier inverts the input signal. If in addition the amplifier gain is less than unity, it behaves as if it were a negative conductance.

Moon [moon40] described a radiant flux analogue of a linear amplifier with positive  $m$ . It consisted of an opal glass diffuser with lamps behind it that were controlled by a photocell and dimmer. The radiant flux emitted from the diffuser is directly proportional to the incident flux. An equivalent linear amplifier with negative  $m$  (representing negative conductance) can easily be constructed by proportionately dimming the lamps in response to increased incident flux.

This of course requires that the linear amplifier be biased such that it produces a constant non-zero flux output in the absence of incident flux. Otherwise, the amplifier will have to produce "negative" light, a physical impossibility.

This situation is not unusual in electrical networks. For example, tunnel diodes have nonlinear transfer characteristics that include negative conductance behavior over a

restricted range of positive input voltages. Within this range, the transfer characteristic is approximately  $i = mv + c$ , where  $m$  is negative and  $c$  is a positive constant. By applying a constant bias voltage to the diode, it behaves as if it were a negative conductance resistor over a range of input voltages.

The same situation may apply to an RT network that includes an inverting linear amplifier – the adjoining conductances may sufficiently “bias” the amplifier such that it behaves like a negative conductance over a range of initial patch exitances.

From Equations 3.18 and 3.19, it is therefore evident that any approximate conductance matrix  $\mathbf{G}'$  can be represented by a radiative transfer network, even if it contains loop and negative conductances. Whether this network is physically realizable (i.e., it does not require “negative” light) depends on the network conductances and the initial patch exitances.

Another problem with negative conductances in both electrical and radiative transfer networks is that they can lead to uncontrolled positive feedback where the solution goes to infinity. Whether this situation will occur in an RT network depends on both the initial patch exitances and the relative magnitude of the negative conductances. (In particular, the non-zero diagonal elements may result in a radiosity matrix that is not diagonally dominant.) This is unlikely to occur however where the differences between  $\mathbf{G}$  and  $\mathbf{G}'$  are small.

### **3.6 Approximation Accuracy**

Knowing that an approximate conductance matrix can be represented by a radiative transfer network does not unfortunately indicate how accurate the approximation will be. While this will depend on the particular network, it is instructive to consider the empty

rectangular room example presented in Figure 3.2 (and where the detailed calculations are presented in Appendix A).

Suppose that the initial exitance vector is:

$$\mathbf{M}_o = [1.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^T$$

Solving for  $\mathbf{M}$  directly using the equation:

$$\mathbf{M} = (\mathbf{I} - \mathbf{R}\mathbf{A}^{-1}\mathbf{G})^{-1} \mathbf{M}_o \quad (3.20)$$

we have:

$$\mathbf{M} = [1.2343 \ 0.3684 \ 0.3684 \ 0.3713 \ 0.3713 \ 0.1296]^T$$

If we construct the approximate conductance  $\mathbf{G}'$  using only the two largest magnitude eigenvectors of  $\mathbf{G}$ , we have:

$$\mathbf{G}'_2 = \begin{bmatrix} 0.6316 & 1.6643 & 1.6643 & 2.6153 & 2.6153 & 5.4514 \\ 1.6643 & 0.9107 & 0.9107 & 1.4311 & 1.4311 & 1.6643 \\ 1.6643 & 0.9107 & 0.9107 & 1.4311 & 1.4311 & 1.6643 \\ 2.6153 & 1.4311 & 1.4311 & 2.2488 & 2.2488 & 2.6153 \\ 2.6153 & 1.4311 & 1.4311 & 2.2488 & 2.2488 & 2.6153 \\ 5.4514 & 1.6643 & 1.6643 & 2.6153 & 2.6153 & 0.6316 \end{bmatrix}$$

which in comparison with  $\mathbf{G}$  does not look at all promising. However, substituting  $\mathbf{G}'_2$  for  $\mathbf{G}$  in Equation 3.20 yields:

$$\mathbf{M}'_2 = [1.2431 \ 0.3695 \ 0.3695 \ 0.3484 \ 0.3484 \ 0.1322]^T$$

and subtracting  $\mathbf{M}'$  from  $\mathbf{M}$  gives:

$$\mathbf{M} - \mathbf{M}'_2 = [-0.0088 \ -0.0011 \ -0.0011 \ +0.0229 \ +0.0229 \ -0.0026]^T$$

Somewhat remarkably, the maximum per-element error between  $\mathbf{M}$  and  $\mathbf{M}'_2$  is less than 6.2%, and the relative 2-norm error between the two solution vectors is only 2.7%.

Thus, while the approximate conductance matrix  $\mathbf{G}'$  may not resemble  $\mathbf{G}$  through visual inspection, it produces almost the same results when used to solve the radiosity equation.

While it difficult to extrapolate these results to more complex environments, they clearly illustrate the potential of approximate conductance networks.

### 3.7 Eigenvector Radiosity

Approximating the radiosity matrix  $\mathbf{G}$  with the rank deficient approximation  $\mathbf{G}'$  does not assist in solving the radiosity equation using conventional solvers such as Gauss-Seidel or Southwell iteration [e.g. cohe94]. This is because  $\mathbf{G}'$  is still an  $n \times n$  matrix. However, Equation 3.5 can be reformulated to take full advantage of  $\mathbf{G}'$ .

To begin with, Equation 3.4 can be solved iteratively as:

$$\mathbf{M}^{(r)} = \mathbf{M}_o + \mathbf{RFM}^{(r-1)} \quad (3.21)$$

which, following Equation 3.15, can be reformulated as:

$$\mathbf{M}^{(r)} = \mathbf{M}_o + \mathbf{SGM}^{(r-1)} \quad (3.22)$$

where  $\mathbf{S} = \mathbf{RA}^{-1}$ ,  $\mathbf{G} = \mathbf{AF}$ , and  $\mathbf{M}^{(0)} = \mathbf{M}_o$ .

Because  $\mathbf{G}$  is real and symmetric, its eigenvectors  $\mathbf{v}_i$  form an orthonormal set spanning an  $n$ -dimensional space, where  $n$  is the matrix order. An arbitrary  $n$ -dimensional vector  $\mathbf{u}$  can always be expressed as a weighted sum of these eigenvectors as:

$$\mathbf{u} = \sum_{i=1}^n a_i \mathbf{v}_i, \quad i=1, \dots, n \quad (3.23)$$

where the coefficients  $a_i$  are given by:

$$a_i = \mathbf{u}^T \mathbf{v}_i, \quad i=1, \dots, n \quad (3.24)$$

The initial exitance vector  $\mathbf{M}_o$  can therefore be expressed in terms of  $\mathbf{v}_i$  as:

$$\mathbf{M}_o = \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{v}_i \quad (3.25)$$

Substituting this result into Equation 3.22 gives the first estimate of the exitance vector  $\mathbf{M}$  as:

$$\mathbf{M}^{(1)} = \mathbf{M}_o + \mathbf{S} \mathbf{G} \mathbf{M}_o = \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{v}_i + \mathbf{S} \mathbf{G} \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{v}_i \quad (3.26)$$

From the definition of an eigenvector as:

$$\mathbf{G} \mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad i = 1, \dots, n \quad (3.27)$$

we also have:

$$\mathbf{G} \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{v}_i = \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{G} \mathbf{v}_i = \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_i \mathbf{v}_i \quad (3.28)$$

which gives:

$$\mathbf{M}^{(1)} = \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{v}_i + \mathbf{S} \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_i \mathbf{v}_i = \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{v}_i + \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_i \mathbf{w}_i \quad (3.29)$$

where  $\mathbf{w}_i = \mathbf{S} \mathbf{v}_i$ .

Using Equation 3.22 again, the second estimate of the exitance vector  $\mathbf{M}$  is:

$$\mathbf{M}^{(2)} = \mathbf{M}_o + \mathbf{S} \mathbf{G} \mathbf{M}^{(1)} \quad (3.30)$$

where from Equations 3.23 and 3.24, we have:

$$\mathbf{M}^{(1)} = \sum_{i=1}^n (\mathbf{M}^{(1)})^T \mathbf{v}_i \mathbf{v}_i \quad (3.31)$$

which gives:

$$\mathbf{M}^{(2)} = \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{v}_i + \sum_{i=1}^n (\mathbf{M}^{(1)})^T \mathbf{v}_i \lambda_i \mathbf{w}_i \quad (3.32)$$

Using Equation 3.29 to expand coefficients, we have:

$$\left(\mathbf{M}^{(1)}\right)^T \mathbf{v}_i = \left(\mathbf{M}_o + \mathbf{S}\mathbf{G}\mathbf{M}_o\right)^T \mathbf{v}_i \quad (3.33)$$

which becomes:

$$\left(\mathbf{M}^{(1)}\right)^T \mathbf{v}_i = \mathbf{M}_o^T \mathbf{v}_i + \sum_{j=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_j \mathbf{v}_i^T \mathbf{w}_j, \quad i = 1, \dots, n \quad (3.34)$$

Therefore:

$$\mathbf{M}^{(2)} = \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \mathbf{v}_i + \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_i \mathbf{w}_i + \sum_{j=1}^n \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_i \mathbf{w}_i^T \mathbf{v}_j \lambda_j \mathbf{w}_j \quad (3.35)$$

The third estimate is then:

$$\begin{aligned} \mathbf{M}^{(3)} = & \mathbf{M}_o + \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_i \mathbf{w}_i + \sum_{j=1}^n \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_i \mathbf{w}_i^T \mathbf{v}_j \lambda_j \mathbf{w}_j + \\ & \sum_{k=1}^n \sum_{j=1}^n \sum_{i=1}^n \mathbf{M}_o^T \mathbf{v}_i \lambda_i \mathbf{w}_i^T \mathbf{v}_j \lambda_j \mathbf{w}_j \mathbf{v}_k \lambda_k \mathbf{w}_k \end{aligned} \quad (3.36)$$

Applying induction to the first three estimates, it is evident that if:

$$\mathbf{K}^{(0)} = \mathbf{M}_o \quad (3.37)$$

then:

$$\mathbf{K}^{(r)} = \sum_{i=1}^n \left(\mathbf{K}^{(r-1)}\right)^T \mathbf{v}_i \lambda_i \mathbf{w}_i, \quad r \geq 1 \quad (3.38)$$

and:

$$\mathbf{M}^{(r)} = \mathbf{M}^{(r-1)} + \mathbf{K}^{(r)}, \quad r \geq 1 \quad (3.39)$$

where to summarize:

$\mathbf{M}_o$  is the initial exitance vector;

$\mathbf{M}^{(r)}$  is the  $r^{\text{th}}$  estimate of the exitance vector  $\mathbf{M}$ ;

$\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of the conductance matrix  $\mathbf{G}$ ;

$\mathbf{v}_i$  is the  $i^{\text{th}}$  eigenvector of the conductance matrix  $\mathbf{G}$ ; and



$$\mathbf{w}_i = \mathbf{S}\mathbf{v}_i = \mathbf{R}\mathbf{A}^{-1}\mathbf{v}_i.$$

The above reformulation of the radiosity equation in terms of eigenvectors is derived from DiLaura and Franck [dila93]. Their work was based on Equation 3.21 and the form factor matrix  $\mathbf{F}$  rather than Equation 3.22 and the conductance matrix  $\mathbf{G}$ . This limited their derivation to symmetric form factor matrices, which generally occur only in unoccluded environments where all patch elements are the same size. Substituting the conductance matrix  $\mathbf{G}$  generalizes eigenvector radiosity to all physically realizable environments.

There are several unique advantages to eigenvector radiosity:

1. The eigenvalues and eigenvectors only need to be calculated once for the conductance matrix  $\mathbf{G}$ ;
2. Changing the initial exitances of the source elements in the environment only affects the initial exitance vector  $\mathbf{M}_o$ ;
3. Changing the patch reflectances only affects the reflectance matrix  $\mathbf{R}$  and the vectors  $\mathbf{w}_i$ ; and
4. In many environments, relatively few of the eigenvalues  $\lambda_i$  will have significantly large magnitude values.

The fourth advantage has several extremely important consequences. Equations 3.38 and 3.39 show that only the largest magnitude eigenvalues will result in significant contributions to  $\mathbf{K}^{(r)}$  and so to incremental changes in  $\mathbf{M}^{(r)}$ . This being the case, only the eigenvectors with the largest magnitude eigenvalues are required to determine the exitance vector  $\mathbf{M}$  using Equation 3.39.

By using a subset of the eigenvectors of  $\mathbf{G}$ , eigenvector radiosity implicitly uses the approximate conductance matrix  $\mathbf{G}'$ . Understanding the physical significance of these eigenvectors in terms of dominant paths through the RT network provides a better understanding of how and why eigenvector radiosity works.

DiLaura and Franck [dila93] used a symmetric form factor matrix representing some 200 random patch elements. They purposely did not model a physical environment in order to avoid having any geometrical properties of the environment skewing their results. What they found was that only a few eigenvalues had absolute magnitudes exceeding 15 percent of the maximum. They also found that  $\mathbf{K}^{(r)}$  rapidly approached zero in five or so iterations, and then oscillated about an asymptotic value. (This was probably due to negative values in the equivalent approximate conductance matrix.)

It is important to realize that Equations 3.38 and 3.39 converge towards a solution of the approximate radiosity matrix  $\mathbf{G}'$  rather than  $\mathbf{G}$ . Whether this approximation is sufficiently accurate will depend upon the application and how closely  $\mathbf{G}'$  approximates  $\mathbf{G}$ . If more accuracy is required, the solution can be used as the initial exitance vector for further refinement using Gauss-Seidel or Southwell iteration with the radiosity matrix  $\mathbf{G}$ .

### 3.8 Jacobi Iteration

We again consider the empty rectangular room example presented in Figure 3.2, with an initial exitance vector:

$$\mathbf{M}_o = [1.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^T$$

We can iteratively solve for the final exitance vector  $\mathbf{M}$  using Equations 3.21 or 3.22, which are both examples of Jacobi iteration. (As shown Section 2.2.3, Jacobi iteration

exactly models the physical “bouncing” of light between diffuse reflectance surface patches in an environment.)

Convergence to within a 2-norm relative error of less than 0.1% occurs within 13 iterations. (See Appendix A for details). This is typical behavior for Jacobi iteration when applied to the solution of radiative transfer systems [cohe85].

It is important to recognize that Equations 3.37 through 3.39 are simply a reformulation of Jacobi iteration in terms of the eigenpairs of  $\mathbf{G}$ . Consequently, solving for  $\mathbf{M}$  using these equations will produce *exactly* the same convergence behavior if all of the eigenpairs are used.

Using a subset of the eigenpairs to generate the approximate conductance matrix  $\mathbf{G}'$  simply means that Jacobi iteration will be applied to a similar RT network. The convergence behavior will be similar, but the final exitance vector  $\mathbf{M}$  will be different. For example, substituting  $\mathbf{G}'_2$  for  $\mathbf{G}$  in Equations 3.37 through 3.39 gives:

$$\mathbf{M}'_2 = [1.2431 \quad 0.3695 \quad 0.3695 \quad 0.3484 \quad 0.3484 \quad 0.1322]^T$$

after 12 iterations (again to within a 2-norm relative error of less than 0.1%). This is the same result that was obtained by solving for  $\mathbf{M}'_2$  directly using Equation 3.20. (See Appendix A for details.)

### 3.9 Form Factor Matrix Interpretation

While eigenvector radiosity has been formulated and interpreted within the context of the symmetric conductance matrix and Equation 3.22 in Section 3.7, it should also be possible to formulate and interpret it within the context of the unsymmetric form factor matrix and Equation 3.21.

The spectral decomposition theorem expressed in Equation 3.16 applies to any square matrix. For unsymmetric matrices, the eigenprojections are generated using the eigenvectors and their complex conjugates. Substituting the complex conjugate for the eigenvector transpose in Section 3.7 should yield an eigenvector radiosity formulation based on the form factor matrix.

In terms of a physical interpretation, the complex-valued elements of the eigenprojections may be interpreted as admittances rather than conductances. For the electrical circuit analogy, these can be physically realized using passive resistors, capacitors, and inductors or active elements comprising resistors, capacitors, and linear amplifiers.

This interpretation however obscures the physical interpretation of the network as paths for the net flow of radiant flux. In particular, an electrical admittance network has a frequency-dependent impulse response that does not have a physical analogy in the equivalent radiative transfer network.

### **3.10 Time and Space**

As presented in this chapter, eigenvector radiosity offers the potential of significant time and space savings when solving the radiosity equation. However, it has the disadvantage of requiring that the full form factor matrix be known in order to calculate its eigensystem.

The goal is to develop eigenvector radiosity as an adjunct to progressive radiosity, such that the patch form factors do not need to be recalculated when the initial patch exitances or reflectances are changed. Indeed, one of the advantages of progressive

radiosity is that it generally does not calculate the full form factor matrix before converging to a useful solution.

Solving this problem requires a closer look at progressive radiosity and the form factor matrix.

# Chapter 4

## Eigenvector Radiosity – Practice

### 4.1 Eigensolvers

Having developed a theoretical basis for eigenvector radiosity, it remains to consider how we might implement it in software as an engineering tool. This immediately brings up the issue of eigensolvers – given a symmetric conductance matrix, we need to determine its eigensystem.

While there are many eigensolvers to choose from, the choice is largely determined by the conductance matrix order, which is equivalent to the number of patches in the environment.

#### 4.1.1 Small Environments

For small environments consisting of fewer than (say) 400 patches, orthogonal similarity transformation techniques such as Householder tridiagonalization and the QL algorithm are preferred [par198]. They are simple, reliable, and well documented. The standard implementations are the **EISPACK** functions *tred2* and *tql2*, which were originally presented as Algol routines in [wilk71].

The QL algorithm will determine the entire eigensystem of the conductance matrix. This should not be a problem for most applications in that the time and space requirements for small environments are almost insignificant.

Once the eigensystem has been determined, the dominant eigenvectors (that is, those eigenvectors with the numerically largest and smallest eigenvalues) can be selected and the radiosity equation solved using Equations 3.37 through 3.39.

#### **4.1.2 Large Environments**

The situation for medium to large environments is considerably more interesting. Orthogonal similarity transformation techniques quickly become unworkable due to matrix fill-in, and so iterative techniques are necessary. These are dominated by the Lanczos algorithm.

The simple Lanczos algorithm works with exact arithmetic, but suffers from loss of orthogonality using finite precision arithmetic. This has led to numerous variants of the Lanczos algorithm; two of the more useful are Lanczos with Selective Orthogonalization [parl79] and Block Lanczos [unde75, golu77].

Closely spaced eigenvalues may occur for surface elements that are tightly coupled (that is, whose mutual conductance dominates the RT network), and so Block Lanczos is probably the better choice. These eigenvalues occur in pairs, and so a block size of two or four is appropriate [parl98].

The Block Lanczos algorithm has several important advantages as a conductance matrix eigensolver:

1. Unlike orthogonal similarity transformation algorithms, it does not need to store the full matrix in memory. In general, the memory requirements are not much more than that required to store the desired number of eigenvectors;
2. It can compute a user-specified number of eigenpairs at either end of the spectrum, and can be iterated to obtain more if needed; and

3. The matrix is accessed through a user-defined matrix-vector multiplication function.

The Block Lanczos algorithm is not well suited to determining more than about 25 percent of the dominant eigenvectors of an eigensystem. This however is not a concern for eigenvector radiosity, which is useful only if the dominant eigenvectors comprise a small portion of the eigensystem.

## 4.2 Progressive Radiosity

A more difficult problem arises when the full conductance matrix is not available. This situation occurs for example with progressive radiosity techniques [e.g., ashd94, sill94] that use Southwell iteration to solve the radiosity equation. Unlike Gauss-Seidel iteration, Southwell iteration requires only one column of the form factor matrix to be available at each step. This allows form factors to be calculated “on the fly” and then immediately discarded in order to conserve memory.

### 4.2.1 Patches and Elements

A second complication with respect to progressive radiosity is that it commonly relies on a two-level hierarchy of patches and elements (Figure 4.1) to model surfaces.

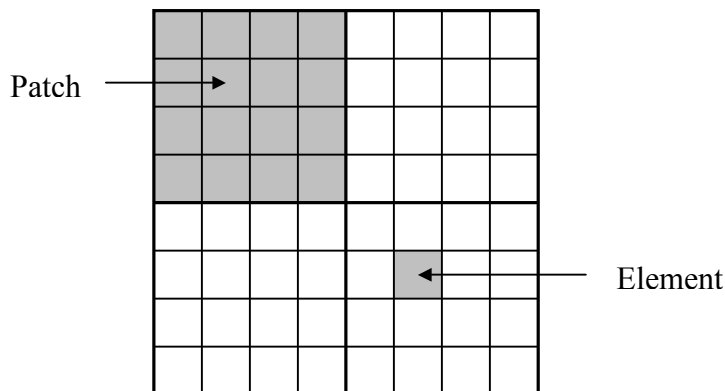


Figure 4.1 – Subdividing a surface into a hierarchy of patches and elements.



The basic idea is that the form factors between a group of adjacent shooting patches and a receiving patch will (in the absence of occluding surfaces) be almost identical. Rather than repeatedly calculating these form factors then, it is reasonable to shoot flux from the center of the group of patches [cohe85].

Rather than group patches, each patch is subdivided into an array of elements. Radiant flux is then sent from each shooting patch to the set of all elements visible to the patch in the environment.

This substructuring technique is useful for progressive radiosity in that it reduces the time complexity of the Southwell iteration from  $\mathbf{O}(n^2)$  for an  $n$ -patch environment to  $\mathbf{O}(mn)$  for an equivalent environment with  $m$  patches and  $n$  elements. It is even more useful for eigenvector radiosity in that form factors are calculated from each finite area element to the center of each shooting patch (modeled as a differential area).

Given a set of patch-to-element form factors as generated by one step of the Southwell iteration, the patch-to-patch form factors can be quickly calculated as the sum of the patch-to-element form factors for each receiving patch. Multiplying each patch-to-patch form factor by the receiving patch area produces a row/column of the symmetric conductance matrix, which has order  $m$ .

This patch-to-patch approach offers two advantages. First, the time complexity of the Block Lanczos algorithm is roughly  $\mathbf{O}(n^2)$ . Calculating the eigensystem of a patch-to-patch conductance matrix rather than a patch-to-element matrix (which must be modeled as a symmetric matrix of order  $n$ ) reduces the time complexity to  $\mathbf{O}(m^2)$ .

Second, the memory required to store the desired number of eigenvectors is reduced by a factor of  $n / m$ . Together, these advantages allow considerably larger environments to be addressed with the same computer resources.

#### **4.2.2 Further Refinement**

The disadvantage of the patch-to-patch approach is that progressive radiosity generates an  $n$ -element exitance vector, where each element corresponds to the exitance of a surface element. While the patch conductance matrix requires less time and memory to determine its eigensystem, the resultant  $m$ -element exitance vector necessarily averages the element exitance values across their parent patches. (That is, each element exitance must be assigned the calculated exitance of its parent patch.)

On the other hand, the  $m$ -element exitance vector should provide an excellent estimate of the final  $n$ -element exitance vector. It can therefore be used as the initial exitance vector for further progressive radiosity steps. Because the initial estimate is close to the final solution vector, progressive radiosity should quickly converge in relatively few steps.

This approach does not appear to be particularly useful in that the radiosity equation is solved by Southwell iteration, leaving no need for an eigenvector radiosity solution. However, there are many applications where a given radiosity equation must be repeatedly solved for different initial surface exitances and reflectances. After using Southwell iteration to determine an approximate sparse matrix representation of the form factor matrix, we can use eigenvector radiosity to more quickly solve for subsequent initial conditions.

### 4.3 Sparse Form Factor Matrices

Southwell iteration often converges to an approximate but useful solution within several hundred to several thousand steps for complex environments with tens of thousands of surface elements. In doing so, it may not calculate the full form factor matrix.

Not having a full form factor matrix available for determination of the patch conductance matrix and determination of its eigensystem might appear to be a serious problem. However, it is actually an advantage.

If a useful solution can be obtained without having to calculate the full form factor matrix, this suggests that a sparse matrix consisting only of those form factors calculated during Southwell iteration can be used to approximate the full form factor matrix. This works well with the Lanczos algorithm, which requires only a user-defined matrix-vector multiply function for sparse matrices.

For very large environments with several million elements, several tens of thousands of Southwell iteration steps may be needed to converge to a solution. This may require impractical amounts of memory to store the calculated form factors, even as a sparse matrix. We therefore need to determine which form factors will produce a sparse matrix that acceptably approximates the full matrix.

### 4.4 Eigenvalue Bounds

A corollary to the Courant-Fischer lemma [e.g., axel96] states that given two symmetric matrices  $\mathbf{A}$  and  $\mathbf{B}$ , changes in each eigenvalue of  $\mathbf{A}$  for the matrix  $\mathbf{A} + \mathbf{B}$  will be between the smallest and largest eigenvalues of  $\mathbf{B}$ . (See Appendix B for a proof.) Suppose then that we have a sparse conductance matrix  $\mathbf{G}_n$  as determined through  $n$  steps of Southwell iteration. The form factors determined by the next step will produce a matrix

$\mathbf{G}_{n+1} = \mathbf{G}_n + \tilde{\mathbf{G}}$ , where  $\tilde{\mathbf{G}}$  is the conductance matrix due the current set of form factors  $F_{ki}$ .

The bounds of the eigenvalues of  $\tilde{\mathbf{G}}$  are determined by the Gerschgorin Circle theorem [axel96]. If the largest magnitude eigenvalue of  $\tilde{\mathbf{G}}$  (i.e., its spectral norm) is much less than the spectral norm of  $\mathbf{G}_n$ , then Equation 3.18 shows that adding  $\tilde{\mathbf{G}}$  to  $\mathbf{G}_n$  will not significantly change this eigensystem, and it can be safely ignored.

Given the structure of  $\tilde{\mathbf{G}}$  (which has only one non-zero row  $g_{ik}$  and column  $g_{ki}$ ), the Gerschgorin Circle theorem reduces to:

$$\lambda_{\max} \leq A_k \sum_{i=1}^n F_{ik} \quad (4.1)$$

where the bound is the sum of the conductances from surface  $k$  to all other surfaces. (See Appendix B for a proof.) In physical terms, this bound represents the degree of radiative coupling between surface  $k$  and its surrounding environment. For enclosed environments where Equation 3.8 holds true, the bound is equal to the surface area.

This makes it practical to compute an upper bound on the spectral norm of  $\mathbf{G}_{n+1}$  for each step of Southwell iteration. By comparing the eigenvalue bounds to those of previously calculated sets of form factors, only those sets that will significantly affect the eigensystem of the approximate conductance matrix  $\tilde{\mathbf{G}}$  need be cached in memory.

An additional benefit of calculating the eigenvalue bounds for each set of form factors is that it provides an a priori estimate of the number of eigenvectors that will need to be calculated using the Lanczos algorithm.

## 4.5 Locally Dominant Paths

The choice of a “shooting” patch for each step of the Southwell iteration is determined by the maximum amount of unsent radiant flux. Therefore, depending on the initial exitance distribution, the sequence of flux transfers between patches and elements may not follow the globally dominant paths of the RT network.

A good example is roadway lighting, where the streetlights may directly illuminate only a small portion of the exterior environment. A progressive radiosity solution for this environment may not consider the form factors between the patches and elements of surfaces that would be illuminated under (say) daylight conditions.

In terms of the associated RT network, the conductance matrix  $\mathbf{G}_n$  for this example will represent locally dominant paths relative to the nodes representing the streetlights. This may not be a concern if changes in the initial exitance distribution are due to changes in the radiant flux from the existing streetlights. However, changes such as adding street lights in previously unlit areas of the environment may involve portions of the RT network that were not considered by the initial progressive radiosity solution. This may require updating  $\mathbf{G}_n$  using progressive radiosity.

The situation with interior lighting is different in that interreflections will generally cause the entire environment to be illuminated. In terms of the associated RT network then, the conductance matrix  $\mathbf{G}_n$  will usually represent globally dominant paths. Changing the initial exitance distribution or surface reflectances should not in general be influenced by the approximation of  $\mathbf{G}_n$  to  $\mathbf{G}$  when solving the associated radiosity equation.

## **4.6 Environment Partitioning**

Given the availability of a conductance matrix (full or sparse), it is possible to select any surface patch and perform a breadth-first path traversal through the associated RT network to determine the total conductance between any two given surfaces. Individual paths (corresponding to multiple reflections of flux between surface patches) need only be followed only as long as their path conductances are significant with respect to the parallel summation of the conductances of the previously determined paths.

This principle can possibly be used to partition an environment into sub-environments that are weakly coupled (in terms of radiative flux transfer) for a given light source. While it is clearly subject to combinatorial explosion, it may be possible to combine the approach with geometric visibility information to quickly and reliably partition large environments.

## **4.7 An Engineering Tool**

This chapter has examined some of the issues involved in implementing eigenvector radiosity as an engineering tool. The Block Lanczos algorithm has been suggested as a suitable eigensolver, but its performance compared to other solvers has not been examined. In particular, more modern variants such as the thick-restart Lanczos method [wu98] may prove more efficient.

The use of eigenvalue bounds to construct an approximate sparse matrix representation of the conductance matrix is also an issue. While it may appear to be a reasonable approach, it is necessary to perform numerical experiments on non-trivial environments to determine its effectiveness.

# Chapter 5

## Numerical Experiments

### 5.1 Motivation

The empty rectangular room examined in Chapter 3 was useful for the purposes of elucidating the principles of eigenvector radiosity. However, it is obviously not representative of the complex environments that are of interest to architects and lighting designers. This leaves open the question of whether a small subset of the eigenvectors of a conductance matrix is sufficient to model the flow of light within complex environments.

In this chapter we examine three such environments: the Warehouse, the Tatami Room and the Ashdown Residence. Each environment is representative of the levels of complexity generally favored by architectural lighting designers – on the order of 5,000 to 50,000 elements.

### 5.2 Warehouse

The Warehouse is a single rectangular room with four long rectangular obstructions that represent stacked warehouse shelves. This deliberately simple design is representative of the environments that lighting designers may generate when they are interested only in the illuminance distribution for building code compliance. In these situations, architectural visualization is not an issue.

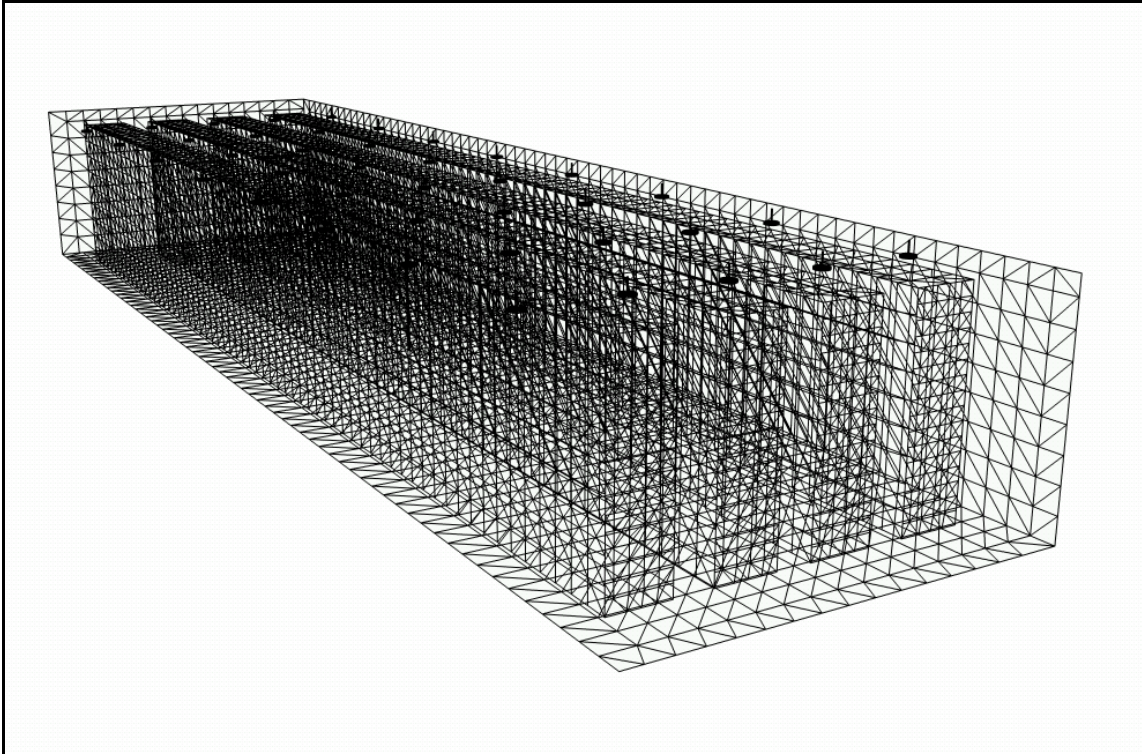


Figure 5.1 – Warehouse (wireframe).

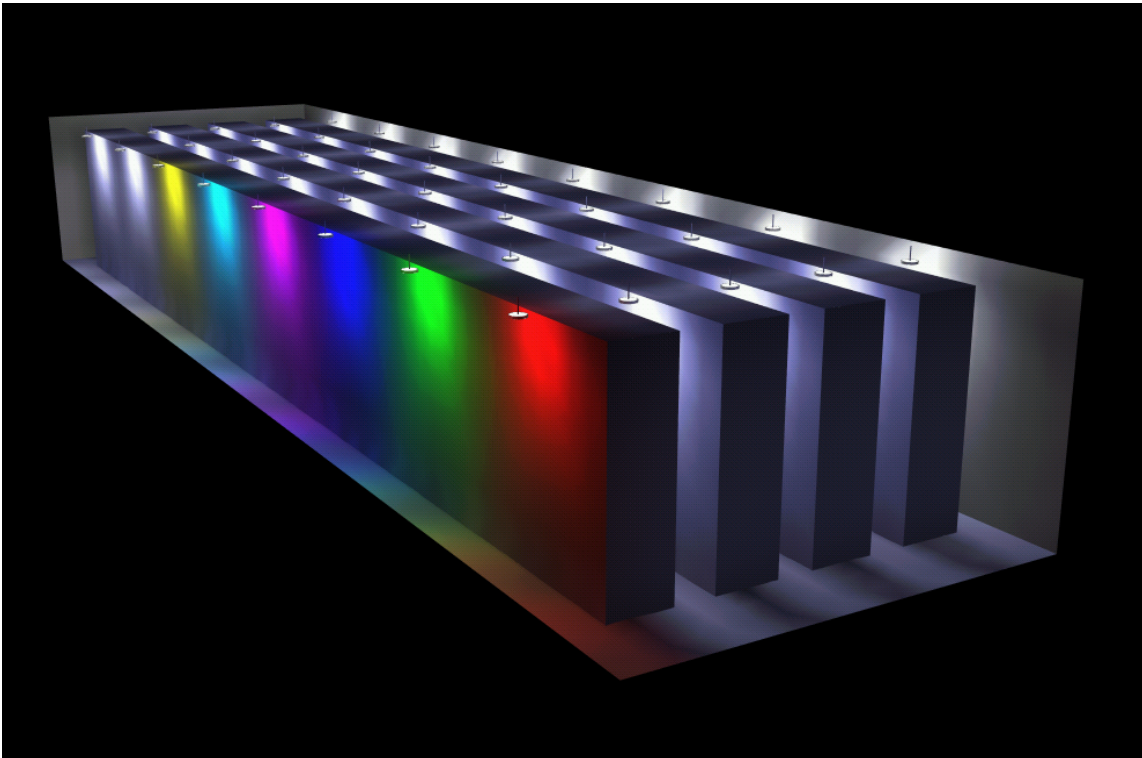


Figure 5.2 – Warehouse (progressive radiosity).



The Warehouse environment consists of 5,952 patches and 26,924 elements, where most of the patches have an area of approximately 5.0 square meters. The warehouse shelves are illuminated by 40 overhead luminaires.

Using progressive radiosity techniques, the environment reached 1.0% convergence in 1,996 steps, including 40 steps for direct irradiance distribution determination due to each of the physical (that is, non-diffuse) light sources. A total of 354 seconds of CPU time were required for the calculations, including 26 seconds for the direct irradiance calculations. (A 450 MHz Pentium II desktop computer was used for the experiments.)

During the progressive radiosity calculations, the eigenvalue bound for each conductance matrix row was calculated in accordance with Equation 4.1. Those rows with the largest magnitude eigenvalue bounds were saved in the conductance matrix row cache. The cache size was set a priori to 200, which means that only 3.3% of the full conductance matrix was saved.

Because the environment is closed, these bounds should be equal to the shooting patch area  $A_i$  for each patch  $i$ . This illustrates a disadvantage of using eigenvalue bounds based on the Gerschgorin Circle theorem. If most of the environment patches have approximately the same area, their corresponding eigenvalue bounds will also be similar. This is evident in Figure 5.3, where the bounds range from 52.4 to 47.2 for the first 200 conductance matrix rows. (The environment is modeled in feet rather than meters.)

The eigenvalues calculated for the sparse conductance matrix indicate that there are 181 eigenvalues with magnitudes greater than 10% of the spectral radius (3.0% of the total), and 194 eigenvalues with magnitudes greater than 5% of the spectral radius (3.3% of the total). However, the eigenvalue bounds indicate that the conductance matrix rows

have probably been selected for caching more or less at random during the progressive radiosity calculations. Given that the cached rows represent only 3.3% of the full conductance matrix, the calculated eigensystem of the sparse matrix may not be similar to that of the full matrix.

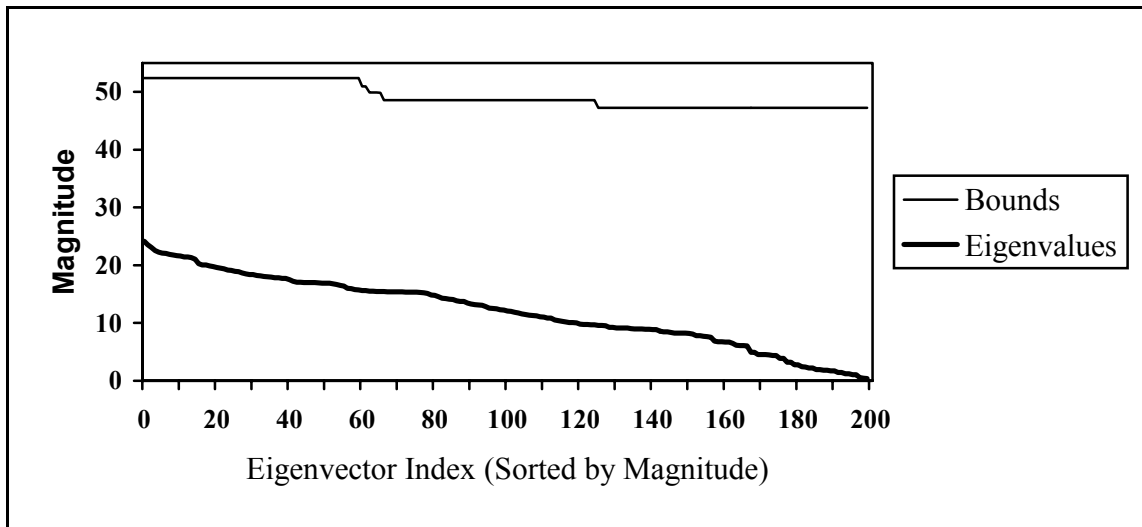


Figure 5.3 – Conductance array eigenvalues for Warehouse environment.

Surprisingly, eigenvector radiosity appears to be a remarkably robust technique. Figure 5.4 shows an eigenvector radiosity solution for the Warehouse using only 40 eigenvectors. Convergence to 0.1% was reached in only 2 iterations, and required 0.38 seconds of CPU time.

The visual artifacts due to displaying the patch exitances as opposed to the element exitances are clearly visible. Otherwise, there appears to be little difference (at least visually) between Figures 5.2 and 5.4.

What is truly remarkable is that the 40 eigenvectors represent only 0.7% of the total and 22% of those with eigenvalues greater than 10% of the spectral radius. The sparse conductance matrix clearly contains most of the dominant path information within its largest few eigenvectors.

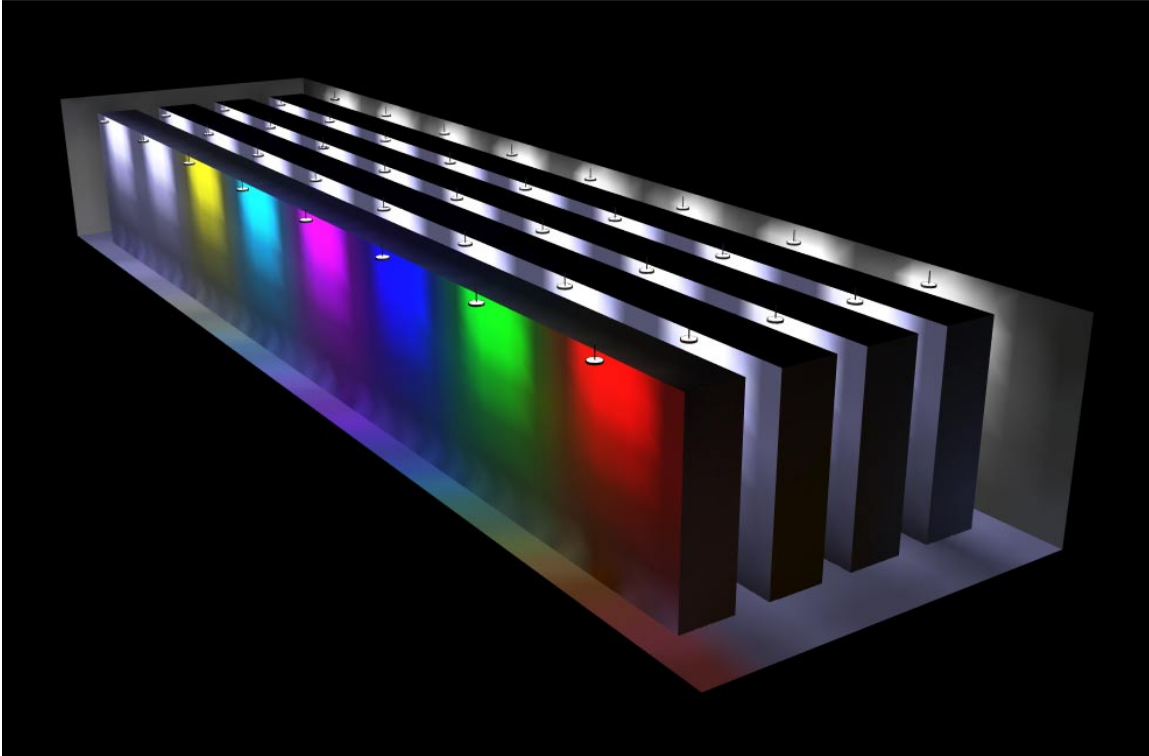


Figure 5.4 – Warehouse (40 eigenvectors).

### 5.3 Tatami Room

The Tatami Room<sup>1</sup> is a single rectangular room with a small number of occluded surfaces. It is representative of a visual environment that an architect or lighting designer would generate for presentation to the client. The emphasis is more on the photorealistic image than it is on illuminance or luminance distributions.

The Tatami Room environment consists of 1,192 patches and 3,798 elements (Figure 5.5). Direct illumination is provided by two electric ceiling luminaires and diffuse daylight through two windows. These light sources are modeled as diffuse self-luminous patches.

---

<sup>1</sup> The word *tatami* means "floor mat" in Japanese. However, a *tatami room* in North American usage generally refers to a private dining room in a Japanese restaurant.

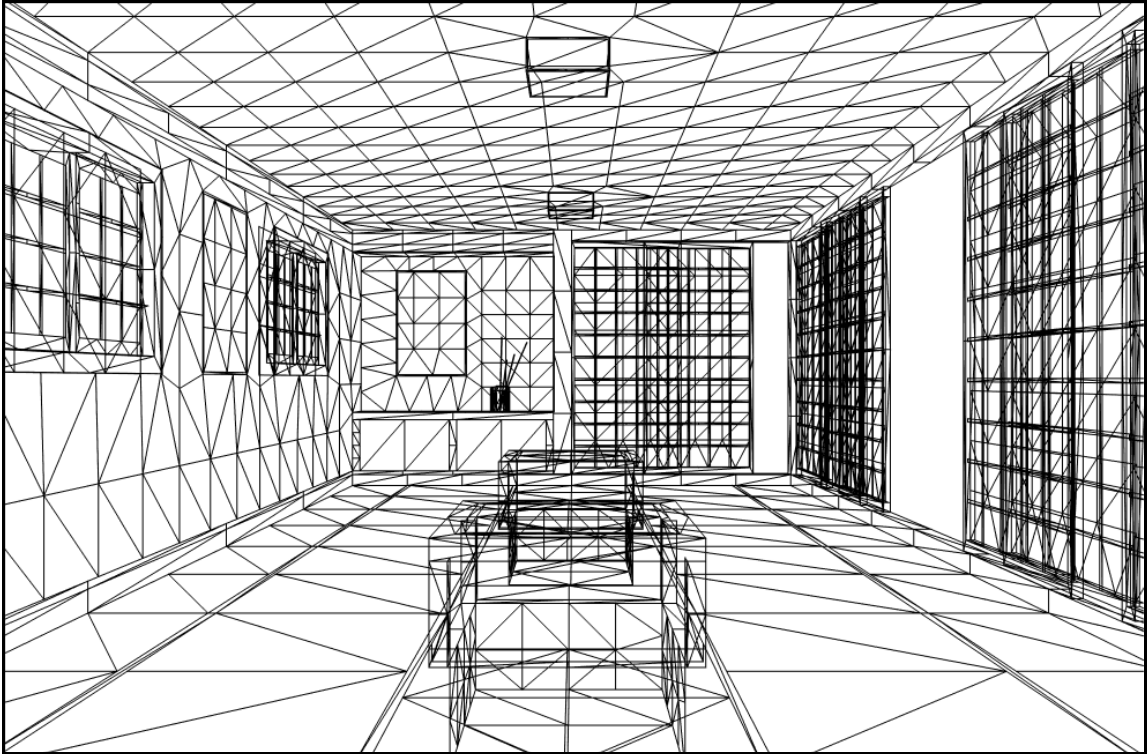


Figure 5.5 – Tatami Room (wireframe).



Figure 5.6 – Tatami Room (progressive radiosity).

Unlike the Warehouse environment, the patch areas in the Tatami Room environment range from approximately 1.8 square meters for the floor and ceiling to  $10^{-3}$  square meters for the branches in the vase. The shoji screens are also partially open, which means that the environment itself is not fully closed.

### **5.3.1 Patch-Element Solution**

Using progressive radiosity techniques, the environment reached 1.0% convergence in 2,228 steps (Figure 5.6). During the calculations, the conductance matrix rows with the 100 largest magnitude eigenvalues were cached, producing a sparse matrix with 8.3% fill-in.

Figure 5.7 shows both the eigenvalue bounds and calculated eigenvalues for this matrix. There are 35 eigenvalues with magnitudes greater than 10% of the spectral radius (2.9% of the total), and 78 eigenvalues with magnitudes greater than 5% of the spectral radius (6.5% of the total). Equally important are the eigenvalue bounds: the Tatami Room environment has only 84 bounds greater than 10% of the maximum bound (7.0% of the total).

The important point here is that the significant conductance matrix rows can be cached using a reasonable amount of memory (in this case approximately 1/2 megabyte). This implies that the sparse conductance matrix is similar to the full matrix, and that their eigensystems should also be similar.

As noted in Chapter 4, it is not necessary that these two eigensystems be identical. All that is needed is an eigensystem for which the iterative eigenvector radiosity calculations converge to a reasonably stable solution. The solution vector can then be iteratively refined using progressive radiosity techniques.

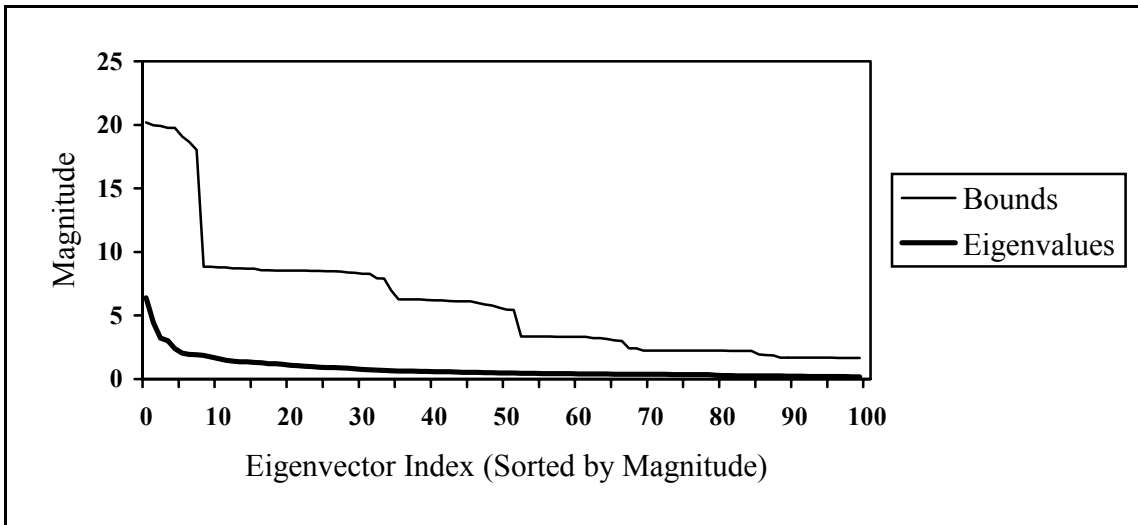


Figure 5.7 – Conductance array eigenvalues for Tatami Room environment.

Given a suitable subset of the dominant eigenvectors, an eigenvector radiosity solution can be calculated using Equations 3.37 through 3.38. Moreover, the environment can be displayed by assigning the calculated exitances of the patches to their respective elements.

Figure 5.8 shows an eigenvector radiosity solution for the Tatami Room using 35 eigenvectors. Convergence to 0.1% was reached in only 11 iterations, and required only 0.55 seconds of CPU time (By comparison, the progressive radiosity solution required 115 seconds of CPU time.)

Figure 5.9 shows an eigenvector radiosity solution for the Tatami Room using 78 eigenvectors. Convergence to 0.1% was reached in 14 iterations, and required 0.60 seconds of CPU time.



Figure 5.8 – Tatami Room (35 eigenvectors).



Figure 5.9 – Tatami Room (78 eigenvectors).

These approximate solutions are surprisingly close to the progressive radiosity solution shown in Figure 5.6. There are some visual artifacts evident (such as the abnormally dark ceiling areas over the shoji screens in Figure 5.8), but otherwise the images are very similar. (As might be expected, the solution using 78 eigenvectors has fewer artifacts than the solution using 35 eigenvectors.)

### 5.3.2 Modified Patch-Element Solution

The preceding experiment was based on progressive radiosity using the conventional patch-element hierarchy. In the following experiment, the geometric database was modified to minimize the number of elements per patch. (The **Helios** radiosity renderer generates quadrilateral patches and triangular elements wherever possible when it subdivides planar surfaces.) The resultant consists of 840 patches and 1,197 elements (Figure 5.10).

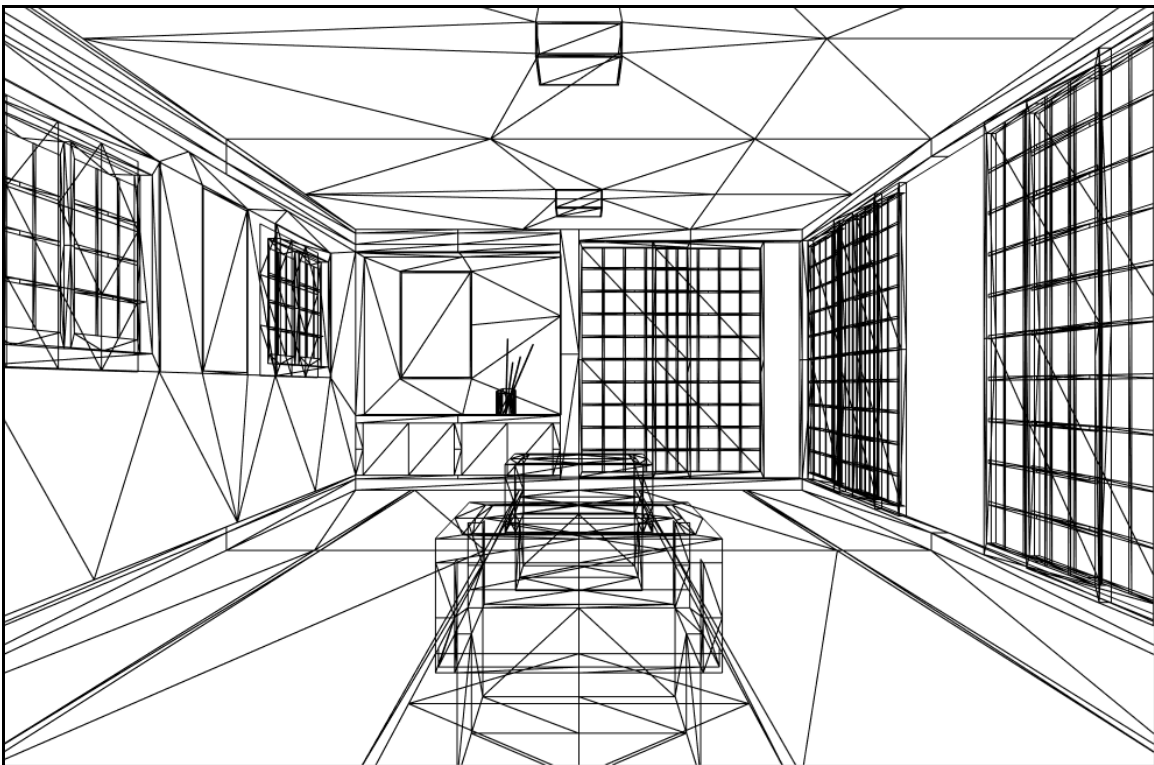


Figure 5.10 – Modified Tatami Room (wireframe).



Again using progressive radiosity, the environment reached 1.0% convergence in 1,112 steps and 41 seconds of CPU time (Figure 5.11). During the calculations, the conductance matrix rows with the 100 largest magnitude eigenvalues were cached, producing a sparse matrix with 11.9% fill-in.



Figure 5.11 – Modified Tatami Room (progressive radiosity).

Figure 5.12 shows both the eigenvalue bounds and calculated eigenvalues for this matrix. There are 27 eigenvalues with magnitudes greater than 10% of the spectral radius (3.2% of the total), and 44 eigenvalues with magnitudes greater than 5% of the spectral radius (5.2% of the total). In addition, the modified Tatami Room environment has only 60 bounds greater than 10% of the maximum bound (7.1% of the total).

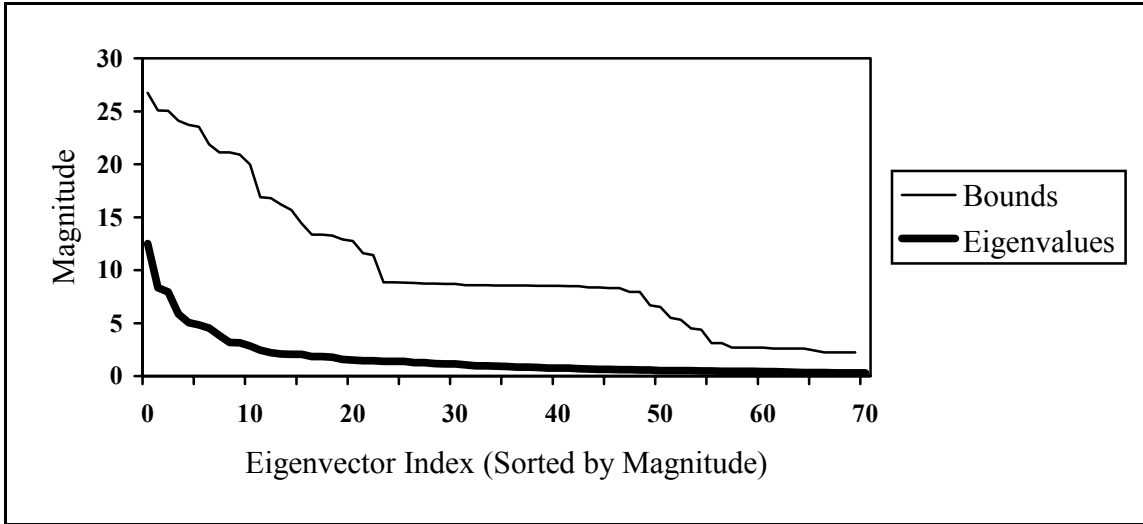


Figure 5.12 – Conductance array eigenvalues for modified Tatami Room environment.

Figure 5.13 shows an eigenvector radiosity solution for the modified Tatami Room using only 20 eigenvectors. Convergence to 0.1% was reached in 8 iterations, and required only 0.11 seconds of CPU time.

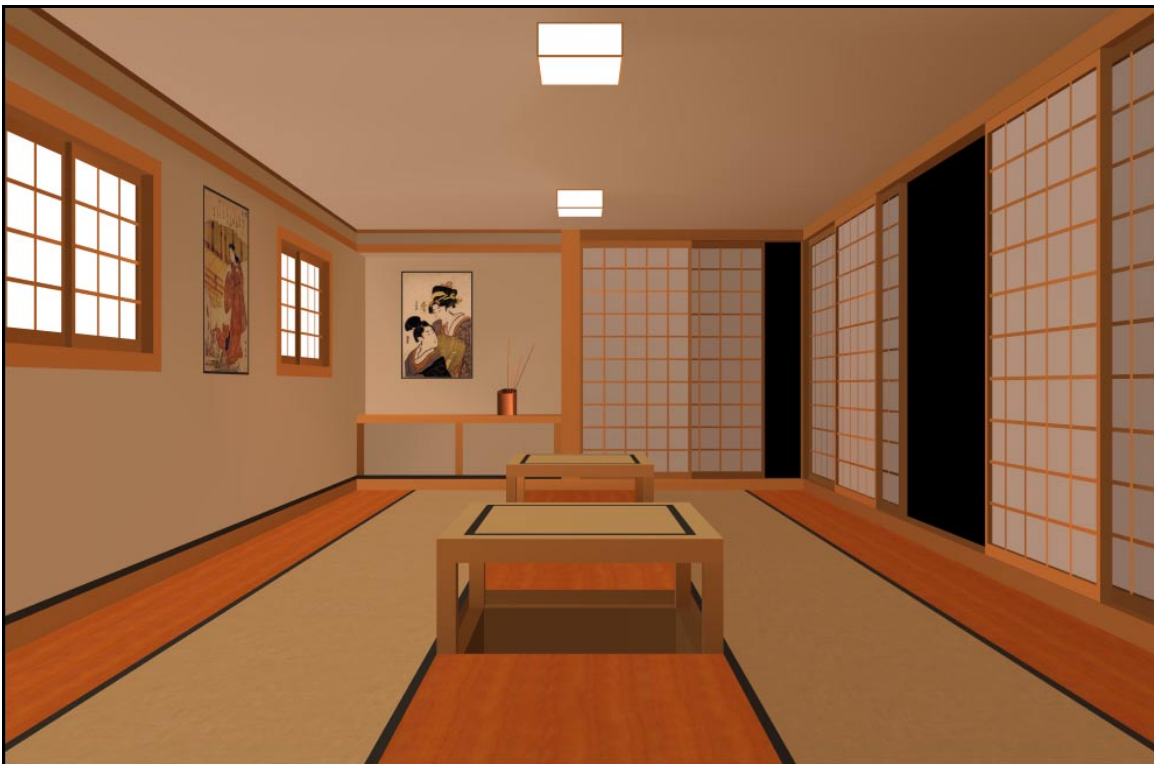


Figure 5.13 – Modified Tatami Room (20 eigenvectors).

Figure 5.14 shows an eigenvector radiosity solution for the modified Tatami Room using 70 eigenvectors. Convergence to 0.1% was reached in 7 iterations, and required only 0.16 seconds of CPU time.



Figure 5.14 – Modified Tatami Room (70 eigenvectors).

These approximate solutions are again surprisingly close to the progressive radiosity solution shown in Figure 5.11. The only obvious visual artifacts are the dark shadows under the tables and the over-illuminated vase. Unlike the previous experiment, there are no visible shading anomalies on the walls or ceiling in Figure 5.14. However, the far wall behind the vase appears somewhat over-illuminated in Figure 5.13.

Looking more closely at both images, it is evident that the overall color cast in Figure 5.11 that is due to multiple reflections from the redwood floor are absent in the eigenvector radiosity images. There is no apparent explanation for this anomaly, which will require further investigation.

### 5.3.3 Daylight Solution

In this experiment, the RGB spectral radiant exitance assigned to the window patches of the modified Tatami Room environment was changed from white (1000.0, 1000.0, 1000.0) to sky blue (500.0, 900.0, 1000.0) to simulate diffuse skylight with no direct solar illuminance. The purpose of this experiment was to examine the eigenvector radiosity solution following changes to the initial (in this case spectral) radiant exitance vector.

Figure 5.15 shows the progressive radiosity solution for the modified Tatami Room. The windows still appear white because they are effectively overexposed in the renderings. However, the effect of the daylight illuminant is clearly evident in the overall greenish color cast.



Figure 5.15 – Modified Tatami Room (progressive radiosity – daylight).

Figure 5.16 shows the corresponding eigenvector radiosity solution using 20 eigenvectors. Convergence to 0.1% was again reached in 8 iterations using 0.11 seconds of CPU time, with visual anomalies similar to Figure 5.13.



Figure 5.16 – Modified Tatami Room (20 eigenvectors – daylight).

#### **5.4 Ashdown Residence**

The Ashdown environment is a multi-room house with a large number of occluded surfaces. It is also representative of a visual environment that an architect or lighting designer would generate for presentation to the client, particularly as a photorealistic three-dimensional model that can be viewed interactively.

The Ashdown Residence environment consists of 2,451 patches and 9,343 elements with 14 physical (non-diffuse) light sources, and with a wide range of patch areas (Figure 5.17).



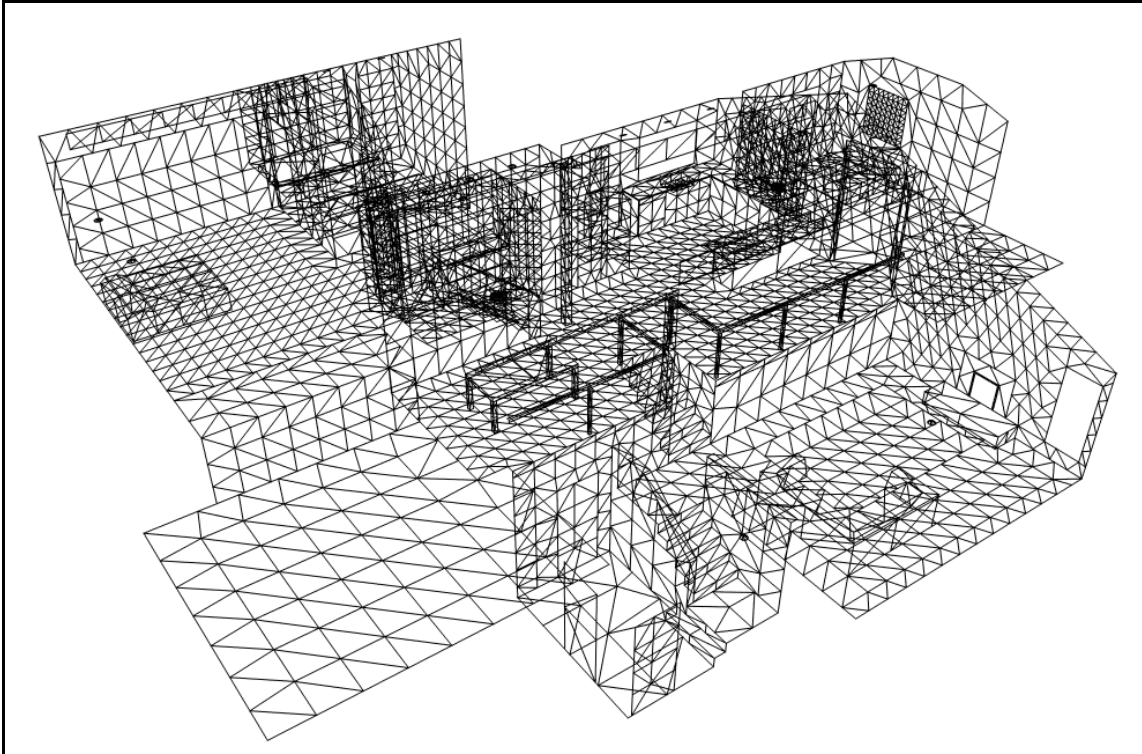


Figure 5.17 – Ashdown Residence (wireframe).

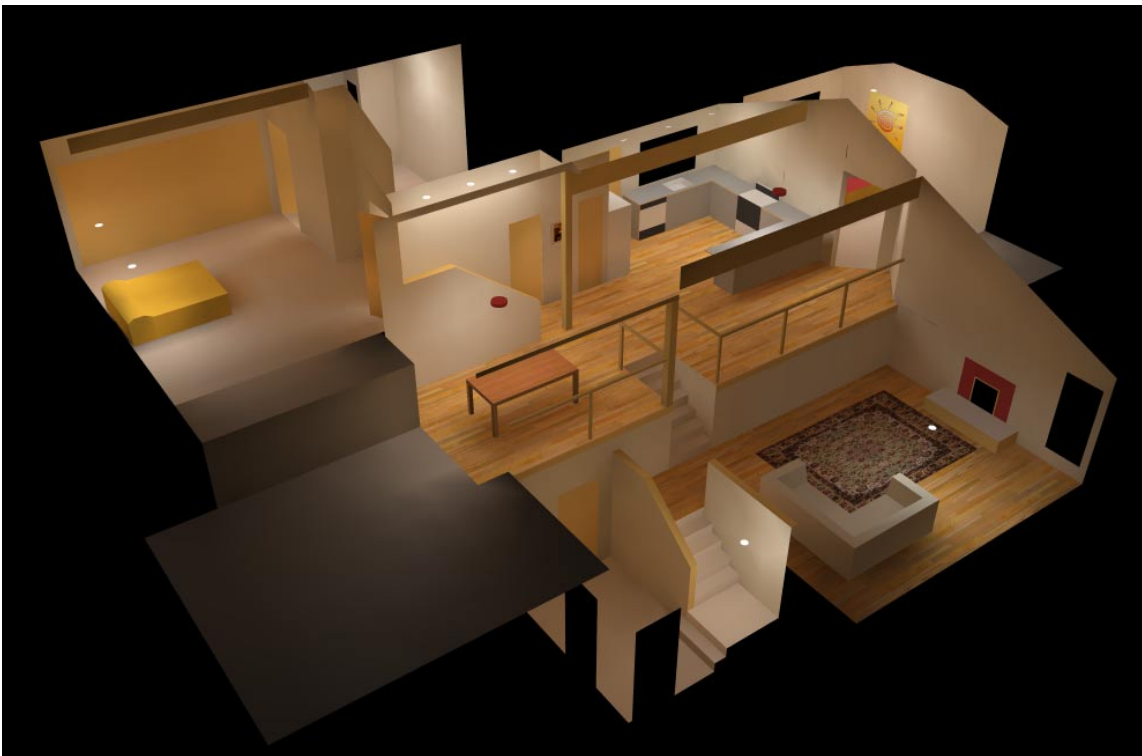


Figure 5.18 – Ashdown Residence (progressive radiosity).

Using progressive radiosity techniques, the environment reaches 1.0% convergence in 6,317 steps (Figure 5.18). The conductance matrix row cache size was 200, which produced a sparse matrix fill-in of 7.8%.

Figure 5.19 shows the eigenvalue bounds and calculated eigenvalues for this matrix. The Ashdown Residence environment has 120 eigenvalues greater than 10% of the spectral radius (4.7% of total) and 149 eigenvalues greater than 5% of the spectral radius (5.8% of total).

The eigenvalue bounds are less encouraging in that the magnitude of the smallest cached bound is 26% of the largest bound. There is therefore less reason to be confident that the eigensystems of the sparse and full conductance matrices are similar. Because most of the environment patches have areas of approximately 0.8 square meters (8.6 square feet), there would be little advantage in increasing the conductance matrix row cache size.

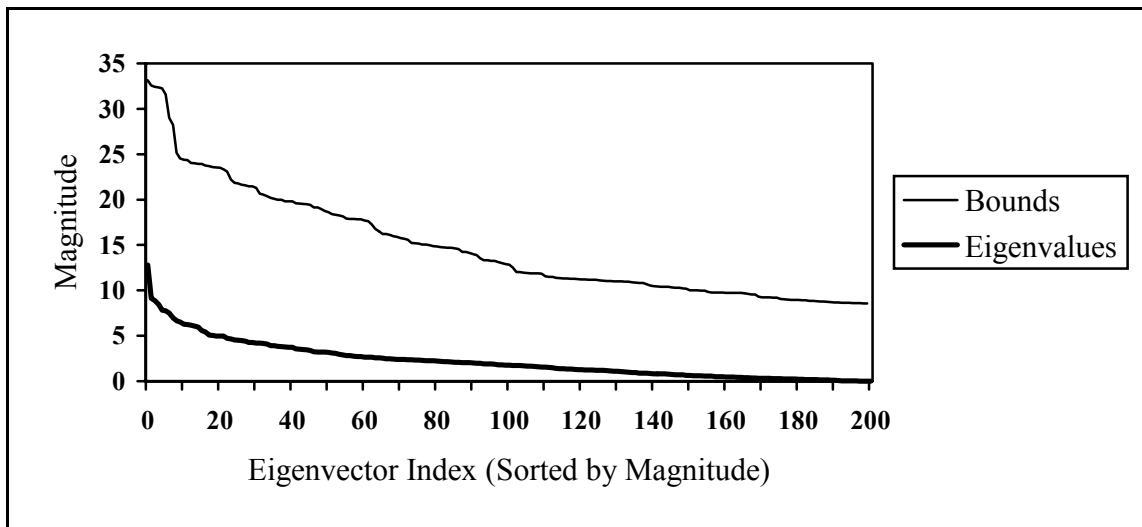


Figure 5.19 Conductance array eigenvalues for Ashdown Residence.

It should be recognized however that the Ashdown Residence is a partitioned environment with five separate rooms. Because of the limited flow of light between these

rooms, it is understandable that more eigenvectors are needed to fully characterize the environment.

An attempt was made to produce an eigenvector radiosity solution for the Ashdown Residence environment using 50 eigenvectors. However, this was clearly insufficient, as the solution converged in 13 iterations to 3% and then oscillated between 2% and 10%.

## **5.5 Conclusions**

The examples presented in this chapter illustrate both the strengths and weaknesses of eigenvector radiosity. In terms of strength, the Tatami Room environment indicates that complex environments have relatively few dominant eigenvectors, and that they can be successfully cached and saved for later use.

The eigenvector radiosity solutions for the Warehouse and Tatami Room environments indicate that the approach is surprisingly robust, even when less than 3% of the conductance matrix eigenvectors are used.

In terms of weakness, the spectral distribution of the eigenvalue bounds is dependent the distribution of patch areas. For extremely regular environments such as the Warehouse, it may not be possible to determine which patches should be cached. (The Warehouse environment was likely an exception because of its symmetry.) Fortunately, the distribution of patch areas can be used to identify such problem environments without the need to perform progressive radiosity calculations.

The Ashdown Residence environment indicates that eigenvector radiosity is not well suited for complex environments that have little symmetry and numerous occlusions. To be useful with such environments, it is likely that the approach will have to be combined with environment partitioning techniques.



## 5.6 Acknowledgments

The Block Lanczos eigensolver used in the development of this thesis was a C++ adaptation of an eigensolver written by Richard Underwood for his PhD thesis, “An Iterative Block Lanczos for the Solution of Large Sparse Symmetric Eigenproblems” [unde75]. The Fortran 77 source code for this program is available as `underwood.f` from the Netlib Repository ([www.netlib.org](http://www.netlib.org)). It requires the **EISPACK** functions `tred2` and `tql2`, which are also available from the Netlib Repository as `tred2.f` and `tql2.f` respectively.

This Block Lanczos eigensolver was used in conjunction with a modified version of the **Helios** progressive radiosity renderer to produce the conductance array eigenspectra and render the images presented in this chapter.

# Chapter 6

## Conclusions

### 6.1 Review

It has been shown that eigenanalysis may be used to decompose a radiative transfer network, and that each of its linearly independent components represents a radiative transfer network.

It has been further shown that the radiosity equation can be iteratively solved using a subset of the eigendecomposition of its associated conductance matrix, and that progressive radiosity techniques (in particular, Southwell iteration) can be modified to provide approximate conductance matrices without the need to calculate the full form factor matrix.

Finally, it has been demonstrated through numerical experiments that eigenvector radiosity can be successfully applied as a software engineering tool for architects and lighting designers.

### 6.2 Related Applications

It may be possible to apply eigenvector radiosity to other scientific and engineering disciplines. Certainly the numerical experiments are applicable to the closely related fields of thermal and aerospace engineering. However, the ability of eigenvector radiosity to model any physical system that can be represented as a linear resistive network is very attractive.

A good example is acoustical engineering, where there are so far no good solutions to the problem of modeling moving sources and/or receivers in arbitrarily complex environments. There are complicating issues such as sound velocity, media absorption, phase cancellation, and diffraction. However, eigenanalysis of the equivalent linear resistive network for an acoustic environment may identify the dominant acoustic paths and so simplify the problem of calculating diffuse interreflections.

More generally, linear resistive networks can be used to model neural networks, gas distribution systems, communication networks, transportation systems, nuclear reactors, and structural frameworks. Because the eigenvector radiosity technique developed in Chapter 3 relies only on network conductances, it should be applicable to all of these disciplines.

### **6.3 Future Work**

Time constraints prevented the investigation of several interested issues related to eigenvector radiosity. It was noted for example in Chapter 3 that eigenvector radiosity is a reformulation of Jacobi iteration. As with Jacobi iteration, it is “embarrassingly parallel.” There may therefore be scientific applications for eigenvector radiosity that can be executed on parallel processor machines while retaining the advantages of working only with the dominant eigenvectors of the conductance matrix.

On another topic, O’Brien and Bobco [obri64] and O’Brien and Gomez [obri67] extended the RT network analogy to include semi-diffuse and specular surfaces, while Oppenheim considered participating media [oppe54, oppe56]. Future work will investigate whether these extended analogies can be incorporated into eigenvector radiosity.

Eigenvalue bounds for the sparse conductance matrix are another topic worthy of investigation. The numerical experiments discussed in Chapter Five indicate that while the Gershgorin Circle theorem produces usable bounds, they are extremely conservative. An algorithm that produces tighter bounds would assist in constructing the sparse conductance matrix.

The relationship between eigenvector radiosity and other global illumination techniques such as hierarchical and wavelet radiosity [e.g., cohe93, sill94] should also be investigated. In particular, it is likely that a hierarchical formulation of eigenvector radiosity can be developed. This may assist in handling complex environments where the conductance matrix becomes too large for the Block Lanczos algorithm to solve in reasonable time.

Finally, it is evident from the numerical experiments that environment partitioning techniques will also assist in handling complex environments. As was discussed in Chapter Four, it may be possible to develop partitioning techniques that take advantage of the local and global path information contained in the conductance matrices.

#### **6.4 Final Words**

The original goal of this thesis was to answer the question, “What is the physical meaning of the eigensystem of a form factor matrix.” As the research work developed, a secondary goal became the development of a useful engineering tool based on eigenvector radiosity.

It can be said with some satisfaction that both goals have been achieved.

# Bibliography

- [aire90] J. M. Airey, J. H. Rohlf, and F. P. Brooks. 1990. “Towards Image Realism with Interactive Update Rates in Complex Virtual Building Environments,” *Computer Graphics* 24(1):41–50 (ACM Workshop on Interactive Graphics Proceedings).
- [ansi96] ANSI/IES. 1996. *American National Standard Nomenclature and Definitions for Illuminating Engineering*. ANSI/IES RP-16-96. New York, NY: Illuminating Engineering Society of North America.
- [ashd94] I. Ashdown. 1994. *Radiosity: A Programmer’s Perspective*. John Wiley & Sons, New York, NY.
- [axel96] O. Axelsson. 1996. *Iterative Solution Methods*. Cambridge University Press, Cambridge, UK.
- [bara97] G. V. G. Baranoski, R. Bramley, and J. G. Rokne. 1997. “Eigen-analysis for Radiosity Systems,” *Proceedings of the Sixth International Conference on Computational Graphics and Visualization (Compugraphics ’97)*, pp. 193–201.
- [bara98] G. V. G. Baranoski. 1998. *Biologically and Physically-Based Rendering of Natural Scenes*. PhD thesis, Department of Computer Science, University of Calgary, Calgary, Alberta.
- [baum89] D. R. Baum, H. E. Rushmeier, and J. M. Winget. 1989. “Improved Radiosity Solutions through the Use of Analytically Determined Form Factors,” *Computer Graphics* 23(3):298–306 (ACM SIGGRAPH ’89 Proceedings).

- [boug29] P. Bouger. 1729. *L'Essai d'Optique*. Paris, France.
- [burd85] R. L. Burden and J. D. Faires. 1985. *Numerical Analysis*. Prindle, Weber & Schmidt, Boston, MA.
- [chen90] S. E. Chen. 1990. "Incremental Radiosity: An Extension of Progressive Radiosity to an Interactive Image Synthesis System," *Computer Graphics* 24(4):135–144 (ACM SIGGRAPH '90 Proceedings).
- [cohe85] M. F. Cohen and D. P. Greenberg, 1985. "The Hemi-Cube: A Radiosity Solution for Complex Environments," *Computer Graphics* 19(3):31–40 (ACM SIGGRAPH '85 Proceedings).
- [cohe86] M. F. Cohen, D. P. Greenberg, D. S. Immel, and P. J. Brock. 1986. "An Efficient Radiosity Approach for Realistic Image Synthesis," *IEEE Computer Graphics and Applications* 6(3):26–35.
- [cohe88] M. F. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg. 1988. "A Progressive Refinement Approach to Fast Radiosity Image Generation," *Computer Graphics* 22(4):75–84 (ACM SIGGRAPH '88 Proceedings).
- [cohe93] M. F. Cohen and J. R. Wallace. 1993. *Radiosity and Realistic Image Synthesis*. Academic Press, San Diego, CA.
- [dale96] B. I. Dalenbäck. 1996. "Room Acoustic Prediction Based on a Unified Treatment of Diffuse and Specular Reflection," *Journal of the Acoustic Society of America* 100(2):899–909.
- [deif91] A. S. Deif. 1991. *Advanced Matrix Theory for Scientists and Engineers, Second Edition*. Abacus Press, New York, NY.

- [dila93] D. L. DiLaura and P. J. Franck. 1993. “On Setting Up and Solving Large Radiative Transfer Systems,” *Journal of the Illuminating Engineering Society*, Vol. 22, No. 2, pp. 3–7, Summer.
- [doba97] Dobashi, Y., T. Nishita, K. Kaneda, and H. Yamashita. 1997. “A Fast Display Method of Sky Color Using Basis Functions,” *Journal of Visualization and Computer Animation* 8(2):115-127.
- [feda92] M. Feda and W. Purgathofer. 1992. “Accelerating Radiosity by Overshooting,” *Proceedings of the Third Eurographics Workshop on Rendering*, Bristol, England.
- [gast70] N. Gastinel. 1970. *Linear Numerical Analysis*. Academic Press, San Diego, CA.
- [golu77] G. H. Golub and R. Underwood. 1977. “The Block Lanczos Method for Computing Eigenvalues,” in *Mathematical Software III*. Academic Press, New York, NY, pp. 361–377.
- [golu96] G. H. Golub and C. F. Van Loan. 1996. *Matrix Computations, Third Edition*. John Hopkins University Press, Baltimore, MD.
- [gora84] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. 1984. “Modeling the Interaction of Light Between Diffuse Surfaces,” *Computer Graphics* 18(3):213–222 (ACM SIGGRAPH ’84 Proceedings).
- [gort93] S. Gortler and M. F. Cohen. 1994. “Radiosity and Relaxation Methods,” *IEEE Computer Graphics and Applications* 14(6):48–58 (November).

- [grei93] G. Greiner, W. Heidrich, and P. Slusallek. 1993. “Blockwise Refinement – A New Method for Solving the Radiosity Equation,” *Proceedings of the Fourth Eurographics Workshop on Rendering*, Paris, France, pp. 233–245.
- [heck91] P. S. Heckbert. 1991. *Simulating Global Illumination Using Adaptive Meshing*. PhD thesis, Technical Report UCB/CSD 91/636, University of California Berkeley, Berkeley, CA.
- [howe82] J. R. Howell. 1982. *A Catalog of Radiation Configuration Factors*. McGraw-Hill, New York, NY.
- [iesn00] IESNA. 2000. *The IESNA Lighting Handbook, Ninth Edition*. Mark Rea, Editor-in-Chief. New York, NY: Illuminating Engineering Society of North America.
- [kaji86] J. T. Kajiya. 1986. “The Rendering Equation,” *Computer Graphics* 20(4):143–150 (ACM SIGGRAPH Proceedings).
- [koen83] J. J. Koenderink and A. J. van Doorn. 1983. “Geometrical Modes as a General Method to Treat Diffuse Interreflections,” *Journal of the Optical Society of America*, Vol. 73, No. 6, pp. 843–850, June.
- [kron63] G. Kron. 1963. *Diakoptics: The Piecewise Solution of Large-Scale Systems*. London, UK: McDonald.
- [lang99] M. S. Langer. 1999. “When Shadows Become Interreflections,” *International Journal of Computer Vision* 34(2/3):193–204.
- [lamb60] J. H. Lambert. 1760. *Photometria sive de mensura et gradibus luminus, colorum et umbrae*. German translation with annotations by E. Anding (1892),



*Ostwalds Klassiker der Exakten Wissenschaften*, Nos. 31–33, Leipzig, Germany.

- [moon40] P. Moon. 1940. “On Interreflections,” *Journal of the Optical Society of America*, Vol. 30, No. 5, pp. 195–205, May.
- [moon81] P. Moon and D. E. Spencer. 1981. *The Photoc Field*. MIT Press, Cambridge, MA.
- [neum94] L. Neumann. 1994. “New Efficient Algorithms with Positive Definite Radiosity Matrix,” *Proceedings of the Fifth Eurographics Workshop on Rendering*, Darmstadt, Germany, pp. 219–237, June.
- [niev97] Y. Nievergelt. 1997. “Making Any Radiosity Matrix Symmetric Positive Definite,” *Journal of the Illuminating Engineering Society*, Vol. 26, No. 1, pp. 165–171, Winter.
- [nime94] Nimeroff, J. S., E. Simoncelli, and J. Dorsey. 1994. “Efficient Re-rendering of Naturally Illuminated Environments,” *Proceedings of the Fifth Eurographics Workshop on Rendering*, pp. 359–373.
- [nish85] T. Nishita and E. Nakamae. 1985. “Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflections,” *Computer Graphics* 19(3):23–30 (ACM SIGGRAPH ’85 Proceedings).
- [nobl69] B. Noble. *Applied Linear Algebra*. Prentice Hall, Englewood Cliffs, NJ.
- [obri55] P. F. O’Brien. 1955. “Interreflections in Rooms by a Network Method,” *Journal of the Optical Society of America* 45(6):419–424.
- [obri64] P. F. O’Brien and R. P. Bobco. 1964. “Interreflections in Mirrored Rooms,” *Illuminating Engineering* 59(5):337–344.

- [obri67] P. F. O'Brien and A. V. Gomez. 1967. "Luminous Transfer in Rooms with Semidiffuse-Specular Surfaces," *Illuminating Engineering* 62(4):180–186.
- [oppe54] A. K. Oppenheim. 1954. *The Network Method of Radiation Analysis*. Heat Transfer and Fluid Mechanics Institute, University of California, Berkeley, CA.
- [oppe56] A. K. Oppenheim. 1956. "Radiation Analysis by the Network Method," *Transactions of the ASME* 78:725–735.
- [parl79] B. N. Parlett and D. S. Scott. 1979. "The Lanczos Algorithm with Selective Orthogonalization," *Mathematics of Computation* 33(145):217–238.
- [parl98] B. N. Parlett. 1998. *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia, PA.
- [rama99] R. Ramamoothi. 1999. *Eigenmodes of Scene Geometry*. Unpublished manuscript, Stanford Computer Graphics Laboratory, Stanford University, Stanford, CA.
- [shao93] M. Shao and N. Badler. 1993. "Analysis and Acceleration of Progressive Refinement Radiosity Method," *Proceedings of the Fourth Eurographics Workshop on Rendering*, Paris, France, pp. 247–258.
- [sieg92] R. Siegel and J. R. Howell. 1992. *Thermal Radiation Heat Transfer, Third Edition*. Hemisphere Publishing, Washington, DC.
- [sill94] F. X. Sillion and C. Puech. 1994. *Radiosity & Global Illumination*. Morgan Kaufmann, San Francisco, CA.
- [spar78] E. Sparrow and R. Cess. 1978. *Radiation Heat Transfer*. Hemisphere Publishing, Washington, DC.

- [thom94] R. E. Thomas and A. J. Rosa. 1994. *The Analysis and Design of Linear Circuits*. Prentice-Hall, Englewood Cliffs, NJ.
- [tsin98] N. Tsingos. 1998. *Simulation de Champs Sonores de Haute Qualité pour des Applications Graphiques Interactives*. PhD thesis, Université Joseph Fourier, Grenoble, France.
- [unde75] R. Underwood. 1975. *An Iterative Block Lanczos Method for the Solution of Large Sparse Symmetric Eigenproblems*. PhD thesis, Technical Report STAN-CS-75-496, Stanford University, Stanford, CA.
- [wilk65] J. H. Wilkinson. 1965. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, UK.
- [wilk71] J. H. Wilkinson and C. Reinsch. 1971. *Handbook for Automatic Computation*. Berlin, Germany: Springer Verlag.
- [wu98] K. Wu and H. Simon. 1998. *Thick-Restart Lanczos Method for Symmetric Eigenvalue Problems*. Technical Report LBNL-41412, Lawrence Berkeley National Laboratory/NERSC, Berkeley, CA.
- [yama26] Z. Yamauti. 1926. "The Light Flux Distribution of a System of Interreflecting Surfaces," *Journal of the Optical Society of America* 13(5):561–571.

# Appendix A

## MATLAB Experiments

### A.1 Does It Really Work?

While the numerical experiments presented in Chapter Five demonstrate the application of eigenvector radiosity to practical illumination engineering problems, they do not provide an unequivocal answer to the question: does eigenvector radiosity really work?

The problem is that the **Helios** progressive radiosity renderer used for the numerical experiments calculates form factors using numerical quadrature rather than analytic techniques. Moreover, these form factors model shooting patch as having differential rather than a finite areas.

For most progressive radiosity applications these approximate form factors are acceptable – the error is typically less than  $\pm 1$  percent [ashd94]. However, it is advisable to implement eigenvector radiosity using analytic form factors, if only to examine its behavior with simple environments.

### A.2 Form Factor Calculations

Despite their apparent simplicity, form factors are notoriously difficult to solve using analytic methods. Johann Lambert, a pioneer researcher in photometry and likely the first person to consider the problem, wrote [lamb60]:

*Although this task appears very simple, its solution is considerably more knotted than one would expect ... the highly laborious computation would fill even the most patient with disgust and drive them away.*

Lambert expressed this opinion in reference to the problem of determining the form factor between two perpendicular rectangles sharing a common edge. However, his comments apply equally well to analytic form factor determination in general. As discussed in Chapter 2, the form factor from a finite area patch  $A_i$  to another finite area patch  $A_j$  (Figure A.1) is given by the double area integral equation:

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_j dA_i \quad (\text{A.1})$$

In general, this equation cannot be solved directly. However, it is often possible (with “highly laborious computation”) to determine closed-form solutions for simple geometries.

Mechanical and aeronautical engineers have long used published tables of formulae for specific area-to-area geometries in their radiant heat transfer studies, including those by Howell [howe82], Siegel and Howell [sie92], and Sparrow and Cess [spar78]. These include simple shapes such as parallel and perpendicular rectangles, circle, and hollow tubes. More complex geometries can be analyzed using form factor algebra [e.g., sie92] to geometrically add and subtract these shapes and their associated form factors.

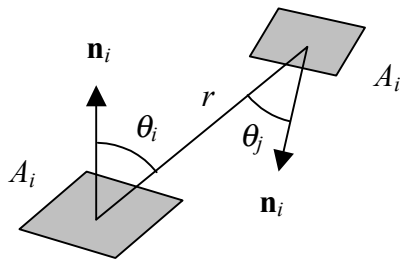


Figure A.1 – Patch-to-patch form factor geometry.

Two analytic form factor equations are required for the purposes of this appendix: one for adjoining perpendicular rectangles (Figure A.2), and another for parallel rectangles (Figure A.3). Both equations are obtained from Howell [howe82].

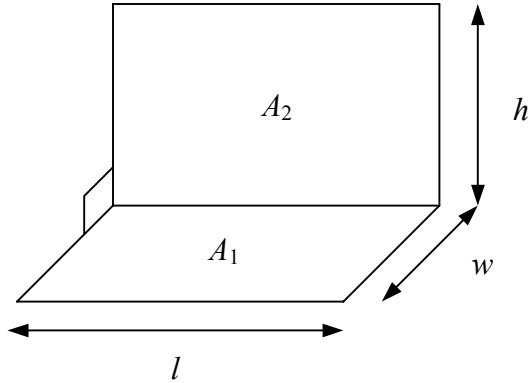


Figure A.2 – Perpendicular rectangles with common edge.

$$\begin{aligned}
 H &= h/l, \quad W = w/l \\
 F_{12} &= \frac{1}{\pi W} \left[ W \arctan\left(\frac{1}{W}\right) + H \arctan\left(\frac{1}{H}\right) - \sqrt{H^2 + W^2} \arctan\frac{1}{\sqrt{H^2 + W^2}} \right. \\
 &\quad \left. + \frac{1}{4} \ln \left\{ \frac{(1+W^2)(1+H^2)}{1+W^2+H^2} \left[ \frac{W^2(1+W^2+H^2)}{(1+W^2)(W^2+H^2)} \right]^{W^2} \left[ \frac{H^2(1+W^2+H^2)}{(1+H^2)(W^2+H^2)} \right]^{H^2} \right\} \right] \quad (A.2)
 \end{aligned}$$

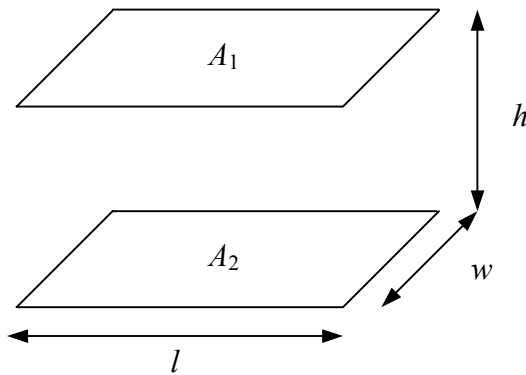


Figure A.3 – Parallel rectangles.

$$\begin{aligned}
 X &= w/h, \quad Y = l/h \\
 F_{12} &= \frac{2}{\pi XY} \left[ \ln \sqrt{\frac{(1+X^2)(1+Y^2)}{1+X^2+Y^2}} + X\sqrt{1+Y^2} \arctan\left(\frac{X}{\sqrt{1+Y^2}}\right) + \right. \\
 &\quad \left. Y\sqrt{1+X^2} \arctan\left(\frac{Y}{\sqrt{1+X^2}}\right) - X \arctan X - Y \arctan Y \right] \quad (A.3)
 \end{aligned}$$

With these two equations, the form factors between any two surfaces in an empty rectangle room (Figure A.4) can be determined.

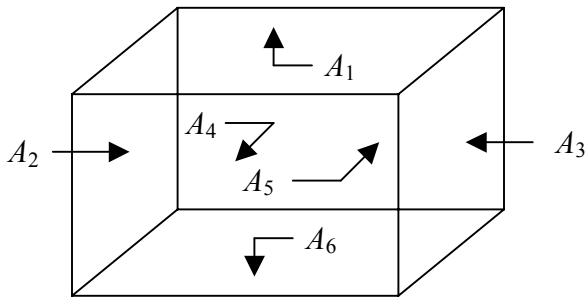


Figure A.4 – Empty rectangular room.

The following MATLAB listings implement Equation A.2 for perpendicular surfaces and Equation A.3 for parallel surfaces:

```

% Calculate form factor for two finite rectangles of
% the same length with one common edge at right angles
% to each other

function FF = calc_perp(width, length, height)

H = height / length;
W = width / length;

SH = H * H;
SW = W * W;

A = 1.0 / (pi * W);
B = W * atan(1.0 / W) + H * atan(1.0 / H);
C = sqrt(SH + SW) * atan(1.0 / sqrt(SH + SW));
D = (1.0 + SW) * (1.0 + SH) / (1.0 + SW + SH);
E = SW * (1.0 + SW + SH) / ((1.0 + SW) * (SW + SH));
F = SH * (1.0 + SW + SH) / ((1.0 + SH) * (SW + SH));

FF = A * (B - C + 0.25 * log(D * E^(SW) * F^(SH)));

```

### Listing A.1 – MATLAB perpendicular rectangles form factor determination

```
% Calculate form factor for identical parallel directly
% opposed rectangles

function FF = calc_para(width, length, separation)

X = width / separation;
Y = length / separation;

SX = X * X;
SY = Y * Y;

A = 2.0 / (X * Y * pi);
B = log(sqrt((1.0 + SX) * (1.0 + SY) / (1.0 + SX + SY)));
C = X * sqrt(1.0 + SY) * atan(X / sqrt(1.0 + SY));
D = Y * sqrt(1.0 + SX) * atan(Y / sqrt(1.0 + SX));
E = X * atan(X) + Y * atan(Y);

FF = A * (B + C + D - E);
```

### Listing A.2 – MATLAB parallel rectangles form factor determination

#### A.3 An Analytic Implementation

Using the above two functions, the following MATLAB listing calculates the radiant transfer system conductance matrix and its eigensystem for an empty room measuring 5.0 meters long by 3.0 meters wide and 2.5 meters high, then calculates the approximate final exitance vectors for a given initial exitance vector.

```
% Calculate radiosity conductance matrix and its eigensystem

% Calculate analytic form factors for empty room measuring
% 5.0 m long by 3.0 m wide by 2.5 m high
l = 5.0;
w = 3.0;
h = 2.5;

ff11 = 0.0000;
ff22 = ff11;
ff33 = ff11;
ff44 = ff11;
ff55 = ff11;
ff66 = ff11;

ff12 = calc_perp(l, w, h);
ff13 = ff12;
ff62 = ff12;
ff63 = ff12;

ff14 = calc_perp(w, l, h);
```



```

ff15 = ff14;
ff64 = ff14;
ff65 = ff14;

ff16 = calc_para(w, l, h);
ff61 = ff16;

ff21 = calc_perp(h, w, l);
ff26 = ff21;
ff31 = ff21;
ff36 = ff21;

ff23 = calc_para(h, w, l);
ff32 = ff23;

ff24 = calc_perp(w, h, l);
ff25 = ff24;
ff34 = ff24;
ff35 = ff24;

ff41 = calc_perp(h, l, w);
ff46 = ff41;
ff51 = ff41;
ff56 = ff41;

ff42 = calc_perp(l, h, w);
ff43 = ff42;
ff52 = ff42;
ff53 = ff42;

ff45 = calc_para(h, l, w);
ff54 = ff45;

% Construct form factor matrix
fprintf(1, 'Form factor matrix\n')
FF = [ ff11 ff12 ff13 ff14 ff15 ff16;
        ff21 ff22 ff23 ff24 ff25 ff26;
        ff31 ff32 ff33 ff34 ff35 ff36;
        ff41 ff42 ff43 ff44 ff45 ff46;
        ff51 ff52 ff53 ff54 ff55 ff56;
        ff61 ff62 ff63 ff64 ff65 ff66 ]

% Assign surface reflectances (ceiling 80%, walls 70%, floor 20%)
fprintf(1, 'Reflectance matrix\n')
R = [ 0.8 0.0 0.0 0.0 0.0 0.0;
      0.0 0.7 0.0 0.0 0.0 0.0;
      0.0 0.0 0.7 0.0 0.0 0.0;
      0.0 0.0 0.0 0.7 0.0 0.0;
      0.0 0.0 0.0 0.0 0.7 0.0;
      0.0 0.0 0.0 0.0 0.0 0.2 ]

% Assign surface areas
fprintf(1, 'Surface area matrix\n')
A = [ (l*w)   0.0   0.0   0.0   0.0   0.0;
      0.0 (w*h)  0.0   0.0   0.0   0.0;
      0.0   0.0 (w*h)  0.0   0.0   0.0;
      0.0   0.0   0.0 (l*h)  0.0   0.0;

```

```

0.0  0.0  0.0  0.0 (l*h)  0.0;
0.0  0.0  0.0  0.0  0.0 (l*w) ]

% Calculate conductance matrix
fprintf(1, 'Conductance matrix\n')
G = A * FF

% Calculate symmetric radiosity matrix eigensystem
[V,D]= eig(G);

% Get eigenvalues
eval = [ D(1,1); D(2,2); D(3,3); D(4,4); D(5,5); D(6,6) ];

% Sort eigenvalues by decreasing absolute magnitude
Q = [ abs(eval(1)); abs(eval(2)); abs(eval(3));
      abs(eval(4)); abs(eval(5)); abs(eval(6)) ];
[SV,I] = sort(Q);

% Extract eigenvalues
d1 = eval(I(6));
d2 = eval(I(5));
d3 = eval(I(4));
d4 = eval(I(3));
d5 = eval(I(2));
d6 = eval(I(1));
fprintf(1, 'Eigenvalues\n\n%f %f %f %f %f %f\n\n', ...
        d1, d2, d3, d4, d5, d6)

% Extract eigenvectors
v1 = V(:,I(6));
v2 = V(:,I(5));
v3 = V(:,I(4));
v4 = V(:,I(3));
v5 = V(:,I(2));
v6 = V(:,I(1));
fprintf(1, 'Eigenvectors\n')
V

% Construct conductance matrix components according to
% spectral theorem
fprintf(1, 'Conductance matrix spectral components\n')
G1 = d1 * (v1 * v1')
G2 = d2 * (v2 * v2')
G3 = d3 * (v3 * v3')
G4 = d4 * (v4 * v4')
G5 = d5 * (v5 * v5')
G6 = d6 * (v6 * v6')

% Reconstruct conductance matrix (validation)
%
% NOTE: Eigenvectors are ordered according to decreasing
%       eigenvalue absolute magnitude
%
H1 = G1;
H2 = H1 + G2;
H3 = H2 + G3;
H4 = H3 + G4;

```

```

H5 = H4 + G5;
H6 = H5 + G6;

% Construct initial exitance vector
fprintf(1, 'Initial exitance vector\n')
Mo = [1.0; 0.0; 0.0; 0.0; 0.0; 0.0 ]

% Solve directly
fprintf(1, 'Direct solution vector\n')
M = inv(eye(6,6) - (R / A ) * G) * Mo

% Determine approximate final exitance vectors
fprintf(1, 'Approximate solution vectors\n')
M6 = inv(eye(6,6) - (R / A ) * H6) * Mo
fprintf(1, 'Error = %f\n', norm((M - M6) / M), 2))
M5 = inv(eye(6,6) - (R / A ) * H5) * Mo
fprintf(1, 'Error = %f\n', norm((M - M5) / M), 2))
M4 = inv(eye(6,6) - (R / A ) * H4) * Mo
fprintf(1, 'Error = %f\n', norm((M - M4) / M), 2))
M3 = inv(eye(6,6) - (R / A ) * H3) * Mo
fprintf(1, 'Error = %f\n', norm((M - M3) / M), 2))
M2 = inv(eye(6,6) - (R / A ) * H2) * Mo
fprintf(1, 'Error = %f\n', norm((M - M2) / M), 2))
M1 = inv(eye(6,6) - (R / A ) * H1) * Mo
fprintf(1, 'Error = %f\n', norm((M - M1) / M), 2))

% Solve using Jacobi iteration
fprintf(1, '\nJacobi iteration\nIter\tError\n')
Mj = Mo;
I = 0;
while (norm(Mj - M) > 0.001 & I < 100)
    Mj = Mo + R * FF * Mj;
    I = I + 1;
    fprintf(1, '%d\t %f\n', I, norm((Mj - M) / M), 2))
end
Mj

% Precalculate S-array and w vectors
S = R / A;
w1 = S * v1;
w2 = S * v2;
w3 = S * v3;
w4 = S * v4;
w5 = S * v5;
w6 = S * v6;

% Solve using eigenvector radiosity with 6 eigenvectors
fprintf(1, 'Eigenvector radiosity (6)\nIter\tError\n')
Me = Mo;
K = Mo;
I = 0;
while (norm(Me - M) > 0.001 & I < 15)
    K = K' * v1 * d1 * w1 + ...
        K' * v2 * d2 * w2 + ...
        K' * v3 * d3 * w3 + ...
        K' * v4 * d4 * w4 + ...
        K' * v5 * d5 * w5 + ...

```

```

        K' * v6 * d6 * w6;
    Me = Me + K;
    I = I + 1;
    fprintf(1, '%d\t%f\n', I, norm((Me - M) / M),2))
end
Me

% Solve using eigenvector radiosity with 5 eigenvectors
fprintf(1, 'Eigenvector radiosity (5)\nIter\tError\n')
Me = Mo;
K = Mo;
I = 0;
while (norm(Me - M) > 0.001 & I < 15)
    K = K' * v1 * d1 * w1 + ...
        K' * v2 * d2 * w2 + ...
        K' * v3 * d3 * w3 + ...
        K' * v4 * d4 * w4 + ...
        K' * v5 * d5 * w5;
    Me = Me + K;
    I = I + 1;
    fprintf(1, '%d\t%f\n', I, norm((Me - M) / M),2))
end
Me

% Solve using eigenvector radiosity with 4 eigenvectors
fprintf(1, 'Eigenvector radiosity (4)\nIter\tError\n')
Me = Mo;
K = Mo;
I = 0;
while (norm(Me - M) > 0.001 & I < 15)
    K = K' * v1 * d1 * w1 + ...
        K' * v2 * d2 * w2 + ...
        K' * v3 * d3 * w3 + ...
        K' * v4 * d4 * w4;
    Me = Me + K;
    I = I + 1;
    fprintf(1, '%d\t%f\n', I, norm((Me - M) / M),2))
end
Me

% Solve using eigenvector radiosity with 3 eigenvectors
fprintf(1, 'Eigenvector radiosity (3)\nIter\tError\n')
Me = Mo;
K = Mo;
I = 0;
while (norm(Me - M) > 0.001 & I < 15)
    K = K' * v1 * d1 * w1 + ...
        K' * v2 * d2 * w2 + ...
        K' * v3 * d3 * w3;
    Me = Me + K;
    I = I + 1;
    fprintf(1, '%d\t%f\n', I, norm((Me - M) / M),2))
end
Me

% Solve using eigenvector radiosity with 2 eigenvectors
fprintf(1, 'Eigenvector radiosity (2)\nIter\tError\n')

```

```

Me = Mo;
K = Mo;
I = 0;
while (norm(Me - M) > 0.001 & I < 15)
    K = K' * v1 * d1 * w1 + ...
        K' * v2 * d2 * w2;
    Me = Me + K;
    I = I + 1;
    fprintf(1, '%d\t%f\n', I, norm(((Me - M) / M),2))
end
Me

% Solve using eigenvector radiosity with 1 eigenvector
fprintf(1, 'Eigenvector radiosity (1)\nIter\tError\n')
Me = Mo;
K = Mo;
I = 0;
while (norm(Me - M) > 0.001 & I < 15)
    K = K' * v1 * d1 * w1;
    Me = Me + K;
    I = I + 1;
    fprintf(1, '%d\t%f\n', I, norm(((Me - M) / M),2))
end
Me

```

### Listing A.3 – MATLAB empty rectangular room conductance matrix eigenanalysis

The annotated output of this program is presented in Listing A.4.

```

EDU> run conductance
Form factor matrix

FF =
    0    0.1249    0.1249    0.2145    0.2145    0.3213
    0.2498    0    0.0800    0.2102    0.2102    0.2498
    0.2498    0.0800    0    0.2102    0.2102    0.2498
    0.2573    0.1261    0.1261    0    0.2331    0.2573
    0.2573    0.1261    0.1261    0.2331    0    0.2573
    0.3213    0.1249    0.1249    0.2145    0.2145    0

Reflectance matrix

R =
    0.8000    0    0    0    0    0
    0    0.7000    0    0    0    0
    0    0    0.7000    0    0    0
    0    0    0    0.7000    0    0
    0    0    0    0    0.7000    0
    0    0    0    0    0    0.2000

Surface area matrix

A =
    15.0000    0    0    0    0    0

```

0	7.5000	0	0	0	0
0	0	7.5000	0	0	0
0	0	0	12.5000	0	0
0	0	0	0	12.5000	0
0	0	0	0	0	15.0000

Conductance matrix

G =

0	1.8733	1.8733	3.2168	3.2168	4.8199
1.8733	0	0.6001	1.5766	1.5766	1.8733
1.8733	0.6001	0	1.5766	1.5766	1.8733
3.2168	1.5766	1.5766	0	2.9132	3.2168
3.2168	1.5766	1.5766	2.9132	0	3.2168
4.8199	1.8733	1.8733	3.2168	3.2168	0

Eigenvalues

12.401950 -4.819853 -2.913177 -2.639829 -1.429002 -0.600088

Eigenvectors

V =

-0.4952	0.7071	-0.4700	-0.1839	0.0000	0.0000
-0.2710	0.0000	0.0303	0.6524	-0.7071	0.0000
-0.2710	0.0000	0.0303	0.6524	0.7071	0.0000
-0.4258	0.0000	0.5274	-0.2013	0.0000	-0.7071
-0.4258	0.0000	0.5274	-0.2013	0.0000	0.7071
-0.4952	-0.7071	-0.4700	-0.1839	0.0000	0.0000

Conductance matrix spectral components

G1 =

3.0415	1.6643	1.6643	2.6153	2.6153	3.0415
1.6643	0.9107	0.9107	1.4311	1.4311	1.6643
1.6643	0.9107	0.9107	1.4311	1.4311	1.6643
2.6153	1.4311	1.4311	2.2488	2.2488	2.6153
2.6153	1.4311	1.4311	2.2488	2.2488	2.6153
3.0415	1.6643	1.6643	2.6153	2.6153	3.0415

G2 =

-2.4099	0.0000	0.0000	0.0000	0.0000	2.4099
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2.4099	0.0000	0.0000	0.0000	0.0000	-2.4099

G3 =

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-1.4566	1.4566	0.0000
0.0000	0.0000	0.0000	1.4566	-1.4566	0.0000

```

0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
G4 =
-0.5832    0.0376    0.0376    0.6544    0.6544   -0.5832
 0.0376   -0.0024   -0.0024   -0.0421   -0.0421    0.0376
 0.0376   -0.0024   -0.0024   -0.0421   -0.0421    0.0376
 0.6544   -0.0421   -0.0421   -0.7342   -0.7342    0.6544
 0.6544   -0.0421   -0.0421   -0.7342   -0.7342    0.6544
-0.5832    0.0376    0.0376    0.6544    0.6544   -0.5832

G5 =
-0.0483    0.1714    0.1714   -0.0529   -0.0529   -0.0483
 0.1714   -0.6083   -0.6083    0.1877    0.1877    0.1714
 0.1714   -0.6083   -0.6083    0.1877    0.1877    0.1714
-0.0529    0.1877    0.1877   -0.0579   -0.0579   -0.0529
-0.0529    0.1877    0.1877   -0.0579   -0.0579   -0.0529
-0.0483    0.1714    0.1714   -0.0529   -0.0529   -0.0483

G6 =
 0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
 0.0000   -0.3000    0.3000    0.0000    0.0000    0.0000
 0.0000    0.3000   -0.3000    0.0000    0.0000    0.0000
 0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
 0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
 0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

```

Initial exitance vector

```

Mo =
 1
 0
 0
 0
 0
 0

```

Direct solution vector

```

M =
 1.2343
 0.3684
 0.3684
 0.3713
 0.3713
 0.1296

```

Approximate solution vectors

```

M6 =
 1.2343
 0.3684
 0.3684
 0.3713
 0.3713

```

0.1296

Error = 0.000000

M5 =

1.2343  
0.3684  
0.3684  
0.3713  
0.3713  
0.1296

Error = 0.000000

M4 =

1.2349  
0.3770  
0.3770  
0.3715  
0.3715  
0.1297

Error = 0.009898

M3 =

1.2431  
0.3695  
0.3695  
0.3484  
0.3484  
0.1322

Error = 0.027335

M2 =

1.2431  
0.3695  
0.3695  
0.3484  
0.3484  
0.1322

Error = 0.027335

M1 =

1.4321  
0.4138  
0.4138  
0.3902  
0.3902  
0.1080

Error = 0.170775



Jacobi iteration

Iter	Error
1	0.368769
2	0.219477
3	0.130254
4	0.077410
5	0.045988
6	0.027324
7	0.016234
8	0.009645
9	0.005731
10	0.003405
11	0.002023
12	0.001202
13	0.000714

Mj =

1.2339
0.3680
0.3680
0.3709
0.3709
0.1294

Eigenvector radosity (6)

Iter	Error
1	0.368769
2	0.219477
3	0.130254
4	0.077410
5	0.045988
6	0.027324
7	0.016234
8	0.009645
9	0.005731
10	0.003405
11	0.002023
12	0.001202
13	0.000714

Me =

1.2339
0.3680
0.3680
0.3709
0.3709
0.1294

Eigenvector radosity (5)

Iter	Error
1	0.368769
2	0.219477
3	0.130254
4	0.077410

5	0.045988
6	0.027324
7	0.016234
8	0.009645
9	0.005731
10	0.003405
11	0.002023
12	0.001202
13	0.000714

Me =

1.2339
0.3680
0.3680
0.3709
0.3709
0.1294

Eigenvector radiosity (4)

Iter	Error
1	0.377013
2	0.223991
3	0.132320
4	0.077172
5	0.044131
6	0.024553
7	0.013456
8	0.008236
9	0.007122
10	0.007678
11	0.008405
12	0.008951
13	0.009312
14	0.009540
15	0.009680

Me =

1.2347
0.3768
0.3768
0.3714
0.3714
0.1297

Eigenvector radiosity (3)

Iter	Error
1	0.393073
2	0.244754
3	0.155064
4	0.101044
5	0.068953
6	0.050360
7	0.039935
8	0.034259

9	0.031203
10	0.029543
11	0.028620
12	0.028095
13	0.027789
14	0.027608
15	0.027500

Me =

1.2429
0.3693
0.3693
0.3482
0.3482
0.1321

Eigenvector radiosity (2)

Iter	Error
1	0.393073
2	0.244754
3	0.155064
4	0.101044
5	0.068953
6	0.050360
7	0.039935
8	0.034259
9	0.031203
10	0.029543
11	0.028620
12	0.028095
13	0.027789
14	0.027608
15	0.027500

Me =

1.2429
0.3693
0.3693
0.3482
0.3482
0.1321

Eigenvector radiosity (1)

Iter	Error
1	0.366843
2	0.210351
3	0.137046
4	0.122279
5	0.131424
6	0.143387
7	0.152744
8	0.159189
9	0.163421
10	0.166137

```
11    0.167861
12    0.168948
13    0.169631
14    0.170060
15    0.170328
```

```
Me =
    1.4318
    0.4135
    0.4135
    0.3898
    0.3898
    0.1079
```

EDU»

Listing A.4 – MATLAB output

#### **A.4 Conclusions**

The MATLAB output presented in Listing A.4 highlights several important points that were discussed in Chapter 4:

1. The approximate solution vector M6 based on all of the eigenpairs of the conductance matrix is identical to the direct solution vector M. (This follows from the spectral decomposition theorem.)
2. The relative 2-norm error between M5 and M is zero. This indicates that the contribution of the smallest eigenvector to the solution is insignificant.
3. The approximate solution vector M2 has a relative 2-norm error of less than 3 percent. This indicates that a reasonably acceptable solution can be obtained using only two of the six eigenvectors (for this particular problem).
4. The approximate solution vector M1 has a relative 2-norm error of 17 percent. While this may not be acceptable for most engineering applications, it indicates that an order of magnitude estimate can be obtained using only the largest magnitude eigenvectors. These are the first eigenvectors to be returned by the iterative block Lanczos eigensolver.

5. The eigenvector radiosity solution utilizing all of the eigenvectors exhibits exactly the same step-by-step convergence as Jacobi iteration. (This follows from eigenvector radiosity being a reformulation of Jacobi iteration in terms of the conductance matrix eigenpairs.) Convergence to less than 0.1 percent error occurs in 13 steps.
6. The eigenvector radiosity solutions utilizing subsets of the largest magnitude eigenvectors exhibit similar convergence behavior in that they also converge to a solution in 13 steps. However, each solution has a constant relative 2-norm error with respect to the direct solution. (This follows from each subset of eigenpairs representing an approximate conductance matrix.)

While these results cannot be directly extended to more complex environments, they do indicate that eigenvector radiosity works in principle.

# Appendix B

## Eigenspectra

### B.1 Eigensystems

Given a homogeneous system of linear equations:

$$\mathbf{Ax} = \lambda \mathbf{x} \quad (\text{B.1})$$

where  $\mathbf{x} \neq 0$  and  $\lambda$  is a complex (and possibly real) scalar, then  $\mathbf{x}$  is an *eigenvector*,  $\lambda$  is an *eigenvalue*, and the *eigenpair*  $\{\mathbf{x}, \lambda\}$  is an *eigensolution* of the matrix  $\mathbf{A}$ . The set of all eigenvalues of  $\mathbf{A}$  is called the *eigenspectrum*, or simply the *spectrum*  $S(\mathbf{A})$ , of  $\mathbf{A}$ , and the set of all eigenpairs is called its *eigensystem*. Further,  $\rho(\mathbf{A}) = |\lambda_{\max}|$  is called the *spectral radius* of  $\mathbf{A}$ .

This appendix outlines various theorems concerning eigenspectra and their bounds as they relate to eigenvector radiosity.

### B.2 Gershgorin Circle Theorem

The Gershgorin Circle theorem states that the spectrum  $S(\mathbf{A})$  of the matrix  $\mathbf{A} = [a_{ij}]$  is bounded by the intersection of the union of two sets of discs  $C_i$  and  $C'_i$  (that is,  $S(\mathbf{A}) \subset C_i \cap C'_i$ ) in the complex plane, where:

$$C_i = \left\{ \left| z - a_{ii} \right| \leq \sum_{j \neq i} |a_{ij}| \right\}, \quad 1 \leq i \leq n \quad (\text{B.2})$$

and

$$C'_i = \left\{ \left| z - a_{ii} \right| \leq \sum_{j \neq i} |a_{ji}| \right\}, \quad 1 \leq i \leq n \quad (\text{B.3})$$

A proof is given by Axelsson [axel96].

For a symmetric matrix,  $a_{ij} = a_{ji}$  and so the Gershgorin Circle theorem reduces to Equation B.2. Furthermore, the eigenvalues of a symmetric matrix are real [gol96], which means that the eigenvalues of  $\mathbf{A}$  are bounded by the intervals:

$$R_i = \left\{ \left| r - a_{ii} \right| \leq \sum_{j \neq i} |a_{ij}| \right\}, \quad 1 \leq i \leq n \quad (\text{B.4})$$

where  $r$  is a real number.

Given a conductance matrix  $\mathbf{G}$  of order  $n$  where  $g_{ii} = 0$  (that is, there are no loop conductances in the associated linear resistive network), Equation B.4 gives:

$$\rho(\mathbf{G}) \leq \max \left( \sum_{j=1}^n |g_{ij}| \right) = \|\mathbf{G}\|_{\infty}, \quad 1 \leq i \leq n \quad (\text{B.5})$$

where  $\|\mathbf{G}\|_{\infty}$  is the  $p$ -norm of  $\mathbf{G}$  for  $p = \infty$ .

Recalling from Chapter 3 that  $\mathbf{G} = \mathbf{A}\mathbf{F}$  where  $\mathbf{A}$  is the diagonal matrix of surface element areas, we have:

$$\rho(\mathbf{G}) \leq \max \left( \sum_{j=1}^n |a_i f_{ij}| \right) = \max \left( a_i \sum_{j=1}^n |f_{ij}| \right), \quad 1 \leq i \leq n \quad (\text{B.6})$$

where  $a_i$  is the area of surface element  $i$ .

However, two properties of form factor matrices (see Chapter 1) are that:

$$\sum_{j=1}^n f_{ij} \leq 1 \quad (\text{B.7})$$

for any row  $i$ ; and:

$$0 \leq f_{ij} \leq 1 \quad (\text{B.8})$$

for any row  $i$  and column  $j$ . Substituting these into Equation B.6 gives:

$$\rho(\mathbf{G}) \leq \max(a_i), \quad 1 \leq i \leq n \quad (\text{B.9})$$

This result is useful in Chapter 4, where it is necessary to quickly determine the eigenvalue bound for a conductance matrix  $\tilde{\mathbf{G}} = \mathbf{A}\tilde{\mathbf{F}}$ , where  $\tilde{\mathbf{F}}$  consists of a single row and column (corresponding to a single shooting element). From Equation B.7, it is evident that the spectral radius of  $\tilde{\mathbf{G}}$  is bounded by the area of the shooting element  $a_i$ .

This assumes however that  $\sum_{j=1}^n f_{ij} = 1$ . If the summation is less than unity (that is, the environment is not closed), then Equation B.6, while more expensive to compute, provides a tighter bound.

### B.3 Courant-Fischer Lemma

Suppose we have a real symmetric matrix  $\mathbf{A}$  of order  $n$  with eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  and the corresponding set of orthonormal eigenvectors  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ .

The Courant-Fischer lemma then states that for any normalized vector  $\mathbf{x}$ :

$$\min_{\mathbf{x} \perp \mathbf{v}_1, \dots, \mathbf{v}_{i-1}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) \leq \lambda_i \quad (\text{B.10})$$

and

$$\max_{\mathbf{x} \perp \mathbf{v}_{i+1}, \dots, \mathbf{v}_n} (\mathbf{x}^T \mathbf{A} \mathbf{x}) \geq \lambda_i \quad (\text{B.11})$$

A proof is given by Axelsson [axel96].

### B.4 Eigenvalue Sensitivity

Wilkinson [wilk65] proved that given two real symmetric matrices  $\mathbf{A}$  and  $\mathbf{B}$  of order  $n$  with respective eigenvalues  $\lambda_1(\mathbf{A}) \leq \lambda_2(\mathbf{A}) \leq \dots \leq \lambda_n(\mathbf{A})$  and  $\lambda_1(\mathbf{B}) \leq \lambda_2(\mathbf{B}) \leq \dots \leq \lambda_n(\mathbf{B})$ , the eigenvalues of the matrix  $(\mathbf{A} + \mathbf{B})$  are bounded by:

$$\lambda_i(\mathbf{A}) + \lambda_{\min}(\mathbf{B}) \leq \lambda_i(\mathbf{A} + \mathbf{B}) \leq \lambda_i(\mathbf{A}) + \lambda_{\max}(\mathbf{B}), \quad 1 \leq i \leq n \quad (\text{B.12})$$



Following Axelsson (axel96), this may be proven as follows: Let  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$  denote the eigenvectors of  $\mathbf{A}$ . The Courant-Fischer lemma shows that for any normalized vector  $\mathbf{x}$ :

$$\begin{aligned} \lambda_i(\mathbf{A} + \mathbf{B}) &\geq \min_{\mathbf{x} \perp \mathbf{v}_1, \dots, \mathbf{v}_{i-1}} (\mathbf{x}^T (\mathbf{A} + \mathbf{B}) \mathbf{x}) \geq \min_{\mathbf{x} \perp \mathbf{v}_1, \dots, \mathbf{v}_{i-1}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) + \min(\mathbf{x}^T \mathbf{B} \mathbf{x}) \\ &= \lambda_i(\mathbf{A}) + \lambda_{\min}(\mathbf{B}) \end{aligned} \quad (\text{B.13})$$

which proves the lower bound. Similarly:

$$\begin{aligned} \lambda_i(\mathbf{A} + \mathbf{B}) &\leq \max_{\mathbf{x} \perp \mathbf{v}_{i+1}, \dots, \mathbf{v}_n} (\mathbf{x}^T (\mathbf{A} + \mathbf{B}) \mathbf{x}) \leq \max_{\mathbf{x} \perp \mathbf{v}_{i+1}, \dots, \mathbf{v}_n} (\mathbf{x}^T \mathbf{A} \mathbf{x}) + \max(\mathbf{x}^T \mathbf{B} \mathbf{x}) \\ &= \lambda_i(\mathbf{A}) + \lambda_{\max}(\mathbf{B}) \end{aligned} \quad (\text{B.12})$$

which proves the upper bound.

This result is useful in Chapter 4, where it is necessary to determine whether a conductance matrix  $\tilde{\mathbf{G}}$  (consisting of a single row and column) will significantly perturb the eigenspectrum of an existing conductance matrix  $\mathbf{G}$ . If  $|\lambda_{\max}(\tilde{\mathbf{G}})| \ll |\lambda_{\max}(\mathbf{G})|$ , then the eigenspectra of  $\mathbf{G}$  and  $(\mathbf{G} + \tilde{\mathbf{G}})$  will be similar. More important, an eigenvector radiosity solution based on  $\mathbf{G}$  will be similar to that based on  $(\mathbf{G} + \tilde{\mathbf{G}})$ .

# Appendix C

## Radiometric and Photometric Units

### C.1 Introduction

The radiometric and photometric definitions employed in this thesis and detailed in this appendix are those commonly used in illumination engineering. They are in accordance with the American National Standards Institute publication ANSI/IES RP-16-96, “Nomenclature and Definitions for Illuminating Engineering” [ansi96].

### C.2 Radiometry

Radiometry is the science of measuring light in any portion of the electromagnetic spectrum. In practice, the term is usually limited to the measurement of infrared, visible, and ultraviolet light using optical instruments.

#### C.2.1 Radiant Energy

Light is *radiant energy*. Electromagnetic radiation transports energy through space. When light is absorbed by a physical object, its energy is converted into some other form. A microwave oven, for example, heats a glass of water when its microwave radiation is absorbed by the water molecules. The radiant energy of the microwaves is converted into thermal energy (heat). Similarly, visible light causes an electric current to flow in a photographic light meter when its radiant energy is transferred to the electrons as kinetic energy.

Radiant energy (denoted as  $Q$ ) is measured in *joules*.

### C.2.1.1 Spectral Radiant Energy

A broadband source such as the Sun emits electromagnetic radiation throughout most of the electromagnetic spectrum, from radio waves to gamma rays. However, most of its radiant energy is concentrated within the visible portion of the spectrum. A single-wavelength laser, on the other hand, is a monochromatic source; all of its radiant energy is emitted at one specific wavelength.

From this, we can define *spectral radiant energy*, which is the amount of radiant energy per unit wavelength interval at wavelength  $\lambda$ . It is defined as:

$$Q_\lambda = dQ/d\lambda \quad (\text{C.1})$$

Spectral radiant energy is measured in joules per nanometer.

### C.2.2 Radiant Flux (Radiant Power)

Energy per unit time is power, which we measure in joules per second, or watts. A laser beam, for example, has so many milliwatts or watts of *radiant power*. Light “flows” through space, and so radiant power is more commonly referred to as the “time rate of flow of radiant energy,” or *radiant flux*. It is defined as:

$$\Phi = dQ/dt \quad (\text{C.2})$$

where  $Q$  is radiant energy and  $t$  is time.

In terms of a photographic light meter measuring visible light, the instantaneous magnitude of the electric current is directly proportional to the radiant flux. The total amount of current measured over a period of time is directly proportional to the radiant energy absorbed by the light meter during that time. This is how a photographic flash

meter works – it measures the total amount of radiant energy received from a camera flash.

The flow of light through space is often represented by geometrical rays of light such as those used in computer graphics ray tracing. They can be thought of as infinitesimally thin lines drawn through space that indicate the direction of flow of radiant energy (light). They are also mathematical abstractions -- even the thinnest laser beam has a finite cross section. Nevertheless, they provide a useful aid to understanding radiometric theory.

Radiant flux is measured in watts.

#### **C.2.2.1 Spectral Radiant Flux (Spectral Radiant Power)**

*Spectral radiant flux* is radiant flux per unit wavelength interval at wavelength  $\lambda$ . It is defined as:

$$\Phi_{\lambda} = d\Phi/d\lambda \quad (\text{C.3})$$

and is measured in watts per nanometer.

#### **C.2.3 Radiant Flux Density (Irradiance and Radiant Exitance)**

*Radiant flux density* is the radiant flux per unit area at a point on a surface, where the surface can be real or imaginary (i.e., a mathematical plane). There are two possible conditions. The flux can be arriving at the surface (Figure C.1a), in which case the radiant flux density is referred to as *irradiance*. The flux can arrive from any direction above the surface, as indicated by the rays. Irradiance is defined as:

$$E = d\Phi/dA \quad (\text{C.4})$$

where  $\Phi$  is the radiant flux arriving at the point and  $dA$  is the differential area surrounding the point.

The flux can also be leaving the surface due to emission and/or reflection (Figure C.1b). The radiant flux density is then referred to as *radiant exitance*. As with irradiance, the flux can leave in any direction above the surface. The definition of radiant exitance is:

$$M = d\Phi/dA \quad (C.5)$$

where  $\Phi$  is the radiant flux leaving the point and  $dA$  is the differential area surrounding the point.

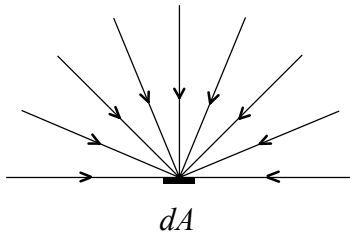


Figure C.1a – Irradiance.

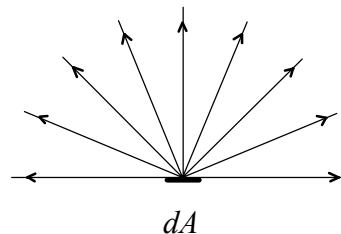


Figure C.1b – Radiant exitance.

The importance of a “real or imaginary” surface cannot be overstated. It means that radiant flux density can be measured anywhere in three-dimensional space. This includes on the surface of physical objects, in the space between them (e.g., in air or a vacuum), and inside transparent media such as water and glass.

Radiant flux density is measured in watts per square meter.

### C.2.3.1 Spectral Radiant Flux Density

*Spectral radiant flux density* is radiant flux per unit wavelength interval at wavelength  $\lambda$ .

When the radiant flux is arriving at the surface, it is called *spectral irradiance*, and is defined as:

$$E_\lambda = dE/d\lambda \quad (C.6)$$

When the radiant flux is leaving the surface, it is called *spectral radiant exitance*, and is defined as:

$$M_\lambda = dM/d\lambda \quad (\text{C.7})$$

Spectral radiant flux density is measured in watts per square meter per nanometer.

#### C.2.4 Radiance

*Radiance* is best understood by first visualizing it. Imagine ray of light arriving at or leaving a point on a surface in a given direction. Radiance is simply the infinitesimal amount of radiant flux contained in this ray. Period.

A more formal definition of radiance requires that we think of a ray as being an infinitesimally narrow (“elemental”) cone with its apex at a point on a real or imaginary surface. This cone has a differential solid angle  $d\omega$  that is measured in steradians.

We must also note that the ray is intersecting the surface at an angle. If the area of intersection with the surface has a differential cross-sectional area  $dA$ , the cross-sectional area of the ray is  $dA \cos\theta$ , where  $\theta$  is the angle between the ray and the surface normal, as shown in Figure C.2. (The ray cross-sectional area  $dA \cos\theta$  is called the *projected area* of the ray-surface intersection area  $dA$ . The same term is used when referring to finite areas  $\Delta A$ .)

With these preliminaries in mind, we can imagine an elemental cone  $d\omega$  containing a ray of light that is arriving at or leaving a surface (Figures C.3a and C.3b). The definition of radiance is then:

$$L = d^2\Phi/[dA(d\omega \cos\theta)] \quad (\text{C.8})$$

where  $\Phi$  is the radiant flux,  $dA$  is the differential area surrounding the point,  $d\omega$  is the differential solid angle of the elemental cone, and  $\theta$  is the angle between the ray and the surface normal  $\mathbf{n}$  at the point.

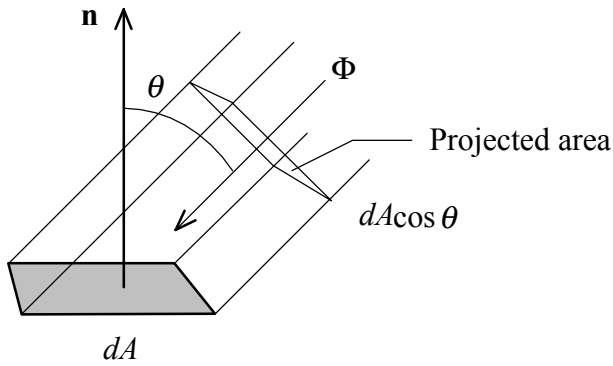


Figure C.2 – A ray of light intersecting a surface.

Unlike radiant flux density, the definition of radiance does not distinguish between flux arriving at or leaving a surface. In fact, the formal definition of radiance [ansi96] states that it can be “leaving, passing through or arriving at” the surface.

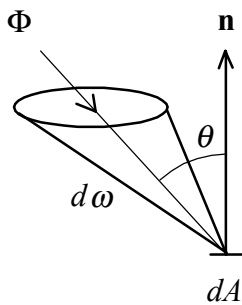


Figure C.3a – Radiance (arriving).

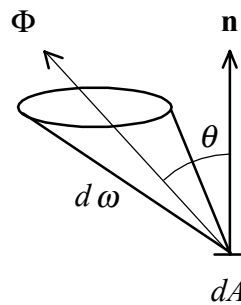


Figure C.3b – Radiance (leaving).

Another way of looking a radiance is to note that the radiant flux density at a point on a surface due to a single ray of light arriving (or leaving) at an angle  $\theta$  to the surface normal is  $d\Phi/(dA \cos \theta)$ . The radiance at that point for the same angle is then  $d^2\Phi/[dA(d\omega \cos \theta)]$ , or radiant flux density per unit solid angle.

Radiance is measured in watts per square meter per steradian.

### C.2.4.1 Spectral Radiance

*Spectral radiance* is radiance per unit wavelength interval at wavelength  $\lambda$ . It is defined as:

$$L_\lambda = d^3\Phi/[dA(d\omega\cos\theta)d\lambda] \quad (\text{C.9})$$

and is measured in watts per square meter per steradian per nanometer.

### C.2.5 Radiant Intensity

We can imagine an infinitesimally small *point source* of light that emits radiant flux in every direction. The amount of radiant flux emitted in a given direction can be represented by a ray of light contained in an elemental cone. This gives us the definition of radiant intensity:

$$I = d\Phi/d\omega \quad (\text{C.10})$$

where  $d\omega$  is the differential solid angle of the elemental cone containing the given direction. From the definition of a differential solid angle ( $d\omega = dA/r^2$ ), we get:

$$E = d\Phi/dA = d\Phi/r^2 d\omega = I/r^2 \quad (\text{C.11})$$

where the differential surface area  $dA$  is on the surface of a sphere centered on and at a distance  $r$  from the source and  $E$  is the irradiance of that surface. More generally, the radiant flux will intercept  $dA$  at an angle  $\theta$  (Figure C.4). This gives us the *inverse square law* for point sources:

$$E = I \cos\theta/d^2 \quad (\text{C.12})$$

where  $I$  is the intensity of the source in the given direction and  $d$  is the distance from the source to the surface element  $dA$ .



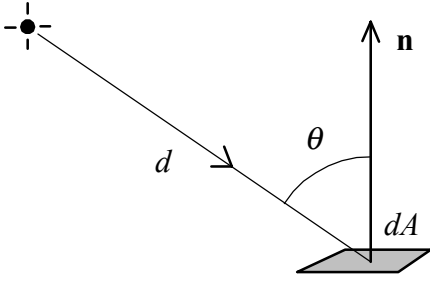


Figure C.4 – Inverse square law for point sources.

We can further imagine a real or imaginary surface as being a continuum of point sources, where each source occupies a differential area  $dA$  (Figure C.5). Viewed at an angle  $\theta$  from the surface normal  $\mathbf{n}$ , the source has a projected area of  $dA \cos\theta$ . Combining the definitions of radiance (Equation C.8) and radiant intensity (Equation C.10) gives us an alternative definition of radiance:

$$L = dI / (dA \cos\theta) \quad (\text{C.13})$$

where  $dI$  is the differential intensity of the point source in the given direction.

Radiant intensity is measured in watts per steradian.

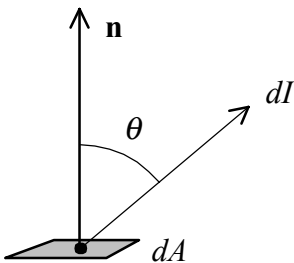


Figure C.5 – Radiance of a point source.

### C.2.5.1 Spectral Radiant Intensity

*Spectral radiant intensity* is radiant intensity per unit wavelength interval at wavelength

$\lambda$ . It is defined as:

$$I_\lambda = dI / d\lambda \quad (\text{C.14})$$

and is measured in watts per steradian per nanometer.

### **C.3 Photometry**

Photometry is the science of measuring visible light in units that are weighted according to the sensitivity of the human eye. It is a quantitative science based on a statistical model of the human visual response to light – that is, our *perception* of light – under carefully controlled conditions.

The human visual system is a marvelously complex and highly nonlinear detector of electromagnetic radiation with wavelengths ranging from 380 to 770 nanometers (nm). We see light of different wavelengths as a continuum of colors ranging through the visible spectrum: 650 nm is red, 540 nm is green, 450 nm is blue, and so on.

The sensitivity of the human eye to light varies with wavelength. A light source with a radiance of one watt/m<sup>2</sup>-steradian of green light, for example, appears much brighter than the same source with a radiance of one watt/m<sup>2</sup>-steradian of red or blue light. In photometry, we do not measure watts of radiant energy. Rather, we attempt to measure the subjective impression produced by stimulating the human eye-brain visual system with radiant energy.

This task is complicated immensely by the eye's nonlinear response to light. It varies not only with wavelength but also with the amount of radiant flux, whether the light is constant or flickering, the spatial complexity of the scene being perceived, the adaptation of the iris and retina, the psychological and physiological state of the observer, and a host of other variables.

Nevertheless, the subjective impression of seeing can be quantified for “normal” viewing conditions. In 1924, the *Commission Internationale d’Eclairage* (International Commission on Illumination, or CIE) asked over one hundred observers to visually match the “brightness” of monochromatic light sources with different wavelengths under controlled conditions. The statistical result – the so-called CIE photometric curve shown in Figure C.6 – shows the *photopic luminous efficiency* of the human visual system as a function of wavelength. It provides a weighting function that can be used to convert radiometric into photometric measurements.

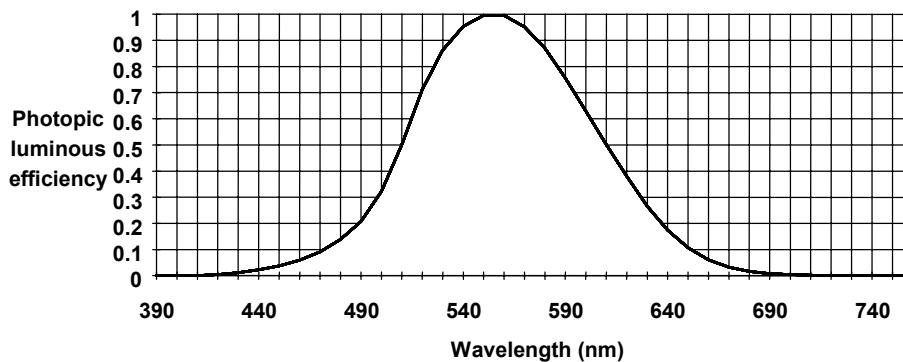


Figure C.6 – CIE photometric curve.

Photometric theory does not address how we perceive colors. The light being measured can be monochromatic or a combination or continuum of wavelengths; the eye’s response is determined by the CIE weighting function. This underlines a crucial point: *The only difference between radiometric and photometric theory is in their units of measurement.*

### C.3.1 Luminous Intensity

The foundations of photometry were laid in 1729 by Pierre Bouguer. In his *L’Essai d’Optique* [boug29], Bouguer discussed photometric principles in terms of the convenient

light source of his time: a wax candle. This became the basis of the point source concept in photometric theory.

Wax candles were used as national light source standards in the 18th and 19th centuries. England, for example, used spermaceti (a wax derived from sperm whale oil). These were replaced in 1909 by an international standard based on a group of carbon filament vacuum lamps and again in 1948 by a crucible containing liquid platinum at its freezing point. Today the international standard is a theoretical point source that has a *luminous intensity* of one *candela* (the Latin word for “candle”). It emits monochromatic radiation with a frequency of  $540 \times 10^{12}$  Hertz (or approximately 555 nm, corresponding with the wavelength of maximum photopic luminous efficiency) and has a radiant intensity (in the direction of measurement) of 1/683 watts per steradian [e.g., iesn00].

Together with the CIE photometric curve, the candela provides the weighting factor needed to convert between radiometric and photometric measurements. Consider, for example, a monochromatic point source with a wavelength of 510 nm and a radiant intensity of 1/683 watts per steradian. The photopic luminous efficiency at 510 nm is 0.503. The source therefore has a luminous intensity of 0.503 candela.

### **C.3.2 Luminous Flux (Luminous Power)**

*Luminous flux* is photometrically weighted radiant flux (power). Its unit of measurement is the *lumen*, defined as 1/683 watts of radiant power at a frequency of  $540 \times 10^{12}$  Hertz. As with luminous intensity, the luminous flux of light with other wavelengths can be calculated using the CIE photometric curve.

A point source having a uniform (isotropic) luminous intensity of one candela in all directions (i.e., a uniform *intensity distribution*) emits one lumen of luminous flux per unit solid angle (steradian).

### **C.3.3 Luminous Energy**

*Luminous energy* is photometrically weighted radiant energy. It is measured in lumen seconds.

### **C.3.4 Luminous Flux Density (Illuminance and Luminous Exitance)**

*Luminous flux density* is photometrically weighted radiant flux density. *Illuminance* is the photometric equivalent of irradiance, whereas *luminous exitance* is the photometric equivalent of radiant exitance.

Luminous flux density is measured in lumens per square meter. (A *footcandle* is one lumen per square foot.)

### **C.3.5 Luminance**

*Luminance* is photometrically weighted radiance. In terms of visual perception, we perceive luminance. It is an approximate measure of how “bright” a surface appears when we view it from a given direction. Luminance used to be called “photometric brightness.” This term is no longer used in illumination engineering because the subjective sensation of visual brightness is influenced by many other physical, physiological, and psychological factors.

Luminance is measured in lumens per square meter per steradian.

#### C.4 Lambertian Surfaces

A Lambertian surface is a surface that has a constant radiance or luminance that is independent of the viewing direction. In accordance with the definition of radiance (luminance), the radiant (luminous) flux may be emitted, transmitted, and/or reflected by the surface.

A Lambertian surface is also referred to as an *ideal diffuse* emitter or reflector. In practice there are no true Lambertian surfaces. Most matte surfaces approximate an ideal diffuse reflector but typically exhibit semispecular reflection characteristics at oblique viewing angles. Nevertheless, the Lambertian surface concept is useful in computer graphics and radiosity theory.

Lambertian surfaces are unique in that they reflect incident flux in a completely diffuse manner (Figure C.7). It does not matter what the angle of incidence  $\theta$  of an incoming geometrical ray is – the distribution of light leaving the surface remains unchanged.

We can imagine a differential area  $dA$  of a Lambertian surface. Being infinitesimally small, it is equivalent to a point source, and so the flux leaving the surface can be modeled as geometrical rays. The intensity  $I_\theta$  of each ray leaving the surface at an angle  $\theta$  from the surface normal is given by *Lambert's cosine law*:

$$I_\theta = I_n \cos\theta \tag{C.15}$$

where  $I_n$  is the intensity of the ray leaving in a direction perpendicular to the surface.

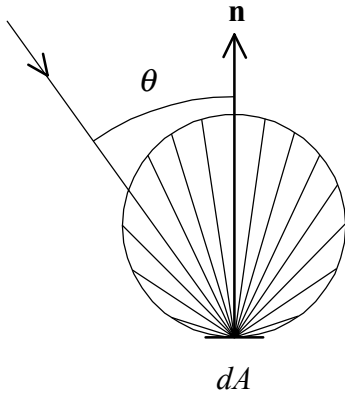


Figure C.7 – Reflection from a Lambertian surface.

The derivation of Equation C.15 becomes clear when we remember that we are viewing  $dA$  from an angle  $\theta$ . For a differential area  $dA$  with a constant radiance or luminance, its intensity must vary in accordance with its projected area, which is  $dA \cos \theta$ . This gives us:

$$L = dI / (dA \cos \theta) = dI_n / dA \quad (\text{C.16})$$

for any Lambertian surface.

There is a very simple relation between radiant (luminous) exitance and radiance (luminance) for flux leaving a Lambertian surface:

$$M = \pi L \quad (\text{C.17})$$

where the factor of  $\pi$  is a source of endless confusion to students of radiometry and photometry. Fortunately, there is an intuitive explanation. Suppose we place a differential Lambertian emitter  $dA$  on the inside surface of an imaginary sphere  $S$  (Figure C.8). The inverse square law (Equation C.12) provides the irradiance  $E$  at any point  $P$  on the inside surface of the sphere. However,  $d = D \cos \theta$ , where  $D$  is the diameter of the sphere. Thus:

$$E = I_\theta \cos \theta / (D \cos \theta)^2 = I_\theta / D^2 \cos \theta \quad (\text{C.18})$$

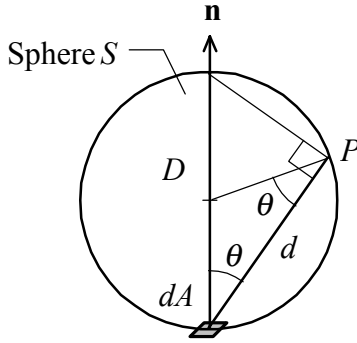


Figure C.8 – A Lambertian emitter illuminating the interior of a sphere.

and from Lambert's cosine law (Equation C.15), we have:

$$E = I_{\theta} \cos\theta / D^2 \cos\theta = I_n / D^2 \quad (\text{C.19})$$

which simply says that the irradiance (radiant flux density) of any point  $P$  on the inside surface of  $S$  is a constant.

From the definition of irradiance (Equation C.4), we know that  $\Phi = EA$  for constant flux density across a finite surface area  $A$ . Since the area  $A$  of the surface of a sphere with radius  $r$  is given by:

$$A = 4\pi r^2 = \pi D^2 \quad (\text{C.20})$$

we have:

$$\Phi = EA = \pi I_n D^2 / D^2 = \pi I_n \quad (\text{C.21})$$

Given the definition of radiant exitance (Equation C.5) and radiance for a Lambertian surface (Equation C.16), we have:

$$M = d\Phi / dA = \pi dI_n / dA = \pi L \quad (\text{C.22})$$

This explains, clearly and without resorting to integral calculus, where the factor of  $\pi$  comes from.