

A PERCEPTUALLY BASED ADAPTIVE
SAMPLING ALGORITHM FOR
REALISTIC IMAGE
SYNTHESIS

by

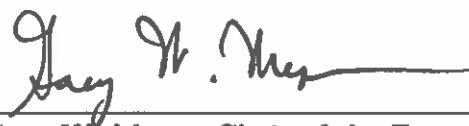
MARK ROBERT BOLIN

A DISSERTATION

Presented to the Department of Computer
and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

March 1999

"A Perceptually Based Adaptive Sampling Algorithm for Realistic Image Synthesis," a dissertation prepared by Mark Robert Bolin in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer and Information Science. This dissertation has been approved and accepted by:



Dr. Gary W. Meyer, Chair of the Examining Committee

4/17/99

Date

Committee in charge: Dr. Gary W. Meyer, Chair
 Dr. Richard M. Koch
 Dr. Amr A. Sabry
 Dr. Kent A. Stevens

Accepted by:

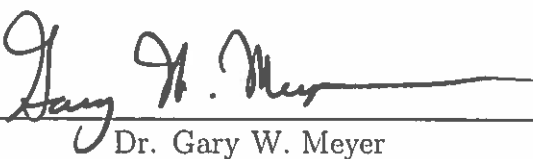


Dean of the Graduate School

© 1999 Mark Robert Bolin

An Abstract of the Dissertation of
Mark Robert Bolin for the degree of Doctor of Philosophy
in the Department of Computer and Information Science
to be taken March 1999

Title: A PERCEPTUALLY BASED ADAPTIVE SAMPLING
ALGORITHM FOR REALISTIC IMAGE SYNTHESIS

Approved: 
Dr. Gary W. Meyer

The developers of previous computer graphic image rendering algorithms have not accounted for the fact that synthesized pictures are ultimately intended to be viewed by a human observer. This has led to the creation of algorithms that can waste large amounts of time refining areas of an image that are already visually acceptable, while neglecting regions of an image containing perceptible artifacts. The human visual system has a varying acuity for error that depends on the context in which the error is viewed. Exploiting the subjective nature of perceptual response holds the key to improving the performance of image synthesis algorithms.

This dissertation presents a new perceptually based adaptive sampling algorithm. This algorithm makes subjective quality assessments during the progression of a rendering algorithm. These assessments are used to focus the effort of the rendering algorithm on the regions of the image containing the most perceptible artifacts. In this manner, images of a given visual quality are produced faster than is possible with existing image synthesis techniques. The new algorithm also allows the user to select a perceptual quality for the output image. This feature eliminates

the guesswork involved in halting a rendering and allows the production of visually consistent results.

This work includes a number of important and novel contributions. The first is the development of a new and comprehensive error metric for Monte Carlo ray tracing. This metric is used to characterize and control the objective accuracy of a rendered image. This metric is additionally employed to determine the optimum number of rays to spawn from each surface intersection. The next major contribution is the design of a new high speed, color visual difference predictor. This predictor is capable of rapidly assessing the perceptual impact of objective differences between two color images. The third contribution is the design of a frequency based adaptive sampling algorithm. This algorithm synthesizes images directly into the frequency domain. This permits the use of a simple, frequency dependent image quality metric to control the placement of samples. The final and most important contribution is the development of a second generation, perceptually based adaptive sampling algorithm. This algorithm employs the high speed, color visual difference predictor to control image sampling in accordance with the perceptibility of error in the reconstructed image. This technique is shown to improve the performance of a realistic image synthesis algorithm.

CURRICULUM VITA

NAME OF AUTHOR: Mark Robert Bolin

PLACE OF BIRTH: Pomona, California

DATE OF BIRTH: July 14, 1966

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon
University of California, San Diego

DEGREES AWARDED:

Doctor of Philosophy, in Computer and Information Science, 1999,
University of Oregon

Master of Science, in Computer and Information Science, 1995,
University of Oregon

Bachelor of Science, in Computer Engineering, 1989, University of
California, San Diego

AREAS OF SPECIAL INTEREST:

Computer Graphics
Human Vision
Image Processing

PROFESSIONAL EXPERIENCE:

Research and Teaching Assistant, Department of Computer and Information Science, University of Oregon, Eugene, Oregon, 1993-1999

Summer Intern, Xerox Webster Research Center, Rochester, New York, 1995

Engineer, IBM Storage Systems Product Division, Tucson, Arizona, 1989-1993

PUBLICATIONS:

Bolin, M. R., "Application of Image Compression Algorithms to Realistic Image Synthesis," *Technical Report CIS-TR-95-19, University of Oregon*, June 1995.

Bolin, M. R. and Meyer G. W., "A Frequency Based Ray Tracer," *Computer Graphics, Annual Conference Series, ACM SIGGRAPH*, pp. 409-418, 1995.

Bolin, M. R. and Meyer G. W., "An Error Metric for Monte Carlo Ray Tracing," *Rendering Techniques '97*, J. Dorsey and P. Slusallek, Editors, Springer-Verlag, New York, pp. 57-68, 1997.

Bolin, M. R. and Meyer G. W., "A Perceptually Based Adaptive Sampling Algorithm," *Computer Graphics, Annual Conference Series, ACM SIGGRAPH*, pp. 299-309, 1998.

ACKNOWLEDGEMENTS

The author expresses sincere appreciation to his advisor and committee chair, Professor Gary Meyer, who provided years of invaluable guidance and support. The author also acknowledges the time, effort, and insightful advice provided by Professors Richard Koch, Amr Sabry, and Kent Stevens, who comprised his doctoral committee. In addition, special thanks are due Jae Kim and Harold Westlund who assisted in the preparation of prior papers and presentations leading to this dissertation. The research was supported in part by National Science Foundation grants CCR 90-08445 and CCR 96-19967, and by generous equipment donations from Hewlett-Packard.

DEDICATION

To my loving wife Dawn, who made this all possible.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
II. THE RADIANCE EQUATION	9
The Transport Equation.....	10
The Integral Equation	12
Radiometry.....	13
The Full Radiance Equation	15
Summary	17
III. GLOBAL ILLUMINATION ALGORITHMS	18
Ray Tracing	18
Radiosity.....	24
Global Illumination as a Foundation for Perceptually Based Rendering	25
Summary	28
IV. ADAPTIVE SAMPLING ALGORITHMS	29
Introduction to Sampling and Reconstruction	30
Base Sampling Patterns	35
Refinement Tests	37
Reconstruction Techniques.....	44
Summary	48
V. AN ERROR METRIC FOR MONTE CARLO RAY TRACING	50
Previous Work	51
Error of Monte Carlo Simulations.....	52
Optimum Splitting and Russian Roulette	58
Terminations	60
Implementation	62
Results	64
Summary	68

	Page
VI. PRIOR PERCEPTUAL IMAGE QUALITY MODELS	70
Contrast Models	71
Fourier CSF Models	75
Spatial Frequency Channel Models.....	77
Orientation Channel Models	80
Masking Models.....	82
Distance Summation	84
Color Models	86
Summary	90
VII. A HIGH SPEED COLOR VISUAL DIFFERENCE PREDICTOR ...	92
Model Description	93
Results	108
Summary	114
VIII. PREVIOUS APPLICATIONS OF PERCEPTION IN COMPUTER GRAPHICS.....	116
Image Quality Models and Applications	117
Tone Reproduction Operators	119
Perceptually Based Sampling Techniques	123
Summary	124
IX. A FREQUENCY BASED RAY TRACER	126
A Frequency Dependent Perceptual Quality Model.....	128
The JPEG Compression Algorithm	134
Synthesis Into the Frequency Domain	141
Results	153
Summary	158
X. A PERCEPTUALLY BASED ADAPTIVE SAMPLING ALGORITHM	161
Basic Perceptual Algorithm	163
Enhanced Perceptual Algorithm	176
Results of the Basic Perceptual Algorithm	180
Comparison with Existing Sampling Algorithms	186
Summary	216

	Page
XI. CONCLUSION	219
APPENDIX	
A. DERIVATION OF THE TRANSPORT, INTEGRAL, AND RADIANCE EQUATIONS	226
Derivation of the Transport Equation	226
Derivation of the Integral Equation	231
Derivation of the Full Radiance Equation	235
B. PROOF OF OPTIMUM SPAWNING FORMULA	240
BIBLIOGRAPHY	242

LIST OF FIGURES

Figure	Page
1. The Five Categories of Gains and Losses	11
2. The Geometry of the Radiance Equation in a Vacuum	16
3. Pseudocode for a Generalized Adaptive Sampling Algorithm	33
4. Low and High Frequency Mountains with Optimal Sampling Rates . . .	65
5. Monte Carlo Rendered Radiosity Box with Optimal Sampling Rates . .	67
6. Contrast Detection Thresholds as a Function of Background Luminance	72
7. The Achromatic Contrast Sensitivity Function	76
8. The Sensitivity of Six Spatial Frequency Selective Channels	79
9. Threshold Elevation Due to Masking	83
10. Achromatic Versus Chromatic Spatial Frequency Response	88
11. Input and Output of the Visual Difference Predictor	94
12. Block Diagram of the Visual Difference Predictor	95
13. The <i>Cone Fundamentals</i> Stage of the Visual Difference Predictor	96
14. The <i>Cortex Filtering</i> Stage of the Visual Difference Predictor	99
15. The <i>Chromatic Aberration</i> Stage of the Visual Difference Predictor . . .	100
16. The <i>Local Contrast</i> Stage of the Visual Difference Predictor	101
17. The <i>Opponents Contrast</i> Stage of the Visual Difference Predictor	102
18. The <i>CSF Filtering</i> Stage of the Visual Difference Predictor	104
19. The <i>Masking Transducer</i> Stage of the Visual Difference Predictor	105

	Page
20. The <i>Spatial Pooling</i> Stage of the Visual Difference Predictor	106
21. The <i>Distance Summation</i> Stage of the Visual Difference Predictor	107
22. The Final Visual Difference Map	108
23. Contrast Sensitivity Illusions	109
24. Contrast Sensitivity Illusions with Noise	112
25. Sarnoff VDM and New Vision Model Visual Difference Predictions	113
26. Spectral Sensitivities for S , M , and L Components of SML Space	129
27. Center and Surrounding Squares of Equal Luminance Difference	130
28. Spectral Sensitivities for A , C_1 , and C_2 Components of AC_1C_2 Space . .	131
29. Achromatic Versus Chromatic Spatial Frequency Response	132
30. Spatial Acuity Wedges	133
31. The Basis Functions of the Discrete Cosine Transform	135
32. Magnitude of Low Frequency Components in the DCT Representation	137
33. A Sample Quantization Table	138
34. Visibility of Noise When Images are Quantized to 4 Bits Per Channel .	139
35. The Zig-Zag Sequence	140
36. Ordering Metric Used to Drive the Selection of Samples	142
37. The Initial 14 Positions Given by the Quad-Tree Sampling Method . . .	144
38. Rate at Which the Number of Rays Spawned Decreases	150
39. Sampling Density for the Contrast Squares in Figure 27	153
40. Sampling Density for the Acuity Wedges in Figure 30	154
41. Images Rendered by Spawning 1000 Rays and 100 Rays	155
42. Images Rendered Using Traditional and Frequency Based Ray Tracer .	156

	Page
43. Block Diagram of the Basic Adaptive Sampling Algorithm	164
44. Illustration of the <i>Refine Cortex Representation</i> Stage	166
45. Illustration of the <i>Construct Boundary Images</i> Stage	168
46. The Stages Employed in the <i>Local Visual Difference Prediction</i>	169
47. An Example of the <i>Update Maximum Error Tree</i> Stage	170
48. An Example of the <i>Determine Next Sample Location</i> Stage	171
49. An Example Image of a Natural Scene	173
50. A Plot of the Two-Dimensional Frequency Spectrum	174
51. RGB Pixel Intensities	175
52. Sampling Densities for Direct Light Source Sampling	181
53. Sampling Density for Monte Carlo Light Source Sampling	183
54. Sampling Densities for Direct and Monte Carlo Light Source Sampling	184
55. Image Rendered at a Visual Tolerance of 7, 5, and 3	185
56. High Quality Image for the Spatial Frequency Example	190
57. Comparison of Sampling for the Spatial Frequency Example	191
58. Sampling Rates for the Spatial Frequency Example	193
59. High Quality Image for the Contrast Example	194
60. Comparison of Sampling for the Contrast Example	195
61. Sampling Rates for the Contrast Example	197
62. High Quality Image for the Chromatic Spatial Frequency Example	198
63. Comparison of Sampling for the Chromatic Spatial Frequency Example	200
64. Sampling Rates for the Chromatic Spatial Frequency Example	202

	Page
65. High Quality Image for the Masking Example	203
66. Comparison of Sampling for the Masking Example	204
67. Sampling Rates for the Masking Example	206
68. Timing Tests for the Spatial Frequency Example	208
69. Timing Tests for the Contrast Example	209
70. Timing Tests for the Chromatic Spatial Frequency Example	210
71. Timing Tests for the Masking Example	211
72. A Summary of the Timing Test Results	216
73. The Visible Surface Function	233
74. The Geometry of the Outgoing Form of the Radiance Equation	239

CHAPTER I

INTRODUCTION

Computer graphics is a budding field within the discipline of computer science. This area of study is just beginning to reach its potential with the advent of modern computers and advanced rendering techniques. These two developments have coupled to allow the synthesis of images containing levels of detail and realism that would have been unimaginable only a few years ago. This realism is a major factor fueling the growth of the computer graphics industry. It has allowed computer graphics to expand into such fields as motion picture entertainment, architectural design, and other areas where the realistic visualization of synthetic scenes is of significant importance.

A recent advancement in rendering technology is the development of photo-realistic rendering techniques. These techniques are used to produce images that portray physically correct views of synthetically defined scenes. The images generated by these methods are more than simple, pretty pictures. Rather, they are the results of an elaborate simulation of the physics of light as it interacts with objects in the environment. These simulations can be employed to create stunning images that are virtually indistinguishable from photographs taken of real environments.

A fundamental formula known as the radiance equation is the foundation of this work [76]. This equation defines a recursive, physically based expression of

how light propagates and scatters within a three-dimensional environment. This expression describes the theoretical relationship that must be solved to produce realistic images of a synthetic scene.

The rendering methods that are based on this equation are termed global illumination algorithms. These algorithms produce images by solving the radiance equation at the surfaces of a scene and projecting the result onto an image plane. This yields a physically correct representation of the amount of light reflected from the scene along a given viewing direction.

Ray tracing is currently the most powerful and widely used global illumination algorithm. This method employs a point sampling technique to solve the radiance equation. This is the only technique currently available that is capable of solving the full generality of the radiance equation. Unfortunately, the applicability of the ray tracing method is currently limited due to the expense of this technique. Improving the performance of this algorithm is therefore key to the broad acceptance of this most powerful, photo-realistic rendering technique. The increased practicality and acceptance of this technique would lead to more realistic renderings, which would in turn foster the continued growth of the computer graphics field.

Adaptive sampling is a powerful technique for reducing the expense of ray traced renderings. In ray tracing, images are produced by sampling the light intensity at numerous locations of the image. Each sample can be very expensive to compute, involving numerous intersection operations, realistic shading calculations, motion translations, and deformations. Adaptive sampling seeks to improve the efficiency of the ray tracing algorithm by minimizing the number of samples that are necessary to accurately reconstruct an image.

In a given rendered image there are typically areas of both high and low spatial frequency variations, as well as areas that produce more or less noisy samples. Areas of high variation require a large number of samples, whereas areas of low variation can be accurately reconstructed with relatively few samples. Adaptive sampling seeks to exploit this by taking the minimum number of samples required to accurately reconstruct each region.

The key element that makes this operation possible is the use of an error metric. Adaptive sampling techniques employ an error metric to make iterative quality predictions during the progression of the rendering algorithm. These predictions are used to determine the appropriate sampling rate in each region of the image.

Previous adaptive algorithms use a number of ad hoc and incomplete error metrics to determine the accuracy of the reconstructed image. This leads to inefficiencies in the placement of samples by these algorithms. The first major development contained in this dissertation defines a new and comprehensive error metric for Monte Carlo ray tracing. This metric can be used to accurately characterize and control the presence of error within a synthetically generated image. The new error metric shows how the quality of an image depends on both the number of samples taken at the image plane and the number of samples spawned from each surface intersection. This work additionally discusses how this error metric is used to derive the optimal number of rays to spawn from each surface intersection. The use of these optimal spawning rates can improve the efficiency of Monte Carlo ray tracing.

Understanding how to characterize and control the accuracy of a synthesized image is an important aspect of developing an efficient adaptive sampling algorithm. However, it is just a starting point for the primary focus of this doctoral work. The

problem with existing adaptive sampling algorithms is that they are based on objective assessments of the accuracy of an image. These objective error measurements are typically performed in RGB energy space. Measuring error in this space neglects the fact that the images produced are ultimately intended to be viewed by a human observer. The human visual system has a widely varying sensitivity to error that depends on the context in which the error is viewed. It is ultimately the perceptibility of this error and not its absolute magnitude that should be used as the basis of adaptive sampling algorithms.

A number of perceptual quality models have been developed by the image processing community. These models are imbedded in what are known as visual difference predictors. A visual difference predictor provides a means to determine the visibility of differences between two input images. This type of algorithm can be used to make the kind of subjective quality assessments that should be employed to drive an adaptive sampling algorithm.

The latest visual difference predictors model a number of significant aspects of the human visual system. These include spatial frequency and orientation selective channels, nonlinear contrast response, variations in contrast sensitivity with spatial frequency, and masking. Each of these components has a pronounced affect on the perceptibility of error.

The developers of previous image quality metrics have gone to great lengths to accurately model the sensitivity of the human visual system. Unfortunately, efficiency is seldom a design criteria in creating these metrics. This limits the utility of the metrics for applications such as adaptive sampling where speed is a primary concern. In addition, the majority of previous visual difference predictors are only

designed to process achromatic images. The models that do include color processing neglect the significant effect of chromatic aberration.

The second major contribution of this dissertation is the design of a high speed, color visual difference predictor. This predictor is based, in part, on a model by Lubin [88]. The new predictor is designed to run efficiently and extended to handle color, including the effect of chromatic aberration. This new color image quality model is shown to execute in a fraction of the time required by existing metrics.

Recently, there has been an increasing amount of interest in exploiting aspects of human visual perception in the field of computer graphics. This is not surprising given that computer graphics is primarily dedicated to the production of visual stimuli. A few authors have investigated the use of visual difference predictors for assessing the accuracy of rendered images, others have utilized models of visual adaptation to develop tone reproduction operators ¹, and a few rudimentary attempts have been made to employ perceptual metrics to control aspects of a rendering algorithm. However, despite the enormous interest in this line of work, the previous applications of human visual perception are very limited.

One major problem with previous attempts to incorporate elements of a perceptual model within an adaptive sampling algorithm is that all prior adaptive algorithms have operated in the spatial domain. This means that the image approximation and the error estimate is calculated at discrete locations within the image plane. The perceptual sensitivity of the human visual system is strongly frequency dependent. Therefore a frequency based estimate of the image and the error is

¹A tone reproduction operator defines an intensity mapping that is intended to match the subjective experience of viewing a real and synthesized image.

necessary to fully exploit variations in visual sensitivity.

The third major contribution of this dissertation develops a new adaptive sampling algorithm that synthesizes images directly into the frequency domain. The synthesis of images into this domain allows a simple frequency dependent perceptual quality model to be used to guide the placement of samples. This model incorporates a number of key aspects of the human visual system. These include contrast sensitivity, spatial frequency response, and a rough approximation of masking. This work is novel in that it is the first time that frequency dependent aspects of the visual system are used to direct the effort of a rendering algorithm. This allows the most perceptible artifacts to be eliminated before those that are less apparent. In addition, the frequency representation that is employed by this algorithm is the same as that used in common image and video compression schemes. Synthesizing images directly into this representation reduces the expense and waste of first rendering a high quality image and then converting it into a compressed form. It additionally allows the algorithm to exploit a number of features of a common image compression technique.

This adaptive algorithm is a preliminary attempt to guide the production of synthetic images based on a perceptual model. The basic ideas contained in this algorithm are refined to create a second generation, perceptually based adaptive sampling algorithm. The new algorithm significantly improves upon the speed and practicality of the original technique. The new adaptive algorithm additionally incorporates a more advanced model of human perception that includes the affect of visual masking and chromatic aberration.

The second generation algorithm uses a high speed, color visual difference pre-

dictor to make perceptual quality assessments during the production of a synthetic image. The use of this predictor allows the algorithm to automatically detect the regions of the image containing visible artifacts. By focussing effort on these areas, the new algorithm is able to produce images of a given visual quality using fewer samples than previous sampling techniques. In addition, this algorithm allows the user to select a perceptual quality for the output image. This feature eliminates the guesswork involved in halting a rendering, and allows consistent results to be achieved regardless of the scene or shading technique that is employed.

The new adaptive algorithm is based on a novel technique that calculates the image approximation and error estimate directly in a pyramidal, wavelet domain. This allows the algorithm to use information about the frequency characteristics of an image and facilitates a rapid application of the perceptual model. It also provides a means to exploit the frequency characteristics of typical images to make informed guesses about the error contribution of unrefined regions.

An enhanced version of the basic technique is also presented. The enhancements incorporated into this final approach significantly increase the performance of the adaptive algorithm by limiting the number of times and locations at which the visual model is applied. These enhancements are shown to allow the algorithm to not only reduce the number of samples required to produce an image of a given visual quality, but the overall execution time as well. A number of timing tests will be presented that demonstrate that the new perceptually based adaptive sampling algorithm is able to synthesize visually accurate images using only a fraction of the time required by existing sampling techniques.

Including this introduction, the dissertation is divided into eleven chapters. In

the second chapter the radiance equation is derived. The third chapter overviews global illumination algorithms and motivates the focus of this dissertation on the visibility tracing technique. In the fourth chapter the existing adaptive sampling algorithms are discussed. This is followed in the fifth chapter by the development of a new error metric for Monte Carlo ray tracing. Chapter six presents the prior perceptual quality models that have been developed by the image processing community. The seventh chapter describes the design of a new high speed, color visual difference predictor. A discussion of previous applications of perception in the computer graphics field is contained in chapter eight. Chapter nine presents the initial frequency based adaptive sampling algorithm. This is followed in chapter ten, by the development of the new perceptually based adaptive sampling algorithm. Final concluding remarks are contained in chapter eleven.

CHAPTER II

THE RADIANCE EQUATION

The radiance equation is the fundamental equation describing how light flows within a three-dimensional environment. This equation is physically based, meaning that it seeks to accurately model the large scale physics of light interaction with the participating media. The basis for this equation comes from the fields of radiative transfer and transport theory. The radiance equation was first introduced to computer graphics by Kajiya in 1986 [76]. It has since formed the foundation of all physically based global illumination algorithms. All such algorithms provide an approximate solution to this equation.

This chapter outlines the important steps in the derivation of the radiance equation. A complete discussion of this equation is included in the appendices. The chapter begins with a description of the transport equation which defines a balance relation between the flow of abstract particles within a three-dimensional environment. This transport equation is then converted to an integral equation in order to incorporate the boundary conditions found at the surfaces within the environment. This is followed by the introduction of some basic radiometric concepts in order to convert from the notion of abstract particles to concrete and measurable quantities of light energy. Finally, the full radiance equation is described along with its most common simplification. The development of this section draws heavily on the works of Glassner [56], Arvo [70] and Kajiya [55].

The Transport Equation

The transport equation seeks to describe the relationship that governs the flow of particles within Euclidean space. In the development of this equation we will leave behind the concept of light transport and instead focus on the interaction of abstract particles with an environment. The particle nature of light is used in the later section on radiometry to tie this equation back to the specifics of light interaction.

The primary quantity of interest in the transport equation is called the flux. Flux describes the quantity of particles passing through an environment per unit time. It is defined as

$$\Phi = \frac{dQ}{dt} \left[\frac{\text{particles}}{S} \right], \quad (1)$$

where Q is the number of abstract particles.

The transport equation is developed by forming a balance relationship between the flux gains and losses in a volume of space and over the directions of interest. There are five categories of gains and losses which are illustrated in Figure 1. They are streaming, emission, absorption, outscattering and inscattering.

Streaming describes the amount of flux that is either injected into or escapes from the surfaces of a volume. Emission and absorption describe the volumetric effects from flux being either produced or consumed from within the volume itself. Finally, outscattering and inscattering are the result of particles being deflected away from or into the direction of interest.

When the system is in equilibrium the flux gains from emission and inscattering

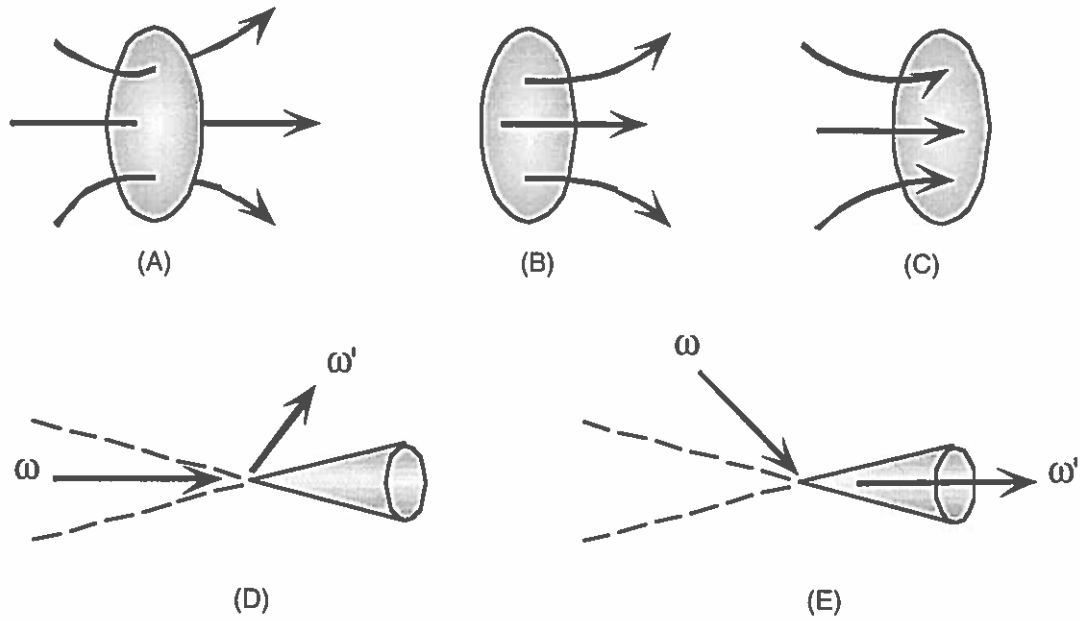


FIGURE 1. The five categories of gains and losses (after Glassner [56]). A) Streaming B) Emission C) Absorption D) Outscattering E) Inscattering

exactly balance the losses from streaming, outscattering and absorption. This yields the expression

$$\epsilon_r + K_r^i \Phi_r^i = \nabla_w \Phi_r + \Phi_r K_r^o + \sigma_r \Phi_r, \quad (2)$$

which describes a balance relation for the flux at an arbitrary point in space r . In this expression, position is denoted by the subscript and inscattering and outscattering are denoted by the superscript i and o respectively. The emission term (ϵ_r) represents the emitted flux at the point r . The inscattering term ($K_r^i \Phi_r^i$) is given by an integral operator (K_r^i) that computes the percentage of incoming flux (Φ_r^i) that is deflected along the direction of interest. The streaming term ($\nabla_w \Phi_r$) is formed by a gradient operator (∇_w) which computes the change in the flux (Φ_r) at the point r . The outscattering term ($\Phi_r K_r^o$) is the product of the flux at the point r and another

integral operator (K_r^o) which computes the percentage of flux deflected away from the direction of interest. Finally, the absorption term ($\sigma_r \Phi_r$) is given by the percentage of the flux absorbed (σ_r) times the flux at the point r . These terms are discussed in detail in Appendix A, pages 226–231.

This equation merely specifies a balance relationship which all points and directions in space must satisfy. In order to actually determine the flux at a specific point we must specify a set of boundary conditions. These conditions explicitly describe the flux at a series of boundary points (the surfaces of objects). The most general form of boundary conditions expresses the flux (Φ_s) at a surface in terms of the emitted and reflected (inscattered) flux along the direction of interest. This expression is

$$\Phi_s = \epsilon_s + K_s^i \Phi_s^i, \quad (3)$$

where the subscript s indicates the surface point, the emission term (ϵ_s) represents the flux emitted from the surface and the inscattering term ($K_s^i \Phi_s^i$) represents the percentage of incoming flux that is reflected from the surface along the direction of interest.

The Integral Equation

The transport equation allows us to compute the flux at an arbitrary point in space. However, the form of this equation is not particularly easy to work with since it contains both an integral and a derivative of the unknown quantity Φ (see Appendix A, pages 226–231).

This equation may be converted to a more useable form by observing that the

transport equation can be expressed as a linear first-order differential equation (see Appendix A, pages 231–232). This equation can be solved for the flux at a specific point (Φ_r) through the use of integrating factors. This yields the integral form of the transport equation which expresses the flux in terms of volumetric effects and the contributions from the nearest surface point backwards along the direction of interest:

$$\Phi_r = \mu_{(s \rightarrow r)} \Phi_s + \int_r^s \mu_{(\alpha \rightarrow r)} (\epsilon_\alpha + K_\alpha^i \Phi_\alpha^i) d\alpha. \quad (4)$$

Within this expression the term μ signifies the percentage of flux lost between two points in space due to outscattering and absorption. The expression $\mu_{(s \rightarrow r)} \Phi_s$ gives the flux contributions to the point of interest r from the surface point s . This is simply the surface flux from the boundary conditions, weighted by the percentage of flux lost as it travels to the point of interest. The integral on the right-hand side of Equation 4 accounts for the flux contributions due to volumetric effects. The term $(\epsilon_\alpha + K_\alpha^i \Phi_\alpha^i)$ represents the flux gains due to volumetric emissions and inscattering at the point α . This is weighted by the term $\mu_{(\alpha \rightarrow r)}$ which is the percentage of flux lost as it travels to the point of interest r . The integral is then the sum of these gains across all points from the surface to the point of interest.

Radiometry

The transport equation is defined in terms of flux, which is the rate of flow of some abstract particle. In this section we explore the field of radiometry in order to relate the abstract notion of flux to measurable quantities of radiant (light) energy.

The fundamental unit of radiant energy is the Joule. In terms of the particle

model, we may think of each particle as being a photon carrying some quantity of radiant energy Q^e . The radiant flux is then

$$\Phi^e = \frac{dQ^e}{dt} \quad \left[\frac{J}{S} = Watt \right]. \quad (5)$$

Flux is a measurement of flow. However for this quantity to have physical meaning we must answer the question: flowing through what? In the radiance equation we will want to be able to measure the flux across a patch of surface and through a set of angular directions. In order to do this we must define a flux density which is the ratio of flux to both area and solid angle. This term is called the radiance and it is defined as

$$L = \frac{d^2\Phi^e}{dA^\Phi d\omega} = \frac{d^2\Phi^e}{dAd\omega^\Phi} = \frac{d^2\Phi^e}{dAd\omega\cos\theta} \quad \left[\frac{W}{sr \cdot m^2} \right], \quad (6)$$

where dA^Φ is the projected area in the direction ω , $d\omega^\Phi$ is the projected solid angle onto the surface patch dA and θ is the angle between the surface normal and ω . Radiance is an important radiometric term because it allows us to compute the flux between two surface patches given only the surface geometry and the radiance along the line between the surfaces.

Since the radiance equation will eventually be defined in terms of radiance, we need to be able to express the scattering that occurs at points in space in the same manner. The scattering term that allows us to do this is called the bidirectional reflectance distribution function (BRDF). It is defined as

$$f_r(r, \omega \rightarrow \omega') = \frac{L(r, \omega')}{L(r, \omega)\cos\theta d\omega} \quad \left[\frac{1}{sr} \right], \quad (7)$$

where $f_r(r, \omega \rightarrow \omega')$ defines the percentage of the radiance at the point r that is scattered from an incident solid angle centered around the direction ω , into the outgoing direction ω' . In this expression $L(r, \omega)$ is the incident radiance at the point r and direction ω , $L(r, \omega')$ is the scattered radiance along direction ω' and $\cos\theta d\omega$ is the incident projected solid angle. This expression implies that if we know the incident radiance and the solid angle over which that radiance is defined, then the BRDF allows us to compute the contribution to the outgoing radiance along any specific direction.

The inscattering that occurs at a point in space r is found by integrating the product of the BRDF and the incident radiance across all incident projected solid angles. The inscattering integral operator B_r^i is defined to perform this operation. This operator is similar to the scattering operator K_r^i discussed earlier, except that it is defined in terms of radiance instead of flux.

The Full Radiance Equation

The full radiance equation is the goal of this section. It provides a complete description of the radiance at a point in space. This equation is based upon the integral form of the transport equation but is defined in terms of radiance instead of the abstract notion of flux.

The radiance equation is defined as

$$L_r = \mu_{(s \rightarrow r)} [L_s^e + B_s^i L_s^i] + \int_r^s \mu_{(\alpha \rightarrow r)} [L_\alpha^e + B_\alpha^i L_\alpha^i] d\alpha. \quad (8)$$

This expression is derived from the integral form of the transport equation (Equation 4) by converting from flux (Φ) to radiance (L) and from the scattering operator

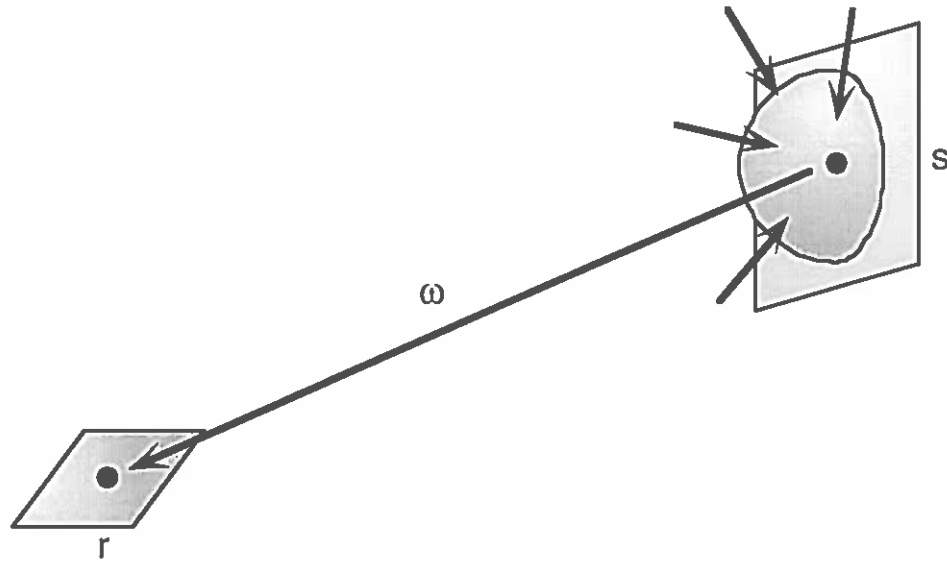


FIGURE 2. The geometry of the radiance equation in a vacuum (after Glassner [56]).

(K) to a BRDF operator (B). It additionally incorporates the boundary conditions from Equation 3 to determine the radiance at the surface point s .

This expression is commonly simplified by assuming that all synthesis occurs in a vacuum. This eliminates all volumetric effects from the expression and yields the final simplified version of the radiance equation:

$$L_r = L_s^e + B_s^i L_s^i. \quad (9)$$

This expression says that the radiance (L_r) at a point r is just the sum of the emitted radiance (L_s^e) and the inscattered radiance ($B_s^i L_s^i$) from the nearest surface point s . The incident radiance (L_s^i) is found by recursively applying Equation 9. This situation is illustrated in Figure 2. This limited form of the full radiance equation is what current realistic image synthesis algorithms are dedicated to solving.

Summary

This chapter discussed the theory underlying physically based realistic image synthesis. This theory is embodied in a fundamental formula known as the radiance equation. The derivation of this equation introduced the basic concepts that facilitate the production of physically correct images. This chapter also laid important groundwork for the remainder of this dissertation.

The derivation of the radiance equation proceeded through a number of basic steps. First, a balance relation was developed that governs the flow of particles in three-dimensional space. Next, the boundary conditions found at the surfaces of objects were incorporated, and the balance relation was solved for the particle flow at a particular point in space. The radiometric concept of radiance was then introduced to convert from an abstract characterization of particle flow to measurable units of light energy. The BRDF was also presented as a means to describe the scattering that occurs at the surfaces of objects. Finally, these radiometric concepts were incorporated to develop the full radiance equation, as well as its most common simplification.

This simplified version of the radiance equation describes a recursive relation whereby the outgoing radiance along a particular direction in space can be calculated by evaluating the incident radiance, at a surface point, from all possible directions. The next chapter describes the photo-realistic techniques that produce images by solving this equation. These techniques are known as global illumination algorithms.

CHAPTER III

GLOBAL ILLUMINATION ALGORITHMS

Global illumination algorithms provide a mechanism to generate physically based renderings of synthetic scenes. These techniques are based on the theory described in the preceding chapter. This type of rendering method produces images by first solving the radiance equation across the surfaces of the scene and then projecting the result onto the image plane. The illumination calculation is global in the sense that all surfaces contribute to the radiance solution.

Global illumination algorithms are state-of-the-art image synthesis techniques. Rendering programs that utilize these methods can synthesize images that are indistinguishable from photographs taken of a real environment. This chapter contains an introductory discussion of these algorithms and provides background information on several approaches to realistic image synthesis. The discussion focuses on the two most prevalent global illumination algorithms: ray tracing and radiosity. A number of variations on the ray tracing algorithm are also presented. This chapter concludes with a comparison of the various algorithms in order to motivate why this dissertation utilizes a form of ray tracing called visibility tracing.

Ray Tracing

Ray tracing was first introduced in papers by Appel in 1968 [4] and by Goldstein and Nagel in 1971 [57]. Ray tracing defines a point sampling solution to the

radiance equation. In ray tracing, infinitely thin rays are intersected with the objects of the scene in order to evaluate the radiance at specific positions within the environment. There are two major forms of ray tracing: visibility and photon tracing. In visibility tracing, rays are used to gather incident light from visible points within the scene. Photon tracing is an alternative technique which uses rays to project energy from the light sources into the scene. These two methods can be combined to produce a technique known as bidirectional ray tracing. These methods as well as two powerful acceleration techniques are described in this section.

Visibility Tracing

In visibility tracing rays are cast from the eye point through the pixels of the image plane in order to point sample the incident light at various positions on the image plane. The rays are intersected with the objects of the scene and the reflected radiance from these points is evaluated. The radiance that is reflected from a surface point is calculated from the radiance equation given in Equation 9. This equation defines the reflected radiance as the sum of the surface emittance and the incident radiance from all directions multiplied by the surface scattering function (BRDF). In order to evaluate the incident radiance, additional rays are spawned from the surface. These rays are similarly intersected with objects of the scene to compute the radiance contributions from these points. The process recurses until the contributions of the rays is below some tolerance. The rays that are generated form a tree which is evaluated in a bottom-up manner in order to ultimately define the incident radiance at a position on the image plane.

Classical Ray Tracing

Classical ray tracing is based on a model presented by Whitted in 1980 [156]. This method is an early solution to the problem of visibility tracing. It provides a simplified approximation to the process outlined above. In classical ray tracing a simple local illumination model is evaluated at the surfaces of objects to determine the outgoing radiance along a ray's direction.

A novel contribution of Whitted's work is that he was the first to spawn rays from the surfaces of objects to evaluate the local illumination model. Whitted cast rays at the light sources to determine the direct lighting contribution. He additionally cast rays in the reflected and transmitted directions in order to determine the radiance contributions from these sources. The reflected and transmitted rays recurse through the scene to form a ray tree. This ray tree provided the first coarse approximation to a global algorithm.

Monte Carlo Tracing

Monte Carlo ray tracing (also known as distributed ray tracing) is based on papers presented by Kajiya [76] and Cook [33, 34]. This method powerfully extends classical ray tracing through the use of stochastic sampling and the distribution of rays across sampling dimensions. Monte Carlo ray tracing makes it possible to simulate the entire range of diffuse inter-reflections. This allows the method to incorporate a much wider range of effects than classical ray tracing which can only simulate point light sources, reflection and transmission.

In the Monte Carlo algorithm, the shading calculation at a surface point is evaluated by randomly distributing rays across all incident directions. This enables

the shading calculation to account for the radiance contributions from all visible surfaces as well as area light sources. This ray tracing method can be viewed as the direct use of a Monte Carlo integration technique to evaluate the shading integral. The full BRDF is required for this calculation.

Path Tracing

Monte Carlo ray tracing spawns many rays at each surface reflection. This causes an exponential growth in the number of rays at the lower levels of the ray tree, where the ray's contribution are the least significant. Path tracing was introduced by Kajiya [76] to eliminate this problem. In path tracing a single randomly positioned ray is spawned from each surface reflection. This method requires more initial rays to be cast from the eye point but because the number of rays does not grow; more effort is concentrated at the higher levels of the tree where the ray's contribution is the greatest.

Photon Tracing

Photon tracing was originally introduced by Appel in 1968 [4] and further explored by numerous authors including Shirley, Rushmeier, Pattanaik and Mudur [108, 109]. Photon tracing simulates the projection of photons from the light source into the scene. In this method rays are originated at the light sources and projected outward into the scene. These rays represent the portion of the light's radiance that is projected along the direction of the ray. These rays are intersected with the objects of the scene and new rays are scattered from the surface. These new rays represent the fraction of the light's radiance that is reflected from the surface

along the outgoing directions. The rays which eventually reach the image plane represent the fraction of the light's radiance which enters the eye of the observer. The contribution of these rays are combined to yield the final image.

Bidirectional Ray Tracing

Visibility tracing excels at sampling the visually important directions, however, it is very poor at evaluating indirect lighting. Photon tracing, on the other hand, excels at sampling the important lighting directions but is very poor at producing rays which actually pass through the image plane. The strengths of these two techniques can be combined to produce a method known as bidirectional ray tracing. This method was introduced by Arvo in 1986 [6].

In bidirectional ray tracing an initial photon tracing pass is performed followed by a visibility tracing pass. The photon tracing pass is used to deposit indirect illumination of the surfaces of objects. This indirect radiance contribution is stored in an illumination map. This illumination map is then used in the subsequent visibility tracing pass in order to determine the indirect lighting contribution along the visually important directions.

Acceleration Methods

Ray tracing is inherently a very expensive process. It is therefore important to perform this process as efficiently as possible. This section will explore two of the more important acceleration methods.

Bounding Volumes

It is estimated that 95% of the time to generate a ray traced image is spent performing ray-object intersection tests. It is therefore imperative to perform these tests as efficiently as possible. The use of bounding volumes has been proposed by a number of researchers to accelerate the ray-object intersection test. An excellent survey of these algorithms is presented by Arvo and Kirk [54].

The basic idea is to enclose the objects of a scene within a simplified bounding volume such as a sphere or a box. The intersection of a ray with this bounding volume can be performed much faster than the intersection with the object itself. The vast majority of these bounding volume tests will fail and only those that succeed will require the more complex object intersection tests. This idea can be expanded further by creating a hierarchy of nested bounding volumes. The intersection test then proceeds by testing the top-most bounding volume. If the test succeeds then the nested bounding volumes will be tested and the process iterates. This technique can reduce the complexity of the intersection tests from the order of the number of objects in the scene to the log of this number.

Diffuse Interpolation

The second acceleration method was presented by Ward [149]. He observed that diffuse reflection generally varied very slowly across the surfaces of a scene. This diffuse reflection is extremely expensive to compute through the use of Monte Carlo ray tracing. He therefore proposed that this value be stored each time it is computed. Then whenever a ray strikes a surface, we can find the nearest points at which this value has already been computed. If these values are within a reasonable

distance then they can be interpolated to find the diffuse reflectance at the surface point. Otherwise a new value will be computed and stored.

Radiosity

Radiosity was introduced to computer graphics by Goral et al. in 1984 [59], Nishita and Nakamae in 1985 [105], and Cohen and Greenberg in 1985 [31]. Radiosity defines an analytic solution to the radiance equation by constructing a set of explicit equations relating the distribution of radiance in the scene.

In order for this solution to be tractable, radiosity makes a number of simplifying assumptions about the environment. It assumes that all surfaces are opaque and that they reflect only diffusely. Radiosity further simplifies the solution by meshing the surfaces of all objects into smaller patches of constant radiance.

These simplifications allow the radiance equation to be re-expressed as

$$B_i = E_i + \rho_i \sum_{k=1}^n B_k F_{i,k}, \quad (10)$$

where B_i is the outgoing flux per unit area from patch i , E_i is emitted flux per unit area from patch i , ρ_i is the reflectivity of the patch and $F_{i,k}$ is a “form-factor” expressing the ratio of flux from patch k that reaches patch i . This expression establishes a system of linear equations which relates the radiosity (B) of a surface patch to the emittance of that patch and the reflected radiosity from all other patches in the environment.

These equations form a matrix relation which can be inverted to solve for the

radiosity at all surface patches:

$$\mathbf{B} = (\mathbf{I} - \mathbf{PF})^{-1} \mathbf{E}, \quad (11)$$

where \mathbf{I} is the identity matrix and \mathbf{P} is a diagonal matrix of patch reflectivities. The solution of this equation gives a view-independent description of radiosity at the surfaces of objects in the scene. The radiosity of the surface patches can then be projected onto a given view plane to form the final image.

Global Illumination as a Foundation for Perceptually Based Rendering

Early graphics algorithms were characterized by very simple shading calculations. These algorithms compute the light reflected from a surface by taking into account only the direct contribution of point light sources. The latest global illumination techniques, on the other hand, are based on the radiance equation, and calculate the reflected light by recursively evaluating the incident light from all angular directions. These techniques are capable of producing stunning, photo-realistic images that contain a number of advanced effects, including diffuse inter-reflections between surfaces, complicated reflection models, and realistic lighting with soft shadows.

Unfortunately, the applicability of these realistic rendering methods are presently limited by their expense. As mentioned in the introduction, this dissertation shows how utilizing knowledge about the perceptual sensitivity of the human visual system can be used to improve the efficiency of a global illumination algorithm. This work will concentrate primarily on the visibility tracing algorithm discussed in pages

19–21 of this chapter. There are a number of reasons for this choice, including the popularity of visibility tracing, the wide range of effects that can be simulated by the algorithm, and most importantly the view-dependent approach this method uses to generate images.

Visibility tracing is the oldest and still the most popular of the global illumination algorithms discussed in this chapter. One reason for the popularity of this algorithm is its simple recursive architecture. In visibility tracing the same technique is employed to determine the contribution of primary rays from the viewpoint, as is used to find the contribution of rays spawned from subsequent reflections and transmissions. Calculating the contribution of a ray is a relatively straightforward operation involving an intersection test and a shading operation. This process is much simpler than the complicated steps of discretizing the surfaces of a scene, evaluating the form factors, and inverting a large matrix that are necessary in the radiosity algorithm. In addition, visibility tracing does not incur the extra overhead of building the illumination map that is required for bidirectional ray tracing. The construction of this illumination map adds extra complexity to the algorithm and significantly increases the amount of memory required. Another reason for the popularity of visibility tracing is its use of point sampling as a technique for evaluating the radiance of the scene. It is far easier to take point samples in order to solve complicated functions, than it is to solve such functions analytically. This point sampling technique allows visibility tracing to easily handle functionally defined objects, texture maps and general BRDF's. These elements are extremely difficult to incorporate into an analytic radiosity algorithm.

The second reason for choosing visibility tracing as the basis for this work

is the wide variety of effects that can be simulated by this algorithm. Visibility tracing is currently the most powerful global illumination algorithm, and it is the only algorithm capable of solving the most general form of the radiance equation. Radiosity, and to a lesser extent bidirectional ray tracing, must restrict the shading calculation to include only opaque surfaces that reflect diffusely. This restriction is necessary in order for these algorithms to calculate the view-independent illumination at the surfaces of the scene. Visibility tracing, on the other hand, needs no such restrictions.

The final and most important reason for selecting visibility tracing as the basis of this work is that it is a view-dependent algorithm. View-dependence implies that the entire rendering process is dedicated to constructing a single view of a scene, as defined by a specific eye point and image plane location. As previously mentioned, this dissertation will show how human perception can be exploited to accelerate the rendering process. The ability of the visual system to perceive error will be shown to vary based on a number of factors that are strongly dependent on specific image content. Therefore, it is impossible to fully exploit the benefits of variations in the visual system's sensitivity in a view-independent illumination calculation. Since radiosity and bidirectional ray tracing spend a significant fraction of their execution time evaluating the view-independent illumination at the surfaces of a scene, there is less opportunity to exploit view-dependent visual effects within these algorithms. Photon tracing is a view-dependent algorithm, since it generates only a single view of a scene. However, the samples in this algorithm are initiated at the light sources and passively collected at the image plane. This process makes it more difficult to alter the sampling of the scene in order to achieve a given accuracy at a particular

location in the final image.

Summary

This chapter has described the major global illumination algorithms developed by the computer graphics community. These powerful, photo-realistic rendering techniques are state-of-the-art image synthesis algorithms. These algorithms produce images by solving the radiance equation at the surfaces of a scene and projecting the result onto an image plane.

There are two major global illumination algorithms that have been discussed: ray tracing and radiosity. Ray tracing uses a point sampling approach to solve the radiance equation, whereas radiosity is based on an analytic technique. A number of variations of the ray tracing algorithm have also been presented. Visibility tracing is a technique that is driven from the image plane, photon tracing is a method where rays are initiated at the light sources, and bidirectional ray tracing is a combination of these two approaches.

The final section of this chapter motivated the focus of this dissertation on the visibility tracing technique. This method stands out as the predominant algorithm in which to exploit the benefits of perceptually based rendering. In addition to its popularity and power, visibility tracing is view-dependent. This fact makes it possible to exploit all of the factors affecting how error is perceived in the final image. Moreover, the sampling in visibility tracing is initiated at the image plane. The following chapter will show how this fact makes it relatively straightforward to alter the sampling distribution in order to efficiently control the error at specific locations of the image plane.

CHAPTER IV

ADAPTIVE SAMPLING ALGORITHMS

Visibility tracing is the most powerful of the global illumination algorithms described in the previous chapter. This technique is capable of solving the general radiance equation and creating images of unrivaled realism and subtlety. However, the quality produced by this algorithm comes with the price of lengthy execution times. It is, therefore, important that this technique computes an image by the most efficient means possible.

Visibility tracing generates an image by evaluating many samples of the light intensity that is reflected from a synthetic scene. These samples can be very expensive to compute since each sample is the result of a recursive, photo-realistic shading calculation. The quantity of samples that are necessary to reconstruct an image is one of the primary factors affecting the speed of the algorithm.

This chapter reviews the previous sampling techniques that have been employed in the visibility tracing algorithm. The nature of these techniques is seen to have a pronounced effect on the expense of the rendering algorithm. The chapter begins with a discussion of the sampling and reconstruction problem that is inherent in the production of images by the visibility tracing technique. A few basic approaches to the solution of this problem are also presented. This is followed by a detailed description of previous adaptive sampling techniques. These techniques are capable of significantly reducing the number of samples required to accurately

reconstruct an image. This is accomplished by varying the sampling density across the image plane so that each area of the image receives only the minimum number of samples necessary to accurately reconstruct it. The discussion of these adaptive methods includes a description of the base sampling patterns, refinement tests, and reconstruction techniques that are the primary components of these algorithms.

Introduction to Sampling and Reconstruction

Visibility tracing is a point sampling and reconstruction problem. In visibility tracing rays are cast through locations on the picture plane to determine the image radiance at a series of discrete points. These samples are then filtered to reconstruct a continuous intensity profile of the image. Finally, this continuous function is resampled at the pixel locations in order to generate the resulting picture.

Point sampling and reconstruction is a classic problem that is discussed at length in many of the standard engineering texts [106, 117]. The theory for the ideal solution to this problem is described in Shannon's sampling theorem [129]. The sampling theorem states that a signal must be sampled at twice the rate of the highest frequency present in the signal in order to accurately represent it. This rate is known as the Nyquist rate. If a signal has been uniformly sampled at the Nyquist rate it is possible to exactly reproduce the original function by convolving the samples with a sinc reconstruction filter. Sampling a signal below the Nyquist rate results in an artifact known as an alias. Aliasing is produced by the overlap of the frequency spectra in under-sampled signals. In computer graphics, aliasing is commonly manifested as "jaggies," which are jagged intensity contours. This situation can result when the sample from one pixel falls on one side of an intensity

contour and the sample from an adjoining pixel falls on the other. This produces a harsh discontinuity between the two pixels, although the contour may gradually transition between them.

The sampling theorem provides an important theoretical background for the sampling and reconstruction problem in visibility tracing. However, it is just a starting point for developing practical algorithms. Due to the nature of visibility tracing, it is extremely difficult and inefficient to directly apply the sampling theorem to this problem. The first reason for this difficulty is that the spatial frequency spectrum of an image is generally not known prior to rendering a scene. This makes it impossible to determine a priori what the Nyquist rate should be. In addition, a number of factors can produce step luminance discontinuities in the image. These factors include object edges, shadow boundaries, and discontinuous textures. These discontinuities imply that the spatial frequency spectrum of an image can be, and often is, non-bandlimited. According to the sampling theorem an infinite number of samples are required to exactly reconstruct functions that are not bandlimited (although fewer samples are required to reduce the error below the color resolution of the display). Finally, the sampling theorem only applies to deterministic functions. Monte Carlo integration is a common technique used to evaluate the shading integral defined in the radiance equation. When this technique is employed, rays are spawned in random directions from a surface in order to evaluate the incident radiance. On average, this approach will return the true radiance at a particular location of the image plane. However, there is a certain amount of variance associated with each sample. This non-determinism is not handled by the sampling theorem and alternate techniques must be used to determine the appropriate sampling rate in this situation.

The earliest and simplest approach to sampling the image plane is to take a single sample at the center of each pixel. The intensity of each pixel is simply given by the radiance of the sample taken within it. This method, however, can severely under-sample the image. Modern scenes can be represented by thousands and even millions of polygonal primitives. The complexity of these objects coupled with the widespread use of texture mapping to add detail can produce images with a great deal of spatial variation. In addition, realistic shading calculations and the Monte Carlo integration techniques that are used to solve them can introduce significant variation in the intensity of the samples. A single sample per pixel is rarely enough to adequately sample scenes with large amounts of spatial and intensity variation.

Supersampling is one commonly used approach for dealing with situations where one sample per pixel is inadequate. In supersampling many samples are taken through each pixel of the image plane and the intensity of a pixel is given by the average of the samples. The number of samples per pixel is typically specified by the user. Selecting the appropriate sampling rate involves a good deal of guesswork and a fair amount of expertise on the part of the user. This process can also be inefficient since a number of trial and error attempts may be necessary before the correct sampling rate is determined.

Supersampling the image at a constant rate can be enormously expensive. In order to accurately compute the intensity of each pixel, this approach must sample all pixels at the rate necessary for the greatest spatial and intensity variance. Within a given image the spatial frequency content and the magnitude of the intensity variation can differ significantly based on the location within the image plane. The majority of the pixels within an image can often be accurately calculated with far


```
Divide the image into regions
Sample region
Apply refinement test
If above threshold
    Supersample region
Reconstruct image
```

FIGURE 3. Pseudocode for a generalized adaptive sampling algorithm.

fewer samples than is necessary at the location of peak variance.

In visibility tracing the number of samples used to construct an image is one of the predominant factors affecting the execution time of the algorithm. Each sample can be very expensive to compute, involving numerous intersection operations, realistic shading calculations, motion translations, and deformations. Therefore, it is important to guide this algorithm by the sampler's credo, which states that "every sample is precious [56]."

Adaptive sampling is a powerful technique that seeks to improve the efficiency of visibility tracing algorithms by varying the sampling rate at different regions of the image plane. The goal of this approach is to take the minimum number of samples necessary in each region to calculate that portion of the image to a specified error tolerance. Because the number of samples required to accurately construct a region can differ significantly within a given image and because the samples can be so expensive to compute, this technique can significantly reduce the amount of time required to render an image.

The adaptive sampling algorithms that have been developed share a number of common elements. A generalized outline of these algorithms is contained within Figure 3. The first step in an adaptive sampling algorithm is to divide the image plane into a number of regions. These regions define the resolution of the adaptive

algorithm. A low density set of pilot samples are taken within each region. Then a refinement test is applied to these samples in order to determine whether further sampling of the region is necessary. If the test determines that the error of the region is above a given threshold, more samples are taken within the region. Finally, after all samples have been taken, they are filtered and used to reconstruct the output image.

There are two techniques that are commonly applied to define regions and the supersampling that occurs within them. These are the two-pass and tree-based refinement strategies. In the two-pass method, regions are commonly defined to be either a single pixel or a small group of pixels. If the refinement test determines that further sampling of a region is necessary this approach resamples the area using a fixed rate, high density sampling. This technique is relatively crude, since it only chooses between two fixed sampling densities. However, only a single test needs to be performed in each region. The tree-based refinement strategy defines the initial region to encompass the entire image plane. As the refinement test determines that more sampling is required, new regions are formed by dividing the existing region into quadrants. More pilot samples are taken in each of the newly defined areas and the refinement test is applied independently on each quadrant. This process continues until a maximum resolution is reached. Tree-based refinement provides greater control over the placement of samples, but it requires that the refinement test be applied more often. In choosing between these two methods one must tradeoff the expense of more refinement tests versus the reduction of samples that comes with finer control over the sampling distribution.

There are three important aspects to any adaptive sampling algorithm. These

are the base sampling pattern, the refinement test, and the reconstruction technique. The following sections will review the prior work that has been performed in each of these areas.

Base Sampling Patterns

The base sampling pattern specifies the locations at which samples are placed within a region. These patterns are typically refineable so that they can define the placement of samples for both the initial pilot sampling and the later supersampling stages of an adaptive algorithm. There are two types of sampling patterns: *uniform* and *nonuniform*. Uniform sampling was a common technique used in early adaptive algorithms and bears close ties to classic sampling theory. Nonuniform sampling is a more recent approach. This method is often preferable because it can produce images in which the error is less objectionable than it would be had the samples been placed uniformly.

Uniform sampling patterns place samples at the vertices of a periodic lattice. The simplest such pattern places samples at the vertices of a rectangular grid [74, 122, 156, 163]. This produces the uniformly spaced, rectangular sampling pattern that is the basis of the two-dimensional sampling theorem. If the underlying signal is known to be bandlimited then the sampling theorem specifies how to produce an exact reconstruction of the signal using this sampling pattern.

Other uniform patterns that have been proposed include hexagonal [40, 117] and triangular [133] sample placements. These patterns place samples at the vertices of a hexagonal and triangular lattice respectively. The benefit of these distributions is that they provide a denser packing of samples within a given area. It has been

shown that the hexagonal sampling distribution requires 13.4% fewer samples than a rectangular distribution to accurately represent signals whose spatial frequency spectrum lies within a circle [56].

A pattern which places samples at the vertices of a diamond lattice has been proposed by Bouville *et al.* [18]. This pattern is formed by rotating a rectangular distribution by 45 degrees. In his paper Bouville showed the correspondence between this pattern and the directional sensitivity of the human eye. The directional sensitivity of the human visual system is known to have a diamond shaped cutoff with greatest sensitivity at horizontal and vertical orientations. The increased density of this pattern along the horizontal and vertical dimensions can produce images which are visually superior to those produced with a rectangular sampling pattern.

A difficulty with uniform sampling is that this method can produce highly objectionable error patterns when the signal is under-sampled. This occurs because coherent structure in the sampling distribution can combine with structure in the signal to produce patterns of error in the reconstructed image. The visual system is adept at recognizing patterns within an image. Therefore structured error can produce very visible artifacts within an image.

Nonuniform sampling patterns overcome this problem by randomly distributing samples within a region. This approach eliminates the periodicity of uniform sampling. The result is that the structured aliasing artifacts produced by uniform sampling are traded for noise in the reconstructed image. Noise is much less objectionable to the human visual system since we tend to overlook this incoherent error and instead concentrate on the underlying patterns within the image.

The simplest nonuniform sampling method is to place samples at completely

random locations within a region. This is known as Poisson sampling [34, 40, 93, 130]. A problem with Poisson sampling is that the randomly placed samples can clump together in one area while placing sparse samples in other areas. An optimal pattern is therefore one that places samples randomly, but with an even coverage of a region.

This goal is achieved by the Poisson-disk sampling pattern [33, 40, 98, 99]. In a Poisson-disk distribution, samples are placed randomly such that no two samples are within a specified distance of each other. This pattern is optimal in many senses, however the methods that have been proposed to generate this pattern tend to be somewhat expensive.

A cheaper alternative is to use a jittered sampling pattern [33, 40, 130]. A jittered pattern is achieved by perturbing the location of samples in any of the uniform sampling patterns by a limited, random distance. These patterns are inexpensive to generate and share many common characteristics with the Poisson-disk sampling pattern.

Refinement Tests

The refinement test is the most important element of an adaptive sampling algorithm. It is this test that ultimately governs the sampling density across the image plane. A refinement test is a metric that is applied to the samples within a region in order to determine whether the existing number of samples can produce an accurate representation of the image within the region. These tests are typically based on some measure of the uniformity of the samples within a region. If the samples are similar then the region can be accurately reconstructed with relatively

few samples. However, if the samples vary significantly, a higher sampling density is usually required. If the test result indicates that a portion of the image contains too much error then more samples will be taken within that region. Optionally, the refinement test can be applied again to the new sample values and the process can recurse until a specified tolerance is reached.

The following two subsection will discuss the refinement tests that have been previously developed. These metrics can be divided into two categories based on the type of information used in the test: *object based* and *radiance statistics*. A good review of existing refinement tests can also be found in a recent book by Glassner [56]. In addition to the metrics described in the following subsections, a few tests have been developed that apply a limited perceptual criteria to determine the accuracy of an image within a region. The discussion of these tests will be deferred until Chapter VIII.

Object Based Tests

Object based refinement tests use information about the objects in a scene in order to determine the appropriate sampling rate within a region. In visibility tracing, samples are taken by casting rays into a scene. These rays usually return only the radiance along the ray's direction. However, the information returned by a ray can be easily expanded to include items such as what object was intersected by the ray and what light sources illuminated the object. This information can be used to infer the locations of object edges and shadow boundaries. These areas are often indicative of radiance discontinuities which require a higher sampling rate.

This idea was introduced in an edge following algorithm by Roth [122]. His

algorithm tested neighboring samples to see if they intersected the same object. If two samples fell on different objects, a new sample was taken midway between them. In this manner a binary search is performed to localize the position of object edges.

The simple test used by Roth was extended in papers by Argence [5] and Akimoto *et al.* [2]. They suggested that for neighboring samples we should compare not only the first object intersected, but also the entire ray tree (which can contain a number of levels of reflected and transmitted contributions). Difference at any node in the ray tree was proposed as an indicator of the need for further refinement. In their papers Argence and Akimoto suggested a number of elements to be compared with the ray tree. These elements included the objects that were intersected, which light sources were illuminating the objects in the tree, and whether the same objects were intersected for specular and transmitted rays. Akimoto also proposed that texture mapped objects receive a higher sampling rate. Argence noted that objects are often composed of a number of smaller surface patches. The intersected patch was incorporated in his test as well. Argence additionally included a test for small objects.

Testing for small objects was first suggested by Whitted [156]. He noted that small objects can be missed when the sampling density is low. He suggested enclosing the objects of the scene with a bounding sphere that is at least one pixel wide in screen space. The intersection of rays with such a bounding sphere can be used to indicate when a small object lies between the existing samples, and a higher sampling rate can be triggered.

A similar idea was proposed by Thomas *et al.* [140]. In Thomas' algorithm bounding surfaces were placed around the objects of a scene. These surfaces were

called “covers”. As in Whitted’s method these covers were at least one pixel wide. This aids in the detection of object edges. One novel aspect of Thomas’ work is that he was able to pre-filter object edges by calculating the pixel to object distance in screen space.

Another approach to detecting object edges was described by Hashimoto *et al.* [69]. In his paper Hashimoto suggests that a perspective projection of the object edges onto the image plane be performed prior to rendering a scene. The pixels that contain an object edge are identified as “active pixels,” wherein a high sampling rate is necessary.

Finally, van Walsum noted that additional tests are necessary to determine when high frequency spatial detail is present as a result of texture mapping [143]. He proposed a metric that consisted of three criteria for determining when refinement is required. These criteria were the distance between the samples in texture space, the uniformity of the texture between the samples, and whether the sample values at a specific location in the texture map is close to a filtered local average.

Object based refinement tests are very good at detecting areas of spatial variation caused by object edges. Some of the tests described in this section are also capable of detecting variation caused by shadow boundaries and texture maps. However, all other sources of spatial and intensity variation will be neglected by these algorithms. This includes variations due to shading gradations, surface curvature, Monte Carlo shading calculations, and others. This lack of generality is a significant weakness of object based refinement tests.

A second problem with object based tests is that while the presence of an edge may indicate the need for a higher sampling rate, it does little to specify what this

rate should be. It is therefore up to the user to select the appropriate sampling rate for regions that do and do not contain edges. In the next subsection we will see a number of refinement tests that seek to address these issues.

Radiance Statistics Tests

The second category of refinement tests utilize metrics that are based on the radiance statistics. These metrics analyze the radiance, or light energy, returned by the samples in a region. The appropriate sampling rate is determined from some measure of the uniformity of these samples. Since this type of test does not utilize any a priori knowledge about the location of objects in the scene, it can be more difficult for these algorithms to isolate object edges. However, the generality of these tests makes them applicable to a wide variety of situations.

The simplest refinement test in this category is based on a comparison of the maximum and minimum radiance within a region. This test was first proposed by Whitted [156] and a similar idea was suggested by Jansen and van Wijk [74]. In Whitted's algorithm four samples are taken in the corners of a region. If the maximum and minimum radiance of these samples differs by more than a threshold value then the region is refined. A region is refined by dividing it into quadrants to form four new regions, samples are taken in the corners of the new regions, and the process recurses. This continues until the threshold is satisfied or a maximum resolution is reached. Glassner states that a good value for the threshold difference is given by the minimum displayable color difference of the frame buffer [56].

A more advanced statistical test was presented by Dippé and Wold [40]. They observed that noise is the predominant source of error in images produced by nonuni-

form sampling patterns. They suggested the use of the signal to noise ratio (SNR) as a means to measure the magnitude of this error. The SNR is a common metric used to measure the quality of a communications line. It is equally useful for measuring the quality of an image in the presence of noise. Dippé and Wold cite evidence that indicates that the noise detection threshold of the human visual system occurs at a SNR of about 100 (40 db). In their adaptive algorithm samples are repeatedly taken within a region and the SNR is estimated from the radiance of the samples. Sampling is halted once the SNR reaches this threshold.

A similar test was suggested by Lee *et al.* [81]. In Lee's test variance is used to measure the quality of the image within a region. The variance is a statistical measure of the range of values of a function. When this range is small the function can be accurately estimated with few samples. If the range is large more samples are required. One advantage of variance over SNR is that the variance is very simple to estimate from the sample values. Lee noted that the variance of the image within a region is equal to the variance of the samples divided by the number of samples taken. This rule provides a simple estimate of the quality of the image. Since the variance is only estimated, this value is somewhat imprecise. In order to handle this, Lee used a chi-squared (χ^2) test to determine the probability that the variance is less than a threshold value. In Lee's algorithm samples are iteratively taken within a region until the χ^2 value is less than a user specified threshold. The result of this algorithm is that all regions of an image will be sampled until their variance is below some value and the remaining error will be spread equally across the image plane.

A derivative of Lee's refinement test was described by Purgathofer [118]. In his test Purgathofer utilizes the same approach to estimating the variance within

a region. A t-test is then applied to the variance estimate in order to construct a confidence interval for the estimate of the image. When this test is satisfied the estimated image is known to lie within a given range of the true image with a user specified probability. Purgathofer also points out that the variance estimate is less accurate when the sampling density is very low. He proposes a technique for determining the minimum sampling rate based on an assumption that the worst case image is described by an asymmetric binomial distribution.

Painter and Sloan developed a refinement test that is specifically designed for a tree-based refinement strategy [107]. Their algorithm utilizes a k-D tree as its primary data structure. The root node of this tree is defined to encompass the entire image. As nodes in this tree are selected for refinement, two new child nodes are created by splitting the parent node alternatively along the x and y axis and taking samples within the new regions. At the leaf nodes a heuristic is computed that indicates a refinement priority. They define this heuristic to be the variance within the region of the node times the area of the node. In this manner nodes which cover a large area will tend to be refined before nodes which cover a smaller area. For interior nodes the refinement priority is defined to be the maximum priority of the two child nodes. This allows the algorithm to select the node that most needs refinement by simply traversing the tree in a top-down fashion and selecting the branch with the largest refinement priority. As their stopping criteria, they use a confidence interval test based on the student-t distribution. This test is similar to that described by Purgathofer.

Reconstruction Techniques

After the scene has been sampled, the sample values are used to reconstruct the final image. If a single sample is taken at each pixel, this process can be as simple as setting the radiance of each pixel to the radiance of the sample that was taken within it. For other uniform sample distributions reconstruction is guided by the two-dimensional sampling theorem. This theorem states that an exact reconstruction of the image is possible if the scene has been uniformly sampled at the Nyquist rate. Reconstruction proceeds by first convolving the samples with a sinc reconstruction filter in order to produce a continuous representation of the image. The continuous representation is then lowpass filtered to bandlimit the signal to the resolution of the display. Finally, the signal is resampled at the pixel locations to form the final image.

Unfortunately, practical reconstruction techniques are not so straightforward. Due to the use of adaptive sampling as well as jittered and Poisson sampling patterns, the samples are generally distributed nonuniformly across the image plane. This implies that within a given image some pixels may not receive any samples whereas other pixels may be heavily sampled. The optimal method to reconstruct an image from a set of nonuniformly distributed samples is still an open question. This section will describe a number of reconstruction techniques that have been proposed in the literature.

One of the simplest approaches to reconstructing an image is to generate a *piecewise constant* representation of the image from the sample values. This approach was first described by Whitted [156]. He proposed tiling the image plane with a series of non-overlapping rectangular regions, such that each region contained

a single sample. This approach is especially suited for tree-based refinement strategies, since these algorithms naturally divide the image plane into such regions. In Whitted's method the radiance across each region is defined to be constant at the value of the sample taken within it. The intensity of a pixel is given by the average radiance of these constant patches across the area of the pixel. This average value is found by taking the weighted sum of the fraction of the pixel covered by each region times the radiance of the regions.

A more sophisticated tiling method is presented by Wyvill and Sharp [163]. In the adaptive stage of their algorithm, samples are taken along the edges of a pixel. These samples are analyzed to determine the locations of color changes along this border. This information is used to divide the pixel into a number of triangular regions. Two of the vertices of each triangle are placed along the edge of the pixel at the locations of neighboring color transitions. The third vertex is placed within the pixel at the centroid of the color edges. Each triangular region is defined to have a constant radiance. The intensity of the pixel is given by the area weighted average of the radiance of these regions.

The quality of a piecewise constant reconstruction can often be improved by additional filtering. This technique was used by Painter and Sloan [107]. In their method the image plane is tiled with constant rectangular regions in a similar manner to that described by Whitted. The intensity of each pixel is defined to be the integral of a lowpass filter over the rectangular regions within the filter's support. For a piecewise constant image representation this integration reduces to determining the volume of the filter across each region. For many simple filters, this value can be determined analytically. For more advanced filters, they suggest the use of

a “summed area table” for determining filter volume.

The most widely used reconstruction technique involves the application of *local filtering*. In local filtering the reconstruction, lowpass filtering, and resampling stages are combined into a single operation. This is accomplished by centering a filter over each pixel and applying the filter directly to the sampled values.

The simplest example of this technique is known as box filtering. In box filtering the intensity of a pixel is defined to be simply the average of the radiance of the samples taken within the pixel’s boundaries. Conceptually, this is the same process as centering a box filter over each pixel, summing the product of the filter and the sample values, and normalizing the result by the volume of the filter.

A number of more sophisticated filters have been proposed within the literature. Cook suggested a filter that is the difference of two Gaussians (DOG) [33]. Mitchell and Netravali developed a family of piecewise cubic filters [100]. They employed a panel of experts and an informal experiment to determine the best filter from within this family. This panel judged the quality of these filters based on the amount of ringing, blurring, and anisotropy present within a reconstructed image.

A normalized, raised cosine filter was employed by Dippé and Wold [40]. They suggested varying the filter width based on the amount of noise present within a region. They developed an equation for the filter width that limits the reconstruction error to within a given bounds.

Grain noise can be a significant problem when local filtering is applied to a set of nonuniformly spaced samples. This artifact is introduced when neighboring filter locations contain a widely varying number of samples. In this case, unwanted intensity fluctuations may be present within the reconstructed image. To solve this

problem, Mitchell suggested the use of a multi-stage filter [98]. Mitchell's multi-stage filter involves the repeated application of a box filter to the samples within a region. At each stage of the filtering process the box filter is doubled in width and applied to the results of the previous filtering operation. This technique is better able to handle variations in sampling densities since clumps of samples will be averaged locally before they are combined with neighboring values.

Lee observed that *nonlinear filters* are often able to better eliminate noise from images produced by stochastic sampling [80]. He suggested the use of a nonlinear, alpha-trimmed filter. The alpha-trimmed filter works by discarding the outlying samples within a region and averaging the remaining samples. In this manner very noisy samples will not affect the quality of the reconstructed image.

Rushmeier and Ward developed another nonlinear filter that is energy preserving [124]. They noted that a great deal of effort is expended in the calculation of each sample, and that all samples do contain valid information. Therefore, samples should not be discarded. They instead constructed a method whereby noisy samples are distributed over some region of influence. In their approach samples with outlying values are spread over a wider region than samples with consistent values. This technique preserves the hard-won sample information, but minimizes the presence of noise in the reconstructed image.

The reconstruction algorithms described so far operate solely within the spatial domain. There are, however, a number of advantages to developing reconstruction methods that operate within the *frequency domain*. Two of these advantages are that bandlimiting the image and interpolating between samples are trivial operations within the frequency domain. Bandlimiting can be accomplished by simply

discarding a number of high frequency terms and interpolation is easily achieved with an inverse Fourier transform.

Kim and Bose developed an algorithm that exploits the frequency domain in order to reconstruct an image from a set of nonuniformly spaced samples [78]. In their paper they describe a technique for building a transformation matrix that can map nonuniform samples within the spatial domain to uniform samples within the frequency domain. The final image is produced by performing an inverse Fourier transform on the frequency samples.

Sauer and Allebach developed a high quality reconstruction algorithm based on the technique of “alternating projections onto convex sets” [126]. This technique alternates between projecting the image onto the spatial and frequency domains. At each stage the error in the representation is minimized subject to a constraint. In the spatial domain the representation is constrained to fit the sample values. In the frequency domain the representation must satisfy a bandlimiting constraint. A relaxation technique is employed to converge to the nearest solution of these two antagonistic constraints. The frequency based reconstruction techniques proposed by Sauer and Allebach and by Kim and Bose both can produce an excellent reconstruction of the image. However, these particular frequency based techniques are generally too expensive to be commonly employed.

Summary

This chapter described the previous sampling and reconstruction techniques employed in the visibility tracing algorithm. The earliest such technique is to take a single sample at each pixel in order to determine its value. This approach can

severely under-sample the image and result in noisy or aliased images. Some authors have proposed reducing these artifacts by supersampling the image. However, this technique is inefficient because all areas of the image do not require the same number of samples to be accurately reconstructed. The inefficiency of supersampling has led to the development of a number of adaptive sampling algorithms.

Adaptive sampling algorithms can significantly reduce the number of samples and the total execution time required to generate a synthetic image. This is accomplished by varying the sampling rate in different regions of the image, so that each area receives the minimum number of samples required to refine it within a given tolerance. There are three primary components of any adaptive sampling algorithm. These are the base sampling pattern, the refinement test, and the reconstruction technique. This chapter has discussed the prior work in each of these areas.

The refinement test is the most important element of any adaptive sampling algorithm. This test is based on an error metric that estimates the image accuracy and governs the distribution of samples. Unfortunately, little attention has been paid to the nature of error metrics and the implications they have for constructing optimal sampling distributions. This issue is addressed in the following chapter.

CHAPTER V

AN ERROR METRIC FOR MONTE CARLO RAY TRACING

Understanding the nature of error has significant implications for image synthesis algorithms. It is therefore surprising that so little attention has been given in computer graphics to the theory of error metrics. An error metric describes the relationship that governs the accuracy of synthetically generated images. This metric allows the acceptability of the final renderings to be determined and controlled. Error metrics additionally provide important insights into the factors that affect the accuracy of the image. This information can be used to develop more efficient rendering algorithms.

In Monte Carlo ray tracing an image is produced by casting a number of rays through the pixels of the image plane. These rays are intersected with the objects of the scene and the light reflected from the surfaces is evaluated. The amount of light reflected from a surface is a function of the surface reflectance and the incident light from all other points in the environment. Monte Carlo ray tracing determines the amount of light incident on a surface by spawning a number of new rays from each surface intersection. These rays are similarly intersected with the objects in the environment and the process repeats.

Previous metrics for Monte Carlo ray tracing characterized the accuracy of an image based solely on the number of samples taken at the image plane. However, the quantity of samples spawned from each surface intersection has an equally significant

effect on the accuracy of the image. In order to develop an optimal ray tracing algorithm it is important to understand not only the effect of spawning on the accuracy of the image, but also whether the error is most efficiently minimized by varying the number of initial rays or by varying the number of rays spawned from each surface.

In this chapter a new error metric for Monte Carlo ray tracing is developed. This metric can be used to accurately characterize and control the presence of error within a synthetically generated image. The error metric that is presented is based on the number of rays taken at the image plane, as well as the quantity of rays spawned from each surface intersection. This allows the effect of spawning on the accuracy of the image to be precisely determined. This chapter additionally shows how this error metric can be used to determine the optimal number of rays to spawn at each surface intersection.

This chapter is divided into five additional sections. In the first section previous work concerning the spawning of rays and the categorization of error is discussed. Section two describes the construction of the new error metric. Sections three and four discuss how this error metric can be used to determine the optimal spawning rates. In section five an implementation is presented that was developed to illustrate and test the results of this theory. Finally, section six describes the results.

Previous Work

Monte Carlo ray tracing was introduced by Kajiya as a method for solving the rendering equation [76]. In his paper Kajiya first broached the issue of how many rays to spawn from each surface intersection. He noted that bushy ray trees were

inefficient because they spent too much effort sampling the depths of the ray tree where the contribution is smallest. He suggested path tracing as an alternative.

The number of rays to spawn from surface intersections was further explored by Arvo and Kirk [8]. In their paper they briefly introduced the concept of splitting (spawning many rays from each surface intersection) and Russian Roulette (probabilistically terminating rays). They also suggested that Russian Roulette be applied after the weight of a ray drops below some fixed threshold. Although they did not give any specific guidelines for the use of splitting, they did note that splitting can be more efficient than path tracing under certain circumstances.

The question of how to quantify the accuracy of computer graphic renderings has received surprisingly little attention within the literature. One work in this area is that of Arvo *et al.* [9]. In their paper they outline a general framework for the analysis of error. This discussion, however, mainly focuses on the radiosity method.

Lee, Redner and Usselton [81] discuss the accuracy of distributed ray tracing, but limit their analysis to the image plane. In their paper they show that the variance of the intensity estimate at a pixel is equal to the variance of the incident radiance divided by the number of samples taken within a pixel. The algorithm they describe makes use of this fact by computing a variance estimate in the image plane and employing it to determine when the accuracy of the pixel has been computed to a given threshold.

Error of Monte Carlo Simulations

In this section an equation is developed that describes the accuracy of Monte Carlo simulations. Most of this section and part of the following section are based

upon the work of Kahn [75] and Mikhailov [97]. In his paper Kahn outlined the basic Monte Carlo technique for a simple function of two variables. Mikhailov expanded this result to an arbitrary number of variables for splitting. This work extends their results to an arbitrary number of variables for Russian Roulette, allows the intermixing of splitting and Russian Roulette, and accounts for terminations.

Monte Carlo is an integration technique that is used to approximate the expected value (or mean) of functions. This is accomplished by randomly assigning values to the variables of a function in order to produce samples of the function. These samples are then averaged to determine the approximate expected value.

In order to describe the accuracy of Monte Carlo ray tracing let us first cast it in a functional form. The radiance of a sample in the image plane can be viewed as the evaluation of the function $f(x_0, x_1, x_2, \dots)$. In this expression x_0 represents the position of the sample in the image plane, x_1 indicates the direction of the ray spawned from the first surface intersection, x_2 specifies the direction of the ray spawned from the second intersection, and so on. In general, the number of variables present in this representation is given by the number of levels of recursion in the ray tree plus an additional variable for the image plane location.

The intensity of a pixel is given by the expected value $E[f]$ of this function across the pixel. In a Monte Carlo simulation this value \bar{f} is approximated by averaging N_0 random samples taken across the pixel:

$$E[f] \approx \bar{f} = \frac{1}{N_0} \sum_{i=1}^{N_0} f(x_0^{(i)}, x_1^{(i)}, x_2^{(i)}, \dots). \quad (12)$$

This technique is known as path tracing in the computer graphics literature [76].

Since Monte Carlo integration is inherently a random process, its accuracy can

be expressed in terms of variance. The variance $V[f]$ of a function is given by

$$V[f] = E[f^2] - E[f]^2. \quad (13)$$

The accuracy of the Monte Carlo approximation at a pixel is related to both the variance of the function that is being sampled and the number of samples:

$$V[\bar{f}] = \frac{1}{N_0} V[f]. \quad (14)$$

This well known expression states that the variance of the pixel intensity is equal to the variance of the function being sampled divided by the number of samples taken. This expression implies that if we know the variance of f , then the number of samples required to obtain a desired accuracy $V_d[\bar{f}]$ in the image plane is

$$N_0 = \frac{V[f]}{V_d[\bar{f}]} \quad (15)$$

The discussion thus far has assumed that a single ray is spawned from each surface intersection. However, what is the effect on the variance of spawning more or less than one ray from each surface? This is one of the primary questions that this chapter seeks to address.

In order to evaluate the effect of spawning at the various levels of the ray tree, we must first be able to express the overall variance in terms of the variance at each level of the tree. The expression that allows us to do this is

$$V[f] = VE[f|x_0] + EV[f|x_0]. \quad (16)$$

This expression states that the variance of a function is equal to the variance of the expectation of f given x_0 plus the expectation of the variance of f given x_0 . For Monte Carlo ray tracing, the first term can be thought of as the variance within a pixel given that the radiance is exactly computed at each location. The second term represents the average variance of the rays spawned from the first surface intersection. If we apply this expression recursively we have

$$V[f] = VE[f|x_0] + EV[E[f|x_0x_1]|x_0] + EV[E[f|x_0x_1x_2]|x_0x_1] + \dots \quad (17)$$

where $EV[E[f|x_0 \dots x_L]|x_0 \dots x_{L-1}]$ is given by calculating the expected value of f given variables $x_0 \dots x_L$, computing the variance of this expectation across all values of x_L with variables $x_0 \dots x_{L-1}$ fixed, and finally evaluating the expected value of this variance over all possible values of $x_0 \dots x_{L-1}$. More simply put, this is the expected variance incurred by the L^{th} variable in the function. This equation implies that for the case of path tracing, the variance of f is equal to the sum of the variance across the pixel and the variance incurred at each level L of the ray tree.

Splitting

The technique known as *splitting* in the Monte Carlo literature involves taking more than one sample of a variable for a single sample of a previous variable. The value returned is the average of these samples. For a simple function of two variables, this can be described as

$$f_S^{(i)} = \frac{1}{N_1} \sum_{j=1}^{N_1} f(x_0^{(i)}, x_1^{(j)}). \quad (18)$$

This process is analogous to spawning multiple rays from a surface for each incident ray. In this case the variance of the function given the first variable is reduced by a factor of $\frac{1}{N_1}$:

$$V[f_S|x_0] = \frac{1}{N_1}V[f|x_0]. \quad (19)$$

Only the last term in Equation 16 is affected by this process and the variance of f becomes

$$V[f] = VE[f|x_0] + \frac{1}{N_1}EV[f|x_0]. \quad (20)$$

Therefore the overall variance has been reduced, but at the expense of additional samples. This technique make sense when the $EV[f|x_0]$ is high compared to $VE[f|x_0]$.

Russian Roulette

Russian Roulette is an alternative technique which seeks to increase the efficiency of the simulation by killing off unimportant rays. Thus, for a given sample of a prior variable, we may with some probability decide to discontinue sampling and simply return 0. Otherwise, we sample the variable and increase the weight of the sample to avoid introducing bias. The new function can be represented as

$$f_{RR}^{(i)} = \begin{cases} \frac{f(x_0^{(i)}, x_1^{(i)})}{P_1} & \text{with probability } (P_1) \\ 0 & \text{with probability } (1 - P_1) \end{cases} \quad (21)$$

This has the effect of increasing the variance by a factor of $\frac{1}{P_1}$ plus an additional factor that is related to the square of the expected value of the ray:

$$V[f_{RR}|x_0] = \frac{1}{P_1}V[f|x_0] + \frac{(1-P_1)}{P_1}E[f|x_0]^2. \quad (22)$$

The net result on the variance of f is

$$V[f] = VE[f|x_0] - E[E[f|x_0]^2] + \frac{1}{P_1}(EV[f|x_0] + E[E[f|x_0]^2]). \quad (23)$$

Thus, the overall variance has been increased but the expense has been reduced. This technique makes sense when both $EV[f|x_0]$ and $E[E[f|x_0]^2]$ are small compared to $VE[f|x_0]$.

General Case

In general we would like to establish an expression for $V[f]$ that allows an arbitrary number of variables with either splitting or Russian Roulette performed at the various levels of the ray tree. This equation can be derived by recursively applying Equations 16, 19 and 22. This yields the result that

$$V[f] = D_0 + \frac{1}{N_1}D_1 + \frac{1}{N_1N_2}D_2 + \dots \quad (24)$$

where N_L specifies the splitting rate or the Russian Roulette (R.R.) probability at level L ,

$$D_0 = \begin{cases} VE_0 & \text{no R.R. at } L=1 \\ VE_0 - EE_1^2 & \text{R.R. at } L=1 \end{cases}$$

and for $L > 0$

$$D_L = \begin{cases} VE_L & \text{no R.R. at L or L+1} \\ VE_L + EE_L^2 & \text{R.R. at L, no R.R. at L+1} \\ VE_L - EE_{L+1}^2 & \text{no R.R. at L, R.R. at L+1} \\ VE_L + EE_L^2 - EE_{L+1}^2 & \text{R.R. at L and L+1.} \end{cases}$$

This expression has been simplified by assuming

$$\begin{aligned} VE_0 &= VE[f|x_0] \\ VE_L &= EV[E[f|x_0 \cdots x_L]|x_0 \cdots x_{L-1}] \quad \text{for } L > 0 \\ EE_L^2 &= E[E[f|x_0 \cdots x_{L-1}]^2] \quad \text{for } L > 0. \end{aligned}$$

This equation is significant because it provides a complete categorization of the variance of Monte Carlo ray tracing based on the variance of the environment and the sampling that is being performed at each level of the ray tree.

Optimum Splitting and Russian Roulette

The computer graphic literature has previously described the number of rays to spawn at a given level of the ray tree in a somewhat ad hoc fashion. The number of rays to spawn is usually described as one (path tracing) or many, where some number N has been determined to be good enough. When Russian Roulette is described it is usually applied after a fixed number of bounces or when the weight of a ray drops below some fixed threshold. As it turns out, we can do better.

In order to derive the optimal number of rays to spawn at a given level of the ray tree, we must first establish a quality metric. The quality metric described by

Kahn [75] and used in various places within the Monte Carlo literature is

$$Q = cost \times V[f]. \quad (25)$$

An algorithm is considered optimal if it produces a minimal Q . This means that it is producing the most cost effective variance reduction. Since the cost of casting a new ray at each level of the ray tree is essentially equal, the cost can be expressed as

$$cost = 1 + N_1 + N_1 N_2 + \dots \quad (26)$$

Deriving the optimal number of rays to spawn for a function of two variables is easily obtained by differentiating $cost \times V[f]$ with respect to N_1 , setting the result to 0 and solving for N_1 . The extension of this result to an arbitrary number of variables is provable by induction (see Appendix B). The result is that the optimal number of rays to spawn at any given level of the ray tree is described by the relation

$$N_L = \sqrt{\frac{D_L}{D_{L-1}}}. \quad (27)$$

The application of this formula is complicated by the fact that D_L and D_{L-1} are dependent on whether Russian Roulette is being performed at levels $L - 1$, L and $L + 1$. However, whether Russian Roulette or splitting should be applied and at what levels is exactly what we are trying to compute. A simple trick can be used to solve this problem. We start at level 1 and work toward increasingly deep levels. At each level we assume that Russian Roulette is being applied at this level and level

$L + 1$. If N_L is less than 1 we progress to the next level. However, if N_L is greater than or equal to 1 then we should not be doing Russian Roulette at this level. We then recurse backwards to level $L - 1$ and recompute N_{L-1} . If N_{L-1} was less than 1 and is now greater than 1 continue backwards. Otherwise, progress back up to level L computing the final N_L (Note: it is possible to have N_L less than 1 and yet have Russian Roulette be invalid. This can occur when the expected value is too large to allow Russian Roulette even though $VE_L < VE_{L-1}$. In this case N_L should be set to 1.).

Terminations

The preceding discussion assumed that rays from a given level of the ray tree always propagate to the next level. However, this is rarely the case. Rays can terminate at any level of the ray tree by striking light sources or escaping from the environment. This fact must be taken into account in order to derive the final expression for the variance and optimal spawning rates.

Terminations can be accounted for by observing that the variance added by the L^{th} bounce of the ray tree is zero for rays that terminate at level $L - 1$. Therefore, the expected value of the variance at level L taken over all rays at level $L - 1$ is equal to the percentage of rays R_L that reach level L from level $L - 1$ times the expected variance taken over only the rays that intersect a surface at level L . This relationship allows us to re-express Equation 16 as

$$V[f] = VE[f|x_0] + R_1 EV[f|x_0]. \quad (28)$$

where $EV[f|x_0]$ is the expected variance across the rays from x_0 which intersect a

surface at level 1 of the ray tree. The complete expression for $V[f]$ that accounts for terminations can be derived by recursively applying Equations 19, 22 and 28. This yields the result that

$$V[f] = D'_0 + \frac{1}{N_1}R_1D'_1 + \frac{1}{N_1N_2}R_1R_2D'_2 + \dots \quad (29)$$

where

$$D'_0 = \begin{cases} VE_0 & \text{no R.R. at L=1} \\ VE_0 - R_1EE_1^2 & \text{R.R. at L=1} \end{cases}$$

and for $L > 0$

$$D'_L = \begin{cases} VE_L & \text{no R.R. at L or L+1} \\ VE_L + EE_L^2 & \text{R.R. at L, no R.R. at L+1} \\ VE_L - R_{L+1}EE_{L+1}^2 & \text{no R.R. at L, R.R. at L+1} \\ VE_L + EE_L^2 - R_{L+1}EE_{L+1}^2 & \text{R.R. at L and L+1.} \end{cases}$$

In this expression the terms VE_L and EE_L^2 are calculated over only the rays which successfully propagate to level L of the ray tree.

The effect of terminations on the optimum splitting and Russian Roulette formula can be derived by noting that cost is reduced by the presence of termination. The new expression for the cost is

$$cost = 1 + N_1R_1 + N_1N_2R_1R_2 + \dots \quad (30)$$

The optimal number of rays to spawn at a given level can be derived by solving for

N_L that minimizes $cost \times V[f]$ as before. This yields the result that

$$N_L = \sqrt{\frac{D'_L}{D_{L-1}}} \quad (31)$$

where we have re-used the definition of D_L from Equation 24 with the modification that the terms VE_L and EE_L^2 are calculated over only the rays which successfully propagate to level L of the ray tree.

Implementation

An implementation was developed to illustrate and test the theory described in the preceding section. This algorithm uses the mean radiance across the spectrum as its primary quantity of interest. The basic approach of the algorithm is to use a set of pilot samples to estimate the optimal number of rays to spawn on a per-pixel basis at each level of the ray tree. Based upon the number of rays to be spawned, the variance in the image plane is estimated. This variance is then used to determine the number of rays that must be cast through each pixel in order to reach a specified tolerance. After this is accomplished the final sampling is performed using the optimal spawning rates. The net result is a method that can render an image to any specified tolerance using the most effective spawning rates at each level of the ray tree.

The algorithm receives as input the pilot sampling rate and target variance within the image plane. Because the algorithm must calculate the variance at each bounce of the ray tree, the pilot sampling rate must spawn many rays from each surface intersection. However, this number can usually be attenuated at the deeper levels of the ray tree since more rays are typically present at these levels.

As pilot samples are taken within each pixel, the variance of these samples is calculated to yield $V[f]$. Additionally, at each surface intersection the variance and the expected value squared is calculated for the rays spawned from the intersection point. This corresponds to the terms $V[f|x_0]$ and $E[f|x_0]^2$ when computed at the first level of the ray tree and $V[f|x_0 \cdots x_{L-1}]$ and $E[f|x_0 \cdots x_{L-1}]^2$ when computed at the L^{th} level of the ray tree. These values are averaged with the variance and expected values from all other surface intersections at the given level to yield $EV[f|x_0 \cdots x_{L-1}]$ and $E[E[f|x_0 \cdots x_{L-1}]^2]$. Additionally, during the pilot sampling stage the ratio of the number of rays spawned at level $L - 1$ to the number of rays that intersect a surface at level L is calculated to estimate R_L .

After all the pilot samples have been taken at a given pixel we can now estimate the VE_L term. The expression that allows us to do this comes from Equation 20. It is

$$VE_0 = V[f] - \frac{1}{N_1^P} R_1 EV[f|x_0] \quad (32)$$

and for $L > 0$

$$VE_L = EV[f|x_0 \cdots x_{L-1}] - \frac{1}{N_{L+1}^P} R_{L+1} EV[f|x_0 \cdots x_L] \quad (33)$$

where N_L^P is the pilot spawning rate at level L .

At this point we have all the necessary terms to calculate the optimum splitting and Russian Roulette rates at each level of the ray tree. This is done by using Equation 31. The variance at the image plane can then be calculated based on this new splitting rate using Equation 29. The number of initial rays necessary to reach a specified tolerance can then be derived from Equation 15. Finally, we cast the

specified number of initial rays, use the optimum splitting and Russian Roulette rates at each level of the ray tree, and compute the pixel intensity.

A few words should be said about the practicality of the method just described. In general it is as expensive to compute the necessary variances and expected values, as it is to compute the final radiance at a pixel. Therefore, this system was implemented primarily to illustrate and evaluate the theory described in the preceding sections. However, the practical benefits of this method can be seen if one considers that a large number of samples are necessary in any Monte Carlo rendering. Therefore, it makes sense to use a portion of the total number of samples to estimate the optimum sampling rates. These estimated rates can be used to bias the original guess toward the optimum rate, where the amount of bias should be based on the size of the pilot set.

Results

The mountain scenes illustrated in Figure 4 provide a good illustration of the nature of the optimum sampling and spawning rates. The images and the sampling rates were computed using the method just described. On the left of this picture the final rendered images of a high and low frequency fractal mountain are shown. The middle panels show the optimal number of initial rays to cast at each pixel. The images on the right show the optimal number of 1st bounce rays to spawn from the surface contained within that pixel. The color scale is linear, with fully saturated green indicating the point of maximal sampling.

It is interesting to note that the interior of the low frequency mountain requires very few pixel samples but should have many rays spawned from each surface inter-

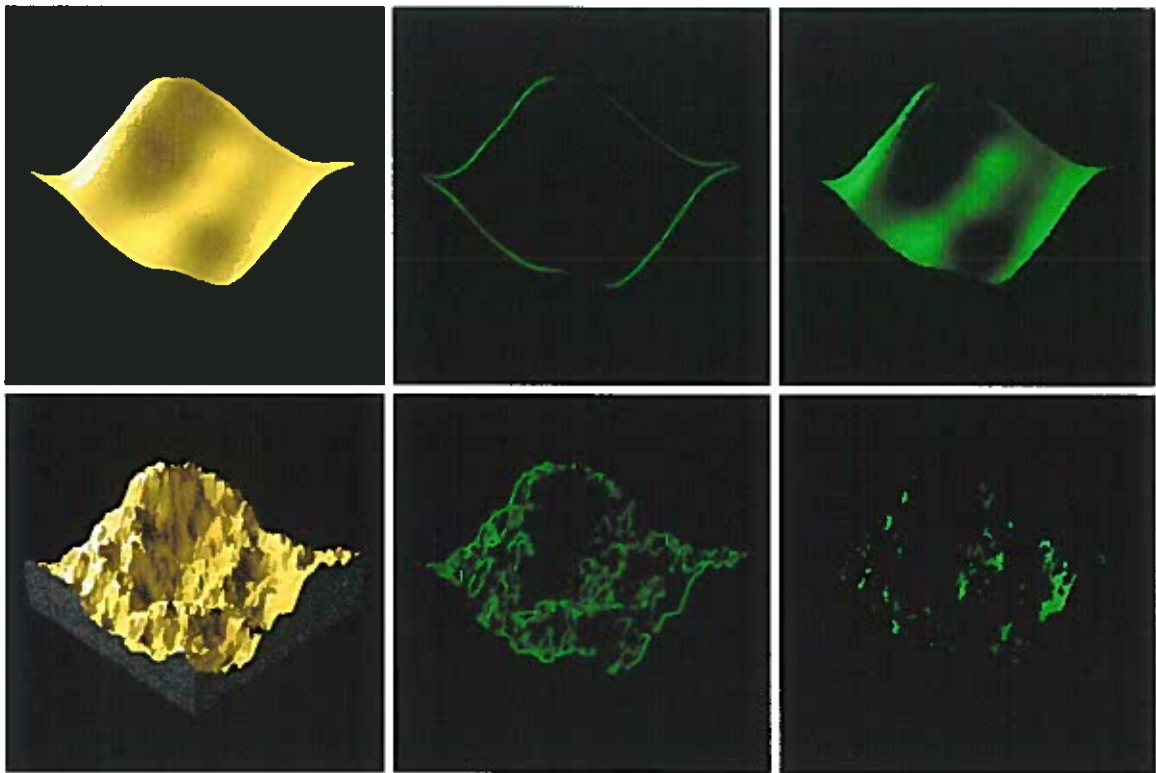


FIGURE 4. Low and high frequency mountains (left) with optimal initial (middle) and first bounce (right) sampling rates.

section. This is because the variance across the pixel (VE_0) is very small compared to the variance of the rays spawned from the surface (VE_1) (which either escape or hit a bright light source). This is the relationship described in Equation 31.

On the other hand, the high frequency mountain and the edges of the low frequency mountain have a higher variance within the image plane. This decreases the ratio of Equation 31 and reduces the optimal number of rays to spawn from the surface. Since there are less rays spawned from the surface and the variance in the image plane is large, the variance of the initial rays ($V[F]$) will be large and therefore require a greater number of rays to reach the specified tolerance. One may notice that there are areas of high surface spawning within the interior of the high

frequency mountain. This is the result of the fact that variances are calculated on a per pixel basis and regions of smoothness within the high frequency mountain are similar in nature to the interior of the low frequency mountain.

The radiosity box shown in Figure 5 serves to further illustrate the point. For this image the optimal splitting and Russian Roulette rates are illustrated. The Monte Carlo rendered radiosity box is on top. The middle panels from left to right indicate the optimal number of rays to cast in the image plane, on the first bounce, and on the second bounce. All values are relative to the pixel at which the ray tree is rooted. Increasing splitting and Russian Roulette is indicated by increasing green and cyan intensity respectively. The bottom panels have had a logarithmic contrast function applied to enhance the visibility of detail.

Within the panels we see that the sampling rate for the initial rays is highest around edges and predominantly around the bright light source. The greater sampling rate at the front edge of the light source than at the back occurs because the variance across a pixel depends on the fraction of the pixel covered by the light source. The middle panel illustrates the spawning rate at the first bounce. Since the variance across the image plane is low, this first bounce spawning rate is high on the floor and walls because they have a good chance of striking the light source. The panel on the far right is perhaps the most interesting. In this panel Russian Roulette is being applied on the floor and walls. This is because on the first bounce rays originating from these regions had a high variance, but on the second bounce, the variance and expected value of rays from these regions is beginning to decline due to attenuation and termination. Therefore, the ratio of Equation 31 is less than one and Russian Roulette is applied. The ceiling and areas that are in shadow, how-

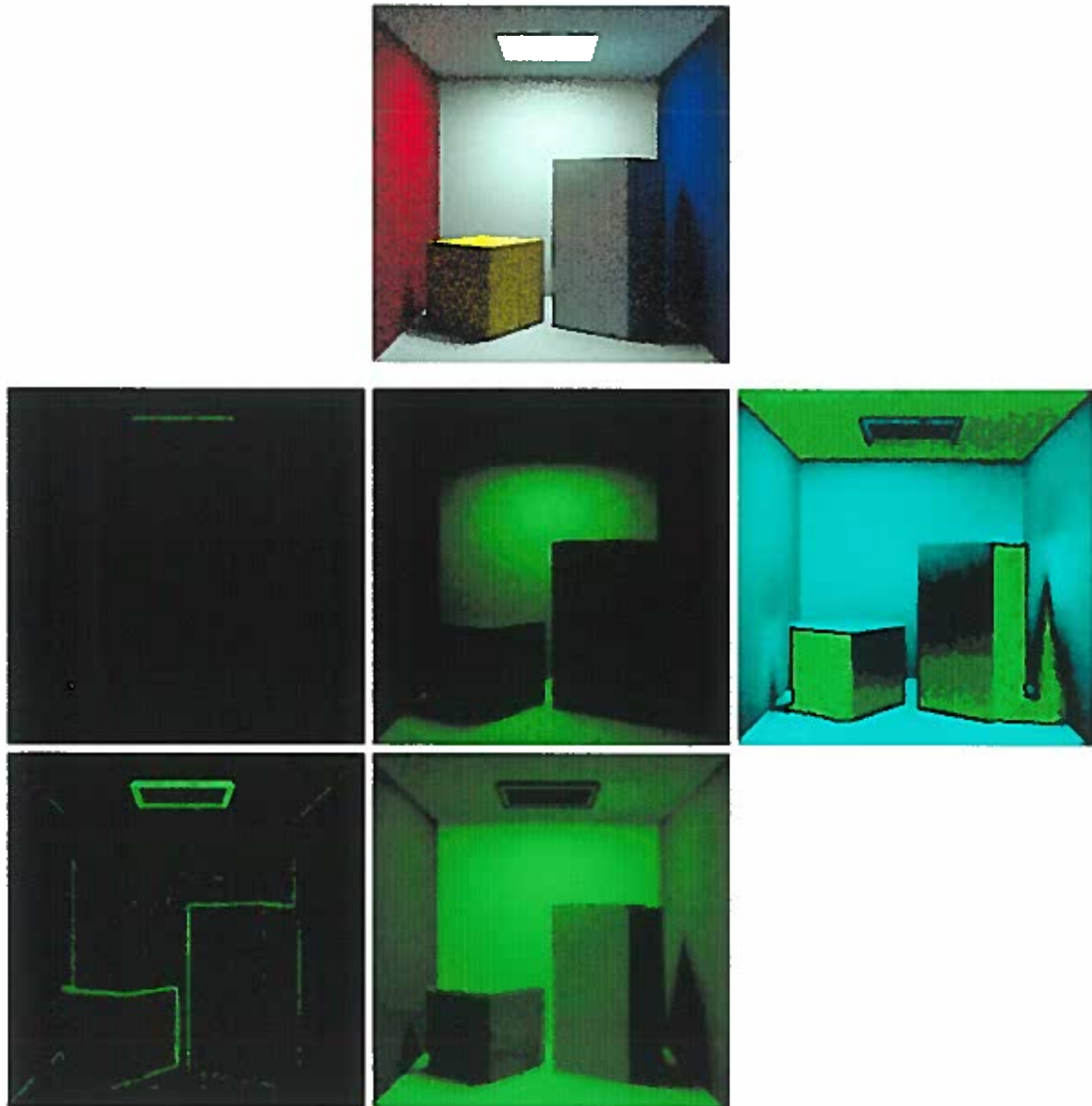


FIGURE 5. Monte Carlo rendered radiosity box (top row) with optimal initial, first, and second bounce sampling rates (middle row, from left to right). The bottom images are contrast enhanced initial (left) and first bounce (middle) rates.

ever, had no chance to hit the bright light source on their first bounce and therefore had a low first bounce variance. The rays that originated from these regions and are on their second bounce now have a chance to strike the light source. As a result the second bounce variance increases and splitting is applied.

Summary

This chapter presented a new error metric for Monte Carlo ray tracing. This metric describes a comprehensive expression that can be used to determine the accuracy of a synthetically generated image. The error metric that was presented is a function of both the number of initial rays taken at the image plane and the quantity of rays spawned at each level of the ray tree. This is a significant advancement over previous error metrics that are only based on the number of initial rays. By including the quantity of spawned rays within an error metric we can, for the first time, evaluate the effect of splitting and Russian Roulette on the accuracy of the image. This chapter has additionally shown how the accuracy of the image can be controlled by varying the sampling rate at the different levels of the ray tree.

Prior to this work, there has been very little information available on how to select the appropriate number of rays to spawn from a surface. This has led to algorithms which apply splitting and Russian Roulette in an ad hoc fashion. The current work has demonstrated how minimizing the amount of error with respect to a quality function can be used to determine the optimal number of rays to spawn at each level of the ray tree. This included the decision to perform either splitting or Russian Roulette. The efficiency of Monte Carlo rendering algorithms can be increased by employing these optimal spawning rates.

Adaptive sampling algorithms seek to control the local accuracy of an image by varying the sampling rate. However, this chapter has shown that accuracy is dependent on both the number of initial rays and the number of spawned rays. This raises the important question of whether an optimal adaptive sampling algorithm should control the accuracy by varying both the quantity of initial and spawned rays. The answer to this question comes from the theory discussed in this chapter. Equation 31 shows that the optimal number of rays to spawn at each level of the ray tree is, in fact, independent of the target variance of the image. Therefore adaptive sampling algorithms should always spawn the optimal number of rays at each level of the ray tree, and they should control the accuracy of the image by varying only the number of initial rays taken through the image plane.

An implementation was developed to illustrate and test the theory discussed in this chapter. Although this implementation is not intended to be practical, it does provide insight into how an optimal Monte Carlo ray tracer should perform. This chapter has also discussed how a more practical algorithm could be developed by using a portion of the early rays to estimate the optimal sampling rates. The estimated rates could then be used in the remainder of the simulation.

This chapter has presented a mechanism whereby the objective accuracy of an image can be determined and controlled. In the following chapter we will see a number of factors that make objective error a poor measurement of the perceptual quality of an image. However, understanding the nature of error and how to control it is a necessary step toward ultimately developing a rendering algorithm that can exploit the inability of the human visual system to perceive certain errors that exist in an image.

CHAPTER VI

PRIOR PERCEPTUAL IMAGE QUALITY MODELS

The two previous chapters have illustrated that error is an important quantity of interest in computer graphic applications. Chapter IV described a number of adaptive sampling algorithms that have been previously developed. These adaptive algorithms use an estimate of the local accuracy of an image in order to efficiently distribute samples at the image plane. Chapter V presented a new, comprehensive error metric and described how it can be used to categorize and control the accuracy of rendered images. These two chapters have shown how the accuracy of a synthesized image can be estimated, and how this estimate can be used to vary the sampling rate to efficiently minimize the error.

The error metrics that have been described in the previous two chapters are objective measurements of the accuracy of an image. Objective error metrics do not take into account how perceptible error is to a human observer. These metrics are instead only concerned with how variable the intensity (typically measured in RGB energy space) is at locations of the image plane. This approach to quantifying error neglects the fact that the images produced are ultimately intended to be viewed by a human observer. In this chapter previous work on quantifying the visibility of objective differences will be reviewed. These perceptual error measures are embedded in algorithms called image quality models.

Contrast Models

In order to determine the perceptual accuracy of an image, a number of image quality models have been developed by the image processing and vision research communities. In general, these metrics seek to quantify the difference between two images by processing each image in a manner similar to that done by the human visual system. The difference between the two images is calculated in the resulting perceptual space. The output of this operation is a measure of the visual difference between two images. The most simple image quality model is one which only takes into account variations in visual sensitivity as a function of brightness. This type of model is termed a *contrast model*.

A classic experiment has been developed for measuring contrast sensitivity over a range of background illuminations [127]. In this experiment the background brightness is set at a fixed level. The intensity of a small disk in the center of the field of view is raised above the background illumination until the two regions are just noticeably different. When this test is performed we get the results contained in Figure 6. In this figure the vertical axis represents the threshold luminance difference between the center and surround (ΔL), and the horizontal axis represents the background luminance of the surround (L). The results of the threshold study are plotted for both the rod and cone receptors found in the human retina. The response of the rods is normally associated with dark adapted vision, whereas the cone response is limited to bright adapted vision. The combination of the output of these two types of receptors yields the overall response characteristic of the visual system. The graph contained in this figure shows that the amount of contrast necessary to produce a perceptible difference increases as a function of background

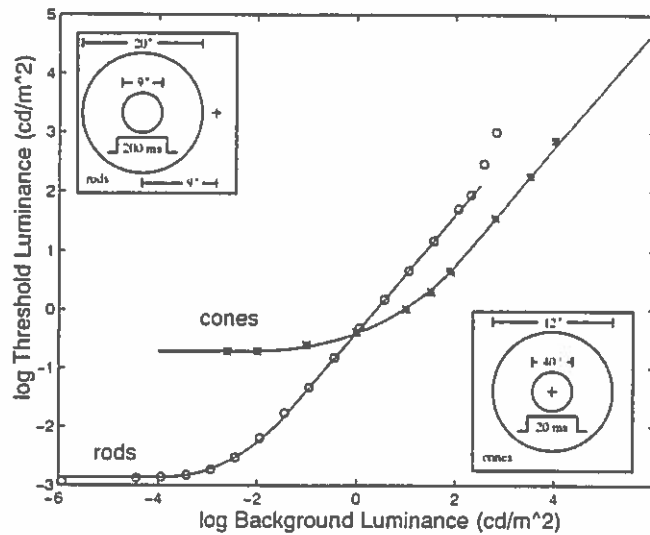


FIGURE 6. Contrast detection thresholds as a function of background luminance for both the rod and cone receptors. The inset figures show the dimensions and duration of the test patterns (from Ferwerda [43]).

luminance. At a background luminance above 2 log units this rate of increase ($\frac{\Delta L}{L}$) is constant and indicates a Weber law behavior.

There are two broad categories of contrast models: *point nonlinearity* and *non-point contrast* models. Point nonlinearity models apply a compressive function to the intensity values of individual pixels in an image. The difference between the pixel intensities of two images is measured after the application of the compressive nonlinearity. This approach captures the effect that differences are less apparent in bright regions than in dark. The disadvantage of this method is that it assumes that the eye can adapt at the resolution of a pixel; however, this model is simple to implement. Non-point contrast models take into account the fact that adaptation to background luminance levels generally occurs over a larger area than a single pixel. These models are somewhat more expensive to implement, but are closer to the true nature of the visual system.

Point Nonlinearity Models

A number of approaches have been taken to modeling the compressive point nonlinearity. A popular, early technique was to take the logarithm of intensity values. This method was used by both Stockham [138] and by Hall and Hall [67]. Power law functions have also been commonly employed. The most popular of these is the cube-root function as was done in Mannos and Sakrison [90].

Recently a number of more advanced point nonlinearities have been proposed. Daly [36] developed a contrast model which adapts the shape of the compressive curve based on the intensity of a pixel. At lower intensities Daly's model performs similarly to a cube-root model, whereas at higher intensities the response is more logarithmic. This technique avoids the prediction inaccuracy of the cube-root and logarithmic models at high and low intensities respectively. Pelli [113] took a very different approach to modeling contrast. He noted that "observers asked to detect or discriminate near-threshold contrasts act as though they are uncertain among many signals and choose the likeliest." This contrast uncertainty could be attributed to the random nature of photon absorptions and neural firing rates, as well as other sources of randomness within the visual system. He developed a contrast model based on a number of decision variables affected by Gaussian noise. A decision between these variables was made by a maximum-likelihood receiver. This metric was shown to have a good correspondence with the contrast response of the human visual system.

Non-Point Contrast Models

In the contrast experiment described at the start of this section, sensitivity is determined as a function of background luminance. The point nonlinearity models assume that background luminance is defined by the intensity of each pixel. However, the eye cannot adapt to arbitrarily fine resolutions. This is the reason for the development of non-point contrast models. These models utilize a larger area of the image plane in order to determine the adapted intensity of an observer.

The first such model is the ratio of Gaussians model developed by Zetzsche and Hauske [164]. In this model the input image is divided into a number of representations by applying a hierarchical set of Gaussian lowpass filters with successively lower cutoff frequencies. This representation is termed a Gaussian pyramid. A series of contrast images are then generated by dividing each Gaussian filtered image by the adjacent, lower frequency, Gaussian filtered image. This produces a set of contrast images, at varying scales, that exhibit a Weber law behavior.

A similar approach was taken by Lubin [88]. Lubin's model utilizes a Gaussian pyramid. However, contrast images are generated by first differencing neighboring levels of the pyramid before dividing by the image at a lower frequency level of the pyramid. This operation is equivalent to determining the magnitude of an edge at a given resolution and dividing that magnitude by the local image intensity. This approach is advantageous because it corresponds with the fact that the visual system is sensitive to edges rather than absolute luminance levels.

Fourier CSF Models

The next stage in the evolution of image quality models was to incorporate variations in perceptual sensitivity as a function of spatial frequency. The easiest technique for accomplishing this is to take a linear systems approach to modeling the human visual system. Fourier theory has shown that an arbitrary image can be represented by a sum of sinusoidal gratings at various frequencies. If a system is linear, then the response to an arbitrary function can be predicted by the sum of the responses to these component gratings. This is the approach taken by the visual metrics discussed in this section. These metrics utilize the sensitivity of the visual system to individual sinusoidal gratings in order to predict perceptual sensitivity to an arbitrary image.

In order to measure the visual system's sensitivity as a function of spatial frequency, a classic experiment has been developed. This experiment measures the magnitude at which a sinusoidal grating is just noticeably different from a constant background. A two alternative forced choice procedure is generally used in this experiment. The point at which the observer is able to correctly identify which of two intervals contain the grating, with a 75% probability, is identified as the threshold contrast for that grating. This experiment is repeated for a range of grating frequencies. Sensitivity is defined as one over the threshold contrast. A plot of sensitivity as a function of spatial frequency is given in Figure 7 for achromatic gratings. In this figure it can be seen that the visual system has a peak sensitivity at around 4 cycles per degree of visual angle, and that sensitivity declines at higher and lower spatial frequencies. This curve is known as the contrast sensitivity function.

This experiment was first introduced by Schade [127], and the results were

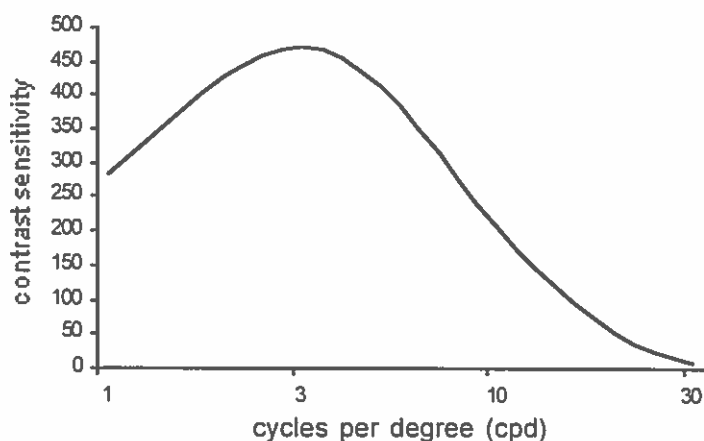


FIGURE 7. The achromatic contrast sensitivity function (plotted from the CSF model presented by Barten [10]).

incorporated into his photoelectric model of the visual system. In his system the lens of a camera was used to simulate the lens of the eye, photoconductive cells were used to model the receptors, and the output of the cells were fed into a computer in order to replicate higher level neural processing. Schade's approach was a very early model of the visual system, however, it included a number of important features, including contrast nonlinearity, contrast sensitivity, and a rudimentary color mechanism.

Schade's model incorporated the contrast sensitivity function by performing a Fourier transform of the input image. The magnitude of the Fourier coefficients were then weighted by the contrast sensitivity of the associated spatial frequency. A similar technique has been used in a number of other image quality models [63, 90, 67, 10].

A problem with the linear system approach to vision modeling is the presence of the contrast nonlinearity. In order to incorporate the sensitivity variations that occur due to both illumination level and spatial frequency, most authors apply a

contrast nonlinearity followed by a weighting by the contrast sensitivity function. This is the approach taken by Schade and by Mannos and Sakrison [90]. The presence of the nonlinearity, however, violates a basic assumption of linear systems. This violation is merely accepted by most authors. Another approach is to eliminate the contrast nonlinearity, as was done by Granger and Cupery [63]. However, this neglects a significant aspect of the visual system. Hall and Hall [67] suggested that the impact of this nonlinearity could be minimized by dividing the contrast sensitivity function into its optical and neural components. The optical component was modeled with a lowpass filter that simulated the defocusing of light caused by the optics of the eye. The neural component is incorporated by a highpass filter that models the effect of lateral inhibition found in the neural pathway. The contrast nonlinearity was placed between these two components. This approach is intuitively appealing since it follows the structure of the visual system (i.e. optics, receptors, neural pathway).

Spatial Frequency Channel Models

The models described in the previous section assume that the visual system is able to perform some type of Fourier decomposition on an input image. This allows the perceptual response to an arbitrary image to be exactly determined by the cumulative response to component gratings. There is now significant evidence that this is not the case. A number of experiments have shown that there exists a limited number of discrete channels in the visual pathway which are tuned to bands of spatial frequencies, and that the contrast sensitivity function is determined by the cumulative responses of these channels.

The first evidence of the existence of channels was presented by Campbell and Robson [25]. They tested the sensitivity of the visual system to square wave gratings. They found that the detectability threshold for a square wave was different from that predicted by a linear systems model. This threshold was instead determined by the threshold of the strongest sinusoidal component of the square wave. This led them to speculate that various sinusoidal components are encoded in discrete channels, and that a threshold response of an individual channel is necessary to achieve a threshold response for the aggregate image. Similar results were found for compound gratings by Graham, Robson, and Nachimias [62].

Further evidence of the existence of spatial frequency selective channels was presented by Blakemore and Campbell [12]. They found that when a subject viewed a grating of a given spatial frequency for a prolonged amount of time, the subject's sensitivity to neighboring spatial frequencies decreased. This would occur if the neighboring frequencies were encoded within the same spatial frequency channel. The final evidence of the existence of channels comes from the masking experiments of Legge and Foley [82]. Masking describes the ability of one signal to interfere with the detectability of another signal. As with the adaptation effect explored by Blakemore and Campbell, masking was found to occur over a band of neighboring spatial frequencies.

Spatial frequency channels were first incorporated into image quality models by Carlson and Cohen [26], and by Wilson and his colleagues [159, 162, 160]. There is still an ongoing debate over the shape and number of channels present in the human visual system. A recent survey by Ahumada [1] lists quality models containing from 3 to 12 spatial frequency channels. As an example Figure 8 illustrates the frequency

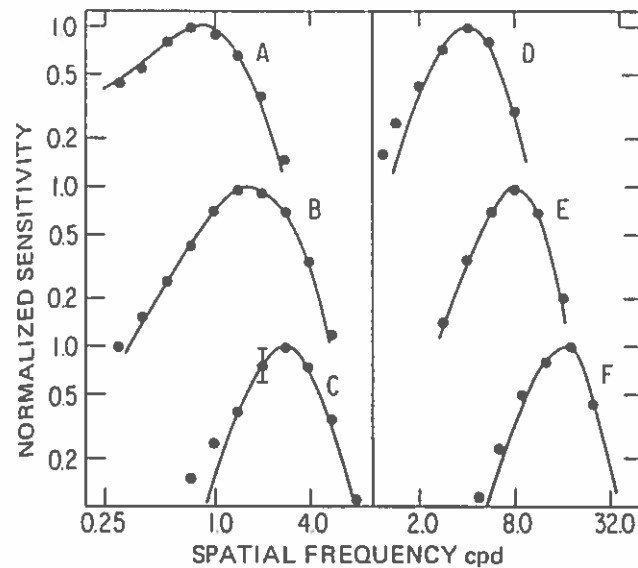


FIGURE 8. The sensitivity of the six spatial frequency selective channels employed by Wilson and Gelb [160].

selectivity of the model proposed by Wilson and Gelb [160]. This metric contains six channels with center frequencies of 0.8, 1.7, 2.8, 4.0, 8.0, and 16.0 cycles per degree.

The typical structure of an image quality model containing spatial frequency channels is as follows. First, the input images are processed by a contrast nonlinearity. Then each image is decomposed into a number of channel images. This is often accomplished by applying a series of spatial filters, with varying frequency selectivity, to the input images. For example, Wilson and Gelb used filters which were the difference of two or three Gaussians. These channel images isolate contrast as a function of spatial frequency. The channel images are then weighted so that the responses of individual channels combine to give the response characteristic of the contrast sensitivity function. Finally, the differences between the channel representations of each of the two input images are combined to yield a measure of the

perceptual difference between the two images. More detail on how these differences are combined will be given in pages 84–86 of this chapter.

Orientation Channel Models

The previous section described quality models that contain a number of spatial frequency selective channels. There is also good evidence that there exist channels in the visual pathway that are tuned to a number of discrete orientations. The models described in this section augment the spatial frequency channel approach with an additional orientation selective component.

The evidence for the existence of orientationally selective channels comes from both adaptation and masking experiments. Adaptation experiments were performed by Campbell and Maffei [24] and by Bradley *et. al.* [19]. In these experiments viewers fixated on sinusoidal gratings of a given orientation. After the fixation period, gratings of differing orientations were presented to the user and the discrimination threshold for these gratings were measured. The results of this experiment showed that adaptation to a grating of a given orientation reduced sensitivity to neighboring orientations. This effect decreased as the differences in orientation increased.

A similar result was found in masking experiments by Campbell and Kulikowski [23] and by Phillips and Wilson [114]. The orientation channel bandwidths predicted from masking experiments are generally somewhat larger than those predicted from adaptation experiments. As with spatial frequency channels, there is still debate as to the appropriate number of orientation channels. Visual metrics have been developed that contain anywhere from 3 to 12 channels spanning the 360 degree range [1].

There have been a number of popular transforms developed that decompose an image into component spatial frequency and orientation selective channels. The first such method is the cortex transform presented by Watson [153]. This method first converts an input image into the frequency domain via the fast Fourier transform. Then component images are generated by applying a set of spatial frequency and orientation selective filters. These filters are defined as the product of difference-of-mesa filters and fan filters. The difference-of-mesa filters isolate spatial frequency bands and the fan filters isolate ranges of orientations. This transform has been successfully incorporated into image quality models by Watson [152] and Daly [36].

Another successful transform has been made by the combination of a Laplacian pyramid [22] and a set of steerable filters [48, 77]. This transform operates in the spatial domain. The first stage of this transform is to decompose an input image into a pyramid of new images, by applying Gaussian lowpass filters. These filters have successively lower frequency cutoffs. Images at neighboring levels of the pyramid are then differenced to yield a new set of images that contain only a specific band of spatial frequencies. Finally, these bandpass images are used to generate a number of orientation selective images by applying filters that are the directional derivatives of Gaussians. This transform is used in Lubin's [88] visual difference predictor.

A number of other similar transforms have been proposed. The Gabor transform is one popular alternative [151, 104, 86]. Zetsche and Hauske [164] suggested a ratio-of-Gaussians pyramid combined with oriented filtering. Watson *et. al.* [154] have utilized the discrete cosine transform. The important commonality of all these models is that they mirror the spatial frequency and orientation selectivity of the human visual system by decomposing the input image into a number of discrete

channel representations.

Masking Models

The next stage in the development of image quality models was to incorporate the effect of masking. The first metric to include the masking effect was created by Carlson and Cohen [26]. This model was introduced shortly after the development of the first spatial frequency channel model. Masking describes the ability of one signal to improve or interfere in the detectability of another. This section will discuss the masking effect and the models that include it.

Legge and Foley [82] presented the basic experiment used to study masking. In this experiment an observer is asked to discriminate between two images. The first image contains only a sinusoidal grating, which is called the mask. The second image contains the mask superimposed with an additional sinusoid called the test. The experiment measures the magnitude at which the mask plus the test is discriminable from the mask alone.

The results of Legge and Foley's study are plotted in Figure 9. The horizontal axis indicates the magnitude of the mask and the vertical axis represents the effect on the detectability of the test. This figure contains the data for a number of masking experiments plotted on an arbitrary scale. The point at which each curve intersects the vertical axis represents the magnitude of test required for discrimination without the presence of the mask. This data shows that low mask contrasts facilitate the detectability of the test (indicated by the "dipper" portion of the curve), and that high mask contrasts can significantly interfere with the perceptibility of the test (indicated by the elevated right side of the curve). This effect is strongest when

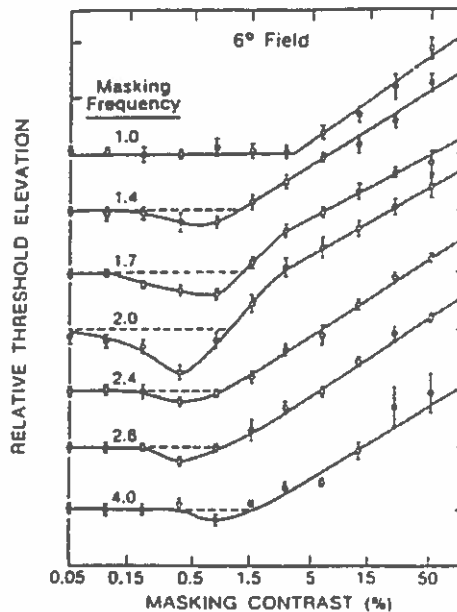


FIGURE 9. Threshold elevation due to masking. In this experiment the “test” grating is a 2.0 cycle/degree sinusoidal. The individual curves show the results for different “mask” frequencies. Each curve is plotted on an arbitrary scale to allow presentation on a single graph. The dotted line indicates the unmasked threshold (from Legge and Foley [82]).

the orientation and spatial frequency of the mask and the test is the same. This can be seen for the case of spatial frequency by comparing the center curve to those above and below. This orientation and spatial frequency selectivity has been cited as evidence for the existence of channels in the previous two sections. The masking effect is significant because it describes how image content affects the visibility of contrast.

There are two primary approaches for incorporating masking into a quality metric. The first method uses the threshold elevation curve illustrated in Figure 9, and the second method utilizes a nonlinear transducer. These methods are generally incorporated after processing by a contrast nonlinearity, decomposition into spatial frequency and orientation channels, and a weighting by the contrast sensitivity func-

tion.

The first technique assumes that threshold experiments can be used to describe the magnitude of the masking effect at all levels, threshold or otherwise. In this approach the contrast difference between an identical channel of the two input images is divided by the amount of threshold elevation. One image is selected as the mask and the other as the mask plus the test. The result is that contrast between the two channels is reduced when the intensity of the mask is high, and increased when the intensity of the mask is low. A number of image quality models have been developed that utilize this technique [162, 36, 154].

The second approach to modeling masking is to use a sigmoid-like nonlinear transducer. This transducer is employed to modify the channel response for each of the input images. The response of a channel is augmented for low contrasts and compressed at high contrasts. The net result is, once again, that differences between channels of the two input images is decreased when the contrast of the bands is high and increased when the contrasts of the bands is low. This is a very similar result to that achieved with the threshold elevation technique. However, this method does a better job of modeling suprathreshold differences. Masking transducers have been utilized in quality metrics by Legge and Foley [82], Watson [152], Zetzsche and Hauske [164], and Lubin [88].

Distance Summation

The final stage of an image quality model is to accumulate differences between the two input images. This stage is equivalent for spatial frequency channel models, orientation selective models, and masking models. For this reason, difference

summation has been deferred until this section.

These three types of models share the idea that the perceptual response to an image is given by the magnitude of a number of neural images called channels. The amplitude of these neural images, at a given location, can be thought of as the response of a detector in the visual pathway. These detectors are tuned to bands of spatial frequencies and orientations, and their output has been modified by a number of levels of visual processing. The visual system is assumed to determine the difference between two images based on changes in the outputs of these detectors. This section describes how variations in the response of these detectors are accumulated across channels to yield the aggregate perceptual difference.

The majority of visual metrics combine differences in detector output with Quick's vector summation rule [119]. This method calculates the total perceptual difference (ΔP) between two images via the expression

$$\Delta P = \left(\sum^c (P1_c - P2_c)^Q \right)^{1/Q}. \quad (34)$$

In this equation $P1_c$ and $P2_c$ indicate the response of the detector in channel c for the first and second image respectively. The variable Q is a free parameter, and there is some debate over the appropriate choice for this variable. If Q is set equal to 1 then differences in the detectors combine additively. However, if Q is set to infinity then the total perceptual difference is based solely on the detector with the largest delta. This second method was proposed in a model by Legge and Foley [82]. It assumes that the channels do not interact in any way and that the response of each detector is measured independently by the visual system. Most authors take a more conservative approach. Setting Q equal to 2 is one popular choice [164, 160, 161].

This equates to using an Euclidean norm. A number of authors have suggested using a Q of 4 [159, 162, 92]. This implies a greater degree of independence between the detectors. Finally, Lubin [88] proposed the compromise that Q should be set to 2.4.

A limited number of alternate methods have been described to combine channel responses. Daly [36] presented a probability summation technique that utilizes a psychometric function to map changes in channel outputs onto a 0 to 1 range that indicates the probability that the difference is detectable. A product rule is then used to combine detection probabilities. Another alternative technique was described by Watson [151]. This method is based on the use of an optimal Bayesian classifier.

Color Models

The quality models described in the preceding sections have focussed on the detectability of differences in achromatic images. There has been far less effort devoted to the development of color image quality models. This section will discuss the preliminary work that has been done in this area.

A very simple technique for handling color was included in a quality model by Wilson [158]. The first stage of this model decomposed the input image into the responses of the three different classes of cone photoreceptors found in the human retina. These three types of receptors are respectively selective of the short, medium, and long wavelengths of light. This stage of the model was followed by a contrast nonlinearity that was applied to each color channel ¹ to simulate each class

¹The wavelength selective bands used to represent a color image are commonly referred to as channels. However, the term channels is generally also used to refer to bands of spatial frequencies and orientations. These concepts are similar. In order to avoid confusion this section will use the

of receptor's nonlinear response to light. Finally, a Euclidean distance metric was used to collapse the color channels into a single achromatic channel for processing by the remainder of the visual metric. Wilson's model can be primarily viewed as an achromatic model. However, it does include the important stages of receptor processing and their subsequent contrast nonlinearity.

The concept of an opponents color representation was first introduced in Schade's visual model (although it was not significantly used) [127]. There is now abundant evidence that the responses of the receptors are processed into an opponents representation in the upper levels of the visual pathway [21, 147, 38]. This opponents representation is made up of a single achromatic and two opponent color channels. The two color channels are commonly referred to as the red/green and yellow/blue channels, since these colors roughly correspond with the opposite ends of their spectra.

The human visual system's contrast sensitivity response, as a function of spatial frequency, is significantly different for the chromatic channels than it is for the achromatic channel. This fact was demonstrated by the contrast sensitivity experiments of Mullen [101]. Figure 10 plots the results of these experiments for the achromatic and chromatic channels. In this figure it can be observed that while achromatic sensitivity is bandpass, chromatic sensitivity is lowpass, with a lower peak sensitivity and a lower spatial frequency cutoff.

This fact was exploited in image quality models by Lloyd and Beaton [86] and by Martin *et. al.* [92]. These models include a transformation into an opponents color space, followed by a decomposition into spatial frequency and orientation selec-

word "channel" to describe wavelength selectivity and the word "mechanism" to describe spatial frequency and orientation selectivity.

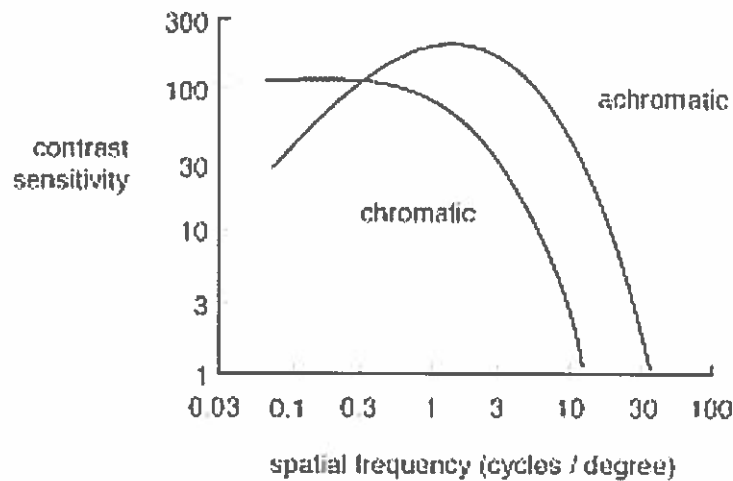


FIGURE 10. Achromatic versus chromatic (either red-green or yellow-blue) spatial frequency response (after Mullen [101]).

tive mechanisms. This decomposition is performed for each of the opponent channels and corresponds with data that indicates that there are chromatic detectors selective of both discrete spatial frequency and orientation bands [19, 87]. In Lloyd and Beaton's model the lower chromatic sensitivity is exploited by utilizing less spatial frequency selective bands for the chromatic than for the achromatic image representation. Martin *et. al.*'s model employs the same number of spatial frequency mechanisms for all channels, but weights the responses of these mechanisms to correspond with the chromatic and achromatic contrast sensitivity functions.

The differences in chromatic and achromatic sensitivity have also been exploited in the recently proposed spatial CIELAB metric [165, 166, 168]. The CIELAB color space is an international standard. Differences in this color space are commonly used to measure color reproduction errors. However, this space was developed based on color appearance judgements of large uniform fields. Spatial CIELAB is an attempt to incorporate spatial variations in perceptual sensitivity into this metric. This approach accounts for differences in spatial frequency sensitivity, but does not

account for the effect of masking. This algorithm proceeds by first transforming the image into an opponents color representation. After this stage, each color channel is lowpass filtered with differing width spatial filters. This process is somewhat analogous to a spatial application of the contrast sensitivity function. The least filtering is applied to the luminance channel and successively lower pass filtering is applied to each of the red/green and yellow/blue channels. The fact that lower frequency filtering is used in the yellow/blue than in the red/green channel models the visual system's lower sensitivity to this channel due to axial chromatic aberration [91]. This result did not appear in Mullen's original data because she corrected for this effect.

The lower spatial frequency cutoff of the yellow/blue contrast sensitivity function has also been incorporated into a quality metric by Faugeras [42]. This metric uses a Fourier CSF model with an additional color processing element. It includes an initial processing by the receptor sensitivities, a logarithmic contrast nonlinearity, a matrix to an opponent color space, and a final spatial filtering that accounts for contrast sensitivity response. An increasing lower frequency cutoff is used for the spatial filtering that is applied to each of the achromatic, red/green, and yellow/blue channels.

Ferwerda *et. al* [45] presented a color image quality model that included the effect of masking. The initial portion of this model utilizes the receptor processing, contrast nonlinearity, and opponent color space developed in the work discussed in Chapter IX. These stages are followed by a decomposition into spatial frequency and orientation tuned mechanisms for each color channel. Finally, a threshold elevation technique is used to incorporate the masking effect. This technique is applied to

the color channels exactly as it has been described for the achromatic channel. This work draws upon the results of Losada and Mullen [87], and of Gergenfurth and Kipper [50] that indicate that equivalent masking effects happen in the achromatic and chromatic channels, and that cross-channel masking does not occur.

Summary

This chapter reviewed previous research on perceptual image quality. This work has produced a number of visual quality models. These models are used to predict the magnitude of visual differences between two input images.

The most recent quality models contain a number of important steps. The first step is to take into account variations in sensitivity with image brightness. This is accomplished with either a point nonlinearity or a non-point contrast metric. This stage is followed by an image decomposition into spatial frequency and orientation tuned mechanisms. The responses of these mechanisms are scaled by their associated contrast sensitivity response. This incorporates differences in perceptual sensitivity as a function of spatial frequency. The affect of masking is then simulated using either a threshold elevation technique or a nonlinear transducer. This stage models sensitivity variations with signal content. These steps are generally applied to each of two input images and differences in the resulting perceptual space are accumulated with a distance summation. This yields the final measure of the visual difference between two images.

A few quality models have been created for color images. The important stages in these metrics are a processing by the short, medium, and long receptor sensitivities, a conversion to an opponents color space, and a processing by the

differences in achromatic and chromatic spatial frequency response. These elements are incorporated in varying degrees within previous color image quality models.

Quality models provide a method to map objective accuracy measurements to subjective image quality. As has been shown in preceding chapters, adaptive sampling is a powerful technique for improving the performance of rendering algorithms. Prior adaptive algorithms are based on objective measurements of the accuracy of the reconstructed image. The use of objective accuracy metrics limits the benefits of adaptive algorithms by neglecting the fact that the images are ultimately intended to be viewed by a human observer. An image quality model provides the means to incorporate the response of a human observer into these adaptive sampling algorithms.

CHAPTER VII

A HIGH SPEED COLOR VISUAL DIFFERENCE PREDICTOR

The preceding chapter reviewed the prior image quality models that have been developed by the image processing and vision research communities. These models provide a means to convert from the objective error measurements used in existing adaptive sampling algorithms, to subjective measurements of perceptual quality. This is the technique that will be used in this dissertation to ultimately develop a rendering algorithm that distributes samples based on the visibility of error in a synthesized image. The current quality metrics provide a good starting point for such an algorithm. However, they are limited in a number of areas.

Previous image quality models have gone to great lengths to accurately simulate the perceptual sensitivity of the human visual system. Unfortunately, efficiency is seldom a design criteria in developing these systems. This limits the utility of these algorithms in applications where speed is a primary concern. The incorporation of a visual model within an adaptive sampling algorithm will require that the metric be used to make many iterative judgements of the quality of an image during the progression of the algorithm. If the perceptual metric takes too long to execute then the gains achieved by the more efficient sample distribution will be offset.

Another weakness of the existing visual quality models is their limited ability to accurately predict differences between color images. The majority of previous models have been designed to only handle achromatic images. The few metrics

that do include color processing have generally neglected the significant effect of chromatic aberration.

In this chapter a new, high speed, color image quality model will be presented. This model will be shown to execute in a fraction of the time required by existing metrics, without a significant loss in accuracy. This new model is also designed to handle color images, and it includes the effect of chromatic aberration.

Model Description

The last chapter described a number of perceptual quality models. Two of the most comprehensive are the Visual Difference Predictor (VDP) by Daly [36] and the Sarnoff Visual Difference Metric (VDM) by Lubin [88]. A recent study by Li [83, 84] compared the results of these two metrics. In this study it was found that although the Sarnoff VDM required somewhat more memory, it executed faster and produced better difference predictions. Another advantage of the Sarnoff model is its use of a pyramidal transformation to isolate spatial frequency and orientation selective channels. The nature of this type of transform offers substantial efficiency benefits as will be seen in the adaptive sampling algorithm described in Chapter X.

For these reasons, the Sarnoff VDM was selected as a starting point for the development of the quality model discussed in this section. The new model has been modified to run efficiently, and it has been extended to handle color. This color extension is necessary because the original Sarnoff metric was only designed for achromatic images.

The perceptual metric that will be described has been imbedded into a visual difference predictor. This predictor receives as input two images represented in CIE

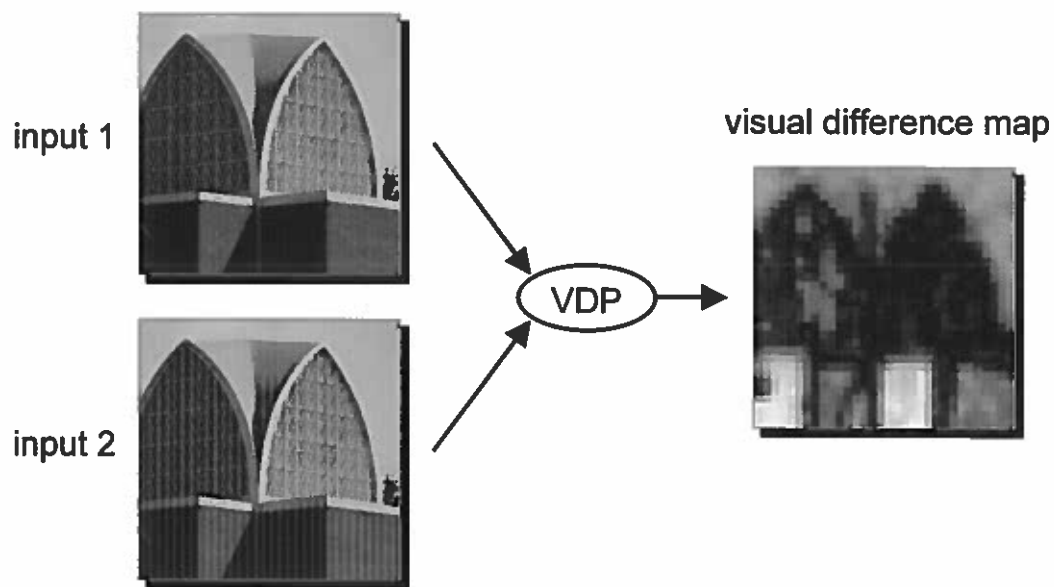


FIGURE 11. Input and output of the visual difference predictor.

XYZ color space. It returns as output a map, specified in terms of just noticeable differences (JND's), of the perceptual difference between the two images. One JND corresponds to a 75% probability that an observer viewing the two images would be able to detect a difference, and the units correspond to a roughly linear magnitude of subjective visual differences [88].

The input and output of this predictor are illustrated in Figure 11. In this example input 1 contains a chapel image, and input 2 is the same image distorted by an equal energy sinusoidal grating. It should be apparent that while the grating is uniform, its perceptibility is not. The distortion is most visible in the dark areas at the base of the chapel and less perceptible in the bright regions at the top of the image. The grating is also completely invisible inside the upper right archway because the lattice work in this area hides, or masks, the detectability of the grating. The output of the predictor is shown in the visual difference map on the right side of the figure. This image utilizes increased brightness to indicate areas

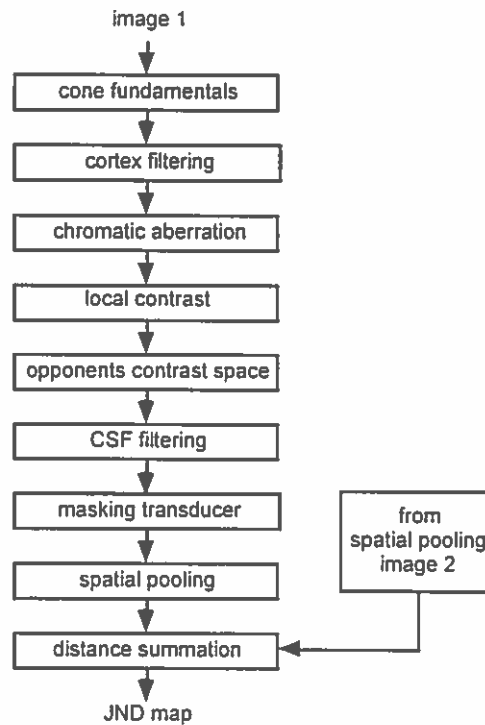


FIGURE 12. Block diagram of the visual difference predictor.

with more perceptible differences. The difference map can be seen to have a good correspondence with a subjective comparison of the two inputs.

A block diagram of the visual difference predictor is given in Figure 12. This diagram illustrates the various stages of processing that are involved in this predictor. The steps *cone fundamentals* through *spatial pooling* are carried out independently on both input images. The differences between the two images are accumulated in the *distance summation* step.

In the first stage of the vision model entitled *cone fundamentals*, the pixels of the input image are encoded into the responses of the short (S), medium (M) and long (L) receptors found in the retina of the eye. Figure 13 illustrates this process. The input image is on the left, the spectral sensitivities of the short, medium, and long receptors are drawn in the center graph, and the resulting receptor response

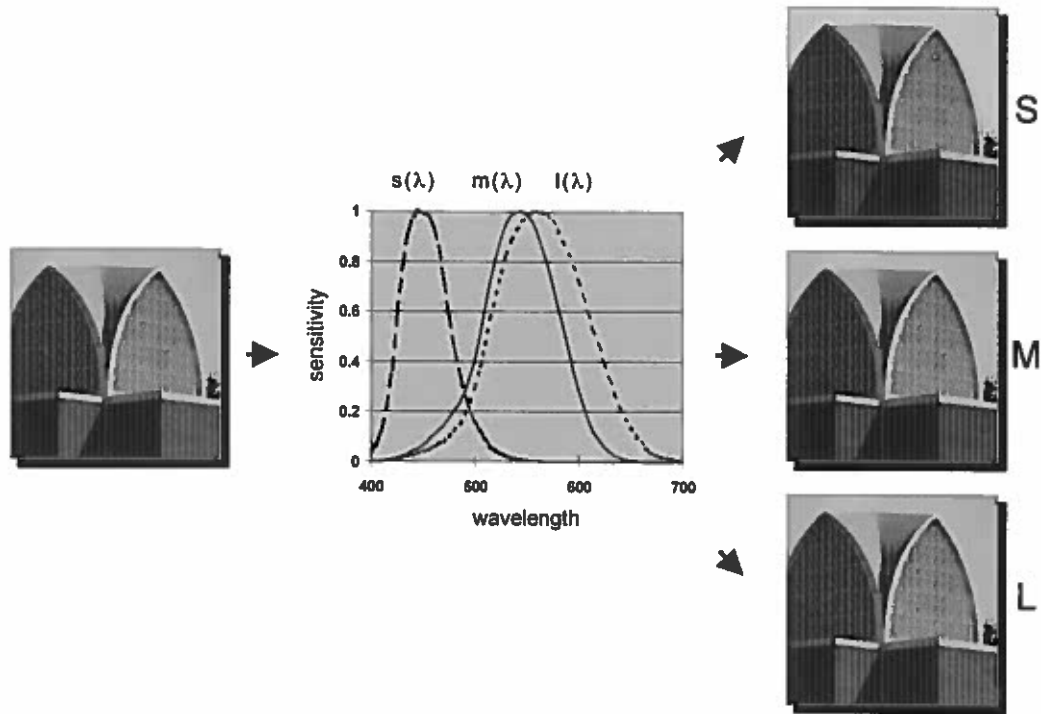


FIGURE 13. Flow graph of the *cone fundamentals* stage of the visual difference predictor.

images are on the right. This transformation converts from CIE XYZ to SML space via the following matrix equation [94]:

$$\begin{bmatrix} S \\ M \\ L \end{bmatrix} = \begin{bmatrix} 0.0000 & 0.0000 & 0.5609 \\ -0.4227 & 1.1723 & 0.0911 \\ 0.1150 & 0.9364 & -0.0203 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (35)$$

The next step in the model is to apply a *cortex filtering* operation. This process decomposes the receptor images into the responses of spatial frequency and orientation tuned mechanisms present in the visual pathway. The evidence that indicates the existence of such mechanisms was presented in the previous chapter.

The decomposition of an image into spatial frequency and orientation tuned

channels is the most expensive operation performed by a visual model. Therefore, in order to significantly improve the execution time of a model, a high speed transform must be selected. The choice of this transform should also be influenced by the desire to incorporate the quality model within an adaptive sampling algorithm. During the progression of an adaptive algorithm it is necessary to make numerous, iterative judgements about the quality of an image. Successive judgements are often made after modifying only a small region of the image. It would therefore be advantageous if small image modifications had a limited effect on the cortex representation, and if this effect could be rapidly calculated.

In order to satisfy these requirements, the Haar wavelet transform was selected to model the spatial frequency and orientation selectivity of the human visual system. This transform provides the fastest mechanism capable of decomposing an image into these selective channels. The Haar transform can be computed in $O(N)$ time, and as will be shown in Chapter X, it can be updated in $O(\log N)$ time during the progression of an adaptive sampling algorithm.

A number of other transforms were considered for this stage of the model. The cortex transform by Watson [153] was one option. The disadvantage of this method is that it is based on a Fourier transform of the image. This transform requires $O(N \log N)$ time to compute. In addition, iterative refinement is slow because modifying the intensity of a single pixel affects all of the terms in this representation. A variety of other pyramidal transforms were also investigated. These included the steerable pyramid used in the Sarnoff model [88], Daubechies' family of wavelets [37], and the biorthogonal bases of Cohen, *et. al.* [30]. These methods were deemed undesirable because of the larger, overlapping spatial filters

that are used in the transforms. The size of these filters slows down both the direct and iterative calculation of the transform, and the fact that the filters overlap would have complicated the error estimation stage of the adaptive sampling algorithm discussed in Chapter X.

The Haar transform employed is the two-dimensional non-standard decomposition. This transform can be expressed as:

$$\begin{aligned}
 c_{l-1}\left[\frac{x}{2}, \frac{y}{2}\right] &= (c_l[x, y] + c_l[x + 1, y] + \\
 &\quad c_l[x, y + 1] + c_l[x + 1, y + 1])/4 \\
 d_{l-1}^1\left[\frac{x}{2}, \frac{y}{2}\right] &= (c_l[x, y] - c_l[x + 1, y] + \\
 &\quad c_l[x, y + 1] - c_l[x + 1, y + 1])/4 \\
 d_{l-1}^2\left[\frac{x}{2}, \frac{y}{2}\right] &= (c_l[x, y] + c_l[x + 1, y] - \\
 &\quad c_l[x, y + 1] - c_l[x + 1, y + 1])/4 \\
 d_{l-1}^3\left[\frac{x}{2}, \frac{y}{2}\right] &= (c_l[x, y] - c_l[x + 1, y] - \\
 &\quad c_l[x, y + 1] + c_l[x + 1, y + 1])/4,
 \end{aligned} \tag{36}$$

where c_l specifies the lowpass coefficients of the level l Haar basis, d_l^1 , d_l^2 and d_l^3 are the detail coefficients of the three two-dimensional level l Haar wavelets, and $c_{levels-1}[x, y]$ corresponds to the response of either the small, medium or long receptors at a pixel location (where *levels* represents the number of levels in the quad-tree). This decomposition is carried out for each of the S, M and L channels and is stored in a quad-tree representation with the highest frequency details at the bottom and lowest frequency at the top. The detail coefficients of the Haar transform constitute the cortex representation. These detail terms indicate variations in

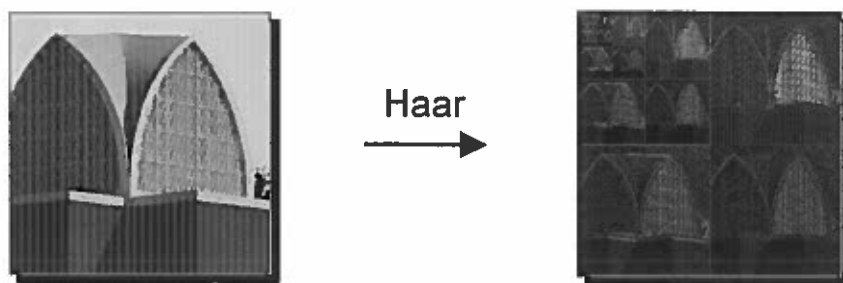


FIGURE 14. Flow graph of the *cortex filtering* stage of the visual difference predictor.

the image that are localized in space, frequency, and angular direction.

The frequency selectivity of the detail terms at a given level of the tree is defined as the frequency in cycles per degree (cpd) to which the wavelet at that level is optimally responsive. The detail coefficients are additionally tuned to three angular directions. The d^1 and d^2 terms are respectively selective of horizontal and vertical variations, and the d^3 terms isolate changes along both diagonal orientations. The poor frequency selectivity and limited orientation tuning of the Haar wavelet is acknowledged as a limitation of this approach. However, the efficiency gains are substantial.

Figure 14 illustrates the Haar transform applied to one of the color channels. In the image on the right, increased brightness indicates larger detail magnitudes. The highest frequency terms are arranged in the bottom and right side of the image and the lowest frequency term is in the upper left. At each level there are three blocks of detail coefficients. The top, left, and lower right blocks respectively contain the horizontal, vertical, and diagonally selective terms.

The next step in the image quality model incorporates the effect of *chromatic aberration*. This model is novel in its inclusion of this effect. Chromatic aberration describes the defocusing of light as a function of wavelength by the optics of the

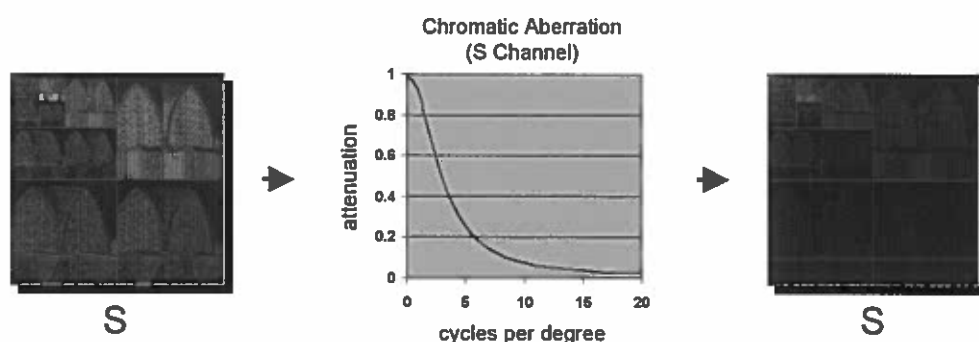


FIGURE 15. Flow graph of the *chromatic aberration* stage of the visual difference predictor.

eye. This defocusing most strongly affects the response of the short wavelength receptors, and severely attenuates the visibility of high spatial frequency detail in this channel. The center graph of Figure 15 depicts a plot of the sensitivity loss due to chromatic aberration in the short wavelength channel. This plot shows that sensitivity drops to less than half its original value at 4 cpd and is virtually non-existent at frequencies higher than 8 cpd. The original chromatic contrast sensitivity experiments performed by Mullen [101] corrected for chromatic aberration. In order to accurately apply the results of her work at the latter stages of the model it is necessary to reintroduce this effect.

Figure 15 illustrates how chromatic aberration is included in the image quality model. The unmodified cortex representation of the S cone receptors is illustrated on the left side of the figure. The response of these receptors are attenuated by the effect of chromatic aberration as a function of spatial frequency. The lowpass filter used is contained in the center graph. This filter was generated by a fit to the data of Marimont and Wandell [91]. The lowpass filtering operation can be performed very rapidly because the cone responses are stored in a frequency based representation. Filtering in this domain is accomplished by merely scaling the detail

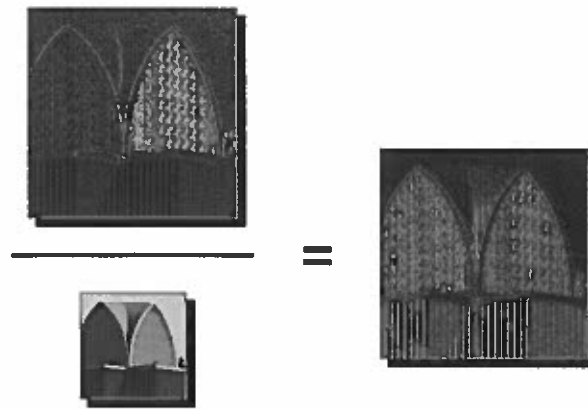


FIGURE 16. Flow graph of the *local contrast* stage of the visual difference predictor.

coefficients by the amount of attenuation at the associated spatial frequency. The decreased response at high spatial frequencies can be seen in the resulting image on the right of the figure.

In the next stage, labeled *local contrast*, the eye's non-linear contrast response to light is modeled. This is accomplished by dividing the detail coefficients of each color channel by the associated lowpass coefficient one level up in the quad-tree. This operation produces a local contrast value which is functionally equivalent to the standard cone contrast calculation of $\frac{\Delta S}{S}$, $\frac{\Delta M}{M}$, and $\frac{\Delta L}{L}$. It additionally avoids the assumption, found in other models [36, 45], that the eye can adapt at the resolution of a pixel.

Figure 16 demonstrates this process. In this example the local contrast calculation can be seen to increase the relative magnitude of the detail coefficients in the dark regions at the base and left side of the chapel. This corresponds with the fact that the visual system is more sensitive to variations at low illuminations than at high illuminations.

The following stage in the model consists of a transformation of the cone

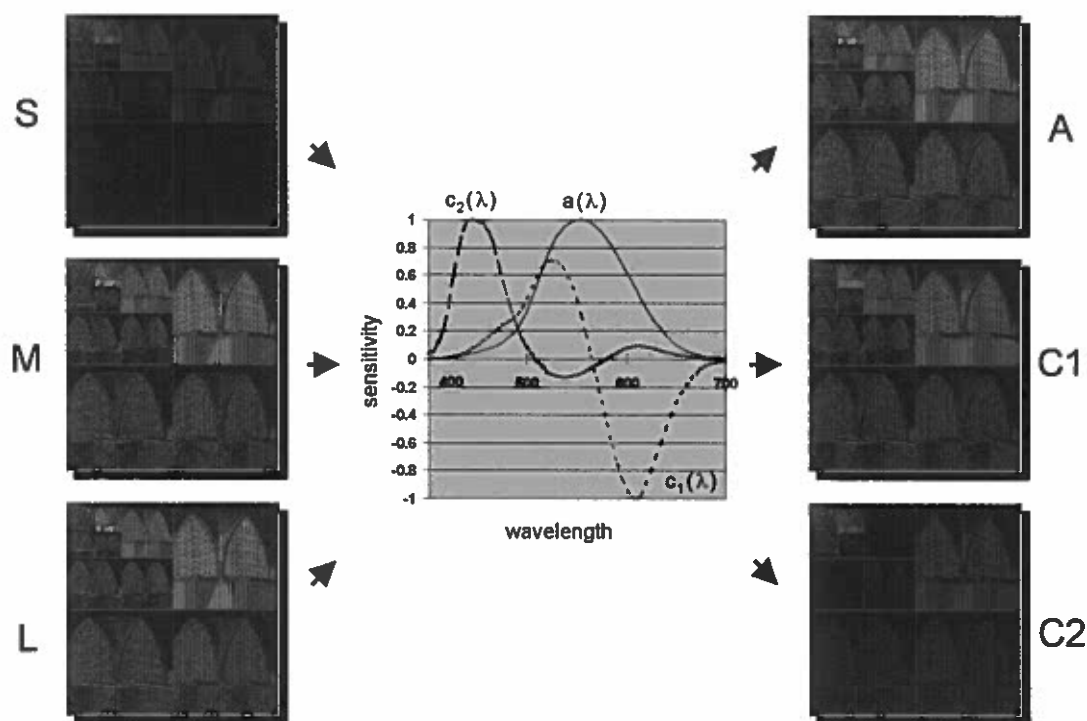


FIGURE 17. Flow graph of the *opponents contrast* stage of the visual difference predictor.

contrasts to an *opponents contrast space*. This space consists of a single achromatic (A) and two opponent color channels (C_1 and C_2). As discussed in the preceding chapter, there is significant evidence that the signals produced by the cones undergo this type of transformation.

Figure 17 illustrates this stage of the model. The local contrasts for the short, medium, and long receptors are contained on the left side of the figure, the spectral sensitivities of the A, C_1 , and C_2 channels are plotted in the center graph, and the resulting contrasts of the opponent channels are on the right. The conversion of the cone contrasts to an opponents contrast space is accomplished using the

transformation matrix [94]:

$$\begin{bmatrix} A \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0.0001 & 0.2499 & 0.7647 \\ 0.0018 & 2.9468 & -2.5336 \\ 1.0111 & -0.3877 & 0.2670 \end{bmatrix} \begin{bmatrix} S \\ M \\ L \end{bmatrix}. \quad (37)$$

This equation shows that the achromatic channel is primarily determined by the combined responses of the medium and long wavelength receptors, the C_1 channel is composed of the difference in the responses of the medium and long wavelength receptors, and the C_2 channel largely isolates the responses of the short wavelength receptors.

The sixth step of the quality model, labeled *CSF filtering*, incorporates variations in achromatic and chromatic contrast sensitivity as a function of spatial frequency. A diagram of this stage is contained in Figure 18. Different contrast sensitivity functions are used for the achromatic and chromatic channels. For the achromatic channel the human visual system has a peak sensitivity to signals of around 4 cpd, and significantly less sensitivity at higher and lower spatial frequencies. The model uses the equation for the achromatic contrast sensitivity function that was presented by Barten [10]. For the chromatic channels visual sensitivity is strictly lowpass, with a lower peak sensitivity and a lower frequency cutoff than is present in the achromatic channel. The chromatic contrast sensitivity function that is used in the model is implemented with a Butterworth filter that has been fit to the chromatic sensitivity data from Mullen [101].

At this stage in the algorithm the square of the contrast for each of the A , C_1 and C_2 channels is multiplied by the square of that channel's contrast sensitivity as

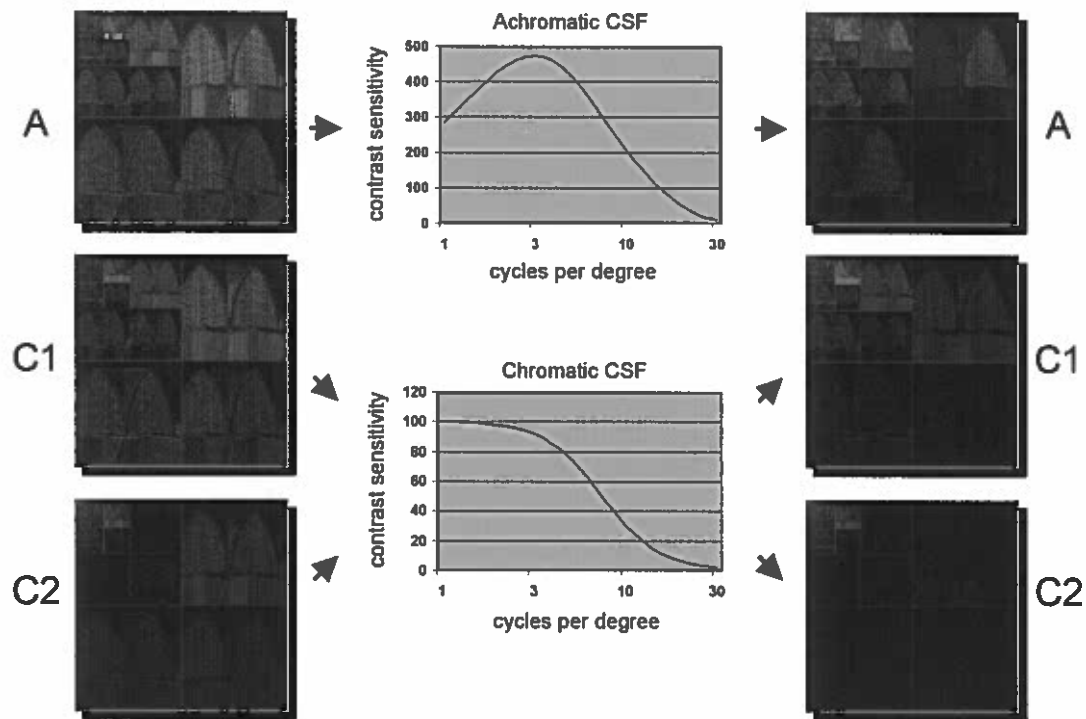


FIGURE 18. Flow graph of the *CSF filtering* stage of the visual difference predictor.

a function of spatial frequency. The square of the contrast and contrast sensitivity function is used to model the energy response that occurs for complex cells, as described in the Sarnoff VDM. This transformation has the result of making the model less sensitive to the exact position of an edge, which is a property shared by the human visual system as well [88].

The images on the right side of Figure 18 show the results of applying the achromatic and chromatic contrast sensitivity functions to the opponent contrast images. In the achromatic image, contrast response has been attenuated for both low and high spatial frequencies. For the chromatic channels contrast response declines with increasing spatial frequency. The fact that the C_2 channel has a lower frequency cutoff than the C_1 channel is the result of attenuation due to chromatic

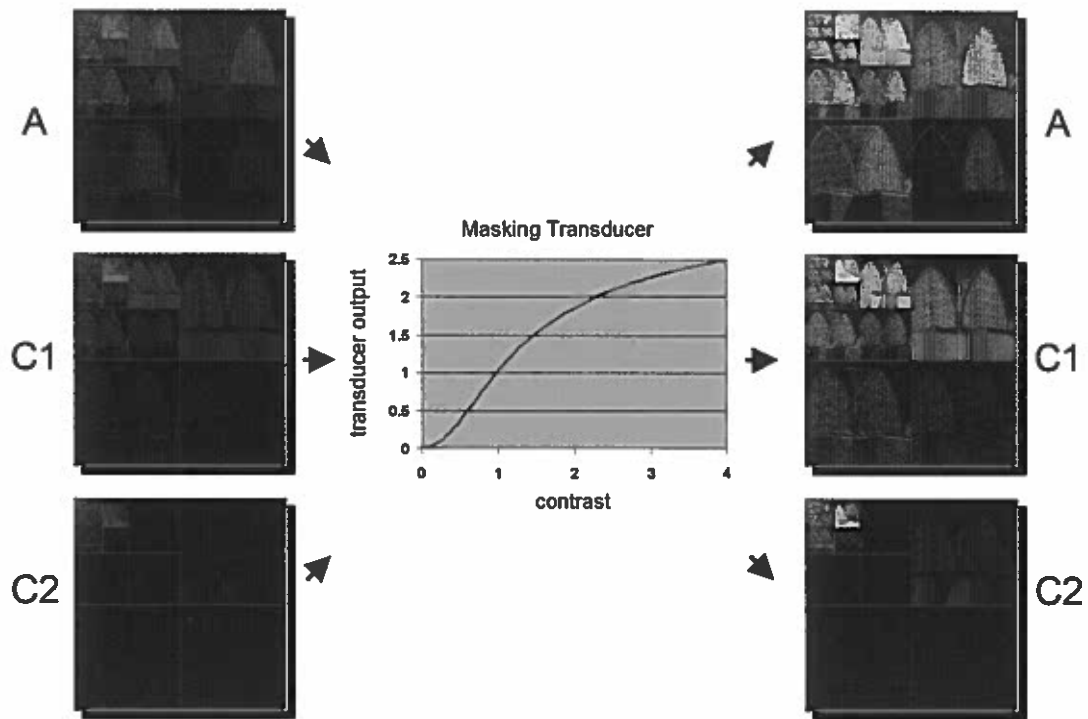


FIGURE 19. Flow graph of the *masking transducer* stage of the visual difference predictor.

aberration that was modeled at an earlier stage of the algorithm.

The next stage of the model labeled *masking transducer*, incorporates the effect of visual masking. Masking describes the phenomena where strong signals of a given color, frequency, or orientation can reduce the visibility of similar signals. This property of the visual system is incorporated through the use of a non-linear, sigmoid transducer described in the Sarnoff VDM:

$$T(A) = \frac{2A^{2.25/2}}{A^{2.05/2} + 1}. \quad (38)$$

In this equation, $T(A)$ is the transducer output and A is the weighted contrast produced from the previous stage of the model. This transducer is applied indepen-

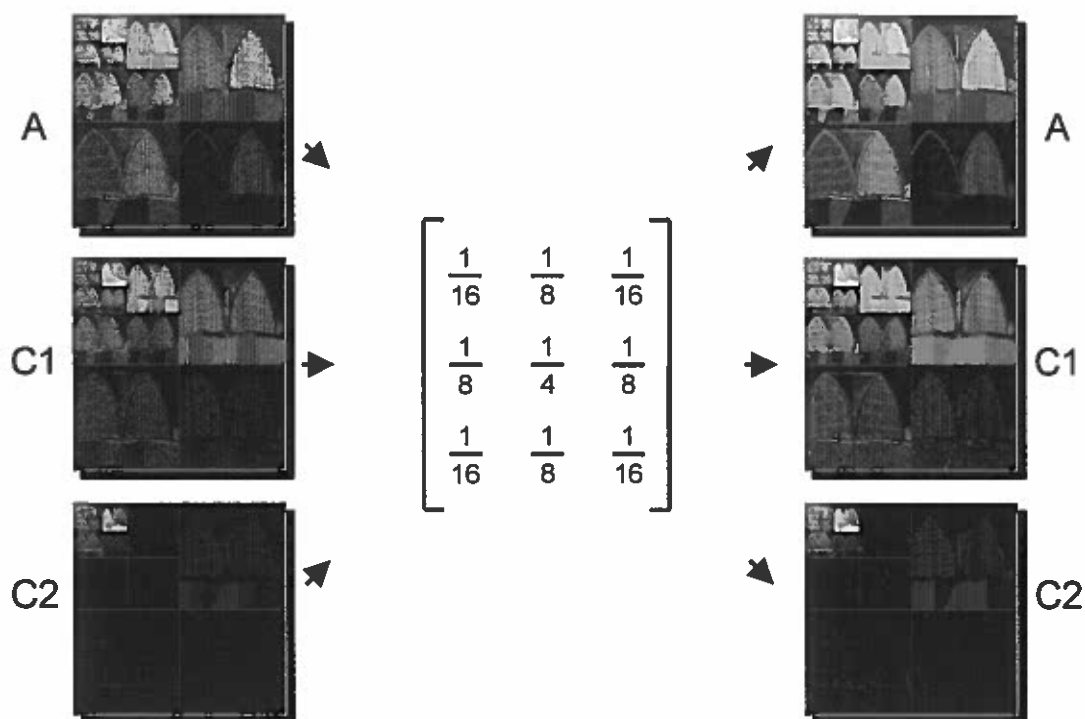


FIGURE 20. Flow graph of the *spatial pooling* stage of the visual difference predictor.

dently to the contrasts of each of the A , C_1 , and C_2 color channels.

This step is illustrated graphically in Figure 19. A plot of the transducer is contained in the center diagram. This function augments low contrasts and compresses high contrasts. The net result is that differences between high contrast signals are reduced, whereas low contrast differences are increased. This simulates the masking and facilitation effect described by Legge and Foley [82].

In the next stage of the model labeled *spatial pooling*, the transducer outputs are filtered over a small neighborhood of surrounding nodes at each level of the quad-tree. This is similar to the pooling operation performed in the Sarnoff VDM. It captures the fact that foveal human sensitivity is larger for sine wave gratings containing multiple cycles than it is for single cycle gratings. The inputs and outputs



FIGURE 21. Local visual differences calculated in the *distance summation* stage of the visual difference predictor.

of this stage are illustrated on the left and right sides of Figure 20 respectively. The pooling filter used in this model is contained in the center of the figure. The decision to use a 3x3 filter rather than the 5x5 filter specified in the Sarnoff VDM was made to improve the speed of the algorithm. This filter also corresponds better with the results of Wilson [158], who indicated that sensitivity reaches its peak for gratings containing 2.5 cycles.

In the final *distance summation* stage the differences between the pooling stages of the two input images are computed and used to generate a visual difference map. The local visual difference (LVD) at each node of the quad-tree is defined to be the sum across all orientations (θ) and color channels (c) of the differences of the pooling stages (P_1 and P_2) of the two images raised to the 2.4 power:

$$LVD = \sum_{\theta=1}^3 \sum_{c=1}^3 (P_1[\theta, c] - P_2[\theta, c])^{2.4}. \quad (39)$$

The output of this process for the two example images is shown in Figure 21. Brighter regions denote areas where the difference between the two images are more

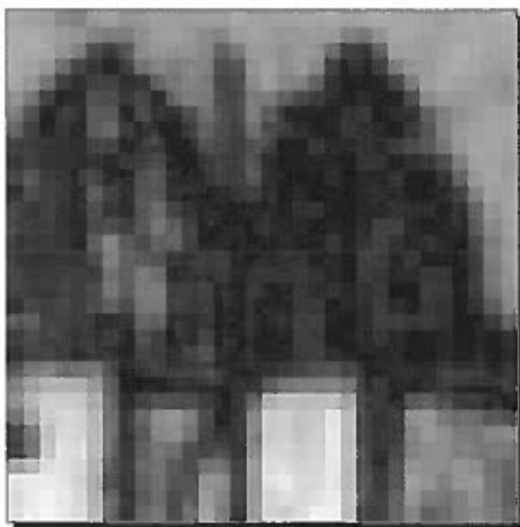


FIGURE 22. The final visual difference map.

perceptible.

The final difference map is generated by accumulating local visual differences across levels. This is accomplished by summing the local difference down each path in the quad-tree and storing the result in the leaves. The output of the algorithm is given by the leaf differences raised to the $1/2.4$ power. This distance summation stage is an application of Quick's vector summation technique with a 2.4 exponent [119]. The resulting visual difference map is contained in Figure 22.

Results

This section discusses the results of applying the new visual difference predictor to a number of demonstration images. The first example tests the ability of the predictor to accurately model how achromatic and chromatic contrast sensitivity varies as a function of spatial frequency. The images on the left side of Figure 23 show contrast sensitivity illusions for each of the opponent color channels. These

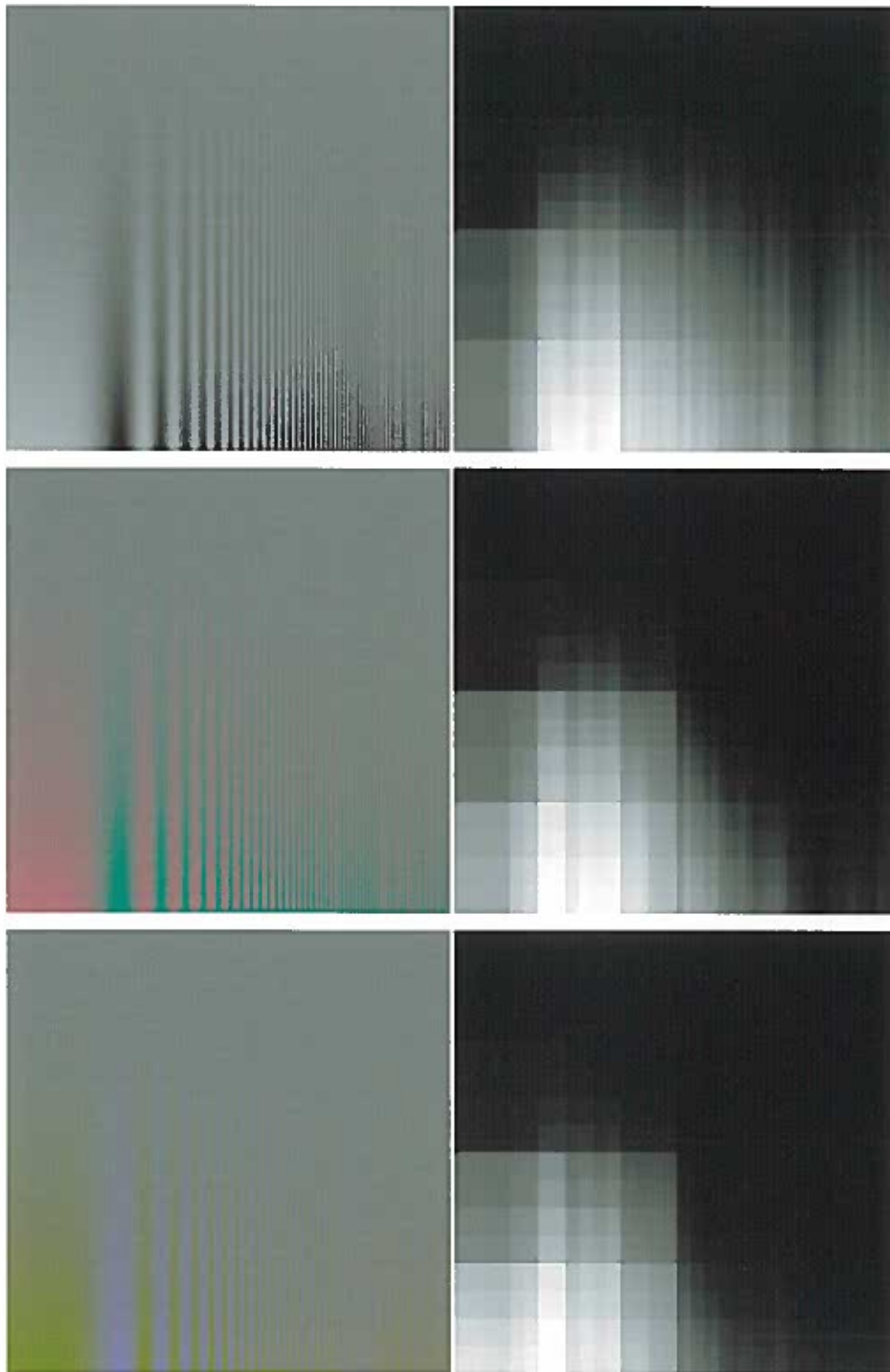


FIGURE 23. Achromatic and chromatic contrast sensitivity illusions and comparison against a uniform gray field.

images contain sinusoidal gratings that vary around a neutral gray. The contrast of the gratings increases logarithmically from the top to the bottom of the images, and the spatial frequency of the gratings increases logarithmically from left to right. The gratings vary along the achromatic axis in the top image, the C_1 axis in the middle image, and the C_2 axis in the bottom image.

These illusions demonstrate the contrast sensitivity function for each of the opponent color channels. Within these images a subjective contour is generated at the points where the contrast of the grating becomes just noticeably different from the gray background. For the achromatic illusion this contour has the shape of an inverted "U" that can be seen along the top of the image. This indicates the bandpass contrast sensitivity response of the achromatic channel. In the chromatic channels the subjective contour is strictly lowpass, with no drop-off at low spatial frequencies. It should also be apparent that the peak sensitivity to chromatic contrast is less than for achromatic contrast, and that the spatial frequency cutoff for the chromatic channel occurs at a much lower point than for the achromatic channel. The fact that the cutoff for the C_2 color channel is less than that for the C_1 is the result of chromatic aberration due to the optics of the eye.

The illustrations on the right side of Figure 23 show the output of the visual difference predictor when comparing the contrast sensitivity illusions on the left side of this figure with a constant gray image. White indicates areas of large visual difference while black denotes regions of low visual difference. In these images we see that the algorithm is able to correctly predict the shape and cutoff of the subjective contours for each of the color channels.

The next demonstration illustrates the ability of the model to predict the

visibility of noise in the presence of another signal. In computer graphic renderings, error primarily is manifested in the form of noise. Therefore, it is worthwhile to give special attention to the issue of noise masking. Noise in the achromatic channel is often the result of aliasing due to under-sampling or can result from poor Monte Carlo light source integration. An illustration of the grayscale contrast sensitivity illusion perturbed by the introduction of random noise is given in the upper left of Figure 24. In this image the noise is readily apparent above and to the sides of the subjective contrast sensitivity contour, but is less perceptible in areas where the sinusoidal grating is visible. This result occurs because the strong visual sensitivity to these frequencies masks the presence of a portion of the frequency spectrum of the noise. The image in the upper right of this figure shows the output of the visual difference predictor when comparing the original contrast sensitivity illustration to the contrast sensitivity illustration with noise added. In this image we see that the perceptual metric has correctly predicted that the error is less visible in the lower center region where masking is strongest.

Noise in the chromatic channels can arise when Monte Carlo integration is performed with multiple colored lights or is used to compute diffuse inter-reflections. Fine grained noise is not masked significantly in the color channels due to the lower frequency cutoff for the chromatic contrast sensitivity function. However, masking can still have a strong affect on the visibility of coarse grained noise. In the middle left and bottom left images in Figure 24 we have overlaid the chromatic contrast sensitivity illusions with coarse grained noise. In these illustrations the noise is very apparent in regions where sensitivity to the chromatic grating is low (top and right of the images), but less visible in regions where the chromatic grating is very

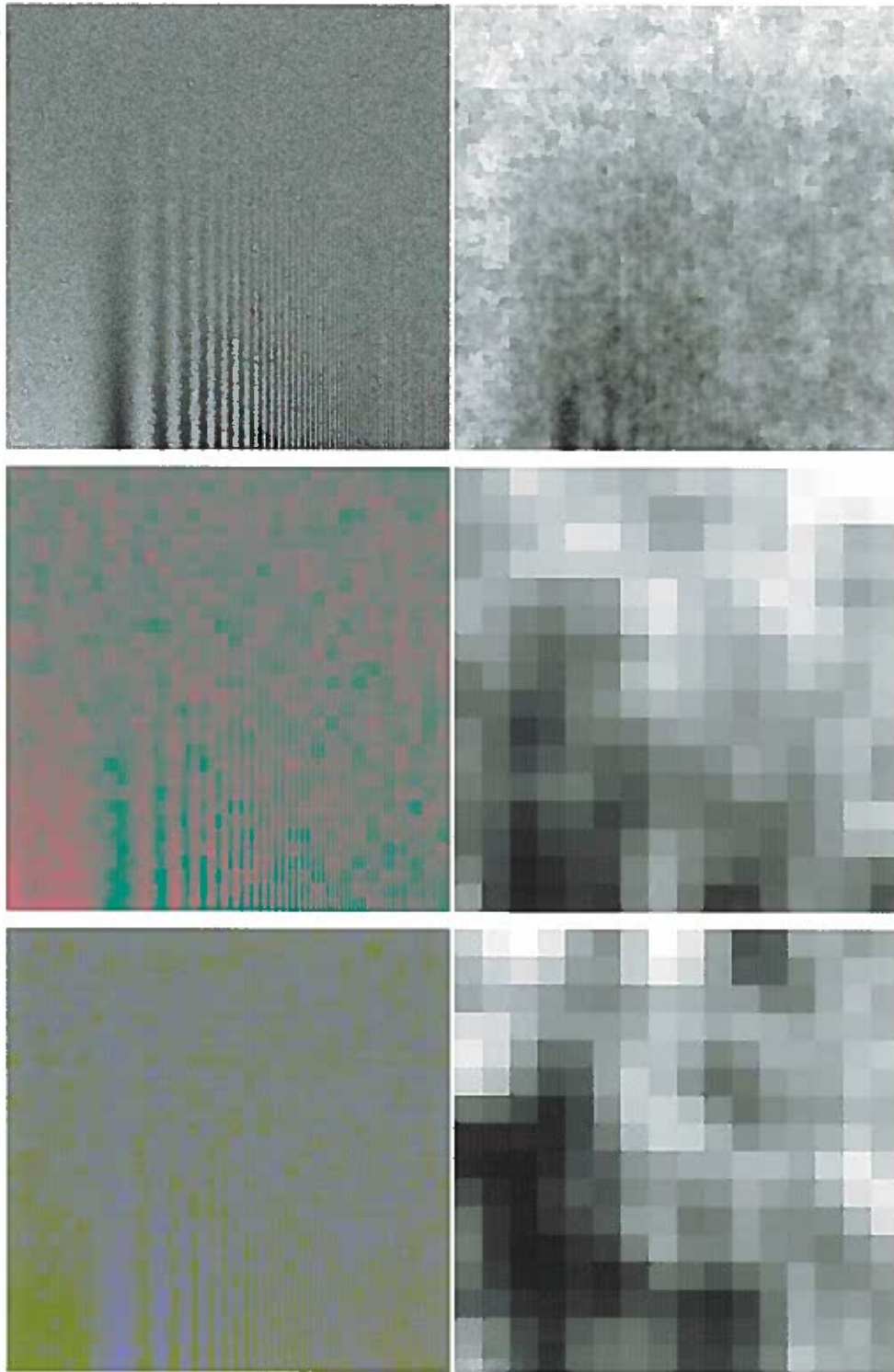


FIGURE 24. Achromatic and chromatic contrast sensitivity illusions with noise, and comparison with noiseless contrast sensitivity illusions.

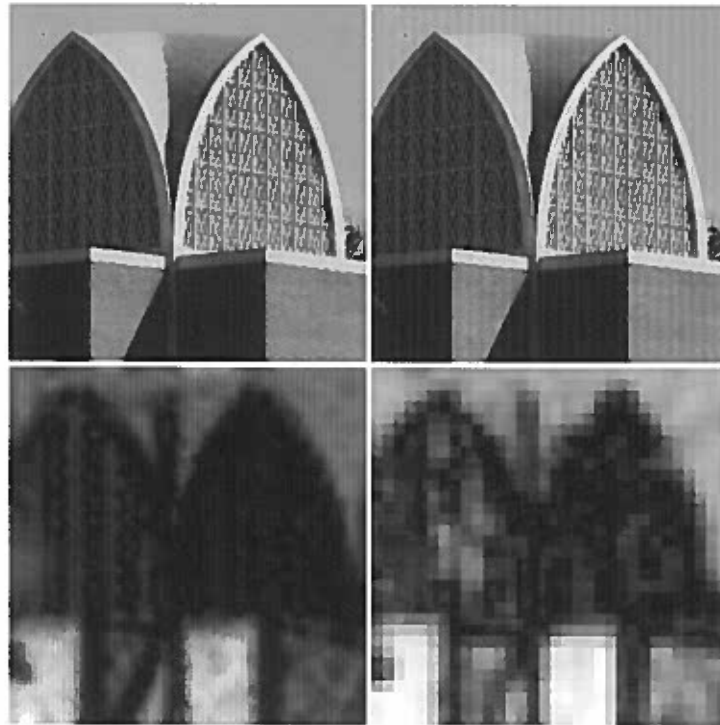


FIGURE 25. Top - Original chapel (left) and chapel with sinusoidal distortion (right). Bottom - Results of the Sarnoff VDM (left) and new vision model (right) visual difference predictions.

perceptible (lower left of the images). The images on the right once again show the output of the visual difference predictor when comparing the images with noise to the original chromatic contrast sensitivity illusions. In these illustrations we see that the algorithm has correctly predicted that the coarse grained noise is less perceptible in the lower left region of the images.

The final demonstration compares the results of the original Sarnoff VDM with the results of the new model for a set of complex images. This comparison is shown in Figure 25. The two input images are illustrated at the top of the figure. These images are achromatic versions of the chapel images used in the previous section. The results of the Sarnoff model's prediction is contained in the bottom left image

and the output of the new model is shown in the bottom right.

The visual difference map that is produced by the new algorithm contains a number of blocking artifacts that are caused by the Haar wavelet decomposition. However, the results of both algorithms are similar and correspond well with a subjective comparison of the input images. The Sarnoff VDM processed one channel in a gray-scale image representation and the new model processed three color channels. The new difference predictor executed in $1/60^{\text{th}}$ of the time of the original Sarnoff metric.

Summary

This chapter has presented a new, high speed, color visual difference predictor. This predictor is capable of rapidly assessing the perceptibility of objective image differences. This subjective measurement is calculated by processing the compared images in a similar manner to the human visual system.

The new predictor is an efficient implementation of an existing image quality model. This predictor utilizes the Haar wavelet transform to decompose the input images into spatial frequency and orientation tuned channels. This allows the model to execute in a fraction of the time required by previous metrics without a significant loss in accuracy. The use of the Haar transform also allows the model to make rapid, iterative predictions, as will be discussed in Chapter X.

The new model has been extended to perform color difference predictions. This addresses an important aspect of the visual system that has been neglected in the majority of previous quality models. It has additionally advanced the state of the art for color image quality models by including a novel chromatic aberration

stage. This stage corresponds with the structure of the visual pathway and allows the model to make accurate contrast sensitivity predictions for both opponent color channels. As discussed in the previous section, the speed of the algorithm is still superior to existing achromatic image quality models even though the new predictor is able to process color images.

CHAPTER VIII

PREVIOUS APPLICATIONS OF PERCEPTION IN COMPUTER GRAPHICS

There are only a limited number of prior applications of human visual perception in the field of computer graphics. This is surprising, given that computer graphics is primarily dedicated to producing images that are viewed by a human observer. The previous two chapters discussed how variations in visual sensitivity significantly affects the perceived quality of an image. Exploiting the varying sensitivity of the visual system has the potential to benefit a wide variety of computer graphics applications.

The importance of this line of study is underscored in a recent survey of leading computer graphics researchers [131]. In this survey, "dealing with human perceptual issues" is listed as one of the top two unsolved problems in the field of realistic image synthesis. The limited amount of work that has been done in this area, coupled with the significance of the problem, illustrates the novelty and importance of this topic.

This chapter reviews previous applications of perception in the field of computer graphics. The existing work in this area falls into three categories. These groupings are 1) the evaluation and application of image quality models, 2) the development of tone reproduction operators, and 3) the design of perceptually based sampling techniques. Each of these areas will be discussed in turn.

Image Quality Models and Applications

The use of visual metrics to measure the perceptual quality of an image is a relatively new idea in the field of computer graphics. This topic has received increasing amounts of attention in recent years. This section discusses various works that evaluate and apply image quality models for computer graphics applications.

Rushmeier *et. al.* [125] presented one of the first papers to suggest the use of perceptually based image quality models. The focus of this work was to evaluate the accuracy of a realistic image synthesis algorithm. This was accomplished by comparing luminance measurements of a real scene to the luminances computed by rendering an image of the same environment. A number of metrics were considered for this task. These included quality models by Mannos and Sakrison [90], Gervais *et. al.* [51], and a modified version of Daly's Visual Difference Predictor (VDP) [36]. The results of these models were analyzed and compared to an objective measurement of the mean squared error (MSE). In this study MSE was found to be a poor measurement of image quality. This was illustrated by an example in which MSE found the real luminances to be a closer match to a random noise image than to the rendered image. This was obviously not the case. The perceptual metric by Gervais *et. al.* was the next worst performer. This metric was adversely affected by its inclusion of phase information. Differences in phase caused edge mis-alignments to be a significant factor, which is not a property shared by the human visual system. The modified Daly VDP was found to produce the most accurate results of the three quality models. This metric was found to have a good correspondence with a subjective impression of image quality. The excellent results of the Daly VDP illustrates the utility of an image quality model for accessing the accuracy of

synthetically generated images.

A more comprehensive study of two image quality metrics was performed by Li [83, 84]. This work compared the results of the VDP by Daly with the results of the Sarnoff Visual Discrimination Metric (VDM) by Lubin [88]. Computer graphics images were utilized in this comparison. A variety of distortions were applied to these images and the two metrics were used to compute the visibility of the distortions. The two image quality metrics produced similar results. The Sarnoff model required more memory, but executed faster and produced slightly more accurate difference predictions.

Ferwerda *et. al.* [45] developed a new color visual difference metric by modifying the Daly VDP and adding the ability to handle color. This model was discussed in Chapter VI. The new metric was used to demonstrate how the presence of a texture can mask the appearance of faceting on a polygonalized object. The use of this model allows textures or polygonal detail to be selected so as to produce images of an acceptable visual quality.

The final work in this area was performed by Myszkowski [102]. His paper evaluated the accuracy of the Daly difference predictor and applied this metric to a number of aspects of an image synthesis algorithm. The accuracy of the VDP was evaluated by performing quality predictions of images produced at subsequent stages of a progressive radiosity algorithm. A number of components of the model were varied during these tests. In general, the VDP was found to produce quality assessments that correspond with a subjective impression of the images. However, superior results were obtained by replacing the point nonlinearity employed in the VDP with the local contrast metric used in the Sarnoff VDM. The VDP was then

applied to the image synthesis tasks of measuring convergence rates, determining when an image has been rendered to a specified quality, and controlling adaptive meshing in a radiosity-algorithm. The techniques proposed for these operations used a brute-force application of the VDP that severely impacts the execution time of the rendering algorithm.

Tone Reproduction Operators

Models of human visual perception have also been used to develop tone reproduction operators. Tone reproduction operators provide a mapping from the luminance values computed by a rendering algorithm to the luminance values that are actually reproducible on a given display. This mapping is constructed so that the visual experience of viewing the displayed image is as close a match as possible to the experience of viewing the real scene. In order to provide this “sensation preserving” mapping, tone reproduction operators have been developed that utilize a model of visual adaptation. These adaptation models enable the operators to account for perceptual differences caused by the limited dynamic range of the viewing device. Adaptation models simulate a somewhat different aspect of the visual system than is necessary for image quality comparisons. However, tone reproduction represents what is currently one of the most successful applications of visual perception in the field of computer graphics.

The first tone reproduction operator was developed by Tumblin and Rushmeier [141]. They noted that luminances encountered in the real world can range from between 10^{-5} to 10^5 cd/m^2 . However a typical CRT only has a luminance range of 1 to 100 cd/m^2 . Reproducing the visual experience of a real world scene with

the limited dynamic range of modern display devices is made possible by the nature of the human visual system. Humans are very poor judges of absolute luminance. Instead, the visual system tends to adapt to the average luminance and differentiate only spatial intensity variations. Therefore it is possible to remap the average luminance of the real scene to within the range that is displayable on a given device. This shift in the mean intensity is largely ignored by the visual system. However, when performing this transformation, it is necessary to account for how changes in illumination affect the visibility of contrast. Tumblin and Rushmeier account for this effect by employing a model of brightness and contrast perception developed by Stevens [137, 136]. This model is used to match the apparent contrast for the two different adaptation levels of the real and displayed images.

A very similar tone reproduction operator was created by Ward [148]. This method also employs a model of luminance adaptation, but it is based on a linear mapping from real to displayed intensities. This mapping is founded on a study by Blackwell [29]. The results of this study were used to derive a scalefactor that can be multiplied with the image radiances to generate the final display values. This technique provides a simple mechanism capable of reproducing both contrast and feature visibility.

The two previous methods assume that the eye adapts globally to the average intensity of an image. However, adaptation actually occurs locally over a small region of the image plane. Chiu *et. al.* [27] exploited this by developing a spatially nonuniform tone reproduction operator. The amount of scaling performed by the operator, at a given location, is based on a local average of the image intensity. An alternate approach was proposed by Tumblin *et. al.* [142]. In this method the

viewer is assumed to fixate on a specific point in the scene. This point is selectable by the user. The adapted luminance is determined by the average intensity in a small region surrounding this point. The size of this region is selected to cover the foveal portion of the eye where adaptation occurs. A global mapping is then applied to the entire image based on the intensity in this area.

A number of authors have created tone reproduction operators that include the effect of glare [27, 150, 111]. Glare is caused by scattering of light in the cornea, lens, and retina, and by diffraction in the cell structures on the outer radial areas of the lens. This creates a number of visible artifacts around small, bright objects such as stars and light sources. These artifacts include bloom, flare lines, and colorful lenticular halos. A physical and psychological model of glare was created by Spencer *et. al.* [134]. This model applies a digital point spread function to an image in order to simulate these effects.

The effects of luminance adaptation on color sensitivity and spatial acuity were included in a tone reproduction operator by Ferwerda *et. al.* [44]. At high illumination colors appear bright and saturated, and spatial frequency sensitivity is quite good. However, when the lighting is very dim, colors are perceived as being very desaturated and tend toward a neutral gray. The spatial frequency cutoff of the visual system also declines under low light conditions, causing images to appear blurred. Ferwerda's operator can be used to produce images that capture these effects over a wide range of lighting conditions. Color saturation is modeled by simulating changes in the spectral sensitivity of the rods and cones as a function of luminance. The loss of spatial acuity under low light conditions is reproduced by applying a lowpass filter to the image. The cutoff of this filter decreases with the

illumination.

A similar tone reproduction operator was developed by Ward Larson *et. al.* [150]. Their operator uses a novel histogram adjustment technique. The technique allows clusters of adaptation levels to be easily located and mapped to display values in a manner that preserves local contrast visibility. This mapping uses a model of human contrast sensitivity, glare, spatial acuity, and color sensitivity to match subjective viewing experiences between real and displayed images.

The latest tone reproduction operator was presented by Pattanaik *et. al.* [111]. This operator bears many similarities to the one developed by Ferwerda *et. al.*. However, the new operator utilizes a more advanced model of visual adaptation. This model incorporates an opponents representation of color and a contrast sensitivity function based on spatial frequency selective channels. The highlights of this model include its ability to simulate both chromatic and achromatic adaptation effects, and to modify the appearance of an image based on changes in both the shape and cutoff of the contrast sensitivity function.

Gibson and Hubbard [52] utilized a tone reproduction operator to determine display intensities during the progression of a radiosity algorithm. This problem is difficult because radiosity computes a view-independent measure of the illumination at the surfaces of a scene. This requires an a priori estimate of the adaptation luminance. However, by using a suitable approximation they were able to map scene radiances to display intensities during the progression of the algorithm. An error metric based on differences in these display intensities in a perceptually uniform color space was used to control a number of aspects of the radiosity algorithm. These included adaptive patch refinement, shadow detection, and a posteriori mesh

optimization.

Perceptually Based Sampling Techniques

There have been a few previous attempts to incorporate aspects of a vision model within an adaptive sampling algorithm. These methods have sought to vary the sampling density across the image plane so that effort is focussed on the regions of the image containing the most perceptible error. The primary limitation of these techniques is that they use very simplistic visual metrics.

Mitchell [98] proposed the first adaptive sampling algorithm to utilize a perceptually based refinement metric. In this algorithm the image plane is divided into a number of small cells and a low density set of samples is taken within each area. A test is then applied to determine which regions of the image require further refinement. Mitchell's test utilized a contrast calculation based on the maximum and minimum luminance returned by a sample within a cell. This measure of the contrast accounts for nonlinear variations in perceptual sensitivity as a function of background luminance. The contrast calculation is performed for each color channel and collapsed to a single quality metric by differentially weighting the responses of the red, green, and blue channels. This provides a simple, although incomplete, model of sensitivity changes as a function of wavelength. Finally, the cells with a weighted contrast larger than a user specified threshold were resampled at a fixed, high density sampling rate.

Meyer and Liu [95] developed an adaptive sampling algorithm that exploited limitations in the visual system's chromatic contrast sensitivity response. The algorithm that they created was based on Painter and Sloan's [107] tree-based refinement

strategy discussed in Chapter IV. This method iteratively refines an image from low to high spatial frequencies by repeatedly subdividing regions of the image as more samples are taken. Meyer and Liu extended this algorithm by utilizing an opponents space to store color information. In this work, they showed that less refinement was necessary in the chromatic than in the achromatic channel. This is the result of the lower spatial frequency cutoff of the chromatic contrast sensitivity function.

The issue of bias in a perceptually based sampling algorithm was addressed in a paper by Tamstorf and Jensen [139]. Bias is a statistical artifact of adaptive sampling that causes the intensity of a pixel to converge to a value slightly different than the true intensity mean [79]. They modified an adaptive algorithm by Purgathofer [118] so that the refinement test included a model of nonlinear contrast response. The bias of the algorithm was then estimated using a nonparametric bootstrapping method. They found that in the majority of cases the effect of bias was negligible, and that it was less than the display's color resolution.

Summary

This chapter reviewed previous applications of perception in the field of computer graphics. This is a new area of exploration that has generated significant interest, but has yet to be fully exploited. The existing work has focussed on the evaluation and application of image quality models, the design of tone reproduction operators, and the development of perceptually based sampling algorithms.

Perceptually based image quality models have been evaluated by a number of authors. These evaluations have been decidedly favorable. Two of the most complete and widely used image quality models are the Visual Difference Predictor and the

Sarnoff Visual Discrimination Metric. Image quality models have been mainly used to perform image comparisons. However, the application of these models to other aspects of the rendering process is beginning to be explored.

Tone reproduction operators have been developed in order to create images that match the subjective experience of viewing a real scene. These operators are based on models of visual adaptation. Adaptation models serve a different purpose than the models used to determine image quality. However, the success of tone reproduction illustrates one of the many benefits of incorporating aspects of human visual perception within computer graphics algorithms.

Perceptually based sampling algorithms that utilize aspects of an image quality model have been developed. These algorithms seek to focus effort on the regions of an image containing the most visible artifacts. However, existing work in this area has been limited, and it has yet to incorporate many of the key features of visual sensitivity. This issue will be addressed in the final two chapters of this dissertation.

CHAPTER IX

A FREQUENCY BASED RAY TRACER

In Chapter VIII, the previous attempts to incorporate aspects of a perceptual model within an adaptive sampling algorithm were described. These existing techniques are significantly limited by the fact that they operate in the spatial domain. This means that samples are distributed based on an estimate of the image and its error at discrete locations within the viewing plane. The difficulty with this approach is that the perceptual response of the human visual system is strongly frequency dependent. There are, therefore, few aspects of visual sensitivity that can be exploited by spatially based adaptive techniques.

This chapter describes the design of a new adaptive sampling algorithm. This algorithm synthesizes images directly into a spatial frequency based representation. The synthesis of images into this representation allows a simple frequency dependent perceptual quality model to be used to direct the placement of samples. This enables the algorithm to reduce the number of samples that are necessary to produce an image of a given visual quality by eliminating the most apparent artifacts first, and progressing to the less apparent artifacts as time and resources permit.

The frequency domain that is chosen as the basis of this algorithm is the same as that employed by common image and video compression schemes. This reduces the steps necessary to compress the resulting image and allows the algorithm to exploit a number of advantageous features of these compression techniques. One

significant feature of these compression methods is that they utilize the perceptual phenomenon of visual masking to determine the necessary accuracy of each of the frequency terms. This aspect of the visual system will be used in a similar fashion in the new adaptive sampling algorithm to determine the sample accuracy that is required to resolve each of the terms in the frequency representation of an image.

The direct synthesis of images into a frequency representation also offers a number of advantages when considered solely as a sampling and reconstruction technique. This chapter shows how changes in the frequency spectrum can be used to estimate the amount of aliasing present in regions of an image. This provides a novel error metric that is used to guide the adaptive sampling process. By careful placement of additional samples, the new algorithm also provides a method to refine images from low to high spatial frequency. This method offers a unique solution to the difficult problem of reconstructing a bandlimited image from a set of nonuniformly spaced samples. Finally, the development of an image in a frequency representation is shown to simplify the final filtering of the image. This step is necessary to remove any residual aliasing artifacts from the resulting image.

The remainder of this chapter is divided into five major sections. The first section discusses the design of a simple, frequency dependent perceptual quality model. In the second section the JPEG image compression algorithm is described. This is the particular compression algorithm that is most directly exploited in this work. The third section presents the design of the new frequency based adaptive sampling algorithm. Results are discussed in section four, and the work is summarized in section five.

A Frequency Dependent Perceptual Quality Model

This section discusses the design of a simple, frequency dependent perceptual quality model that will be used to direct the placement of samples by the adaptive sampling algorithm. This model falls into the class of Fourier CSF models discussed in Chapter VI, pages 75–77, and includes an additional color processing element. The model that is described is most closely related to the work of Faugeras [42] and utilizes the color spaces and spatial filters presented by Meyer and Liu [95].

This model is similar in many respects to a number of image quality models that have been successfully employed by the image processing and vision research communities. It includes an initial receptor stage, followed by the application of a contrast nonlinearity, a matrix to an opponents color representation, and a final spatial frequency filtering. These stages simulate the processing that is performed by the visual pathway on incident light. The measurement of differences in the resulting perceptual space can be used to determine the visibility of changes in the incoming light intensity.

The first stage of the model simulates the spectral absorptions of the three types of cone receptors found in the retina. These three classes of receptors are sensitive to the short (S), medium (M), and long (L) wavelengths of visible light. The wavelength selectivity of these receptors is the fundamental building block of human tri-chromatic color vision. In order to determine the response of these receptors, the incident light is linearly transformed from the device independent CIE XYZ space

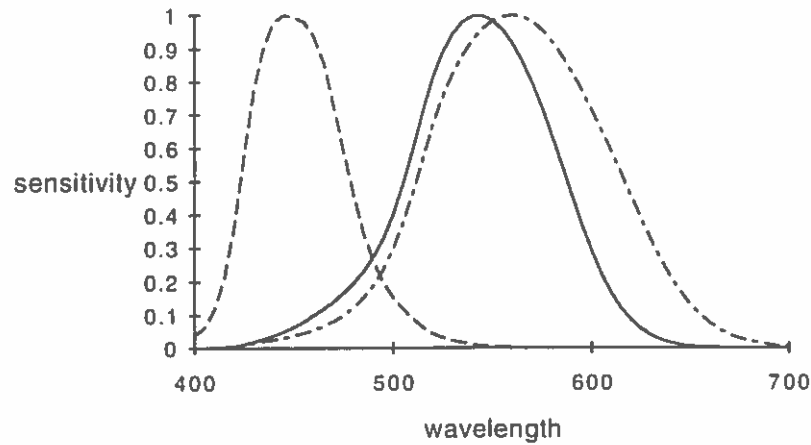


FIGURE 26. Spectral sensitivities for S (dashed), M (solid), and L (dot-dashed) components of SML space.

to SML space via the expression

$$\begin{bmatrix} S \\ M \\ L \end{bmatrix} = \begin{bmatrix} 0.0000 & 0.0000 & 0.5609 \\ -0.4227 & 1.1723 & 0.0911 \\ 0.1150 & 0.9364 & -0.0203 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (40)$$

Figure 26 illustrates the short, medium and long spectral sensitivities that result from this operation.

The next stage of the model accounts for the receptors nonlinear contrast response. This aspect of the visual system causes equal differences in light energy to be more perceptible in dark regions than in bright. This effect is demonstrated in Figure 27. The interior squares in both the left and right examples are equally different from the surrounding region in terms of luminance (light energy). However, the perceptible difference between the center and surrounding squares is larger in the left example than in the right.

This variation in sensitivity is accounted for in the model by applying a com-

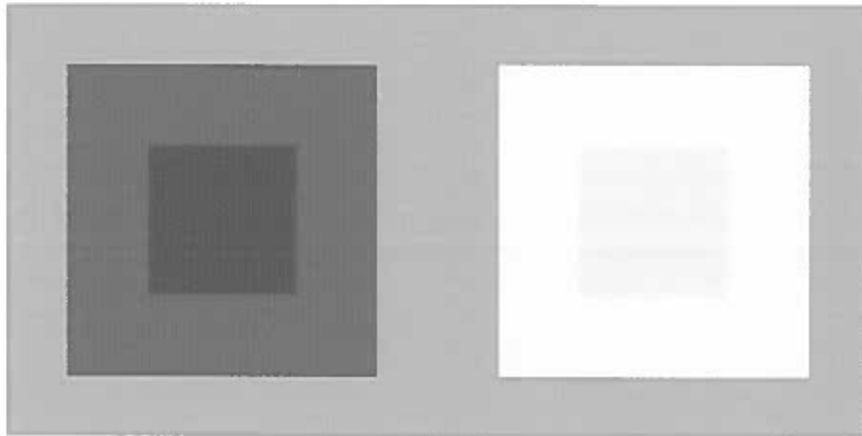


FIGURE 27. Center and surrounding squares of equal luminance difference but not equally perceptible difference.

pressive point nonlinearity to the responses of the short, medium and long receptors.

The nonlinearity that was selected is given by a power law with a $\frac{1}{2.2}$ exponent:

$$\begin{bmatrix} S' \\ M' \\ L' \end{bmatrix} = \begin{bmatrix} S \\ M \\ L \end{bmatrix}^{\frac{1}{2.2}} . \quad (41)$$

This has the effect of reducing the sensitivity of the cone responses to differences at increasing illumination levels.

At the next stage in the model the cone responses are transformed into an opponents color space. This space consists of a single achromatic (A) and two opponent (C_1 and C_2) color channels. The response of these channels are given by

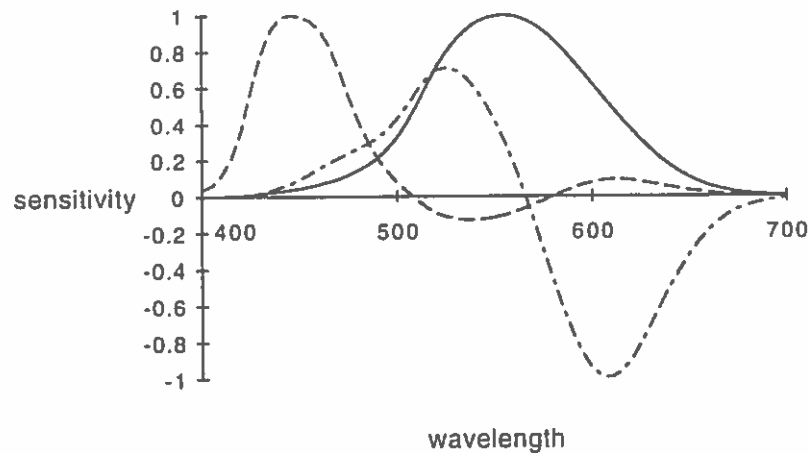


FIGURE 28. Spectral sensitivities for A (solid), C_1 (dot-dashed), and C_2 (dashed) components of AC_1C_2 space.

the following transformation of the cone responses:

$$\begin{bmatrix} A \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0.0001 & 0.2499 & 0.7647 \\ 0.0018 & 2.9468 & -2.5336 \\ 1.0111 & -0.3877 & 0.2670 \end{bmatrix} \begin{bmatrix} S' \\ M' \\ L' \end{bmatrix}. \quad (42)$$

Figure 28 depicts the spectral sensitivities of the color channels that result.

There is strong evidence that the signals produced by the cones undergo this type of transformation (see pages 86–90 of Chapter VI and [21, 39, 71, 73]). A number of different opponents spaces have been proposed [117]. The space that was selected for this model was developed by Buchsbaum and Gottschalk [21]. They applied the discrete Karhunen-Loeve expansion to the SML fundamentals to develop a space that is optimal in terms of statistical communication theory and is consistent with other spaces that have been derived psychophysically. This space has additionally been used to select the wavelengths at which to perform synthetic image generation [94].

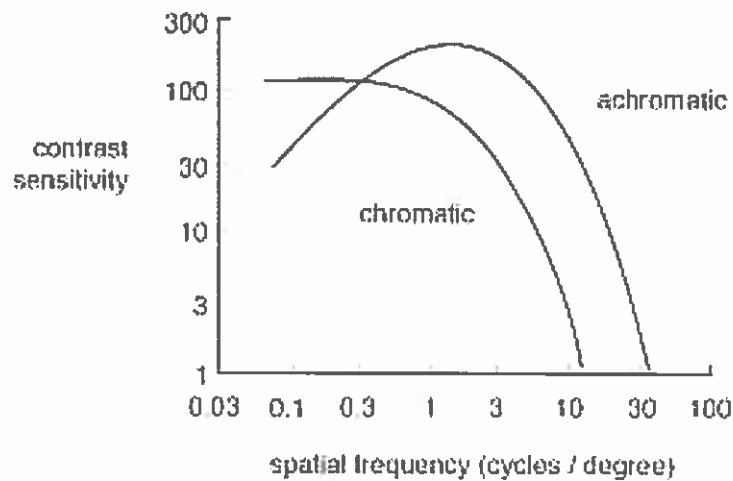


FIGURE 29. Achromatic versus chromatic (either C_1 or C_2) spatial frequency response (after Mullen [101]).

The final stage of the model accounts for differences in the visual system's achromatic spatial frequency sensitivity. The response of the visual system is strongly dependent on both the color and the spatial frequency of the stimulus. This is partially the result of axial chromatic aberration due to the optics of the eye. Chromatic aberration describes the defocusing of light as a function of wavelength that is caused by the lens optics. This effect is strongest for the short wavelengths of light and severely limits the visibility of high frequency detail in this channel. The relative densities of the cone receptors also contributes to the differences in chromatic spatial frequency response. This is because a denser packing of the receptors is required to adequately sample higher spatial frequency detail. Cicerone and Nerger [28] have shown that there are approximately twice as many long wavelength receptors as there are medium wavelength receptors, and Williams *et. al.* [157] have found that the spacing of the short wavelength receptors is very sparse (approximately one in every ten minutes of visual arc).

The visual system's chromatic and achromatic spatially frequency response

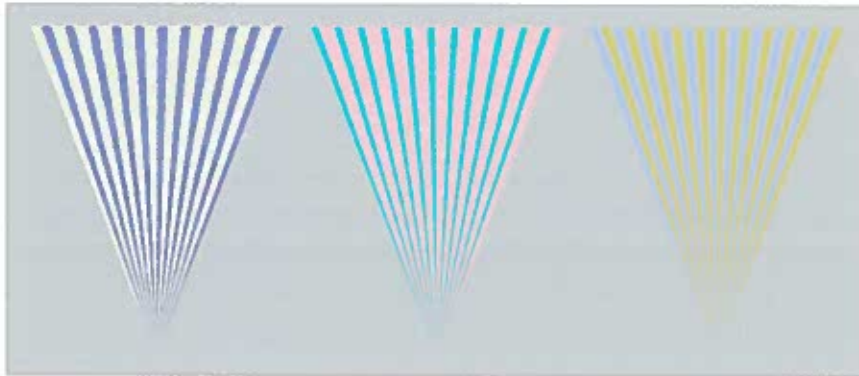


FIGURE 30. From left to right, spatial acuity wedges depicting variation in the A , C_1 and C_2 channels.

has been measured by Mullen [101]. The results of her experiments are depicted in Figure 29. This figure plots contrast sensitivity (one over the magnitude of the sinusoidal grating that is required to produce a threshold response) versus spatial frequency for the achromatic and chromatic (either C_1 or C_2) channels. These experiments show that the cutoff for the achromatic channel is 34 cycles/degree and for the chromatic channels is 11 cycles/degree. More recent work by Poirson and Wandell [115] that accounts for chromatic aberration has shown that the cutoff for the C_2 channel may be as low as 4 cycles/degree.

The differences in these visual cutoffs is demonstrated in Figure 30. This figure depicts three acuity wedges that vary along the A , C_1 , and C_2 axes. In this illustration it can be seen that it is easier to resolve the high frequency detail at the bottom of the achromatic wedge than it is for either the C_1 or C_2 wedge.

This aspect of the visual system is incorporated into the model by transforming the opponent signals into the frequency domain and applying a set of spatial frequency filters. This operation is performed independently to each of the A , C_1 , and C_2 channels. The spatial frequency response of each channel is modeled as a

Butterworth filter with the frequency cutoffs mentioned above. The net result is a perceptual representation of the input signal that will be used as the basis of the adaptive sampling algorithm.

The JPEG Compression Algorithm

The goal of image compression algorithms is to represent images of a high visual quality using a minimal number of bits. Adaptive sampling algorithms seek to produce images of a high visual quality using a minimal number of samples. There are a number of similarities between these two operations, and important insights into the design of efficient adaptive sampling algorithms can be achieved by investigating image compression techniques.

The JPEG still image compression standard is one of the most powerful and popular image compression methods. This algorithm is able to achieve substantial compression rates by transforming an input image into the frequency domain and then utilizing the properties of visual masking to determine the appropriate accuracy for each of the frequency terms. These are two important elements that will be exploited in the new adaptive sampling algorithm. The remainder of this section will describe the various stages of the JPEG compression algorithm.

The first step in the JPEG algorithm is to break the input image into a series of 8 by 8 pixel blocks. This is done to both increase the tractability of the subsequent frequency transform and to exploit the large correlation of intensity values found in blocks of this size. The values of the 8 by 8 blocks are then transferred into the frequency domain by means of the Discrete Cosine Transform (DCT).

The DCT decomposes each block into an 8 by 8 sum of harmonically related

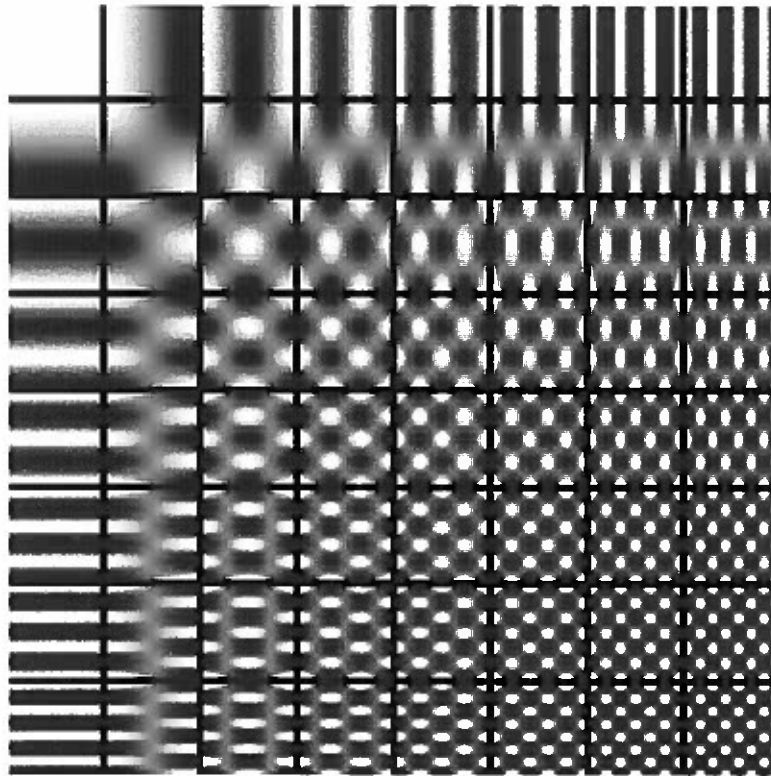


FIGURE 31. The basis functions of the Discrete Cosine Transform.

cosine basis functions in a manner similar to the Discrete Fourier Transform. These basis functions are depicted in Figure 31. The lowest frequency or “DC” term is located in the upper left corner, and the “AC” terms increase in horizontal frequency from left to right and in vertical frequency from top to bottom.

The forward and inverse DCT are respectively given by the following two equations [145]:

$$F(u, v) = \frac{1}{4} \kappa(u) \kappa(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) C_{x,u} C_{y,v} \quad (43)$$

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 \kappa(u) \kappa(v) F(u, v) C_{x,u} C_{y,v}, \quad (44)$$

where

$$C_{A,B} = \cos \left[\frac{(2A+1)B\pi}{16} \right]$$

and

$$\begin{aligned} \kappa(u), \kappa(v) &= 1/\sqrt{2} \quad \text{for } u, v = 0 \\ \kappa(u), \kappa(v) &= 1 \quad \text{otherwise.} \end{aligned}$$

The forward transform takes as input an 8 by 8 block of intensity values $f(x, y)$ from the input image and returns an 8 by 8 block of frequency coefficients $F(u, v)$. These frequency coefficients give the magnitudes of the cosine basis functions which, when summed, will exactly reproduce the original image block. This summation is given by the inverse transform that takes the frequency coefficients $F(u, v)$ and returns the original intensity values $f(x, y)$.

An important aspect of the DCT is its ability to represent the majority of the image's energy within the first few low frequency terms. This is demonstrated in Figure 32. This figure depicts the magnitude of the 4 by 4 low frequency coefficients. The magnitude of all 64 coefficients were not shown due to the imperceptibility of the higher frequency terms. The inset images were generated by dividing the input image into 8 by 8 blocks, applying the DCT to each block, and collecting the magnitudes of a specific frequency coefficient. The DC terms are depicted in the upper left image. The remaining images show the AC terms for bases that increase in horizontal and vertical frequency from left to right and top to bottom respectively.

This figure demonstrates that in a DCT representation of an image, the major-

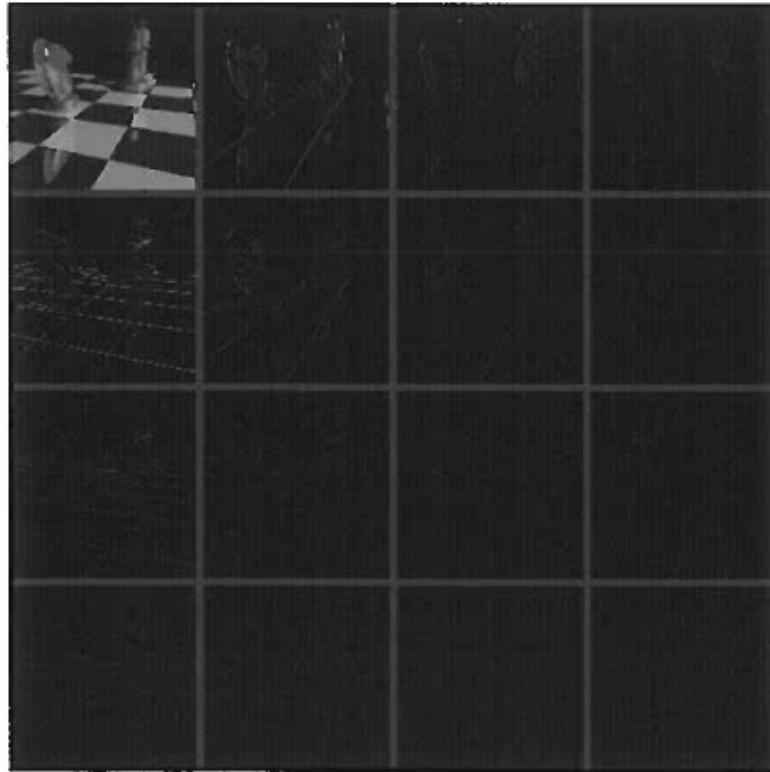


FIGURE 32. Magnitude of the 4 by 4 low frequency components in the DCT representation of a chess board image.

ity of the frequency coefficients will be very small and that the magnitude of these coefficients rapidly declines with increasing spatial frequency. Rao and Yip [120] have shown that the DCT is asymptotically equivalent to the optimal Karhunen-Loeve transform in its ability to pack the energy of the image into the low frequency terms. The fact that most of the higher frequency coefficients are small or zero imparts a large amount of redundancy to this representation that will be exploited at a latter stage of the compression algorithm. This aspect of the transform is also interesting from an adaptive sampling standpoint, since the areas of the image that contain high frequency information are the only areas that require a high sampling rate in order to be accurately reconstructed.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

FIGURE 33. A sample quantization table (after Wallace [145]).

The next stage of the compression algorithm reduces the accuracy of the frequency coefficients by means of a scalar quantization table. This lowers the bit rate that is required to encode the image by further increasing the redundancy of the frequency terms. An example quantization table is shown in Figure 33. This table specifies the quantizer step sizes $Q(u, v)$, that are applied to determine the final coefficients $F(u, v)$ that will be encoded. This is accomplished by the formula

$$F^Q(u, v) = \text{Integer Round} \left[\frac{F(u, v)}{Q(u, v)} \right]. \quad (45)$$

The quantization table applies different step sizes to the various frequency terms. This allows the algorithm to achieve larger compression rates by discarding information that is the least perceptually significant. High frequency coefficients can be represented much less accurately than low frequency coefficients without adversely affecting the perceived quality of the image. This is largely due to the effect of visual masking.

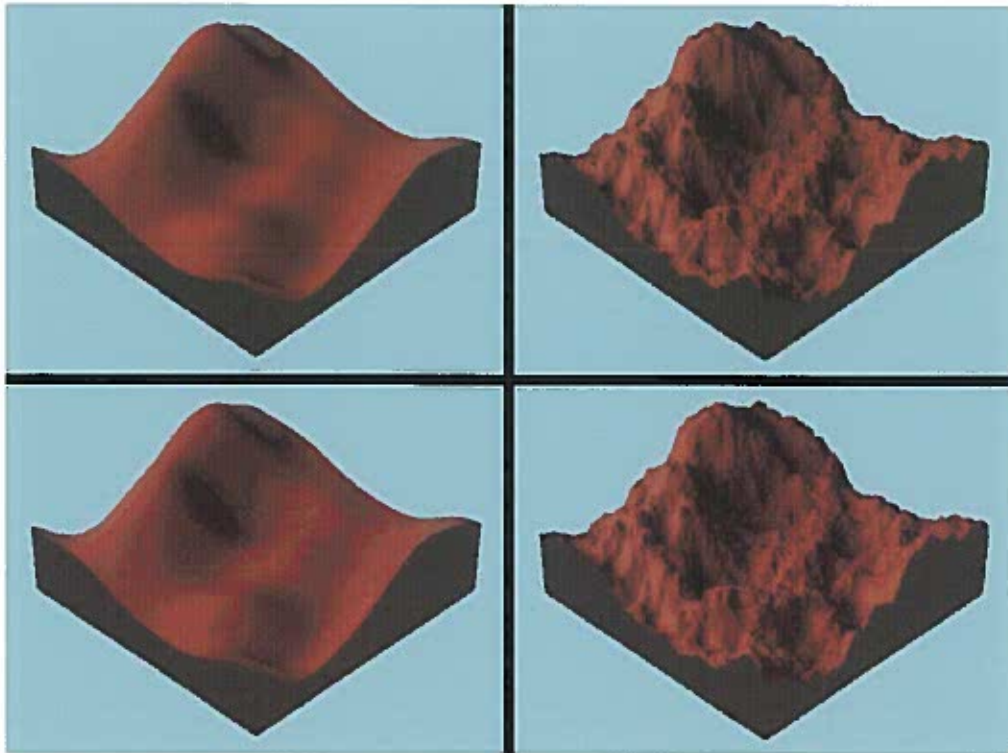


FIGURE 34. Visibility of noise when images with different spatial frequency content (top row) are quantized to 4 bits per color channel (bottom row).

One aspect of this effect is that the presence of high frequency information can reduce the detectability of other high frequency stimuli [128]. This is demonstrated in Figure 34. The top two images in this figure show the original versions of low and high frequency fractal mountains. In the bottom row the intensity of the images has been quantized to 4 bits per color channel. The quantization error is clearly evident in the low frequency mountain. However, the quantized high frequency mountain is virtually indistinguishable from the original image.

At the next stage in the compression algorithm the frequency coefficients in each block are arranged in a zig-zag order. This ordering is illustrated in Figure 35. This operation facilitates the following entropy encoding step by placing low fre-

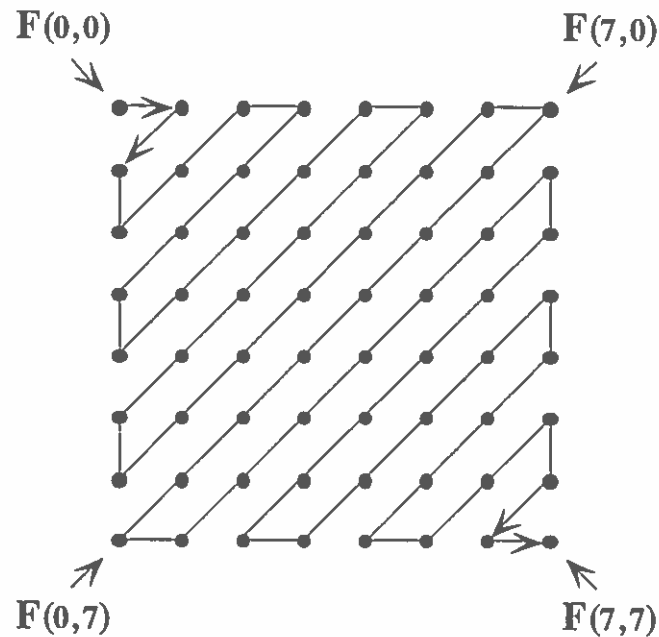


FIGURE 35. The zig-zag sequence (after Wallace [145]).

quency coefficients (which are more likely to be non-zero) before high frequency coefficients. This ordering also emphasizes horizontal and vertical frequencies more than diagonal frequencies and is therefore consistent with the anisotropic spatial frequency response of the human visual system.

The final step in the JPEG algorithm applies a number of conventional compression techniques to remove the redundancy from the ordered frequency terms. The DC values are handled separately from the AC values. The DC values are encoded as the difference from the DC value of a neighboring block. This technique is known as Differential Pulse Code Modulation, and exploits the strong correlation in the average color of adjacent blocks. The AC values are entropy encoded using either a Huffman or an Arithmetic technique, and the result is run-length encoded. The output of these operations forms the final bit stream that is stored as the compressed representation of the image. Further detail on these final techniques can be

found in a paper by Wallace [145].

Synthesis Into the Frequency Domain

This section describes a new algorithm that synthesizes images directly into the DCT frequency domain. The simple, frequency dependent perceptual quality model will then be used to determine the visibility of image error in this domain. This enables the algorithm to focus the effort of the rendering engine on the regions of the image containing the most perceptible artifacts and to channel the remaining error into the areas of the image where it is the least apparent. In addition, the synthesis of images into the DCT domain will allow the algorithm to exploit the prevalence of low frequency energy, and the property of visual masking that have been utilized in the JPEG compression algorithm.

The algorithm begins by dividing the image into a series of 8 by 8 pixel blocks. A sorted list of these blocks is maintained with the block at the head of the list being the next to receive a sample. Once a sample is taken, a least squares technique is used to find the AC_1C_2 frequency representation that best interpolates the data. As more samples are placed within a given block, frequency terms are added to the representation in a low to high frequency order. The algorithm takes advantage of the masking effect by reducing the precision of the illumination calculation as higher frequency terms are added. The vision model described in pages 128–134 of this chapter is used to determine where to put the updated block back on the list of sorted blocks. After the image has been refined to the desired tolerance, a frequency based filtering is performed to minimize any residual aliasing. The remainder of this section provides additional details on these various steps.

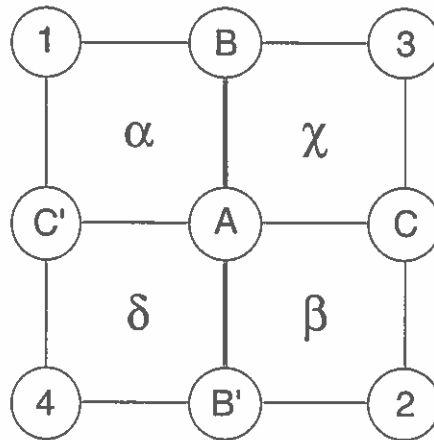


FIGURE 36. Ordering metric used to drive the selection of samples in the quad-tree subdivision method.

Sampling the Scene

Once a block has been selected from the sorted list, a decision must be made about where to place a sample within the block. A fixed sampling sequence is employed for all blocks. This will be seen to minimize the computation required to generate the frequency representation in pages 144–149 of this chapter. As samples are taken they are placed as far as possible from any neighboring samples. This provides the best coverage of the sample space and constrains the interpolation of samples to values that are within the range of the existing samples.

The sampling sequence for each block is based on a quad-tree subdivision technique. The root node of this tree is defined to encompass the entire 8 by 8 block. The first four samples in the sequence are taken through the outermost corners of the block in the order indicated in Figure 36. After these samples have been taken the repeated portion of the sequence is begun. The next sample is placed at position A in the center of the block. A random choice is then made between positions B and C to determine the location of the next sample. The subsequent sample is placed

at whichever location was not chosen initially. This random ordering is employed to evenly distribute samples in both the horizontal and vertical directions.

After samples A, B and C have been taken within a given node, it is subdivided into nodes α , β , χ , and δ which form four new leaves of the tree. A breadth first search is then performed to select the leaf that will receive the next sample. This process then repeats, placing one sample in each new node at either location A, B, or C before returning to take subsequent samples.

There is one additional step that is necessary to complete the sampling sequence. Inspection of the existing sequence shows samples are evenly distributed throughout the interior of the block and across the top and right edges. However, except for the initial samples, no locations are chosen at the left and bottom edges of the block. In order to remedy this problem, whenever a sample is taken at the top or right edge a subsequent location is inserted into the sequence at the opposing edge. Thus, positions A' and B' would immediately follow positions A and B respectively. The first fourteen locations in this sampling sequence are illustrated in Figure 37.

This sequence is stored in a list that is indexed into at varying positions for all blocks. If a block requires a sample position that is not currently on the list, the location is calculated and added to the tail of the list. Otherwise, sample locations are obtained by merely stepping through the existing entries.

Samples that are taken on the edge of an 8 by 8 block are used to refine the frequency representation of all blocks that they border. A cache is employed for each block to store the values of bordering samples returned for neighboring blocks. This is necessary to avoid taking redundant samples since a strict ordering has been

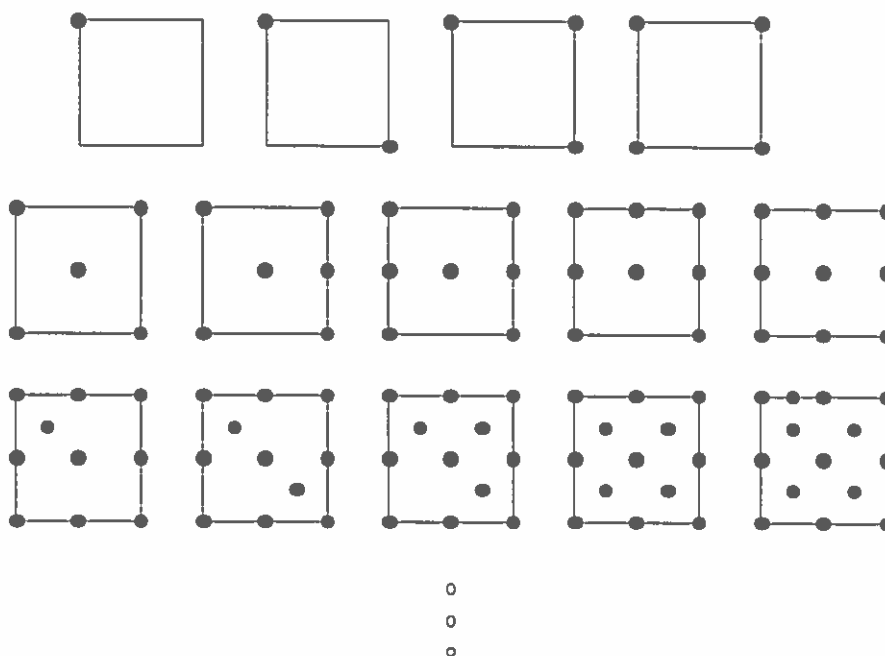


FIGURE 37. The initial 14 sample positions given by the quad-tree sampling method.

placed on the sample sequence.

Computing the Frequency Representation

As each new sample is taken within a given block, its value is first transformed into the AC_1C_2 color space as described in the first three stages of the vision model. The final stage of the model will be applied at a later point in the algorithm. A least squares fit is then used to determine the DCT frequency representation that best interpolates the existing sample values. This representation is iteratively refined with the introduction of each sample and is calculated independently for each of the A , C_1 , and C_2 color channels.

From Equation 44 we see that within a specific channel, a block of the image

can be reconstructed from N cosine frequency terms by the equation

$$f(x, y) = \sum_{i=0}^{N-1} G(u_i, v_i) C_{x, u_i} C_{y, v_i}. \quad (46)$$

In this expression the subscript i imposes a zig-zag ordering on the frequency terms u and v and

$$G(u_i, v_i) = \frac{1}{4} \kappa(u_i) \kappa(v_i) F(u_i, v_i). \quad (47)$$

The sum of the squared error between an arbitrary DCT representation and S sample values is given by the expression

$$\sum_{s=0}^S [f(x_s, y_s) - \alpha(x_s, y_s)]^2, \quad (48)$$

where (x_s, y_s) is the position of a sample within the block, and $\alpha(x_s, y_s)$ is the sampled value. The least squares method is used to find the frequency coefficients that minimize this residual error. These coefficients are given by the equation

$$[X] = [J]^{-1} [Y], \quad (49)$$

where X and Y are $N \times 1$ column vectors, J is an $N \times N$ matrix, and

$$\begin{aligned} J_{ij} &= \sum_{s=0}^S C_{x_s, u_i} C_{y_s, v_i} C_{x_s, u_j} C_{y_s, v_j} \\ Y_i &= \sum_{s=0}^S \alpha(x_s, y_s) C_{x_s, u_i} C_{y_s, v_i} \\ X_i &= G(u_i, v_i). \end{aligned}$$

The actual frequency coefficients $F(u, v)$ are obtained from the column vector X by inverting Equation 47.

The solution of Equation 49 requires the inversion of an N by N matrix for each new sample location. However, the cost of this operation can be reduced since the frequency representation is being iteratively refined as additional samples are taken. The matrix J can be iteratively inverted in $O(N^2)$ time using a rank one inverse matrix modification technique that is tuned to suppress the accumulation of error [116]. This technique is used to calculate the inverse matrix $(J')^{-1}$ that solves the least squares equation for $(S + 1)$ samples from the inverse matrix $(J)^{-1}$ that solves the equation for S samples. This is accomplished by the formula

$$(J')^{-1} = (J + AA^T)^{-1} \quad (50)$$

$$= J^{-1} + \frac{(A - J^{-1}\gamma)A^T J^{-1}}{(A^T J^{-1}\gamma)}, \quad (51)$$

where A is an $N \times 1$ column vector and

$$\gamma = A(A^T A) + JA$$

$$A_i = C_{x_{S+1}, u_i} C_{y_{S+1}, v_i}.$$

The expense of computing the least squares solution is also reduced by observing that the inverse matrix in Equation 49 is solely dependent on the sample locations and not the actual sample values. This is the reason that the same sampling sequence is used for all blocks of the image. The use of a fixed sampling sequence implies that this matrix only needs to be inverted once for each position in the sequence and can be used to calculate the frequency representation of all blocks

of the image. The inverse matrices are stored on a shared list. This list is grown dynamically with the next matrix inverse being calculated for the first block to reach the new sampling position. The frequency representation for a given block is then determined by simply retrieving the inverse matrix from the list and performing a matrix multiplication to yield the frequency coefficients for the A , C_1 , and C_2 color channels.

Adding Additional Frequency Terms

The blocks of an image are initially represented with only a DC term after the placement of a single sample within each block. As the algorithm progresses and additional samples are taken within a given block, terms are added to the blocks representation in a low to high frequency manner. This provides a gradually resolving, bandlimited image representation that interpolates the existing sample values.

The zig-zag pattern specified in the JPEG compression algorithm is used to determine the order in which frequency terms are added. This pattern introduces the terms that are most likely to be non-zero first, and additionally emphasizes the more perceptually significant horizontal and vertical frequency terms. A new term is added to the frequency representation after the placement of every 2 or 3 samples. Since the sampling theorem specifies that sampling be done at twice the highest frequency present in the solution, this prohibits introducing a frequency term until there are enough samples to represent it. It also prevents the presence of singularities in the inverse matrix, which can occur if the frequency coefficients are not uniquely constrained. New terms are added to the frequency representation as

required, up to the maximum 64 terms. No new terms will be added after the 64, but the existing terms can continue to be refined with the placement of additional samples.

The calculation of an additional frequency coefficient requires the expansion of the $(J)^{-1}$ matrix and the Y column vector. Frobenius' relation [13] is used to compute the expanded inverse matrix. This relation allows the matrix expansion to also be performed in $O(N^2)$ time. Frobenius' relation shows that the inverse matrix $(J_{N+1})^{-1}$ that computes the least squares solution for $(N + 1)$ terms can be calculated from the inverse matrix $(J_N)^{-1}$ for N terms by the equation

$$\begin{aligned} (J_{N+1})^{-1} &= \begin{bmatrix} J_N & B \\ B^T & D \end{bmatrix}^{-1} \\ &= \begin{bmatrix} (J_N)^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} (J_N)^{-1} B \Delta^{-1} B^T (J_N)^{-1} & -(J_N)^{-1} B \Delta^{-1} \\ -\Delta^{-1} B^T (J_N)^{-1} & \Delta^{-1} \end{bmatrix}, \quad (52) \end{aligned}$$

where B is an $N \times 1$ column vector and

$$\begin{aligned} \Delta &= D - B^T (J_N)^{-1} B \\ B_i &= \sum_{s=0}^S C_{x_s, u_i} C_{y_s, v_i} C_{x_s, u_{N+1}} C_{y_s, v_{N+1}} \\ D &= \sum_{s=0}^S C_{x_s, u_{N+1}} C_{y_s, v_{N+1}}. \end{aligned}$$

The resulting inverse matrix can then be multiplied by the expanded Y vector to yield the vector X that contains the $(N + 1)$ frequency coefficients as shown in Equation 49. The 64 term versions of J and Y are maintained to yield B , D and the expanded Y vector in an expedient manner.

The first section of this chapter discussed how visual masking reduces the detectability of high frequency error in the presence of other high frequency information. This aspect of the human visual system is exploited in the JPEG algorithm by reducing the accuracy of the higher frequency coefficients. The resulting error does not adversely affect the perceived image quality. This effect is similarly employed by the new rendering algorithm to determine the appropriate sample accuracy when computing each of the frequency terms for a given block.

A Monte Carlo ray tracer is used as the basis of this rendering algorithm. This permits physically correct illumination calculations, as well as providing a means to vary the sample accuracy. In a Monte Carlo ray tracer, numerous rays are spawned from each surface intersection to evaluate the amount of light that is incident on the surface. The spawning of many rays will produce an accurate estimate of the incident light and thus return an accurate sample of the light reflected from the scene. However, as the number of spawned rays decreases the initial samples become increasingly noisy.

The new rendering algorithm exploits the affect of visual masking by reducing the number of rays that are spawned from each surface intersection as more frequency terms are added to a block's representation. This produces error in the high frequency coefficients. However, due to the presence of high frequency image content, the error will be imperceptible and the expense of the rendering algorithm will be decreased. Figure 38 shows a plot of the two-dimensional function that is used to attenuate the spawning rate based on the latest frequency term that has been added to the representation. This function is modeled after the quantization table used in the JPEG algorithm.

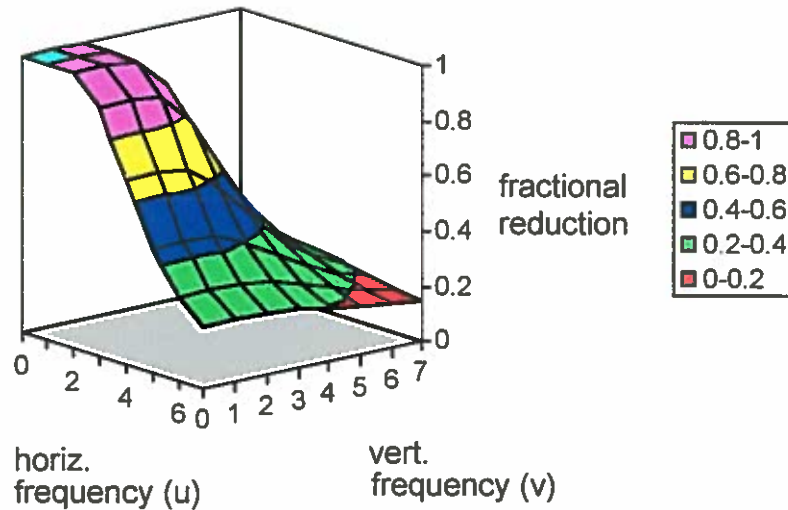


FIGURE 38. Rate at which the number of rays spawned decreases as higher frequency terms are added.

Choosing the Next Block

Within a given image, all regions of the image do not generally contain the same amount of spatial detail, nor do they require the same number of frequency terms to be accurately represented. This can be seen by inspecting Figure 32, that was used to demonstrate the energy packing ability of the DCT. This image shows that most regions of the image can be accurately represented using only a limited number of low frequency terms. Only the few areas of the image that contain high frequency energy need the dense sampling that is required to reconstruct the high frequency detail.

The perceptual sensitivity of the human visual system also has a significant effect on the accuracy that is required in each region of the image. Areas of the image where the error is very apparent should receive a higher sampling rate than areas where the error is less perceptible. The areas of the image that contain the

most visible artifacts should also be scheduled for processing first in order to produce the most visually pleasing images in the allotted time.

In order to achieve these goals, a priority is calculated for each block in the image. This priority is used to arrange the sorted list of blocks according to the order in which they should receive an additional sample. The priority of a given block is updated after a sample is taken in the block and its frequency representation has been refined.

The priority of a block is based on the application of the visual metric (described in pages 128–134 of this chapter) to determine the perceptibility of image error. The first three stages of this model were applied to transform the sample values into the AC_1C_2 color space. The remaining stage is applied as part of the priority calculation and does not affect the actual image representation.

The first step in the priority calculation is to filter the frequency representation of each AC_1C_2 channel by the associated spatial frequency response of the visual system as described in the final stage of the visual model. Next, the root mean squared change in the filtered frequency representation that resulted from the current sample is calculated. This value is a measure of the perceptual significance of the aliasing that was removed by the current sample. It is used as an indicator of the need for further refinement. The validity of this indicator is based on the assumption that blocks in which a large amount of aliasing has occurred tend to need refinement the most and blocks in which a small amount of aliasing has occurred tend to be well defined. This value is combined with decreasingly weighted previous values and an initial uncertainty measure to generate the final priority. The initial uncertainty measure is added to this calculation to assure that blocks

whose frequency representation is not changing still receive an adequate minimum sampling. The priority is then used to sort the current block back onto the list of blocks such that the block at the head of the list should be the next to receive a sample.

Displaying the Final Result

The frequency based synthesis technique described in the preceding sections generates and refines a functional representation of an image. This representation interpolates the sample values and can be used to construct a full resolution image even at the early stages of the algorithm. Before the image is displayed, however, it should be low pass filtered to remove any residual aliasing due to under-sampling.

In the initial stages of the algorithm, aliasing can occur in the low frequency terms. As the algorithm progresses and more samples are taken within a block the aliasing decreases and is only present in the high frequency terms. In order to minimize this aliasing a lowpass filter with a variable cutoff is applied to the frequency representation of each block.

The frequency coefficients of each block are transformed back into a linear luminance space before a simple two-dimensional Butterworth filter is applied. The cutoff of this filter increases with the square root of the number of samples that have been taken. As the sampling rate increases the width of the filter grows large and its effect is greatly diminished. The final image is then produced by transforming the filtered coefficients back into the spatial domain using the inverse DCT in Equation 44.

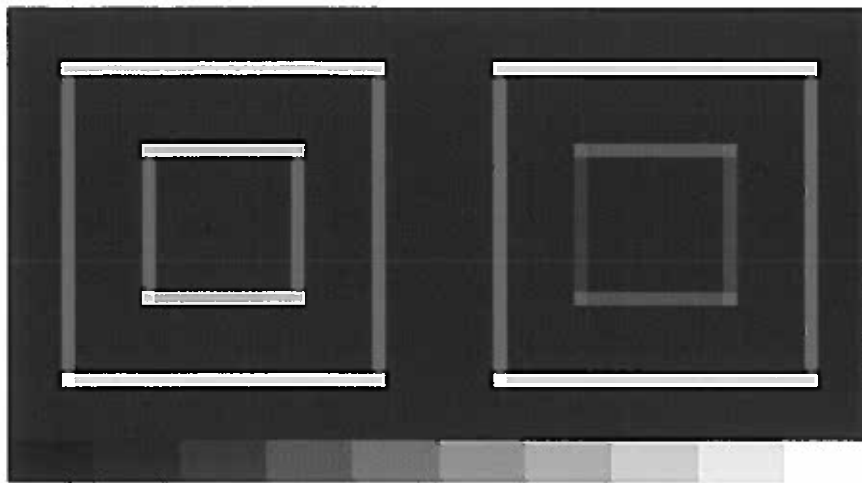


FIGURE 39. Sampling density for the contrast squares in Figure 27.

Results

The image displayed in Figure 27 was created using the new frequency based ray tracing technique. The sampling densities that were used in generating this image are shown in Figure 39. In this figure, lighter areas indicate regions of denser sampling. This example consists of two sets of squares. The intensity of the interior square in each set is an equal luminance difference from the surrounding region. However, due to the nonlinear contrast sensitivity of the human visual system, the interior square on the dark surround is more apparent than the interior square on the bright surround.

Figure 39 shows that the new algorithm correctly detected that the edges of the squares were the only areas of the image containing high frequency content and, therefore, required a higher sampling rate. It additionally shows that more samples were used to reconstruct the interior edge in the left set of squares than in the right. This demonstrates how the algorithm incorporates nonlinear contrast response to determine that edge artifacts are more apparent in dark regions than in bright.

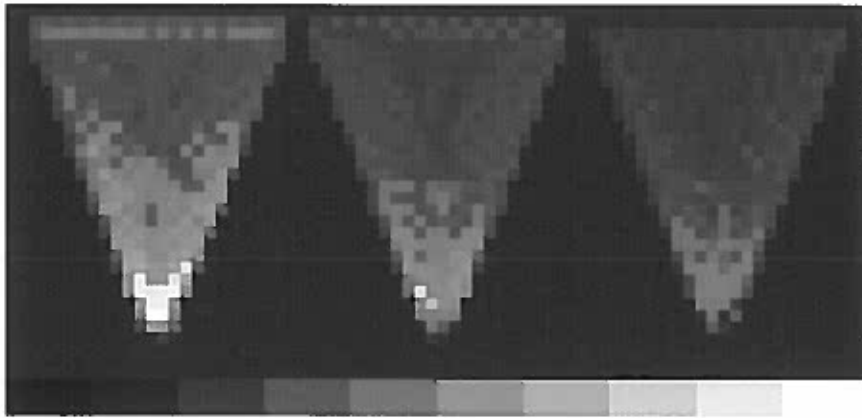


FIGURE 40. Sampling density for the acuity wedges in Figure 30.

The acuity wedges shown in Figure 30 were also generated using the new rendering algorithm. The sample densities that were employed are displayed in Figure 40. In this example the wedges increase in spatial frequency from top to bottom. The left wedge varies along the A axis of AC_1C_2 space, the middle wedge varies along the C_1 axis, and the right wedge varies along the C_2 axis. The amplitude of the oscillation in each wedge is equal. The fact that the oscillations can be resolved further down the achromatic wedge than the chromatic wedges illustrates the differences in the cutoffs of the visual system's spatial frequency response curves in these channels.

Figure 40 shows that the frequency based rendering algorithm has detected that more samples are required to accurately reconstruct the higher spatial frequencies at the bottom of the wedges. However, less samples are required to reconstruct the C_1 and C_2 wedges than are required for the achromatic wedge. The resulting artifacts are not perceptible to a human observer. This demonstrates that the algorithm is able to exploit the fact that achromatic differences are more visually apparent than chromatic differences.

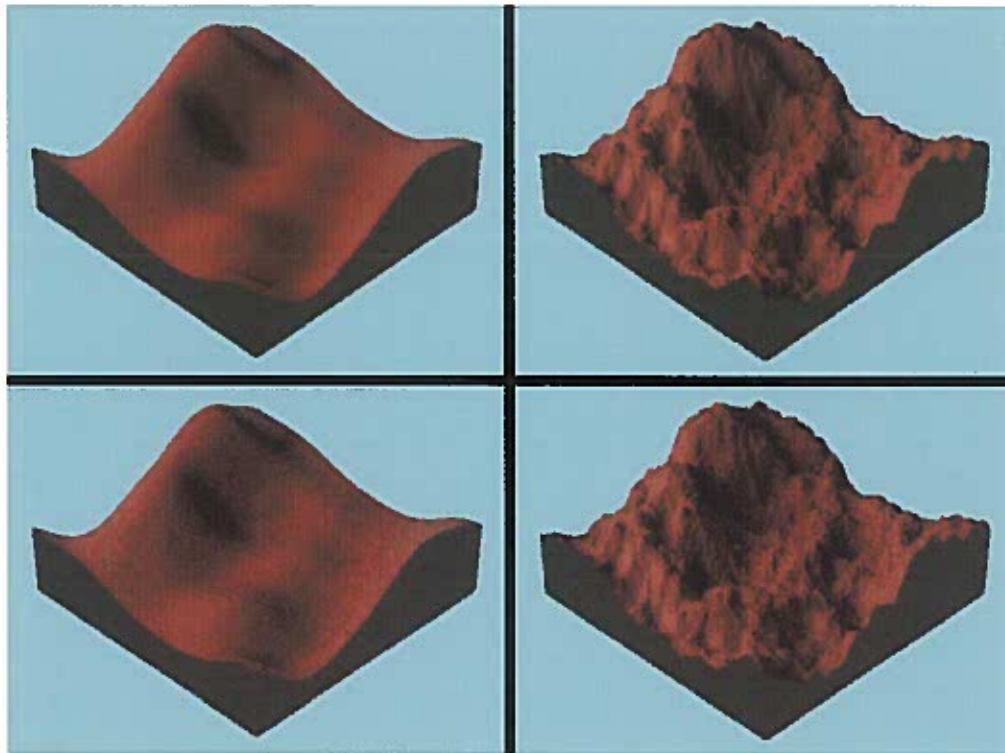


FIGURE 41. Images with different spatial frequency content rendered by spawning 1000 rays for every intersection (top row) and 100 rays for every intersection (bottom row).

The fractal mountains in Figure 41 were rendered using a version of the frequency based ray tracer that did not attenuate ray spawning as discussed in pages 147–149 of this chapter. The geometry for these mountains was created using a Fourier synthesis technique to produce two-dimensional fractional Brownian motion. The frequency distribution has been cut off at a lower frequency for the mountain on the left than the mountain on the right. The lighting for this scene is very directional and is produced by a small area light source. The nature of this lighting creates a significant amount of noise in the Monte Carlo intensity calculation.

In the top two images, 1000 rays were spawned at each surface intersection to evaluate the incident light. In the bottom two images, only 100 rays were spawned.

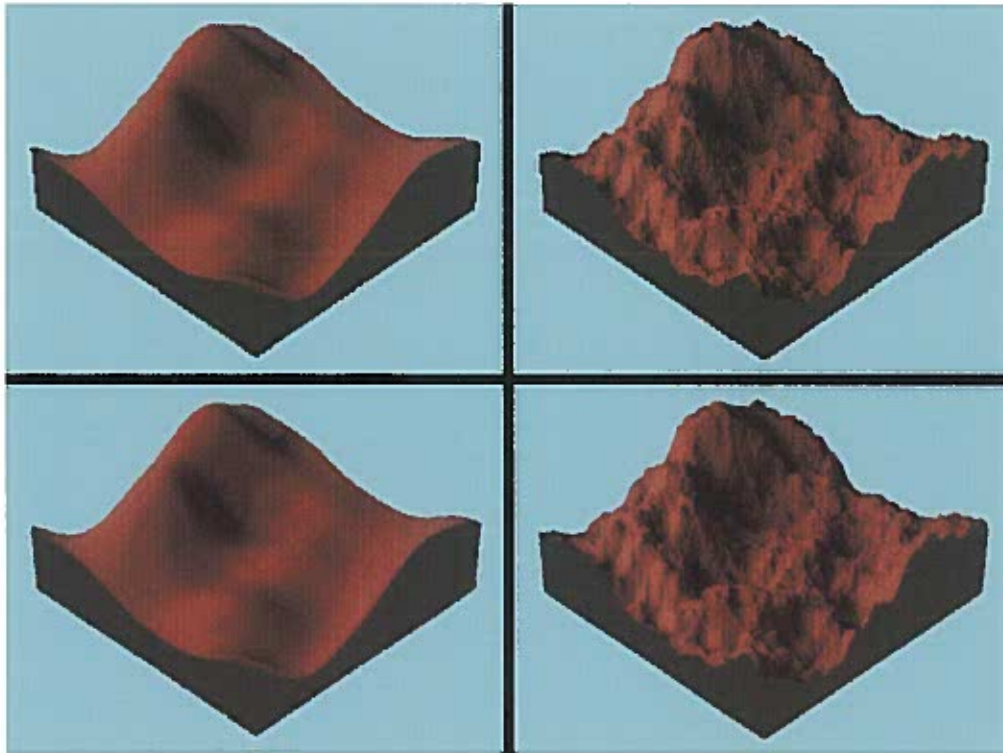


FIGURE 42. Images with different spatial frequency content rendered using a traditional Monte Carlo ray tracer and one ray per pixel (top row). Same images rendered using the frequency based ray tracer and one half ray per pixel (bottom left) and one ray per pixel (bottom right).

The resulting noise is apparent in the image of the low frequency mountain in the bottom left. This demonstrates that masking affects the visibility of noise in a similar manner to how it affects the visibility of quantization error. It additionally shows that fewer rays can be spawned from surface intersections when high frequency image detail is present without adversely affecting the perceived image quality.

The final demonstration, shown in Figure 42, compares the results of the complete frequency based ray tracer with the results of a standard Monte Carlo ray tracer. The top two images were produced by the standard ray tracing algorithm. These images were generated by taking one sample per pixel and spawning 1000

rays from each surface intersection. In these images, note that noise is visible in the interior of the low frequency mountain and that aliasing is evident around the circumference of both mountains.

The bottom two images were created by the complete frequency based ray tracing algorithm. This algorithm attenuates the number of rays spawned from surface intersections as discussed in pages 147–149 of this chapter. The low frequency mountain in the bottom left of the figure was generated by taking an average of one sample for every two pixels. There is less noise and aliasing present in this image than in the image produced by the standard technique even though half as many samples were used in its production. This demonstrates the benefit of the band limited interpolation that is done by the frequency based technique. The effect of this interpolation is somewhat similar to the ray cache developed by Ward *et. al.* [149], but is performed in image space.

The high frequency mountain displayed in the bottom right was created by taking one sample per pixel. However, as more frequency terms were added to the solution, the number of rays spawned from each surface intersection was rapidly reduced from an initial rate of 1000 rays according to the rolloff curve depicted in Figure 38. The imperceptibility of differences between this image and the image created by the standard ray tracer shows that the affect of visual masking allows fewer rays to be spawned in high frequency regions without reducing the perceived quality of the image.

Summary

This chapter has presented the design of a new ray tracing algorithm that synthesizes images into the spatial frequency domain. This technique, for the first time, allows the spatial frequency characteristics of an image and its error to be directly exploited within an image synthesis algorithm. The availability of this spatial frequency representation enables the new algorithm to use elements of human perception and modern compression algorithms to reduce the number of samples that are required to produce high quality images.

The new ray tracing algorithm employs a simple, frequency dependent perceptual quality model to direct the placement of samples. This model consists of an initial receptor stage, followed by a contrast nonlinearity, a matrix to an opponents color space, and a final filtering by the spatial frequency response of each color channel. The use of this model allows images of a given visual quality to be produced with fewer samples than existing techniques. In deciding where to place samples, the new algorithm takes more samples for luminance differences in regions of low intensity than in regions of high intensity because of the visual system's nonlinear contrast response. Achromatic variations also receive more samples than chromatic variations due to differences in the cutoffs of the perceptual spatial frequency response for these channels.

The new algorithm synthesizes images into the DCT frequency domain that is used in the first stage of the JPEG compression algorithm. The synthesis of images into this domain reduces the number of steps that are necessary to compress the final image. Additionally, it allows the algorithm to exploit a number of features of this compression algorithm. One feature that is particularly exploited is the

frequency quantization that is made possible by the affect of visual masking. This property of the visual system is used in the new rendering algorithm to determine the appropriate number of rays to spawn from each surface intersection when evaluating the different frequency terms. Low frequency regions of the image are produced by interpolating a few high quality samples, each of which is averaged from a large number of spawned rays. Conversely, high frequency regions are produced with many samples, but the quality of these samples is reduced by attenuating the spawning rate. The resulting high frequency noise is not perceptible to a human observer.

The synthesis of images into the frequency domain has also been seen to offer a number of advantages when considered solely as a sampling and reconstruction technique. The detection of changes in the frequency spectrum of an image provides a novel means of detecting aliasing and directing the effort of an adaptive sampling algorithm. In addition, the least squares technique that is used to resolve images from low to high spatial frequency detail offers a unique method of reconstructing a bandlimited image from a set of nonuniformly spaced samples. Finally, the direct filtering of the frequency representation of an image provides a rapid means of removing aliasing in accordance with where it is occurring within the frequency spectrum.

The work described in this chapter is novel in several respects. However, there are areas of the algorithm that could be improved. A number of techniques have been employed in this chapter to reduce the cost of synthesizing images into the frequency domain from $O(N^3)$ to $O(N^2)$. In the following chapter a new technique will be presented that reduces this cost to $O(\log N)$. This performance enhancement will significantly improve the practicality of frequency based image synthesis.

In addition, the simple perceptual metric described in this chapter has been enhanced in subsequent work. This work led to the development of the high speed, color visual difference predictor presented in Chapter VII. This new image quality model incorporates a number of advanced features, and will enable a more refined application of the affect of visual masking. This new model will be incorporated into the perceptually based adaptive sampling algorithm discussed in the following chapter.

CHAPTER X

A PERCEPTUALLY BASED ADAPTIVE SAMPLING ALGORITHM

In this chapter a new rendering algorithm is developed. This algorithm utilizes a high speed, color visual difference predictor to make perceptual quality assessments during the construction of a synthetic image. The use of a perceptually based quality metric enables the algorithm to automatically detect the regions of the image that contain visible artifacts. By focussing effort on these areas, an image of a given visual quality can be produced with fewer samples and a lower overall execution time than with previous techniques. In addition, this algorithm allows the user to select the perceptual quality of the output image. This feature eliminates the guesswork involved in halting a rendering, and allows different scenes and shading techniques to produce consistent, predictable results.

In order to accomplish these tasks the new algorithm is required to make many, iterative predictions of the perceptual image quality. These predictions must be performed very rapidly in order to avoid offsetting the gains achieved by the more efficient sample distribution. Chapter VII has already discussed the design of a high speed, color visual difference predictor. This predictor is an integral part of the new rendering algorithm. A novel image synthesis technique is developed to render images directly into a Haar wavelet representation. This representation corresponds with the cortex transform performed at an early stage of the visual metric. The

synthesis of images directly into this frequency based domain facilitates expedient predictions of image quality. This technique has the additional benefit of allowing the algorithm to exploit the statistical properties of natural images to estimate values in unrefined regions.

Two variations of the new perceptually based sampling method are presented. The basic perceptual algorithm is discussed first. This technique employs the image quality model to place each sample at the location of the most perceptible error. This simplified version of the final approach is used to discuss key features of the algorithm and provides a basis for direct comparison with existing objective adaptive sampling techniques. The enhanced perceptual algorithm is then described. This complete version of the adaptive sampling algorithm employs a number of performance enhancing techniques to further improve the overall execution time of the perceptually based algorithm.

This chapter is divided into five major sections. The first section describes the basic perceptually based adaptive sampling algorithm. This section also discusses how the frequency statistics of natural images are used to make informed guesses in unrefined areas. In the second section the final enhanced version of the perceptual algorithm is developed. The third section presents the results of the new perceptually based algorithm. In section four the new algorithm is compared with two previous sampling techniques. This comparison covers both the required sampling rates and the execution time that is necessary to produce images of a given visual quality. Finally, the major developments contained in this chapter are summarized in section five.

Basic Perceptual Algorithm

An adaptive sampling algorithm has been developed that is based on the perceptual image quality model described in Chapter VII. This algorithm receives sample values as input and specifies the placement of samples at the image plane as output. The goal of this basic adaptive algorithm is to iteratively place each sample at the location that currently contains the most visible error.

The key to developing this perceptually based adaptive sampling algorithm comes from two primary insights. The first is that an estimate of the image and its error can be used to construct two boundary images that may be used as input into a visual difference predictor. The output of this difference prediction can then be used to direct the placement of subsequent samples. The second insight is that a given sample only affects the value and accuracy of a very limited number of terms at each pyramidal level of a Haar wavelet image representation. This fact makes the algorithm tractable because it implies that only a small number of operations are necessary to refine the image approximation, its error estimate, and the visual difference prediction for any given sample.

The algorithm proceeds through a few basic steps. First, as samples are taken of the scene, a Haar wavelet image approximation is generated and refined. Next, a multi-resolution error estimate is developed and similarly refined. This error estimate is an objective measurement of the image accuracy and is expressed in terms of the variance of the detail terms in the Haar representation. The image approximation and error estimate are then used to construct two boundary images which serve as input to the visual difference predictor. The output of the difference predictor is accumulated in a hierarchical tree. The nodes of this tree specify the maximum

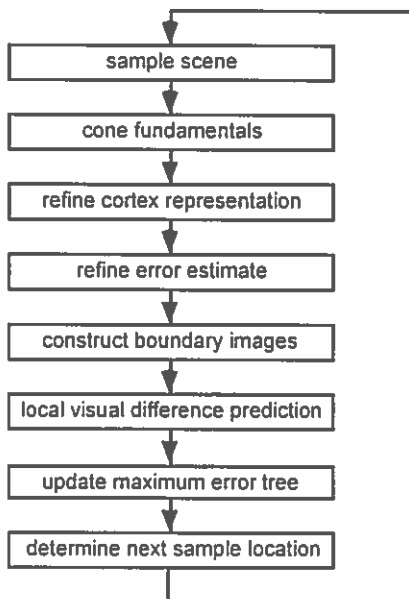


FIGURE 43. Block diagram of the basic adaptive sampling algorithm.

visual difference present at the current nodes and the children below it. This tree is traversed choosing the branch with the largest visual difference in order to determine the location on the image plane with the greatest perceptual error.

A block diagram of the algorithm is illustrated in Figure 43. As samples are taken their values are first transformed from CIE XYZ to SML space in the step labeled *cone fundamentals*. The Haar image representation and its error estimate are constructed in this space.

In the *refine cortex representation* stage the Haar image approximation is created and refined. This is done through a technique similar to the “splat and pull” method used by Gortler, *et. al.* [60]. The Haar image representation is stored in a quad-tree data structure. The leaves of this structure are defined to contain the intensity of single pixels in the image plane and the interior nodes contain the lower resolution lowpass and detail terms of the Haar representation. As a sample is passed

into this stage it is “splatted” at the leaf containing the sample. The intensity at this leaf is simply the average of the samples taken within the pixel it is defined to cover. The lower resolution lowpass and detail terms are generated by “pulling” the updated leaf intensity up through the tree. During this process, if all children of a node contain at least a single sample, then the lowpass and detail terms are given by a local application of the Haar transform contained in Equation 36. If only a single child contains a sample, then the detail terms are left undefined and the lowpass term is set equal to the lowpass of the child containing the sample. If only two or three children contain samples, then a simple scheme is used to fit the lowpass and one or two detail terms, respectively, to the values of the defined children. In this manner the image representation is gradually resolved as samples are taken of the scene. It is also worth noting that this process is very fast since the addition of a sample only requires the updating of a single path up the tree.

Figure 44 contains a demonstration of the results of this technique. The number of samples used to construct the inset images increases from left to right, and from top to bottom of this figure. As more samples are taken of the scene, new detail terms are added and the existing terms are refined. This process resolves the image from low to high spatial frequencies.

At the next step labeled *refine error estimate*, the error of the current Haar approximation is determined. This process is similar in some respects with the algorithm described by Painter and Sloan [107]. The error estimate is expressed in terms of the variance of the lowpass and detail coefficients. For leaf nodes containing at least two samples the variance of the (lowpass) pixel approximation ($V[leaf]$) is



FIGURE 44. Illustration of the *refine cortex representation* stage. The number of samples used to construct these images increases from left to right, and from top to bottom.

given by the relation

$$V[leaf] = \frac{V[samples]}{\# samples}, \quad (53)$$

where $V[samples]$ and $\# samples$ respectively indicate the variance and quantity of the samples taken in the leaf region (see pages 52–55 of Chapter V and references [16, 81]). The error of the lowpass and detail terms in the interior nodes is defined with respect to the error of their children. If the variance is defined for all children of a node, then the variance of the lowpass and detail terms at the node ($V[interior]$) is determined by the expression

$$V[interior] = \frac{1}{16} \sum_{i=1}^4 V[child_i]. \quad (54)$$

In this equation ($V[child_i]$) is the variance of the lowpass terms for the i^{th} child and the index i sums over all four children. This result comes from the rule $V[\sum^i a_i x_i] = \sum^i a_i^2 V[x_i]$ and inspection of Equation 36. If the error is not defined for all children of a node and at least 2 samples have been taken, then the variance is given by the variance of the samples taken within the node divided by the number of samples in the node. As in the case of refining the Haar representation, updating the multi-resolution error estimate requires that only a single path in the tree be modified for the addition of each sample.

The next stage in the algorithm labeled *construct boundary images* is concerned with defining the two input images for the visual difference predictor. In an ideal situation the accuracy of the current image would be determined by comparing an exact version of the image to the current approximation. However, an exact version of the image is not known during the progression of the rendering algorithm. Therefore a statistical technique is employed that uses the current image approximation and an estimate of its error to construct two images that represent the boundaries of what the actual image could be. These boundary images are then used as input to the difference predictor, and on average the perceptual difference between these two images is equal to a subjective measurement of the accuracy of the current image approximation.

The boundary images are constructed in a wavelet representation in order to facilitate the application of the quality model. In this representation the input images are described by the magnitude of their detail coefficients which are used to determine the local contrast at an early stage in the perceptual model. The details for the two boundary images are derived from the details in the current image

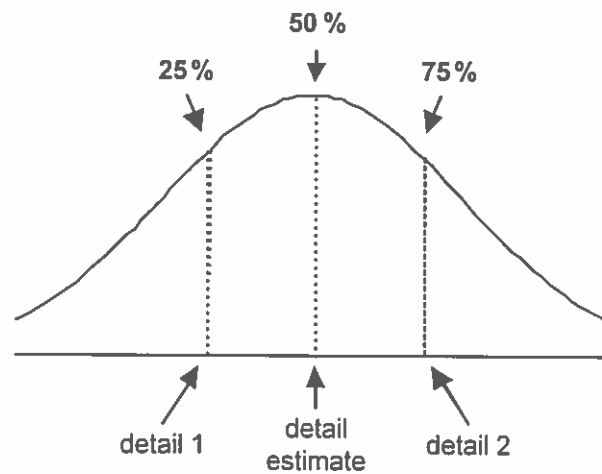


FIGURE 45. Illustration of the selection of detail terms at the *construct boundary images* stage of the adaptive algorithm.

approximation and the variance of those details. The magnitude of the approximated detail specifies a mean value and the square root of the variance defines the spread of the standard deviation curve. The magnitudes of the details for the boundary images are taken from the 25% and 75% points on this curve. This process is illustrated in Figure 45. In this manner two boundary images are specified which should contain the true value 50% of the time. The boundary images are organized so that image 1 contains the detail of minimum energy contrast and image 2 contains the detail of maximum energy contrast. These boundary images can also be rapidly refined for the introduction of each new sample. Since a sample only affects the image approximation and error estimate along a single path in the tree, the detail terms for the boundary images only need to be updated along this path as well.

A *local visual difference prediction* is performed at the updated nodes in the next step of the algorithm. This is accomplished by passing the detail terms of the boundary images through the *chromatic aberration* to the first portion of the *distance summation* stages of the image quality model. These stages are highlighted

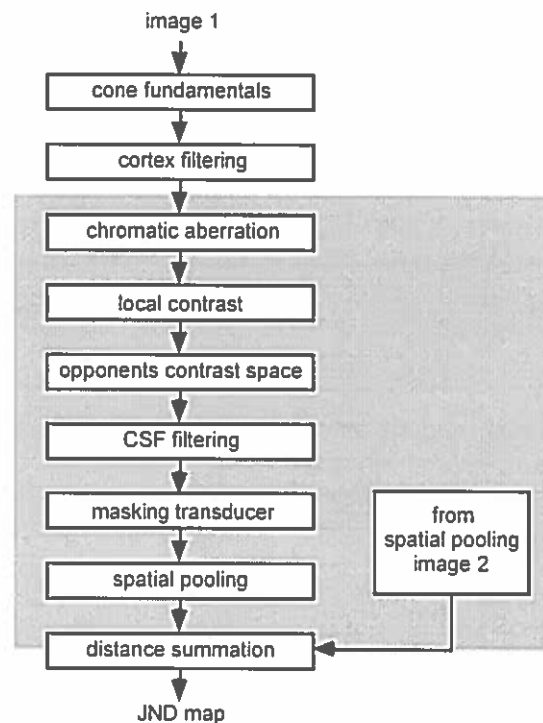


FIGURE 46. A block diagram of the image quality model. The stages of the model employed in the *local visual difference prediction* are contained in the gray region.

in Figure 46. The output of this operation is a measure of the local visual difference (LVD) contribution at each of the updated nodes. This difference contribution is defined by Equation 39 in Chapter VII. It equals a local sum of the perceptual distance between the boundary images across each detail and color channel raised to the 2.4 power. During this process the transducer outputs at the current node are stored in the tree for fast re-use in the pooling stages of neighboring nodes.

The next step, labeled *update maximum error tree* in Figure 43, accumulates the updated LVD's into a hierarchical tree. At each node a measure of the maximum visual difference (MVD) is stored. This value is recursively defined to be the sum of the LVD at a node and the largest MVD contained in either of the four child nodes. The magnitude of the MVD indicates the largest cumulative perceptual

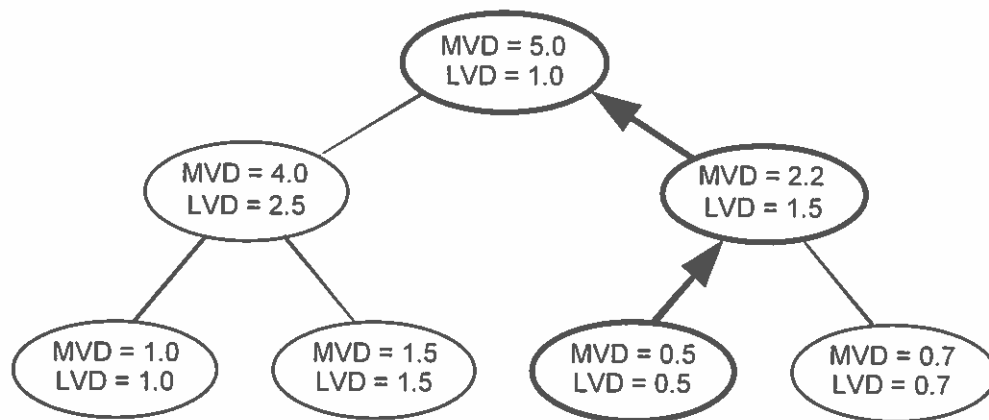


FIGURE 47. An example of the *update maximum error tree* stage. The updated path is denoted by the bold arrows. The values of the maximum visual difference (MVD) and the local visual difference (LVD) are indicated for a number of nodes in the quad-tree. The quad-tree is represented by a binary tree for simplicity.

error contained at and below a given node in the tree. The MVD of the root node is raised to the $1/2.4$ power and represents the magnitude of the most perceptible error contained at any location within the image plane (in units of just noticeable differences). This value is tested against a user specified threshold to determine when the image has been rendered to the desired visual quality. An example of this process is depicted in Figure 47.

In the final stage, labeled *determine next sample location* in Figure 43, a sample location is selected at the point in the image plane containing the most perceptible error. The location is selected by traversing the quad-tree in a top-down fashion and, at each node, selecting the branch of maximum visual difference. This traversal continues until a leaf node is encountered or an interior node is found which contains less than eight samples. If a leaf node is reached, a sample is randomly placed within it. If the traversal stops at an interior node, then a sample location is chosen randomly from a child's quadrant so that the number of samples in each child node

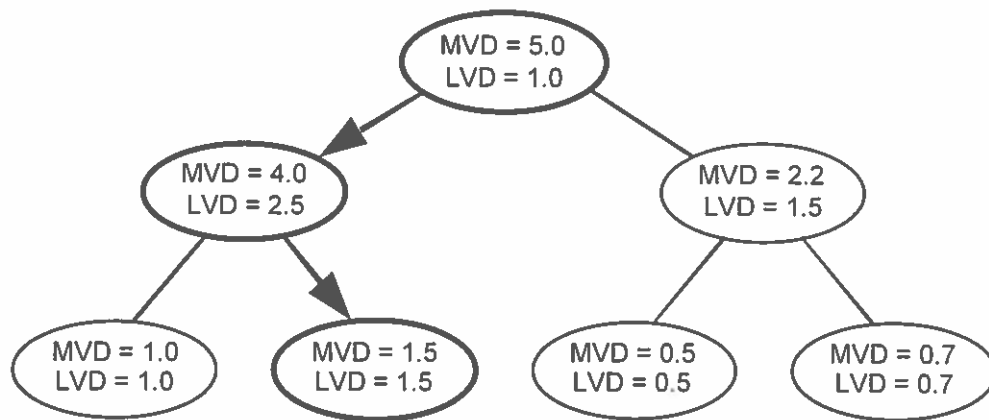


FIGURE 48. An example of the *determine next sample location* stage of the adaptive algorithm. The traversed path is indicated by the bold arrows.

is balanced to a tolerance of one sample. An example of this traversal is contained in Figure 48.

The discussion thus far has assumed that only a single path in the quad-tree is affected by a given sample. However, this is not strictly the case. Due to the local contrast and spatial pooling stages of the vision model the modification of one node in the quad-tree can have an affect on the visual difference at neighboring nodes. One solution to this problem is to update multiple paths up the tree. However, this approach was deemed too expensive. Instead the problem is effectively solved by adding a small amount of randomness to the traversal of the maximum error tree. As each node in the tree is visited, there is some likelihood a neighboring node will be chosen instead. In this manner, if a particular path is traversed often, there is a chance of selecting neighboring paths. This creates the opportunity to incorporate updated values into the local contrast and spatial pooling calculations for these paths.

The algorithm continues recursively until the maximum error of the root node drops below a specified tolerance. The output image is reconstructed by simply

doing an inverse Haar transform of the image representation and converting pixel values from SML to the frame buffer space. This technique can also be used to construct an iterative display of the image during the progression of the algorithm.

Selecting Values for Unknown Quantities

A difficulty with adaptive sampling algorithms that are based on the sample variance is knowing when and to what extent to believe the error estimate obtained from the samples. This is especially true for the hierarchical variance estimation scheme described in this chapter. If the first two samples obtained from the scene return exactly the same values and therefore have zero variance, can we conclude that the image has been computed exactly and stop? If the image has been sampled densely and two samples from within a particular pixel of the image plane are the same, can we say that the intensity of the pixel has been computed correctly? A person analyzing these two situations would certainly believe that the scene has not been adequately sampled in the first case, but would probably be willing to stop sampling in the second case. The reason for this difference stems from the statistics of natural images.

A number of authors have analyzed the statistics of images commonly encountered in nature [46, 47, 123, 128]. These authors have found that the frequency spectra of natural images is not random, but tends to be highly correlated and contains a $1/f$ drop-off in the magnitude of the frequency terms. This fact is illustrated for the example natural scene depicted in Figure 49. A Fourier transform was applied to the two-dimensional intensity values of this image. A plot of the magnitude of the frequency spectrum that results is contained in Figure 50. The $1/f$ decline in

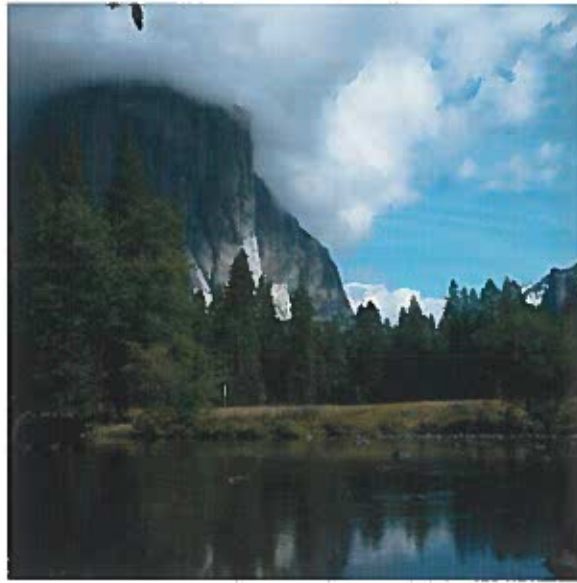


FIGURE 49. An example image of a natural scene.

the magnitude of the coefficients is clearly evident in this graph.

Therefore, if only two samples have been taken of a scene, we have just begun to compute the low end of the frequency spectra. Based on our experience with images found in the natural world, we know that an average image contains higher frequency detail, and therefore believe that the scene has not been adequately sampled. Thus, we have some a priori knowledge about the error of an image approximation. If a portion of the frequency spectra has not been computed, then, on average, the approximation of the image contains an amount of error that is equivalent to the $1/f$ magnitude of the uncomputed spectra. This fact is simple to exploit in the adaptive sampling algorithm because the objective error is estimated for each of the frequency coefficients of the image.

We can also draw upon the statistics of natural scenes when we must determine the error contribution of the chromatic channels. The frequency content of naturally occurring spectral reflectances is known to be very low [89]. This means

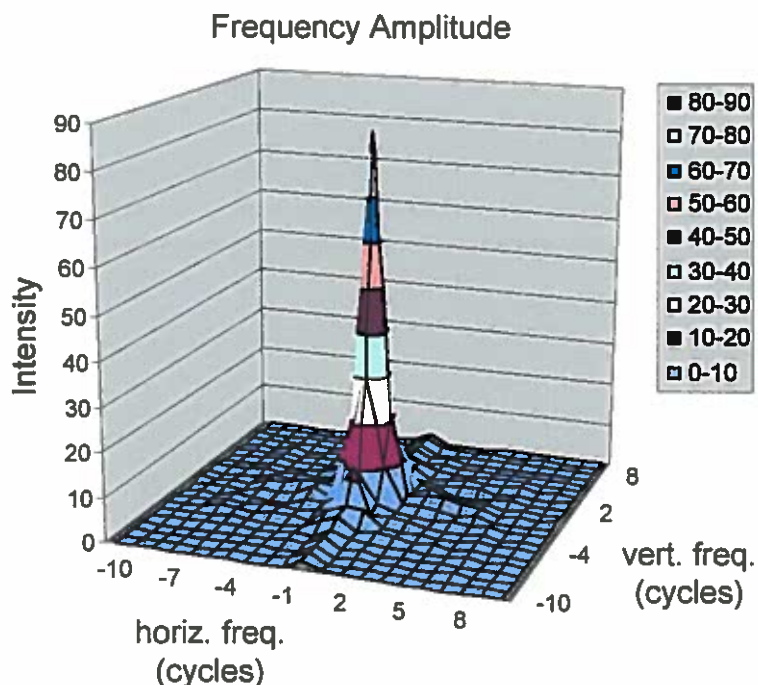


FIGURE 50. A plot of the two-dimensional frequency spectrum of the image contained in Figure 49.

that reflectances are more likely to be uniform across the spectrum than they are to be spiky. The result of this is that the average color in the natural world is quite desaturated. This is illustrated for an example scene in Figure 51. This figure contains plots of the RGB pixel intensities from the image in Figure 49 along the red/green and green/blue axes. These intensity values can be seen to be strongly grouped along the central achromatic axis. This implies that in the absence of other knowledge about the chromatic content of an object, setting the difference contribution of the chromatic channels close to zero is as good a choice as one can make. Again, this statistic is easy to exploit because the adaptive algorithm stores color information in an opponents representation.

As previously mentioned, the statistics of natural images are employed within

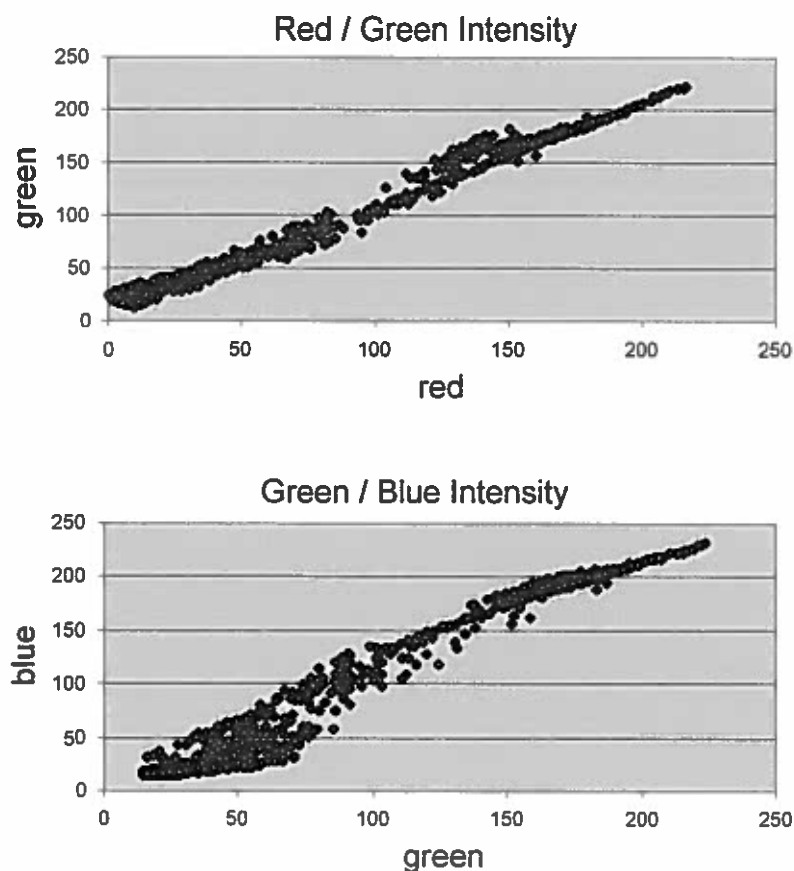


FIGURE 51. RGB pixel intensities from Figure 49, plotted along the red/green axis (top) and green/blue axis (bottom).

our adaptive sampling algorithm. This is accomplished by initializing the two boundary images to a uniform gray for one, and a statistically average image for the other. The visual difference predictor is run on these two input images and the output is used to seed the visual difference at each node in the quad-tree. Initially, the estimated visual difference of the rendering is based on the comparison of the gray and statistically average image. As the algorithm progresses and the image approximation and error estimate is calculated at new nodes in the tree, the visual difference based on the average statistics is traded for the visual difference that is

based on the variance and content of the scene samples.

Enhanced Perceptual Algorithm

This section describes the final enhanced version of the perceptually based adaptive sampling algorithm. This enhanced perceptual algorithm incorporates a number of elements that are omitted in the basic approach. These modifications significantly improve the execution speed of the algorithm by limiting the number of times and locations at which the visual metric is applied.

The basic perceptual algorithm is designed to provide a direct comparison with existing adaptive sampling techniques that are based on objective assessments of the image accuracy (as well as to establish a simple framework in which to discuss the key features of the method). For this reason, the basic technique evaluated the image quality model to direct the placement of each sample. This is similar to a number of objective adaptive sampling algorithms that compute a simple refinement test each time a sample is placed [81, 107, 118]. However, the additional expense of the perceptually based refinement test limits the practicality of this approach.

The remainder of this section will describe three modifications of the basic approach that are used to create the enhanced perceptual algorithm. These elements are designed to increase the number of samples that are placed for each evaluation of the quality metric. This is done in a manner that does not adversely affect the number of samples required to produce an image of a given visual quality. Similar enhancements could be applied to a number of prior objective adaptive sampling algorithms. However, they would have a minor impact on the execution time of these techniques due to the minimal overhead of the simple objective error metrics.

The first modification to the basic perceptual method is to perform a preliminary low density sampling of the image prior to initiating the adaptive sampling algorithm. This modification is somewhat obvious because at the very early stages of sampling the image there is very little information on which to base a quality assessment. By taking an initial sampling, the error characteristics and frequency content of the image can be reasonably determined before undergoing the expense of executing the image quality model. In addition, a minimum sampling of the scene is necessary in order to insure that no relevant image detail is lost. The basic algorithm determines this minimum sampling rate based on the perceptibility of error in a statistically average image. The appropriate choice of this rate can be determined by using the basic algorithm to render a uniform image to the visual threshold. This algorithm requires approximately one sample for every two pixels before it is convinced that the image has been adequately sampled. This rate represents a minimum sample density that is required for all scenes. It therefore is an appropriate choice for the preliminary sampling rate based on the use of the average statistics and the visual metric.

The enhanced algorithm incorporates this improvement by randomly selecting one out of every two pixels to sample. The sample locations and values are stored in the leaves of the quad-tree data structure. After the initial sampling is complete, the Haar image approximation, error estimate, and visual difference prediction are calculated at the affected nodes in the tree in a depth-first order. This modification reduces the number of times that values are calculated for the interior nodes of the tree. In the old method a complete path up the tree was updated for the addition of each sample. This implies that values stored at the interior nodes would be updated

multiple times during this initial sampling period (e.g. the lowest frequency values stored at the root node are affected by every sample). However, with the new method the values at each node are updated only once.

The second modification is to sample all regions of the image with a perceptual error higher than the user specified threshold before updating the image approximation, error estimate, and difference prediction. This change is applied after the preliminary sampling is complete. The basic method placed each sample at the location of most perceptible error. However, all locations with a lower accuracy than desired would eventually receive additional samples. Sampling all of these areas before updating, again reduces the number of times that values are recalculated in the interior nodes of the tree.

This change requires a modification in the traversal of the maximum error tree. In the basic method a value was stored at each node that indicates the largest visual difference contribution at and below that node in the tree. However, the total difference was not indicated at interior nodes. This simple technique allowed the location of the image containing the most perceptible error to be easily located. The enhanced method must be able to rapidly find all locations of the image with more error than the user specified threshold.

This search is accomplished by performing a depth-first traversal of the tree. At each node a value is calculated that indicates the most perceptible error that is contained in a location of the image that is reachable from the node. This value is easily calculated by summing the local visual difference contributions down the traversed path and adding this value to the maximum visual difference stored at the node. This measure of the maximum total error is tested against the thresh-

old to determine which branches contain areas that require refinement and should be traversed further. In this manner all regions of the image that should receive additional samples are rapidly located without investigating areas that are already below tolerance.

After the sampling is complete, values are updated at the affected nodes. Only one update is required per node, whereas before numerous updates could occur at the interior nodes for the same number of samples. This process continues until all regions of the image have been rendered to the desired accuracy.

The final modification to the basic algorithm is to take multiple samples in each area requiring refinement before updating values. This is the most beneficial of the enhancements discussed in this section. Regions of the image that are indicated to need refinement generally require more than a single sample in order to reach the specified accuracy tolerance. This is especially true of very high frequency areas and when Monte Carlo shading techniques are employed. In these situations hundreds of samples can easily be necessary within a given pixel. It is very inefficient to perform hundreds of image quality assessments in such a region. Conversely, taking too many samples when they are not necessary can also negatively impact the execution time.

In order to solve this problem, an oracle was developed that is able to roughly predict the number of samples that should be taken within a given area. This oracle is based on the observation that the perceptual error declines with approximately the square-root of the sampling rate. This can be observed in the plots of visual quality versus sample density contained in pages 188–206 of this chapter. This relationship is obviously a crude approximation due to the complexity of perceptual response. However, it is simple to compute and tends to work well in practice.

This relationship allows the number of samples to be taken in a leaf region to be easily calculated based on the current visual error and number of samples that have already been taken in the area. This oracle is slightly conservative and multiple iterations are generally still necessary before all regions of the image are computed to the desired accuracy. However, by taking the number of samples estimated by this relationship between updates, it is possible to significantly reduce the number of times that the visual model is executed and utilize very few unnecessary samples.

Results of the Basic Perceptual Algorithm

This section discusses the results of applying the basic perceptually based adaptive sampling algorithm to render a number of three-dimensional scenes. Similar results are achieved with the enhanced algorithm as will be demonstrated in pages 206–216 of this chapter. Simple environments will be considered first followed by a scene with more complex geometry and lighting. Two shading techniques will be used in these examples, direct and Monte Carlo light source sampling. The direct sampling method uses a simple shading algorithm in which point light sources are directly sampled each time a ray strikes a surface. The Monte Carlo method uses area light sources and blind Monte Carlo integration to evaluate the shading integral. In this approach the incident radiance at a surface point is evaluated by spawning a number of rays at random orientations across the positive hemisphere. Blind integration is acknowledged as not being the most efficient means of evaluating the shading integral. However, this technique provides a simple means of demonstrating a situation where noise is present within the illumination calculation.

The first demonstration is illustrated in Figure 52. The images on the left

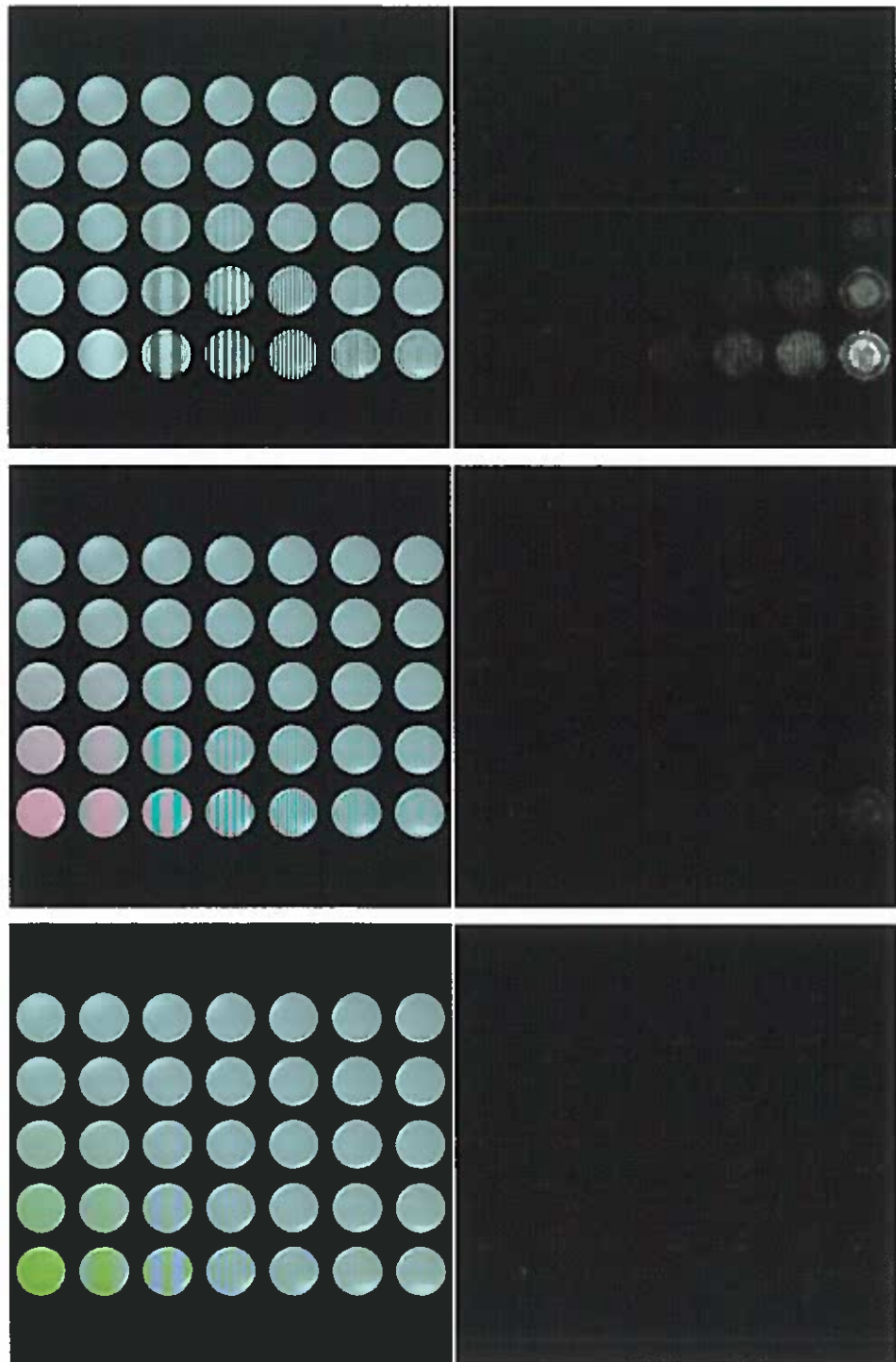


FIGURE 52. Sampling densities for direct light source sampling.

side of this figure have been rendered by the perceptual algorithm to a tolerance of 1 JND. This corresponds with the threshold error visibility of the human visual system. The three scenes consist of arrays of texture mapped disks. The spatial frequency of these textures increases from left to right and the contrast increases from top to bottom. The textures for the top image vary along the achromatic axis in AC_1C_2 space, the textures in the middle image vary along the C_1 axis, and the bottom textures vary along the C_2 axis. The illumination calculation in these examples uses direct light source sampling.

This shading technique produces consistent intensity computations at a given location of the image plane. Therefore, the spatial frequency of the textures is the main determinant of the appropriate sampling rate within a given disk. More samples are required to accurately reconstruct high spatial frequencies than are necessary for low spatial frequencies. However, less samples can be taken in the chromatic disks than in the achromatic disks due to the lower spatial frequency cutoff of the chromatic contrast sensitivity function. The resulting error will not be perceptible to a human observer. Additionally, less samples can be used to reconstruct the C_2 disks than the C_1 disks because of the effect of chromatic aberration.

The images on the right side of this figure illustrate the sampling that was performed by the adaptive algorithm. In these images white indicates the areas receiving the most dense sampling whereas black marks the regions where the fewest samples are taken. These images show that the algorithm has correctly distributed samples in accordance with these aspects of the visual system.

In the next example the scene containing the achromatic disks is rendered again using Monte Carlo light source sampling. The resulting image and sample

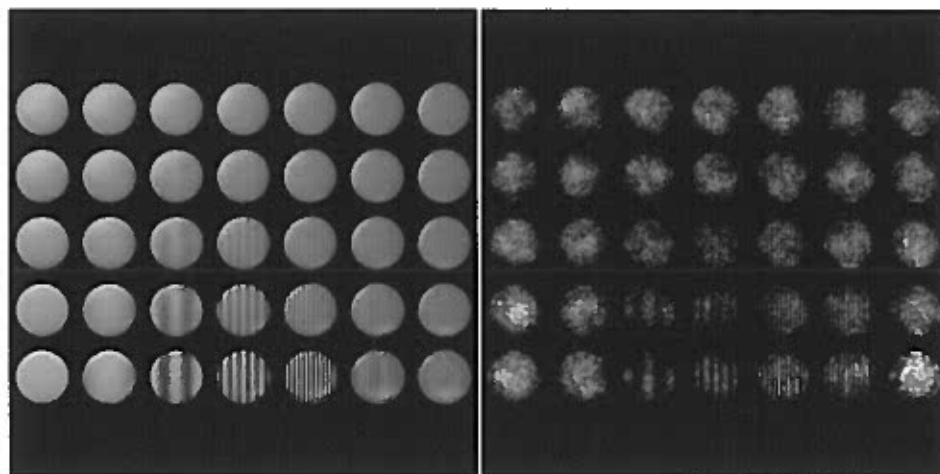


FIGURE 53. Sampling density for Monte Carlo light source sampling.

density map are shown in Figure 53. The use of the Monte Carlo shading technique results in very noisy illumination calculations, and many samples are required to reduce the error to an acceptable threshold. The effect of masking has a significant impact on the visibility of this error. This is reflected in the placement of samples by the new algorithm. Many samples are necessary to accurately reconstruct both the high and low spatial frequency disks because the noise is very apparent in these regions. However, fewer samples are taken in the middle frequencies because the strong visibility of these textures makes the noise less perceptible.

In Figure 54 the environment is made more complex but a similar result is still obtained. For the case of direct light source sampling there is no additional noise and high frequency achromatic transitions receive the most samples. Regions that vary along the C_1 and C_2 axis receive progressively less sampling. For the case of Monte Carlo sampling, many samples are required across the entire image to reduce the noise to an acceptable threshold. However, fewer samples are necessary for the middle achromatic frequencies due to the effect of masking.

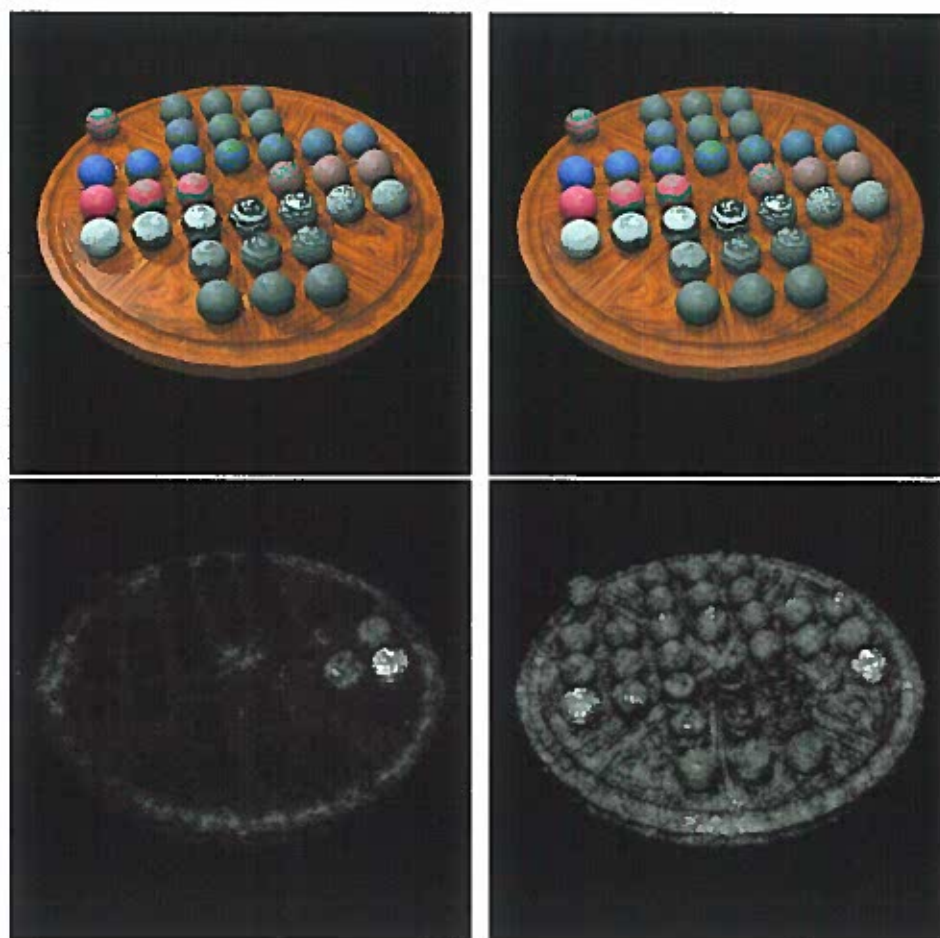


FIGURE 54. Sampling densities for direct (left) and Monte Carlo (right) light source sampling. Color varies in the middle three rows along the C_2 , C_1 , and A axes of AC_1C_2 space. Contrast of the middle three balls in the C_2 and A rows is decreased in the top two and bottom two rows respectively.

The final example illustrates the perceptually based stopping criteria. In this example identical scenes are rendered to a number of visual tolerances using two different shading techniques. The resulting images are shown in Figure 55. The images on the left were produced using direct light source sampling, and the images on the right used a Monte Carlo shading technique. The visual quality of these images increases from the top to bottom of the figure. This demonstration shows that the new algorithm is able to produce perceptually consistent results even though

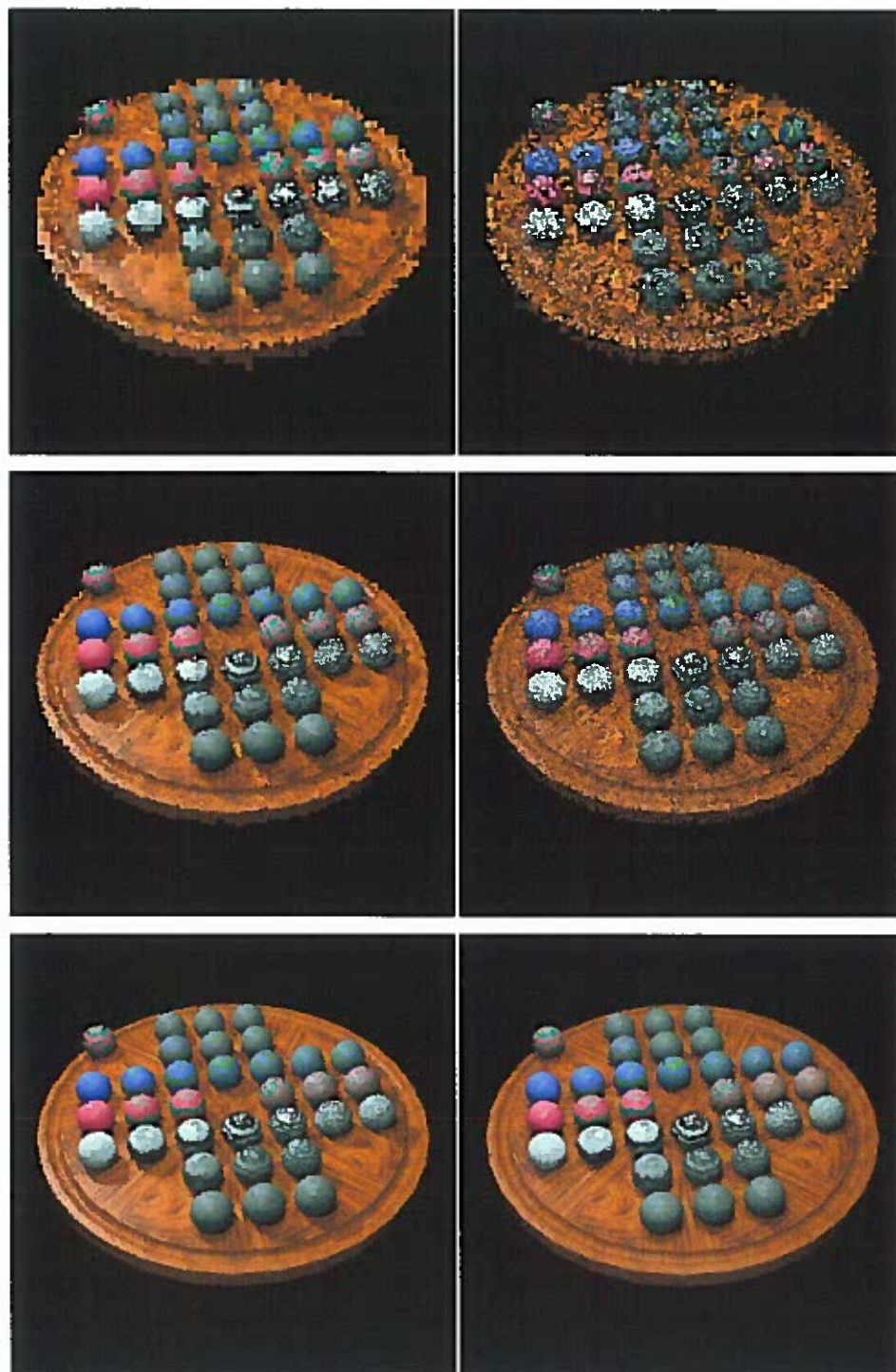


FIGURE 55. Image rendered at a visual tolerance of 7 (top), 5 (middle), and 3 (bottom) using direct light source sampling (left column) and Monte Carlo light source sampling (right column).

radically different sample distributions and shading techniques are employed. In the case of direct light source sampling, aliasing is the significant problem, whereas for Monte Carlo shading, noise is the predominant artifact. However, at a given tolerance, the algorithm is able to hold the worst case appearance of each type of artifact to an equivalent level of visual impact.

Comparison with Existing Sampling Algorithms

This section compares the results of the new perceptually based adaptive sampling algorithms with the results of two commonly used sampling strategies. This comparison will cover both the quantity of samples that are required to produce images of a given visual quality and the overall expense of the algorithms. A number of example renderings will be used to demonstrate the key features of the perceptually based technique. The two other sampling strategies that will be employed in this comparison are uniform sampling and adaptive sampling based on an objective error estimate.

Uniform sampling is the simplest and therefore one of the most prevalent methods for placing samples within the image plane. In this technique a refinement test is not used and an equal number of samples are taken in each pixel. One of the drawbacks of this strategy is that it is the responsibility of the user to determine the sampling rate that produces an image of the desired quality. The usual approach to applying uniform sampling is to take all samples at a given pixel before beginning to sample the next pixel. This minimizes the amount of memory required. However, this method does not allow a fair intermediate image comparison with the other sampling strategies because regions of the image will be undefined until all

samples are taken. Therefore the uniform sampling algorithm that is used in the tests reported here, disperses intermediate samples more evenly across the image plane. In this approach a single sample is first taken at each pixel. This sampling is performed left to right, top to bottom across the image. After all pixels have been sampled once, a second sample is taken in each pixel. This process continues until the final sampling density has been reached. Within a given pixel the samples are randomly distributed and the intensity of a pixel is defined to be the average of the samples taken within it.

The second sampling strategy that will be used for comparison is adaptive sampling based on an objective error estimate. This algorithm uses the variance of the sample's radiance in RGB color space as its error metric. This approach is similar to a number of the prior techniques discussed in pages 41–43 of Chapter IV that use sample statistics as the basis for their refinement test [81, 107]. The actual adaptive algorithm was created by removing the stages that modeled the human visual system from the basic perceptually based adaptive sampling strategy described in pages 163–176 of this chapter.

The adaptive sampling algorithm receives the sample's RGB radiance as input. The goal of this algorithm is to iteratively place each sample at the location containing the largest objective error. This is accomplished by creating and refining a Haar wavelet image approximation and multi-resolution error estimate as described in the *refine cortex representation* and *refine error estimate* stages of the basic perceptual algorithm (see pages 163–176 of this chapter). The local error (*LOD*) at each node of the quad-tree is defined to be the sum of the detail variance (*V*) across all detail

orientations (θ) and RGB color channels (c):

$$LOD = \sum_{\theta=1}^3 \sum_{c=1}^3 V[\theta, c]. \quad (55)$$

A maximum error tree is created by summing the largest local error up the branches of the quad-tree as described previously. The result of this operation is that a value is stored at each node of the tree that represents the largest variance present in the region of the image which that node is defined to cover. The next sample location is determined by traversing the quad-tree in a top-down fashion and selecting the node with the largest variance. In this manner samples are always placed in the location of the image plane containing the largest objective error.

Required Sampling Rates

This section will discuss the number of samples required by the different sampling strategies in order to produce images of a given visual quality. This discussion compares the results of the basic perceptual algorithm with the results of the uniform and objective adaptive sampling techniques described in the previous section. The examples that are described show that the new perceptually based adaptive sampling algorithm is able to produce images of equivalent visual quality using fewer samples than either of the existing uniform or objective adaptive sampling algorithms.

A number of example renderings will be used to illustrate key features of the new algorithm. In each of the examples the placement of samples by the three approaches will be discussed. The images that are produced by the algorithms after an equivalent number of samples will be shown in order to allow a visual inspection.

Additionally, a visual difference prediction will be performed using the algorithm described in Chapter VII to compare the images with a high quality rendering. This will illustrate the areas of the image containing visible artifacts and further verify the new difference predictor. Finally, a graph of the maximum visual difference versus number of samples will be given for each of the three techniques. The maximum visual difference is defined to be the largest difference found at any location in the image, using the new visual difference predictor. A rendering is generally not considered to be of high enough quality until all regions of the image are computed accurately enough so that the error is below the visual threshold. Therefore, if even a small region of the image contains significant perceptible error, the image can be considered unusable. For this reason maximum visual difference is an appropriate quantity in these comparisons.

The first example illustrates the benefit of adaptive sampling. A high quality rendering of this scene is contained in Figure 56. This scene consists of simple texture mapped disks which increase in spatial frequency from the left to right and top to bottom of the image. The shading in this scene is accomplished by directly sampling point light sources every time a ray strikes a surface. In this case there is no variance in the intensity of a sample at a given location and the spatial frequency of the texture is the primary determinant of the sampling rate that is necessary to accurately render the scene.

According to the sampling theorem, more samples are necessary to accurately reconstruct signals containing high spatial frequencies than are necessary for signals containing only low spatial frequencies. For this reason uniform sampling performs badly in this example. The disk in the lower right of the image contains the highest

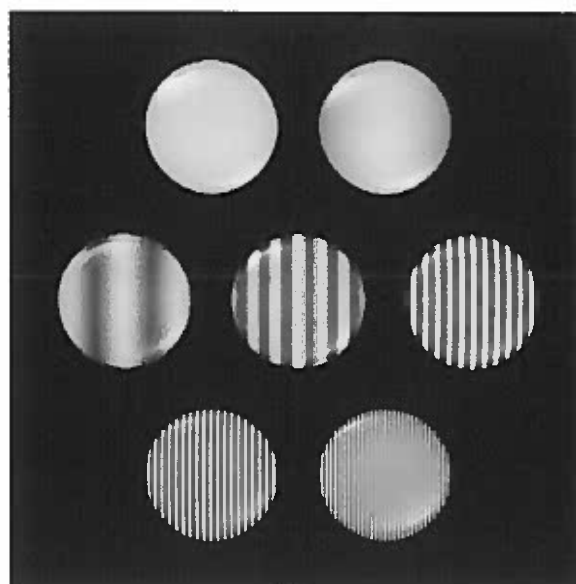


FIGURE 56. High quality image for the spatial frequency example. This scene contains texture mapped disks which increase in frequency from left to right and top to bottom.

frequency texture and therefore requires the most samples to accurately reconstruct it. However, in order for the uniform sampling algorithm to take more samples in this region, it must take more samples in every pixel of the image. This is inefficient because the majority of the samples are spent calculating regions of the image that have already been accurately computed.

The adaptive sampling algorithms are much more efficient in their placement of samples. The images in the top row of Figure 57 show the sample density plots for each of the sampling techniques. The result of uniform sampling is on the left, the result of adaptive sampling based on an objective error estimate is in the center, and the result of the perceptually based algorithm is on the right. Areas containing the most dense sampling are marked in white and areas with the lowest sampling density are marked in black. Intermediate densities are plotted on a square root scale. The sample density plot for uniform sampling is included for reference and

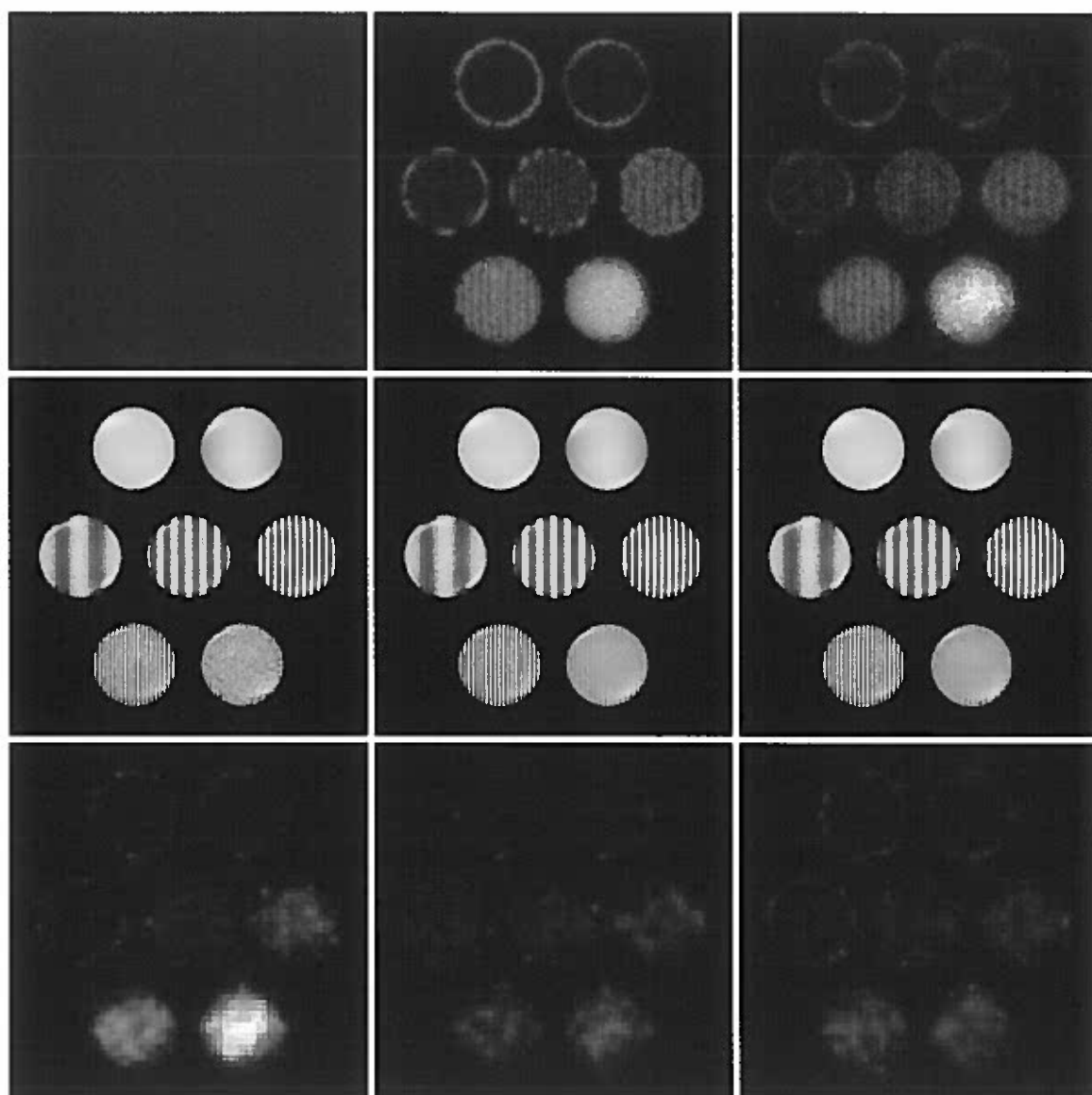


FIGURE 57. Comparison of uniform (left), objective (middle), and perceptual (right) sampling strategies for the spatial frequency example. The rows contain the sample density maps (top), the images produced after an average of 4 samples per pixel (center), and the visual difference comparison with the high quality image (bottom).

indicates that a low number of samples have been taken equally at all pixels. In the plots for the adaptive algorithms we see that both algorithms have correctly placed the most samples within the highest frequency disks and very few samples in the constant background and low frequency disks.

The middle row of Figure 57 shows the images that are produced after an average of 4 samples per pixel have been taken. Again, the image created by uniform sampling is on the left, objective sampling is in the middle, and perceptual sampling is on the right. At this sampling density it is very difficult to discriminate any error within the images produced by adaptive sampling, but the error is still very apparent within the highest frequency disk in the case of uniform sampling.

The results of visual difference predictions between these images and a high quality rendering are contained in the bottom row of Figure 57. In these images increased brightness is used to indicate areas of more perceptible error. The difference predictions provide a good correspondence with subjective impressions of the images in the middle row. Again we see that the error is significantly more visible in the highest frequency disk produced by uniform sampling, and less visible and more evenly distributed for both of the adaptive sampling algorithms.

A graph of peak visual difference versus number of samples is plotted in Figure 58. In this example over a hundred samples per pixel are required to reduce the error of the highest frequency disk to the visual threshold (visual difference = 1). For the uniform sampling method this equates to taking over a hundred samples in each pixel of the image. The two adaptive sampling algorithms, however, are able to significantly reduce the total number of samples required. This is accomplished by taking fewer samples in regions of low spatial frequency and more samples in

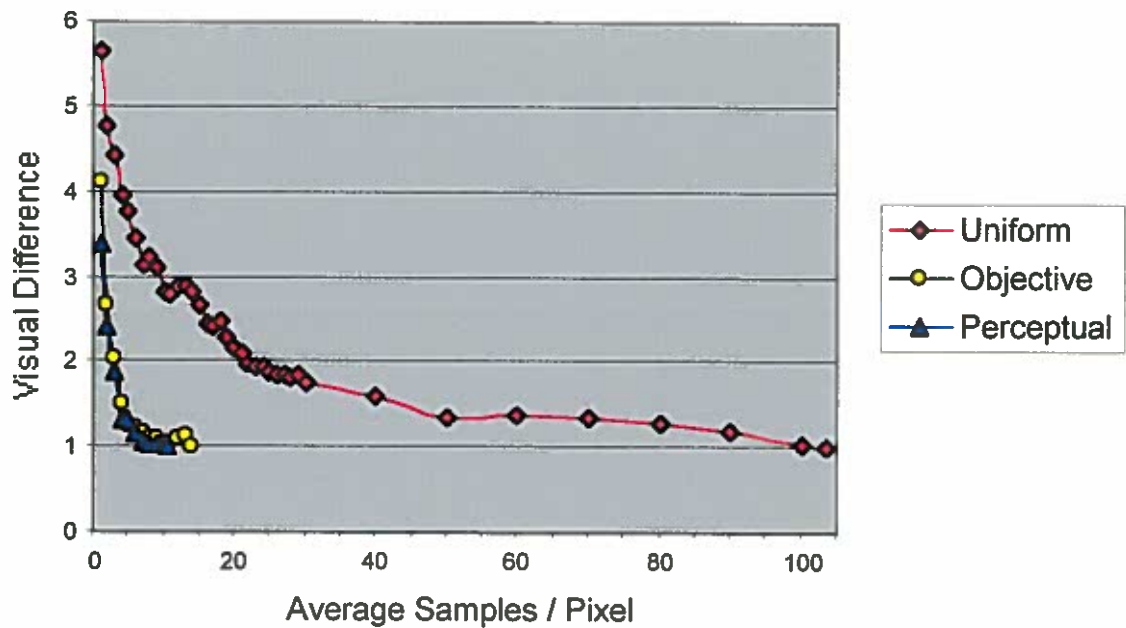


FIGURE 58. Sampling rates for the spatial frequency example depicted in Figure 56.

regions of high spatial frequency. This technique allows the adaptive algorithms to accurately render the image using roughly 12% of the samples required by the uniform sampling algorithm.

The benefits of adaptive sampling are well known in the field of computer graphics. It is often assumed that adaptive algorithms based on objective error will always outperform uniform sampling. However, this is not the case. In the next example we will see a situation where placing more samples in regions of large objective error will create images with more perceptible artifacts than if the same number of samples had been placed uniformly. A perceptually based sampling algorithm is therefore the only method which can efficiently handle this situation.

The scene for the next example consists of the simple sphere illustrated in Figure 59. The sphere is illuminated by a small area light source. Blind Monte Carlo integration is performed to evaluate the shading integral. This shading technique spawns many rays at random directions from each surface intersection in order to



FIGURE 59. High quality image for the contrast example. This scene consists of a simple sphere illuminated by an area light source.

evaluate the radiance that is incident at a point on the sphere. Since this is a random process there is a certain amount of variance in the intensity of the samples taken of the sphere. Because there is very little spatial variation in this image, intensity variation and the visibility of this variation at different illumination levels are the primary factors that govern the appropriate sampling rate.

The sample density maps for the three sampling algorithms are shown in the top row of Figure 60. The map in the center shows the sample density for the objective method. In this image we see that the most samples are taken at the brightest regions of the sphere and the least samples in the dark regions. This is because the standard deviation of the samples scales with the reflectivity of the sphere. Consider for example that the scene is illuminated by a 100 cd/m^2 light source. At each point on the surface rays are spawned to determine the incident light. Rays that strike the light source will return the intensity of the light. Rays

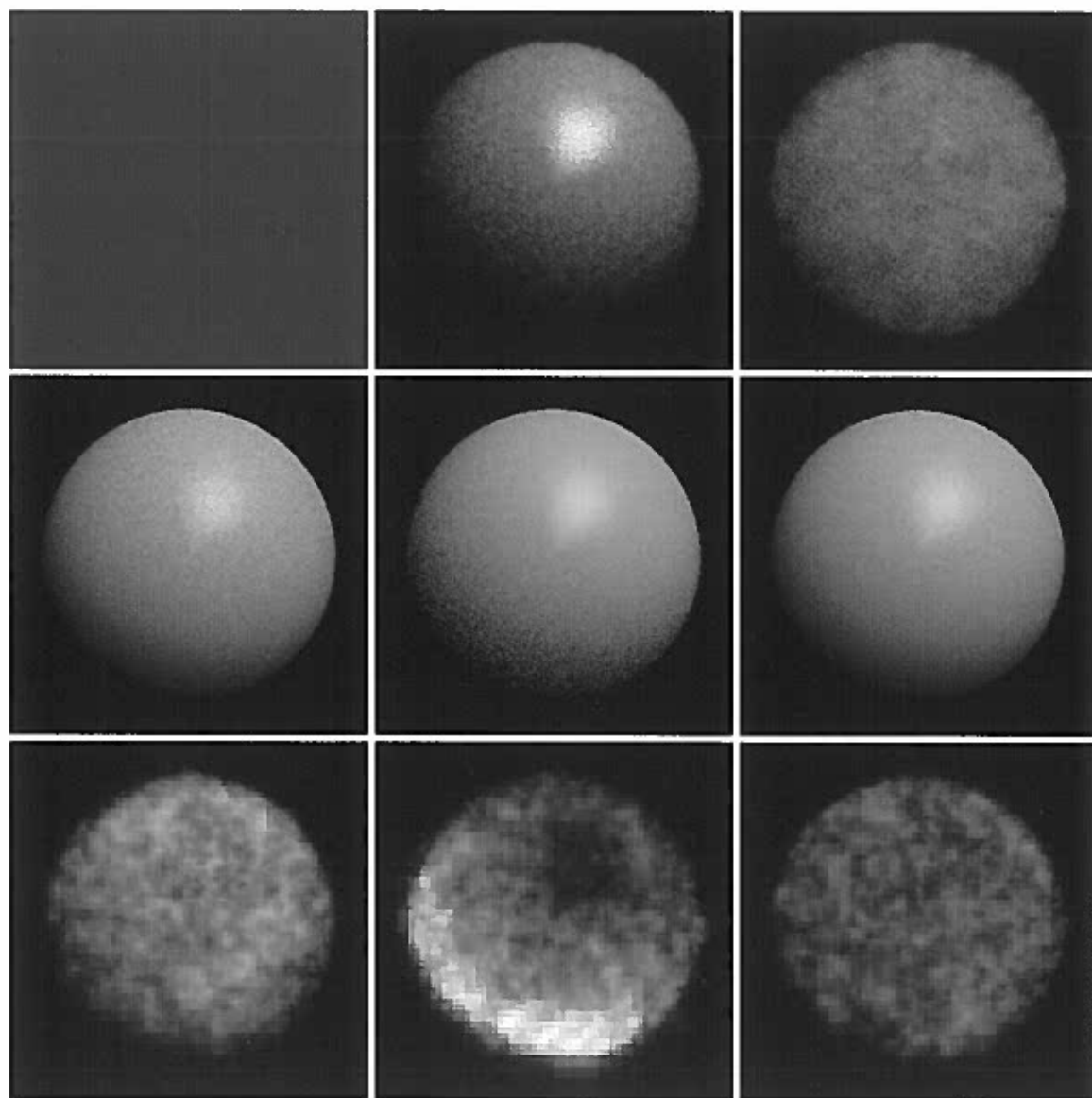


FIGURE 60. Comparison of uniform (left), objective (middle), and perceptual (right) sampling strategies for the contrast example. The rows contain the sample density maps (top), the images produced after an average of 20 samples per pixel (center), and the visual difference comparison with the high quality image (bottom).

that miss the light source will return an intensity of 0 cd/m^2 . Therefore, samples that originate at the image plane and strike a point on the sphere that reflects 100% of the incident light will return noisy values between 0 and 100 cd/m^2 , depending on the number of spawned rays that strike the light source. However, samples from the image plane that intersect a point on the sphere that only reflects $1/100^{\text{th}}$ of the incident light will only vary between 0 and 1 cd/m^2 (the difference in the reflectivity of these two points is the result of the orientation of the surface relative to the light source and eye position). Thus, the amount of noise in the first case will be 100 times greater than the amount of noise in the second case.

As it turns out, the sampling pattern produced by the objective algorithm is extremely inefficient. This is because the sensitivity of the human visual system varies with the local illumination level. The visual system is much more tolerant of error in bright regions than in dark and is equally tolerant of error when $\frac{\Delta L}{L}$ is a constant (where ΔL represents the luminance error and L is the mean luminance). In this example the mean luminance at locations of the image also scales with the reflectivity of the sphere. The net result is that the visibility of the error at a given sampling rate is uniform across the face of the sphere. This implies that uniformly sampling the interior of the sphere is an optimal solution. This is the sampling pattern used by the perceptual algorithm, as can be seen in the rightmost sample density map.

The middle row of Figure 60 shows the images created by the three algorithms after an average of 20 samples per pixel. Note that noise is very visible along the dark underside of the sphere in the image produced by the objective method, whereas it is difficult to discriminate the noise anywhere on the surface of the sphere

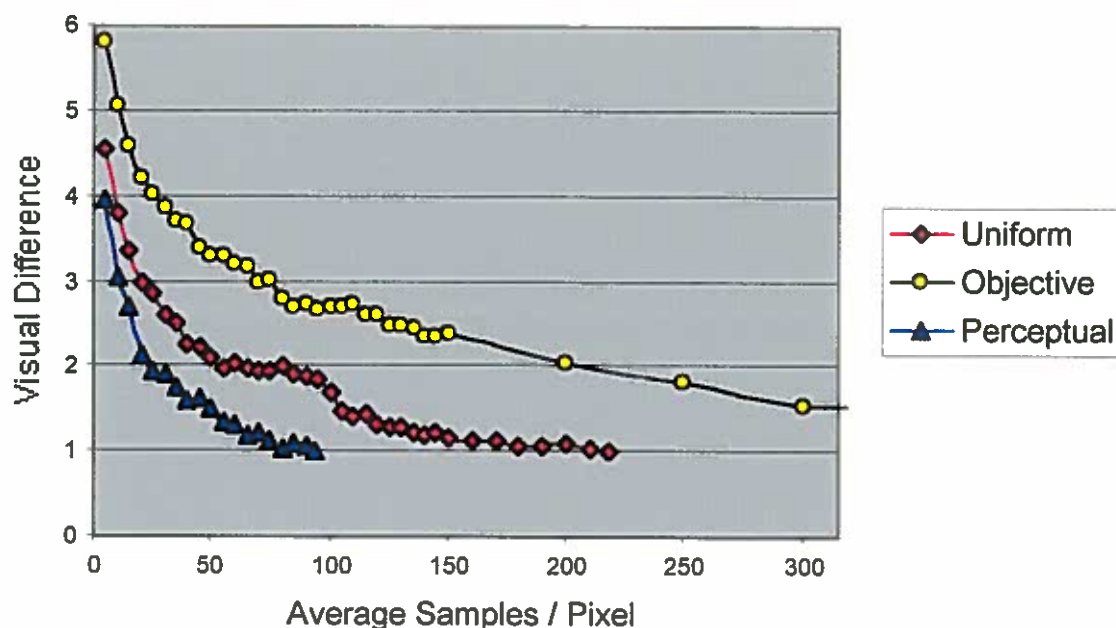


FIGURE 61. Sampling rates for the contrast example depicted in Figure 59.

produced by the perceptual method. Additionally, the image rendered by the perceptual algorithm has a somewhat higher visual quality than the image generated by the uniform sampling strategy. This is because the perceptual algorithm cast fewer samples in the constant background around the edges of the image and, instead, concentrated these samples in the interior of the sphere where they were most needed. These observations are further demonstrated by the visual difference maps contained in the bottom row of this figure.

The visual quality of the images produced by these algorithms is plotted versus the sampling rate in Figure 61. The objective sampling algorithm has the worst performance in this example. Because of the poor sample distribution used by this method a large number of samples are wasted in the bright specular region of the sphere before an adequate number of samples are taken in the darker regions. The uniform sampling algorithm produced better results, requiring only a quarter of the samples of the objective method. This is due to the fact that uniform sampling is

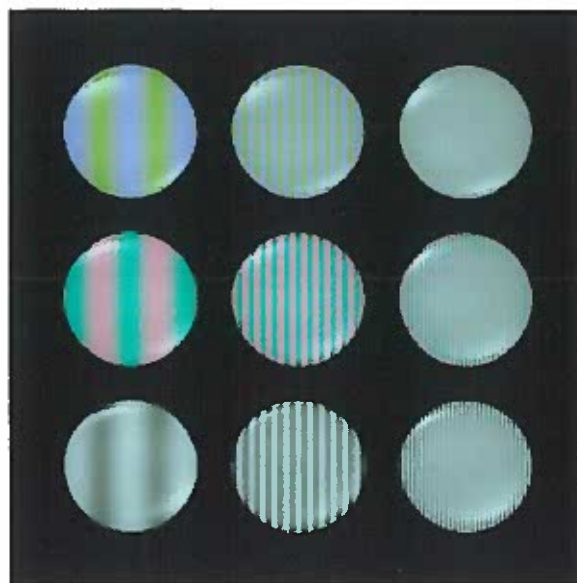


FIGURE 62. High quality image for the chromatic spatial frequency example. This scene contains three rows of texture mapped disks that increase in frequency from left to right. The bottom row varies along the achromatic axis, the middle row along the C_1 axis, and the top row along the C_2 axis.

exactly the right thing to do within the interior of the sphere. Again, the perceptually based algorithm fared the best, requiring only half as many samples as the uniform method.

The third example that will be discussed in this section is illustrated in Figure 62. This example will demonstrate the benefit of exploiting differences in the spatial frequency sensitivity of the achromatic and chromatic channels. This scene consists of three rows of texture mapped disks. The spatial frequency of the textures increases from the left to right disk. In the bottom row the grating varies achromatically, the grating in the middle row varies along the C_1 color axis, and the top row varies along the C_2 axis. The magnitude of each sinusoidal grating is equidistant in RGB radiance space. Direct sampling of point light sources is used in the shading calculation.

The sampling patterns produced by the uniform, objective, and perceptual algorithms are shown in the top row of Figure 63. In this figure we see that both adaptive algorithms have correctly sampled the disks with high spatial frequency textures at a greater rate than the disks with low spatial frequency textures. The difference in the two sampling patterns is that the objective algorithm cast an equal number of samples in each row, whereas the perceptual algorithm took the most samples in the achromatic row and successively less samples in the rows for the C_1 and C_2 color channels. This result occurs because the objective error metric does not take into account the spatial frequency sensitivity of the human visual system or the differences between achromatic and chromatic sensitivity. The perceptual algorithm, on the other hand, has modeled these elements. Because the achromatic contrast sensitivity function cuts off at a higher spatial frequency than the chromatic contrast sensitivity function, high frequency noise is more visible when it occurs along the achromatic axis than either of the chromatic axes. In addition, due to the effect of chromatic aberration, high frequency noise is more visible along the C_1 axis than it is along the C_2 axis. The sample distribution produced by the perceptual algorithm is therefore more efficient than the one employed by the objective algorithm.

The middle row shows the images produced by the three algorithms after an average of 2 samples per pixel. The image produced by the uniform sampling pattern contains the worst artifacts. This is because the high frequency disks are severely undersampled. It should also be apparent that noise on the achromatic disk is more visible than the noise on the C_1 disk which is more visible than noise on the C_2 disk. The image produced by the objective sampling algorithm is better because more samples are concentrated with the highest frequency disks. However, artifacts

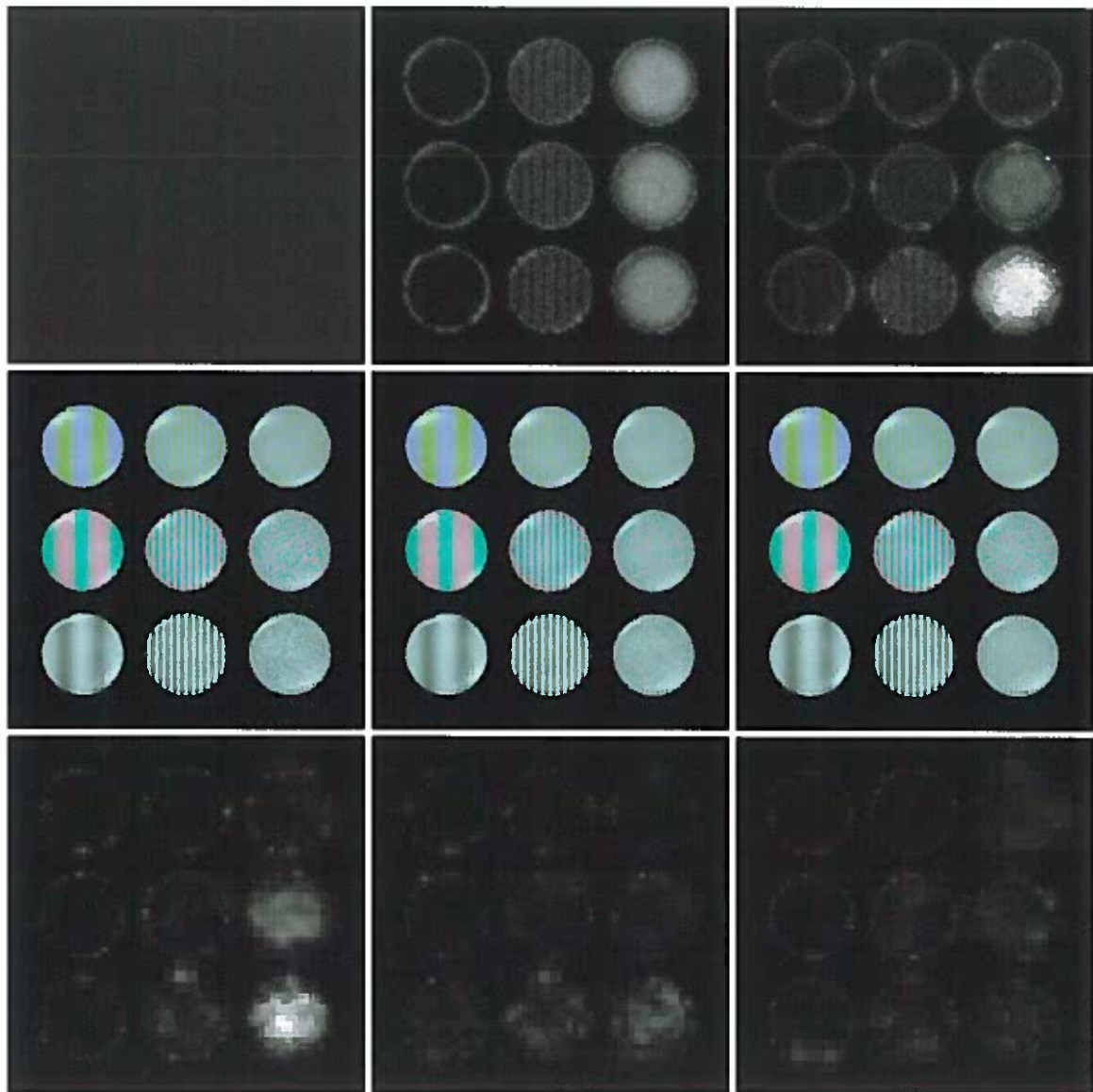


FIGURE 63. Comparison of uniform (left), objective (middle), and perceptual (right) sampling strategies for the chromatic spatial frequency example. The rows contain the sample density maps (top), the images produced after an average of 2 samples per pixel (center), and the visual difference comparison with the high quality image (bottom).

are still visible on the high frequency achromatic disk. The image produced by the perceptually based algorithm contains even fewer artifacts. This is because samples were concentrated in the high frequency achromatic disk rather than in other regions where the error is less conspicuous. It is interesting to note that within this image the high frequency C_2 disk contains rather coarse grained noise. However, the low frequency cutoff of the contrast sensitivity function for this channel and the effect of chromatic aberration make this noise invisible at standard viewing distances.

A visual difference prediction of the results of these algorithms is contained in the bottom row. Again we see that both the uniform and objective methods have left the most significant visual error within the high frequency achromatic disk. The difference map for the perceptually rendered image shows that this algorithm has succeeded in distributing the error equally across all high frequency disks. This implies that an efficient sample distribution was employed. The large blocks that are visible on the high frequency C_2 disk are indicative of the fact that only low frequency error is visible for this color channel.

The graph contained in Figure 64 shows that uniform sampling has the worst performance in this example, requiring approximately 30 samples per pixel to reach the visual threshold. The objective sampling algorithm performs better requiring only 1/3 as many samples. The perceptually based algorithm performs the best, reducing the number of samples again by half through the correct incorporation of contrast sensitivity as a function of spatial frequency and chromatic aberration.

The final example demonstrates the effect of masking. The image utilized in this example is contained in Figure 65. This scene consists of two rectangles. The left rectangle reflects a uniform gray. The right rectangle is texture mapped with the

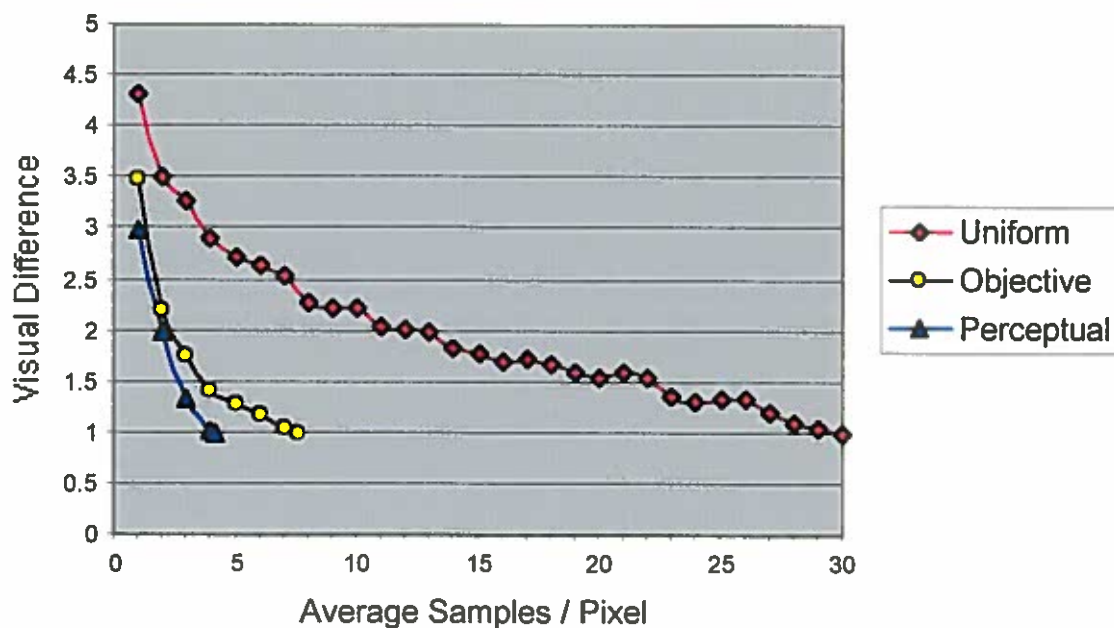


FIGURE 64. Sampling rates for the chromatic spatial frequency example depicted in Figure 62.

top view of a section of carpet. This scene is illuminated with an area light source and blind Monte Carlo integration is performed to evaluate the shading integral. This process results in significant variation in the intensity of the samples at any given location.

The sample density maps for this example are depicted in the top row of Figure 66. The map produced by the objective method shows that more samples have been taken in the right rectangle than were taken in the left. This is because the additional spatial variation of the carpet creates a greater sample variance than in the left rectangle where there is only intensity variation caused by the Monte Carlo integration. However, this sampling pattern is inefficient. The texture map of the carpet contains significant energy at spatial frequencies to which the visual system has a high sensitivity. This energy is additionally distributed across a number of frequencies and orientations. The result of this energy distribution is that the

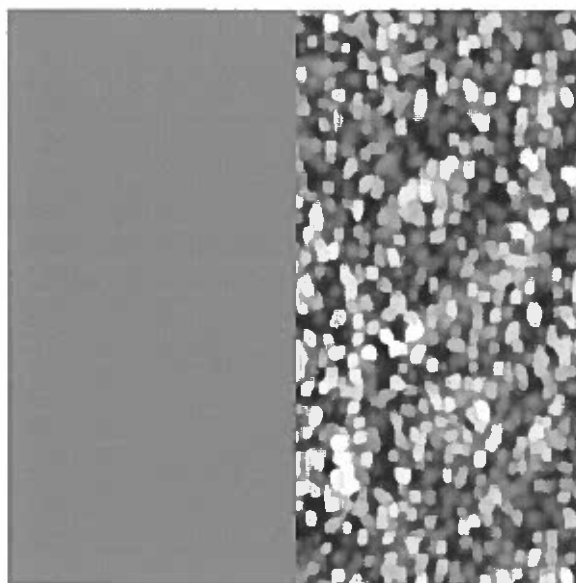


FIGURE 65. High quality image for the masking example. This scene contains two squares. The left square is a uniform gray and the right square is texture mapped with the top view of a section of carpet.

white noise produced by the intensity variation is masked by the presence of the carpet texture. Therefore, an equivalent amount of noise will be less apparent on the texture mapped rectangle than on the uniform one, where no masking occurs. This effect is correctly incorporated by the perceptual algorithm which takes more samples in the left rectangle than in the right.

The images produced by the three algorithms after 10 samples per pixel are shown in the middle row. At this stage noise is still apparent in all of the images. Within the image produced by the objective algorithm the artifacts are the strongest and occur within the left rectangle where there is no masking. The right rectangle contains little perceptible error. The image produced by the uniform sampling distribution is somewhat better because an equal number of samples are taken in each rectangle. However, the error is still more apparent in the left rectangle than in the right. The image from the perceptually based algorithm is the only one with

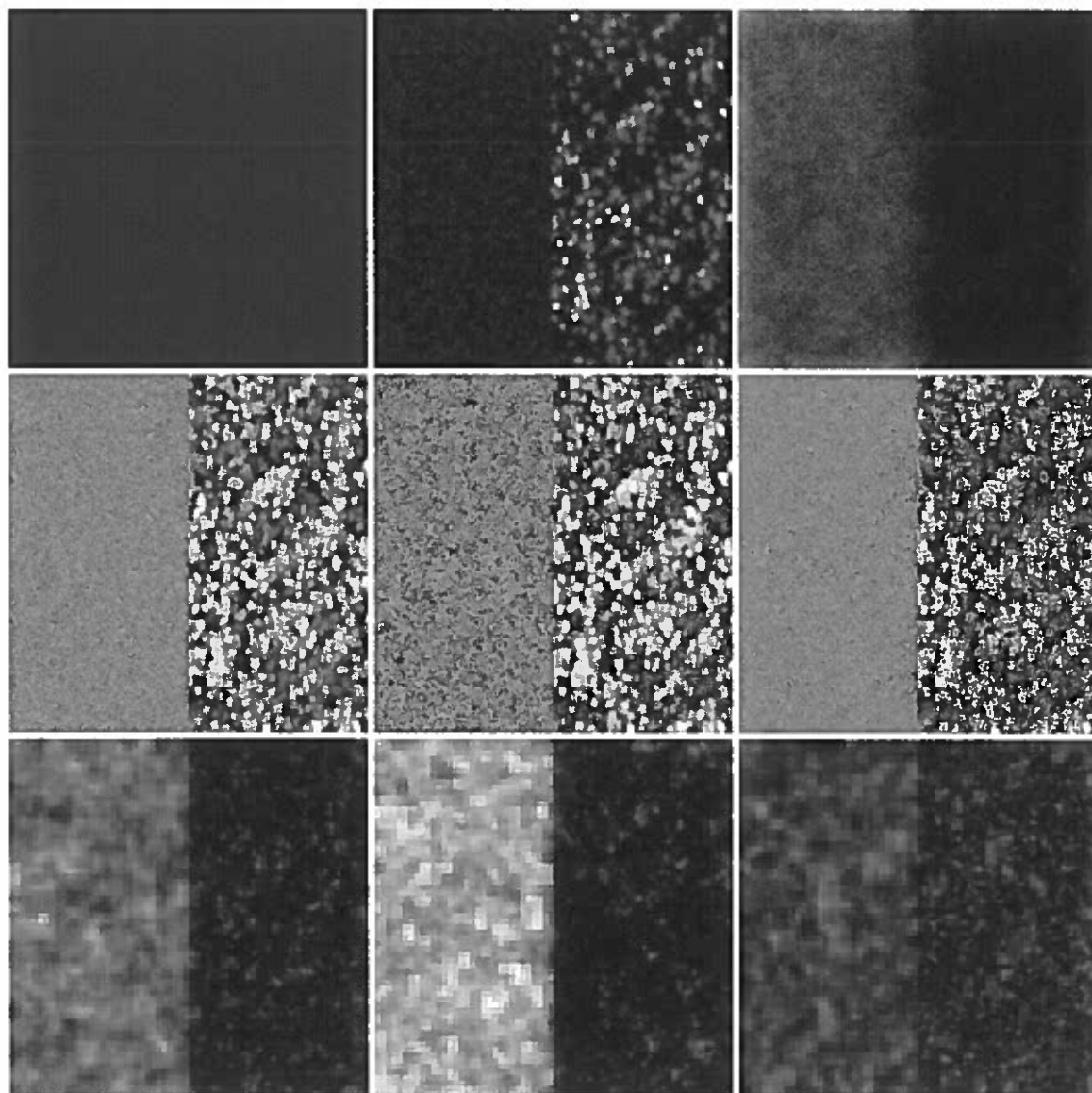


FIGURE 66. Comparison of uniform (left), objective (middle), and perceptual (right) sampling strategies for the masking example. The rows contain the sample density maps (top), the images produced after an average of 10 samples per pixel (center), and the visual difference comparison with the high quality image (bottom).

a visibly uniform error distribution. This approach has significantly more objective error in the textured rectangle than in the non-textured one. Due to the effect of masking, however, the two are of equivalent perceptual quality.

Visual difference maps of images produced by the algorithms are contained in the bottom row. The difference maps for the uniform and objective algorithms show non-uniformity in how visible the error is in the two rectangles. The difference map for the perceptual algorithm is more uniform. In the early stages of sampling, the difference maps for the perceptually based algorithm still exhibit some non-uniformity, with more error on the left rectangle than on the right. This occurs because a number of samples are required before the algorithm can ascertain a reasonable estimate of the spatial frequency spectrum.

Uniformity of perceptible error is a key idea in improving the performance of sampling algorithms. If the perceptibility of error is uniform across the image plane then the peak visual error has been minimized. This concept allows the perceptually based adaptive sampling algorithm to minimize the number of samples required to compute images to a given visual tolerance.

For this example, visual difference is plotted against number of samples in Figure 67. In this graph it is shown that the objective sampling algorithm required the most samples to accurately render the image, requiring roughly 1,000 samples per pixel. This extraordinarily high sampling rate was required because of the large sample variance and the inefficiency of the sampling pattern. Uniform sampling performed better in this instance, requiring a little over half as many samples. By correctly incorporating the effect of visual masking, the perceptual algorithm performed best of all, requiring only a third as many samples as the objective sampling

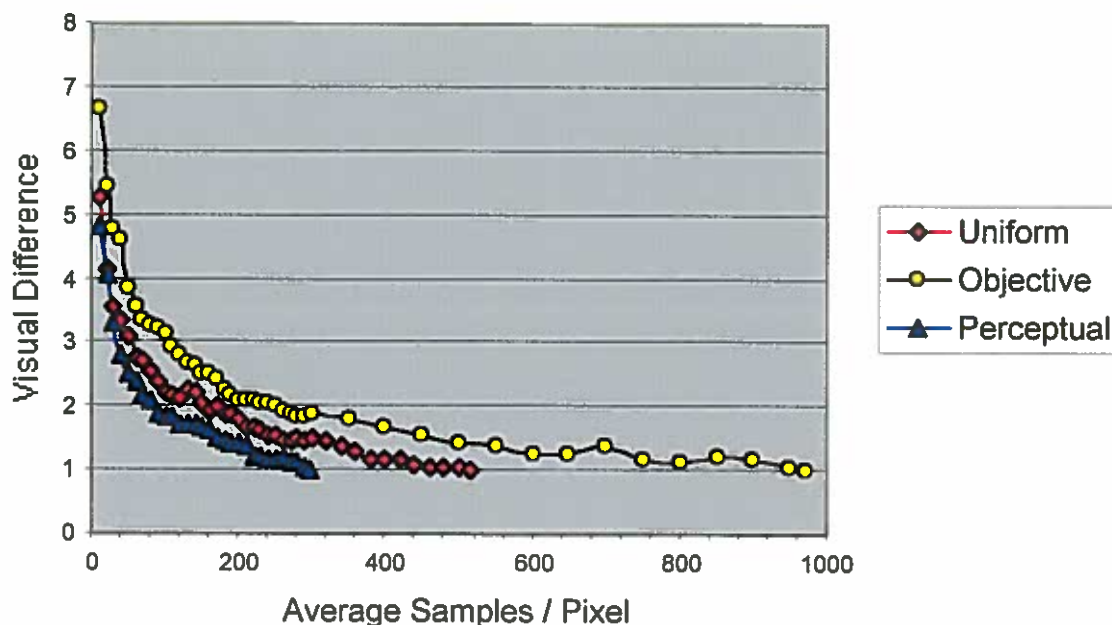


FIGURE 67. Sampling rates for the masking example depicted in Figure 65.

algorithm.

This section has demonstrated that the perceptually based adaptive sampling algorithm can significantly reduce the number of samples required to render all regions of an image to the perceptible threshold. These sample reductions have been shown in comparison to the previously designed techniques of uniform sampling and adaptive sampling based on an objective error estimate. In some cases the sampling rate has been reduced by as much as 80 to 90 percent. In the following section the cost of this algorithm will be discussed, as well as the execution times of the examples in this section.

Timing Tests

This section presents the results of a number of timing tests that compare both variations of the perceptually based adaptive sampling technique with two previous sampling strategies. These tests demonstrate the benefits of the modifica-

tions contained in the enhanced perceptual algorithm. This enhanced algorithm is additionally shown to produce images of a given visual quality in less time than is required by previous sampling methods.

The timing tests employ a similar technique to the one used to determine required sampling rates in the previous section. A number of example scenes will be rendered using the various of sampling algorithms. The images that are produced by these algorithms are output at specified intervals. The perceptual quality of the images is then computed by a comparison to a high quality renderings of the same scene. This comparison utilizes the visual difference predictor discussed in Chapter VII. These results are accumulated in a graph that plots the maximum perceptual difference between the two images versus length of execution time. This is essentially a remapping of the required sampling rate graphs along a time axis.

These graphs plot the results of four different sampling techniques. These techniques are the *basic* perceptual algorithm described in pages 163–176, the *enhanced* perceptual algorithm discussed in pages 176–180, and the existing *uniform* and *objective* methods presented in pages 186–188. The timing tests have been performed for each of the example scenes described in the previous section.

Figures 68, 69, 70, and 71 show the results of these tests for the scenes depicted in Figures 56, 59, 62, and 65 respectively. The plots of maximum visual difference versus execution time are contained at the bottom of these figures. The graph at the top of these figures redisplay the results of the sampling rate tests that plot peak visual difference against the number of samples. These charts are reshown to provide an easy comparison between required sampling rates and length of execution time. The plots of the sampling rates have also been augmented to include the results of

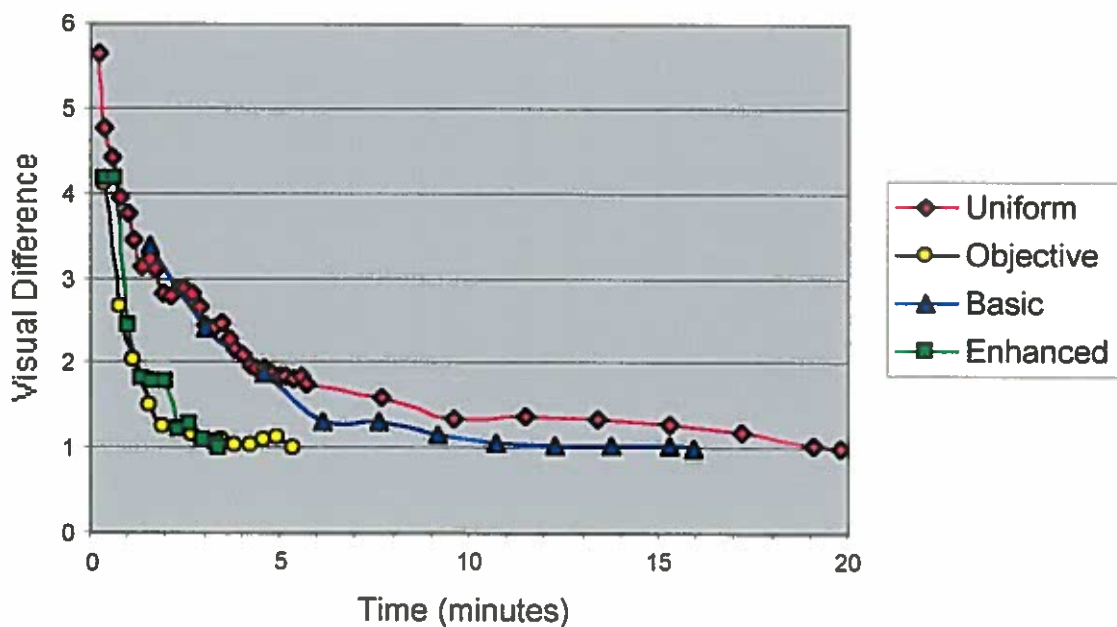
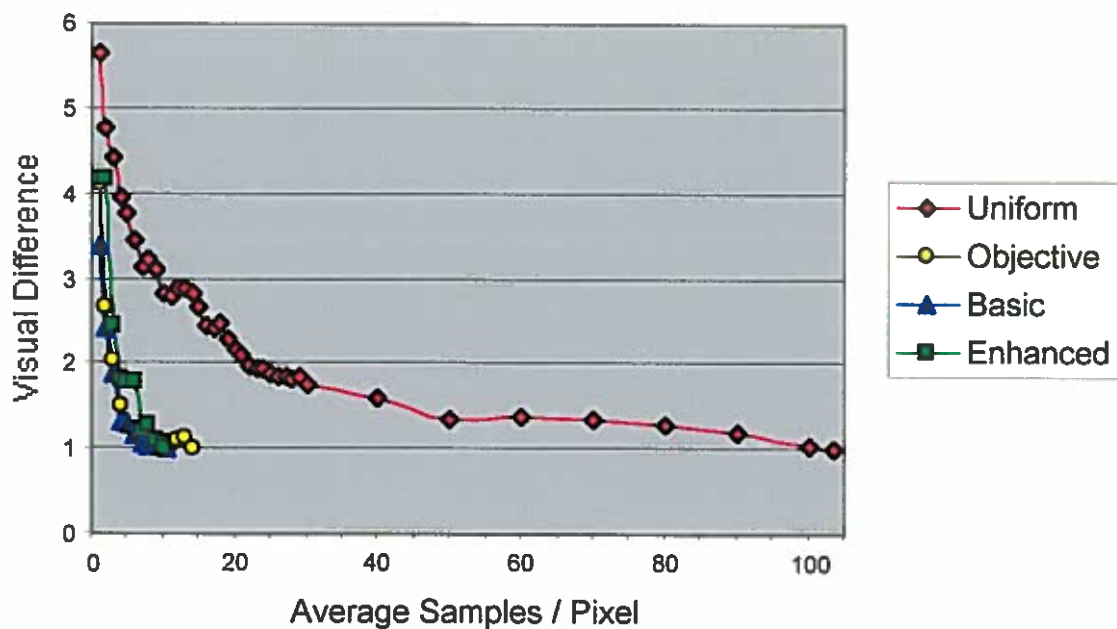


FIGURE 68. Sampling rates (top) and timing tests (bottom) for the spatial frequency example depicted in Figure 56.

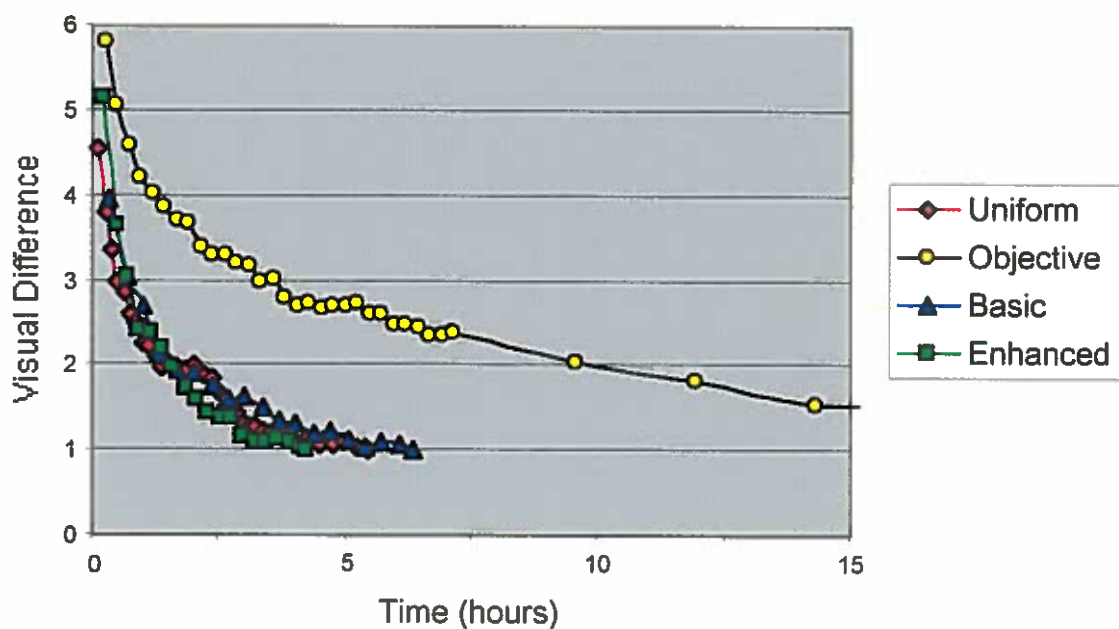
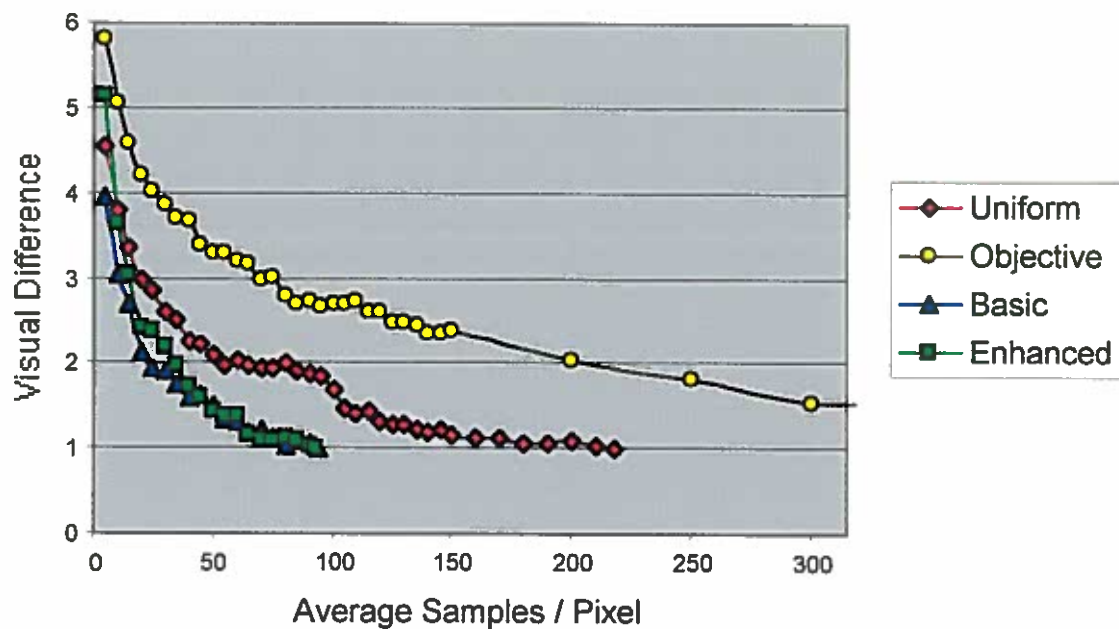


FIGURE 69. Sampling rates (top) and timing tests (bottom) for the contrast example depicted in Figure 59.

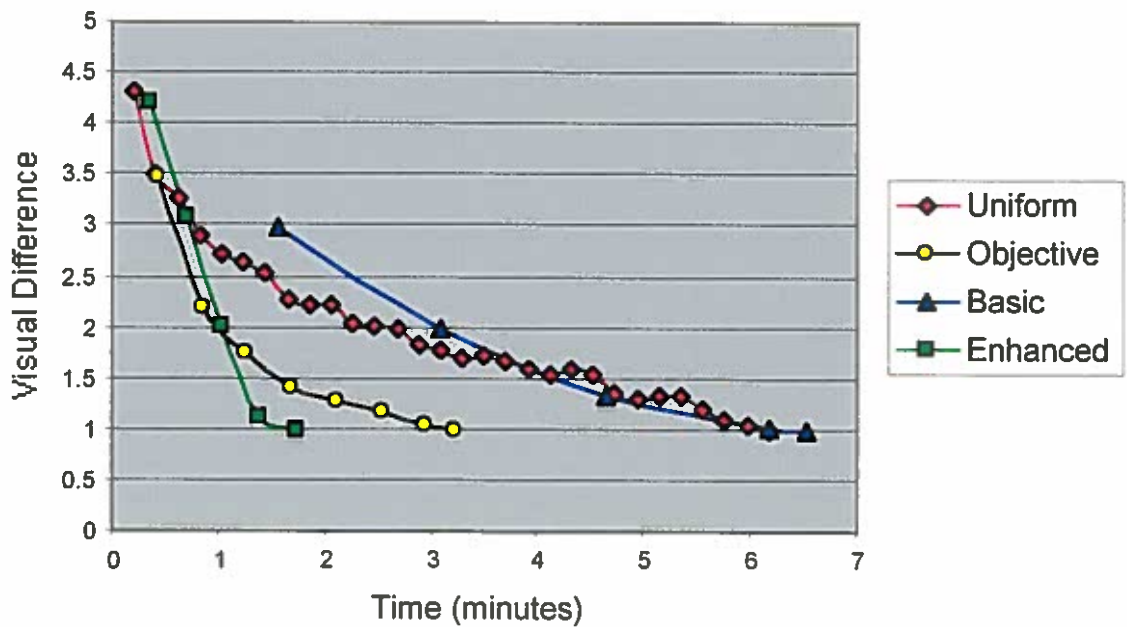
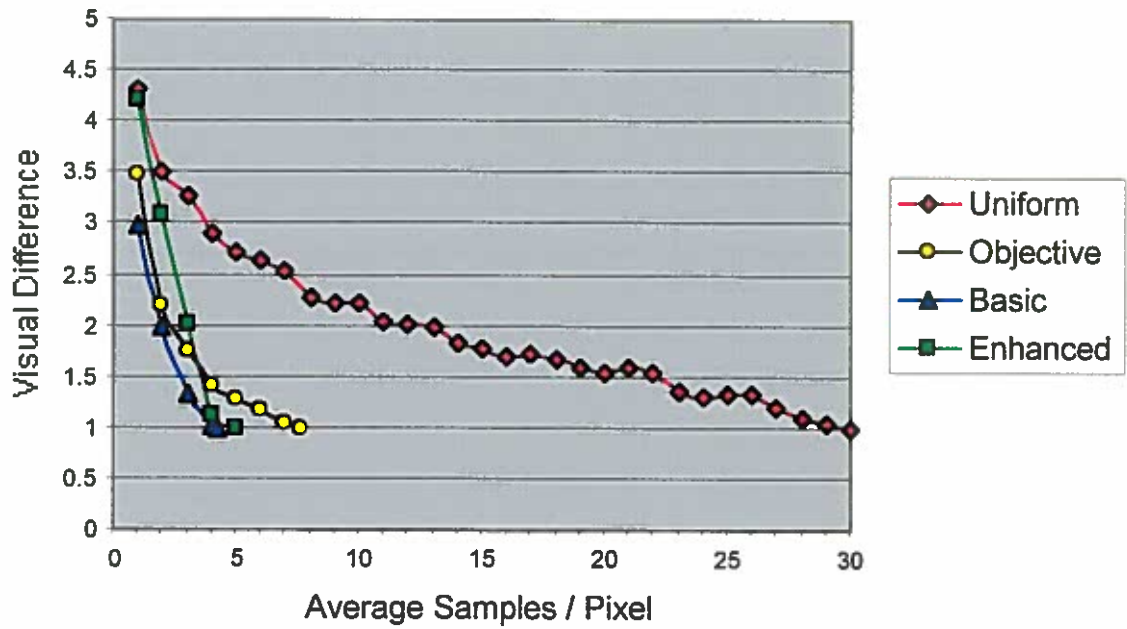


FIGURE 70. Sampling rates (top) and timing tests (bottom) for the chromatic spatial frequency example depicted in Figure 62.

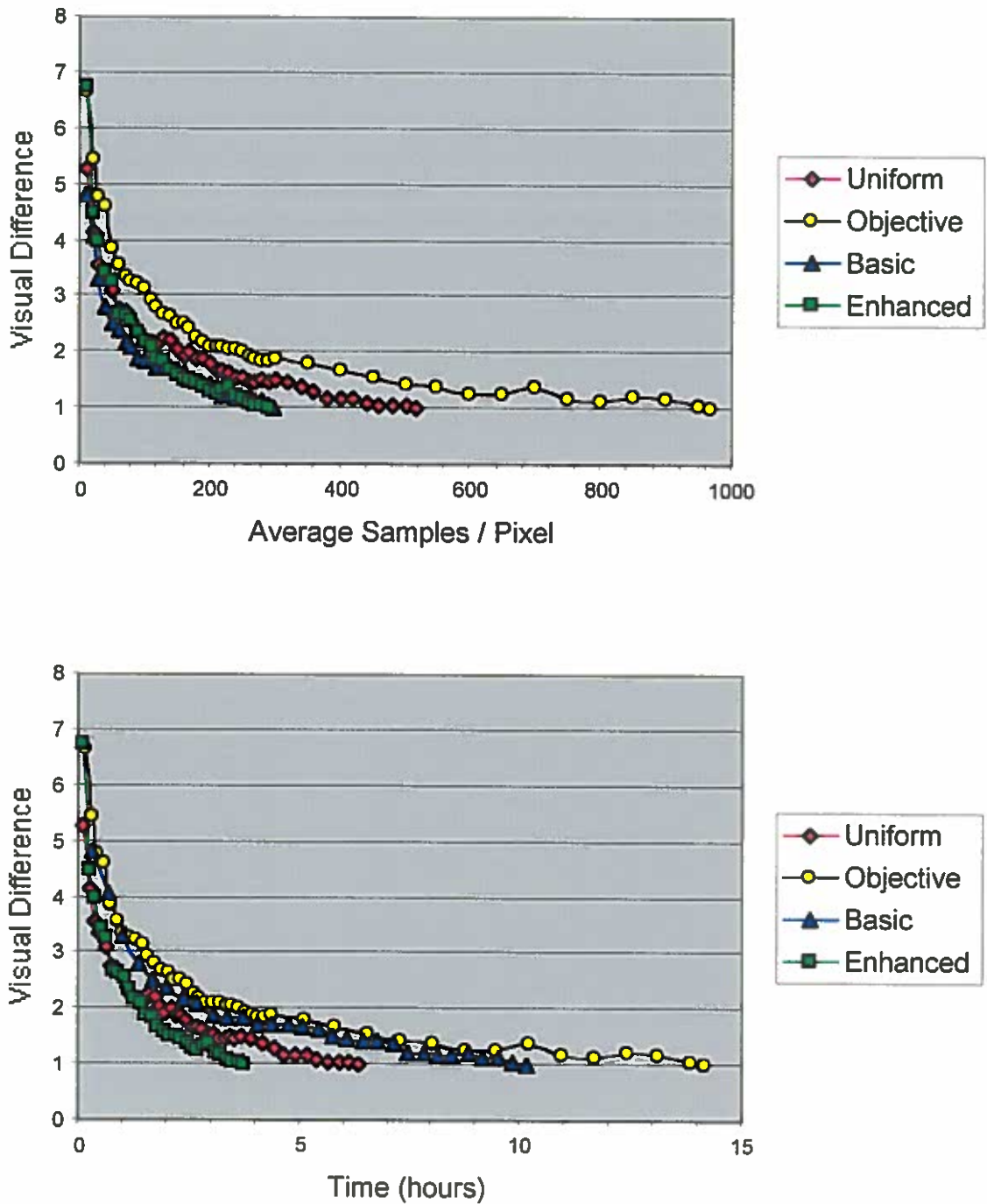


FIGURE 71. Sampling rates (top) and timing tests (bottom) for the masking example depicted in Figure 65.

the final enhanced perceptually based sampling algorithm.

There are a number of important points to observe in these charts. The first is the good correspondence between the required sampling rates for the basic perceptual and the enhanced perceptual algorithms. These two techniques require roughly the same number of samples to produce an image of a given visual quality. This can be seen by inspecting the plots of visual difference versus average samples/pixel.

The modifications described in pages 176–180 of this chapter significantly reduce the number of times that the image quality metric is applied during the course of the adaptive algorithm. This is especially true of the final enhancement that utilizes an oracle to predict the quantity of samples that are necessary in a given area of the image. The fact that an equivalent number of samples are required by both the basic and enhanced algorithms demonstrate that these modifications are able to lower the overhead of the technique without placing unnecessary samples. This results in a faster overall execution time.

The plots of the required number of samples also show that the peak visual difference is occasionally higher at intermediate sampling rates for the enhanced algorithm than it is for the basic perceptual algorithm. This difference comes from the nature of the two approaches. The basic algorithm always places samples at the location of most visible error. This produces the highest quality image possible at all points in the rendering process. However, it requires that the quality prediction is always current and that a search is performed to determine the location of each sample. The enhanced algorithm, on the other hand, tries to minimize the amount of time required to produce an image with the specified error tolerance. This is accomplished by sweeping across the locations of the image that need refinement,

and at each spot taking the number of samples that are estimated to be necessary in order to reach the desired accuracy. This process is far more efficient, but it does imply that some parts of the image will be much more refined than others during the intermediate stages of the algorithm. If a more uniform quality refinement is desired, intermediate target tolerances could be employed. This would be beneficial if the rendering needs to be halted before the image reaches the final accuracy.

The second point concerns the overhead of the sampling techniques. The top charts in Figures 68, 69, 70, and 71 plot the number of samples that each method uses to produce an image of a given visual quality. The bottom charts graph the amount of time required. If all four algorithms took the same amount of time to determine where to place samples, then the top and bottom graphs would exactly coincide except for a change in units along the bottom axis. However, this is not the case. The differences in the horizontal scaling of the curves in the bottom chart therefore indicate the relative amount of overhead incurred by each sampling method.

Let us begin by analyzing the performance of the uniform, objective, and basic perceptual sampling strategies. Uniform sampling has the lowest overhead of any sampling technique. This method does not employ any refinement test. It merely steps across the pixels of the image plane and places samples at each location. This approach therefore represents a baseline by which other techniques should be measured.

The objective sampling algorithm incurs slightly more overhead than uniform sampling because a refinement test is used to determine the placement of samples. This is evidenced by the fact that the objective curve shifts to the right relative to

the uniform curve in the timing tests. The refinement test used by the objective algorithm is based on a simple measurement of the sample variance. This test can be rapidly calculated and therefore does not have an overly significant impact on the total execution time.

The basic perceptual algorithm, on the other hand, utilizes a more expensive refinement test. This test adds approximately 1 millisecond of overhead to the amount of time required to evaluate every sample (on a 100 MHz CPU). The added expense of this approach increases the relative amount of time required to take a given number of samples. This is especially true when the samples can be evaluated very quickly, as in the case of direct light source sampling. The overhead of this technique produces a large shift in the timing curve that is plotted for the basic algorithm. The net result is that while the basic algorithm requires the fewest samples of these three methods (roughly 1/3 to 1/10 the number of samples used by worst method in these examples), it is only a middle performer in terms of overall execution time. This is, in itself, an accomplishment because the basic perceptual technique has the added benefit of allowing the user to specify a perceptual quality for the output image.

Finally, let us look at the performance of the enhanced algorithm. The modifications incorporated in this method are seen to significantly reduce the amount of overhead of the perceptually based sampling strategy. This is observed by comparing the results of the timing tests for the basic perceptual and the enhanced perceptual algorithms. Even though these two methods require approximately the same number of samples to produce an image of a given visual quality, the enhanced technique uses significantly less time. The amount of overhead of the enhanced algorithm is

still larger than the uniform method, but is on a par with the objective sampling technique. These two results are demonstrated by the relative scaling of the enhanced curve in comparison with the uniform and objective curves in the timing tests. This is a significant accomplishment given the complexity of the perceptually based refinement test.

The last point to observe in these charts is the aggregate speed of the enhanced algorithm in comparison to the two previous uniform and objective sampling techniques. This comparison demonstrates the ultimate benefit of the new perceptually based adaptive sampling method. The previous section discussed the number of samples required by the uniform and objective methods to produce an image of a given visual quality. These two approaches were shown to require three to ten times the number of samples necessary with the basic perceptually based technique. The enhanced algorithm has been shown to be able to produce images of a given visual quality using the same number of samples as the basic technique, but with much less overhead. This allows savings in the sampling rate to directly translate into a savings in overall execution time.

This fact can be observed in the results of the timing tests. In these tests the final enhanced algorithm is able to render images to the visible threshold using less time than both the uniform and objective sampling techniques in every example. There is also a variation in which of the uniform or objective methods produce the best results in the different examples. This makes it difficult to determine which of the two methods is preferable. The consistent speed of the enhanced perceptual algorithm overcomes this limitation and produces desirable execution times in all examples.

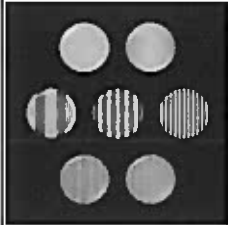
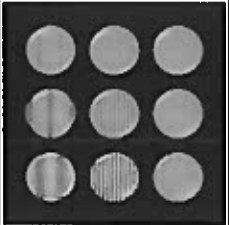

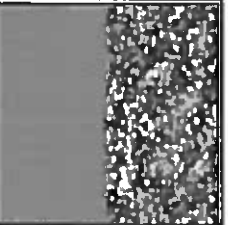
Relative Time				
Direct			Monte Carlo	
				
	Spatial Frequency	Chromatic Spatial Frequency	Contrast	Masking
Uniform	100.0	100.0	12.8	44.7
Objective	27.0	52.1	100.0	100.0
Enhanced	17.0	28.1	10.0	26.5

FIGURE 72. A summary of the timing test results. The values are based on the elapsed time required to render the images to the perceptual threshold. The time are reported as a percentage of the slowest algorithm.

Figure 72 accumulates the results of these timing tests into a single chart. The values are derived by rendering the images to the visible threshold (visual difference = 1 in the timing graphs), and the times are reported as a percentage of the worst case performer. In this chart we see that the enhanced algorithm is able to render the example images using only 10.0 to 28.1 percent of the elapsed time required by the existing algorithms. This is a significant increase in execution speed.

Summary

This chapter has presented a new perceptually based adaptive sampling algorithm. This algorithm utilizes a high speed, color visual difference predictor to

make subjective image quality assessments during the progression of the rendering algorithm. These quality predictions allow the algorithm to determine the regions of the image that contain visible artifacts and to focus effort on these areas. The quality metric is also used to determine when the image has been rendered to a desired perceptual accuracy. This allows the algorithm to be automatically halted when the image reaches a specified threshold. This technique was shown to produce consistent, predictable results regardless of the scene or the shading method employed.

The new adaptive sampling algorithm is based on a novel technique that computes the image approximation and error estimate directly in a Haar wavelet domain. This allows the algorithm to use information about the frequency characteristics of the images and facilitates a rapid application of the visual metric. It additionally provides a means to exploit the frequency statistics of natural images to make informed guesses about the error contribution of unrefined regions.

Two variations of the new perceptually based sampling method were presented. A simplified basic perceptual algorithm was presented first. This approach employs the image quality model to place each sample at the location of most perceptible error. This algorithm was used to describe the major features of the perceptually based sampling technique, and allowed a direct comparison with previous objective sampling methods. The final enhanced perceptual algorithm was described next. This complete version of the adaptive sampling algorithm incorporates a number of techniques that lower the expense of the perceptual algorithm by limiting the number of times and locations at which the visual metric is applied.

A number of examples were presented to both demonstrate the correctness of

algorithm's sample placements and to provide a comparison with existing sampling techniques. This comparison showed that the new method is able to produce images of a given visual quality using far fewer samples than prior algorithms. The quality of these images were measured using the visual difference predictor described in Chapter VII and verified by a presentation of the resulting images.

A further study of the required sampling rates demonstrated that the enhanced algorithm is able to produce images of a given visual quality using an equivalent number of samples as the basic technique. A number of timing tests were then performed to analyze the overall execution time of the new method. The enhanced algorithm was shown to have significantly less overhead than the basic approach. This allows lower sampling rates *and* lower execution times. Finally, the aggregate speed of the enhanced, perceptually based sampling algorithm was compared with the speed of prior sampling techniques. The new algorithm was shown to produce images of an equivalent visual quality using only a fraction of the time required by previous sampling methods. This illustrates the success and benefit of the new perceptually based adaptive sampling scheme.

CHAPTER XI

CONCLUSION

The developers of previous rendering algorithms have ignored the fact that synthesized images are intended to be viewed by a human observer. This has led to the creation of image synthesis algorithms that waste significant amounts of effort refining portions of an image that are already visually acceptable while ignoring regions that contain perceptible artifacts. This dissertation has presented work that resolves this problem through the development of a new perceptually based adaptive sampling algorithm.

The dissertation began with an overview of the theory and algorithms that are used to produce realistic images of synthetically defined scenes. This discussion included a derivation of the radiance equation. The radiance equation is a fundamental formula that must be solved in order to determine the amount of light present at a given position and orientation in space. This equation is based on a physical model of how light propagates and scatters within a three-dimensional environment. A number of global illumination algorithms were discussed next. These algorithms produce images by solving the radiance equation at the surfaces of a scene and projecting the result onto an image plane.

Global illumination algorithms are capable of producing stunningly realistic images. These images are often indistinguishable from a photograph taken of a real environment. However, the applicability of these algorithms is presently limited by

the enormous expense of these realistic simulations. Improving the performance of a global illumination algorithm is therefore crucial to its broad acceptance and the wide availability of realistic rendering systems.

This dissertation has focussed on the visibility tracing variant of the ray tracing algorithm. This is the most powerful and popular of the global illumination methods. Visibility tracing constructs an image from samples of the incident light energy at various locations of an image plane. These samples can be very expensive to compute, and the quantity of samples necessary to accurately reconstruct an image is one of the predominant factors affecting the expense of the algorithm.

The previous approaches to minimizing the required sampling rate were discussed in Chapter IV. These approaches took the form of a number of adaptive sampling algorithms. An adaptive sampling algorithm uses an error metric to determine the accuracy of regions of the synthesized image. The error estimate produced by this metric can be used to determine when an adequate number of samples have been taken in each area. This technique allows samples to be non-uniformly distributed across the image plane so that only the minimum number of samples are taken that are required to accurately reconstruct each region.

Prior error metrics are only based on the sampling that is performed at the image plane. However, the number of rays spawned from each surface intersection has an equally significant effect on the accuracy of the image. In order to quantify this effect, a new and comprehensive error metric for Monte Carlo ray tracing was developed. The development of this error metric is somewhat tangential to the primary thrust of this dissertation. However, a rigorous understanding of how to quantify and control the accuracy of an image is an important building block in the

development of a perceptually based adaptive sampling algorithm.

This error metric also yielded a number of significant results. It provides a more complete method of determining image accuracy than is available with current error metrics. This metric, for the first time, allows the effect of surface spawning on image accuracy to be precisely determined. In addition, this work shows how the metric can be used to derive the optimal splitting and Russian roulette rates at each level of the ray tree. This is far more information than has been previously available on the selection of appropriate spawning rates. The use of these optimal spawning rates can improve the efficiency of Monte Carlo ray tracing.

The problem with the direct application of this type of error metric is that it ignores the perceptual response of the human visual system when determining the accuracy of an image. What is necessary is a means to map these objective error measurements to units of subjective image quality. Toward this end, the existing literature on subject image quality metrics was reviewed.

These metrics are embedded within what are known as visual difference predictors. These predictors are capable of determining the perceptibility of differences between two images. This is accomplished by processing each of the images in a manner analogous to that performed by the human visual system and measuring the differences between the images in the resulting perceptual space. The latest difference predictors model a number of key aspects of the visual system. These include contrast nonlinearity, separation of spatial frequency and orientation selective channels, spatial frequency sensitivity, and masking. These recent models produce results that are very similar to the perceptual response of a human observer.

The existing visual difference predictors are designed to produce quality pre-

dictions that are as accurate as possible, with little consideration given to the speed of the algorithm. This limits the utility of these predictors for applications such as adaptive sampling where speed is a primary concern. In addition, the majority of existing visual difference predictors are only designed to handle achromatic images. This neglects a number of key aspects of human perceptual response.

These two problems are solved by the development of a new high speed, color visual difference predictor. This predictor is a modified version of Lubin's [88] image quality model. The new predictor utilizes a Haar wavelet transform to decompose an image into spatial frequency and orientation selective channels. This approximation improves the speed of both direct and iterative quality predictions without a significant loss of accuracy. The predictor is also extended to perform color difference predictions. This addresses a significant aspect of the visual system that was neglected in most previous image quality models. This extension includes a novel chromatic aberration stage that corresponds with the structure of the visual pathway and allows accurate quality predictions for both chromatic color channels. The net result is a new difference predictor that can assess the perceptual quality of color images in a fraction of the time used by previous metrics for images containing only a single achromatic channel.

There are very few previous applications of perception in the field of computer graphics. This is surprising given that computer graphics is so intimately tied to the subjective impression of a human observer. This is perhaps one of the reasons why dealing with human perceptual issues is viewed by many researchers to be one of the top two unsolved problems in the field of computer graphics. This deficiency is significantly addressed by this dissertation, not only by its development of prac-

tical perceptual algorithms, but also by increasing the awareness of other computer graphics researchers as to the relevance and specifics of human perceptual response.

One major factor that has made it difficult to incorporate perceptual models within a rendering algorithm is that all previous image synthesis techniques have operated in the spatial domain. Without knowledge of the frequency characteristics of an image it is impossible to fully exploit the varying sensitivity of the human visual system. This issue was first addressed by the work presented in Chapter IX.

This chapter describes the development of a new adaptive sampling algorithm that synthesizes images directly into a frequency representation. This allows a simple, frequency dependent perceptual quality model to be used to direct the placement of samples. In this way the most apparent image artifacts are removed before those that are less visible. This work is also significant in that it is the first time that the effect of visual masking is introduced to the computer graphics community, as well as the first time that an attempt is made to synthesize images directly into a compressed format.

The novel ideas presented in this work are refined further to create a second generation, perceptually based adaptive sampling algorithm. This new algorithm significantly improves the speed and practicality of the original technique. The second generation algorithm also employs a more sophisticated perceptual quality model that includes the effects of masking and chromatic aberration.

This new perceptually based sampling algorithm uses a high speed, color visual difference predictor to make subjective image quality assessments during the progression of the rendering algorithm. These assessments allow the algorithm to detect and focus effort on the regions of the image containing the most perceptible

artifacts. These quality assessments are also used to determine when the image has been rendered to a desired visual quality. This enables the algorithm to produce images of consistent visual quality regardless of the scene or shading technique that is employed.

The new adaptive algorithm uses a novel technique that computes the image approximation and its error estimate directly in a Haar wavelet representation. This allows the algorithm to rapidly compute the frequency characteristics of the image and its error. The representation is also equivalent to the cortex transform performed at an early stage of the visual model. This minimizes the overhead of applying this model. In addition, the synthesis of images into the frequency domain permits the use of statistical commonalities of natural images to make informed guesses about the error contribution of unrefined frequency terms.

The simplified, basic adaptive algorithm employs a quality prediction to direct the placement of each and every sample. This is clearly not necessary and adversely affects the performance of the algorithm. The complete, enhanced version of the algorithm includes a number of modifications that greatly reduce the overhead of the adaptive technique. This is accomplished by minimizing the number of times and locations at which the visual model is applied. These enhancements are shown to allow a reduction in the sampling rate to directly translate into savings in the overall execution time.

Finally, a number of tests were performed to evaluate the performance of the new algorithm. These tests compared the results of the perceptually based adaptive algorithm with the results of two previous sampling techniques. These tests show that the new algorithm is able to outperform the previous sampling techniques in

terms of both the number of samples and the elapsed time required to produce images of a given visual quality.

In conclusion, this dissertation describes the benefits of utilizing a model of human visual perception to guide the construction of synthetic images. This technique allows images to be produced in a manner that is consistent with how they are perceived. The result of this work not only produces an interesting theoretical result, but also a practical application that can improve the performance of realistic rendering systems.

APPENDIX A

DERIVATION OF THE TRANSPORT, INTEGRAL, AND RADIANCE EQUATIONS

This appendix provides the complete derivation of the transport, integral and radiance equations. These sections are intended to supplement the abbreviated derivations discussed in Chapter II. The development of these sections is strongly related to the works of Glassner [56] and Arvo [70].

Derivation of the Transport Equation

This section describes the complete derivation of the particle transport equation. This equation describes a relationship that governs the flow of abstract particles within Euclidean space. This particle flow relationship has been used to model a variety of phenomenon, including liquid dynamics, automobile traffic flow and (most importantly) light transport.

A particle moving in space has 5 degrees of freedom; three for position and two for direction. The equations in this and subsequent sections will use \mathcal{R}^3 to signify the three dimensions of Euclidean space and \mathcal{S}^2 to signify the set of all directions defined across an encompassing unit sphere. The product of these two spaces $\mathcal{R}^3 \times \mathcal{S}^2$ is called the phase space. The term r will be used to describe a specific position in \mathcal{R}^3 and ω will be used to describe a direction in \mathcal{S}^2 . Additionally, the term V will be used to describe a volume within \mathcal{R}^3 and Γ will specify a solid angle in \mathcal{S}^2 (i.e.

a subset of the directions defined across the unit sphere).

The primary quantity of interest in the transport equation is called the flux. Flux describes the quantity of particles passing through an environment per unit time. It is defined as

$$\Phi = \frac{dQ}{dt} \left[\frac{\text{particles}}{S} \right], \quad (56)$$

where Q is the number of abstract particles. A related quantity is the phase space flux $\Phi(r, \omega)$. The phase space flux is simply the flux present at a specific point in space r , flowing in the direction ω .

In order to develop the transport equation we seek to define a relationship governing the flow of particles through an arbitrary volume V in an arbitrary set of directions Γ . This relationship is based upon quantifying the flux gains and losses over this specific region of phase space. A balance equation will then be developed that relates these quantities and forms the final transport equation. There are five categories of gains and losses which are illustrated in Figure 1 of Chapter II. They are streaming, emission, absorption, outscattering and inscattering.

Streaming is a measure of the net flow of particles through a volume. It may be thought of as the amount of flux that is either injected into or escapes from the surfaces of a volume. The net change in flux due to streaming may be found by integrating the flux across the surface of a volume. Since flux may be viewed as a vector quantity, this value will be either positive or negative depending on whether there is a net flow of particles into or out of the volume. The equation for the net

change in flux due to streaming is

$$\Phi_s = \int_{\Gamma} \int_S \Phi(s, \omega) ds d\omega, \quad (57)$$

where we have additionally integrated this quantity over the directions of interest.

The emission term is a description of the flux contributions due to particles emitted from within the volume itself. In order to describe this term we must assume to have a volumetric emission flux function $\epsilon(r, \omega)$ which describes the flux emitted at all points and directions within the volume. The total flux emitted is then simply the sum of emissions at all points and angles of interest:

$$\Phi_e = \int_{\Gamma} \int_V \epsilon(r, \omega) dr d\omega. \quad (58)$$

The absorption term describes the total flux absorbed within a volume. It is similar in nature to the emission term except that we define a function $\sigma_a(r, \omega)$ which describes the percentage of flux absorbed at a specific point and direction within the volume. The total flux absorbed is simply the sum over $\Gamma \times V$ of the percentage of flux absorbed times the flux at that point and direction:

$$\Phi_a = \int_{\Gamma} \int_V \sigma_a(r, \omega) \Phi(r, \omega) ds d\omega. \quad (59)$$

Particles that interact with a media may also be scattered into some new direction ω' . A volume scattering probability function $\kappa(r, \omega \rightarrow \omega')$ is defined to account for this effect. This function describes the probability that a particle at r originally traveling in the direction ω will be deflected into the new direction ω' .

The outscattering term is defined to be the quantity of flux that is scattered from the original region of interest Γ into some other direction. It is given by

$$\Phi_o = \int_{\Gamma} \int_V \Phi(r, \omega) \int_{S^2} \kappa(r, \omega \rightarrow \omega') d\omega' ds d\omega. \quad (60)$$

This is the integral over the volume and angles of interest of the flux at that point times the total probability that a particle is deflected into any other angle.

The inscattering term is similar to the outscattering term except that inscattering accounts for the flux due to particles originally traveling in any direction ω' that are scattered into the region of interest Γ . This term is given by

$$\Phi_i = \int_{\Gamma} \int_V \int_{S^2} \kappa(r, \omega' \rightarrow \omega) \Phi(r, \omega') d\omega' ds d\omega. \quad (61)$$

The transport equation is generated from these terms by assuming that the system is in equilibrium. This means that the illumination levels are not growing brighter or dimmer over time. When this is the case, the gains from emission and inscattering exactly balance the losses due to streaming, outscattering and absorption. This yields the equation:

$$\Phi_e + \Phi_i = \Phi_s + \Phi_o + \Phi_a. \quad (62)$$

If we examine the terms in this expression we see that the outer two integrals are the same for all terms except for the streaming term. This term contains a surface integral rather than a volume integral. However, we can make use of Gauss' theorem to convert this term to a volume integral. When we apply Gauss' theorem

we get the result that

$$\Phi_s = \int_{\Gamma} \int_V \omega \cdot \nabla \Phi(r, \omega) dV d\omega, \quad (63)$$

where ∇ is the del operator.

Since the outermost integral of all terms is now the same, we can simply remove them. In essence this is saying that the relationship that we have built for a volume V and solid angle Γ also holds for all specific points r and directions ω . This yields the standard one-speed particle transport equation:

$$\begin{aligned} \epsilon(r, \omega) + \int_{S^2} \kappa(r, \omega' \rightarrow \omega) \Phi(r, \omega') d\omega' = \\ \omega \cdot \nabla \Phi(r, \omega) + \Phi(r, \omega) \int_{S^2} \kappa(r, \omega \rightarrow \omega') d\omega' + \sigma_a(r, \omega) \Phi(r, \omega). \end{aligned} \quad (64)$$

This equation describes the balance relationship that the flux at all points and directions in space must satisfy. This balance relationship by itself does not tell us anything about the actual value of the flux at a specific point. It merely describes the flux in terms of the incident flux from all directions. In order to actually determine the flux we must define boundary conditions which explicitly describe the flux at some boundary points in space. The actual flux at all other points can then be determined in relationship to the flux at these boundary points. In terms of computer graphics these boundary conditions can be thought of as describing the emissive and reflective properties of objects within the scene. Once these properties are described it is then possible to compute the flux at all other

points within the scene. The most general form of these boundary conditions is

$$\Phi(s, \omega) = \epsilon_s(s, \omega) + \int_{\Theta_i^+(s)} \kappa(s, \omega' \rightarrow \omega) \Phi(s, \omega') d\omega'. \quad (65)$$

This expression describes the flux at a surface in terms of the emission at that surface and the flux scattered from the surface from all incident directions. This expression combined with Equation 64 allows us to explicitly compute the flux at any point within the environment.

Derivation of the Integral Equation

The transport equation given by Equation 64 allows us to compute the flux at an arbitrary point in space. However, the form of this equation is not particularly easy to work with since it contains both an integral and a derivative (∇) of the unknown quantity Φ . This section describes the conversion of the transport equation to a purely integral form in order to simplify the process of solving for the flux.

The first step in this conversion process is to change the gradient (∇) to a standard derivative. This is accomplished by reparameterizing the position vector r by the expression $r - \alpha\omega \big|_{\alpha=0}$. This expression says that the position r is given by the line that passes through r in the direction $-\omega$ at the point where $\alpha = 0$. This method of parameterization is also useful because we will eventually wish to describe the flux at a point by looking backwards along ω to see where that flux is coming from. The expression $\omega \cdot \nabla \Phi(r, \omega)$ from the streaming term is just the change in the flux at the point r along the direction ω and may therefore be rewritten as

$$\omega \cdot \nabla \Phi(r, \omega) = -\frac{\partial}{\partial \alpha} \Phi(r - \alpha\omega, \omega) \big|_{\alpha=0}. \quad (66)$$

The next step is to simplify Equation 64 by defining some new terms. The outscattering and absorption terms on the right-hand side of Equation 64 are simply constants times the unknown function $\Phi(r, \omega)$ so we can collect them into a single term $\sigma_c(r, \omega)$:

$$\sigma_c(r, \omega) = \int_{S^2} \kappa(r, \omega \rightarrow \omega') d\omega' + \sigma_a(r, \omega). \quad (67)$$

Additionally, the emission and inscattering terms on the left-hand side of Equation 64 are only infinitesimally dependent on the flux through the direction ω and can also be considered constant with respect to $\Phi(r, \omega)$. These terms can be rewritten as a gain term:

$$G(r, \omega) = \epsilon(r, \omega) + \int_{S^2} \kappa(r, \omega' \rightarrow \omega) \Phi(r, \omega') d\omega'. \quad (68)$$

Substituting these expressions into Equation 64 and reparameterizing in terms of α allows us to express this equation in a much simpler form:

$$\frac{\partial}{\partial \alpha} \Phi(r - \alpha\omega, \omega) - \Phi(r - \alpha\omega, \omega) \sigma_c(r - \alpha\omega, \omega) = -G(r - \alpha\omega, \omega). \quad (69)$$

This expression is simply a linear, first-order differential equation. It can be solved for Φ using the technique of integrating factors. This technique tells us that if we define a term

$$\mu(r - \alpha\omega, \omega) = \exp \left[\int_0^\alpha -\sigma_c(r - \tau\omega, \omega) d\tau \right], \quad (70)$$

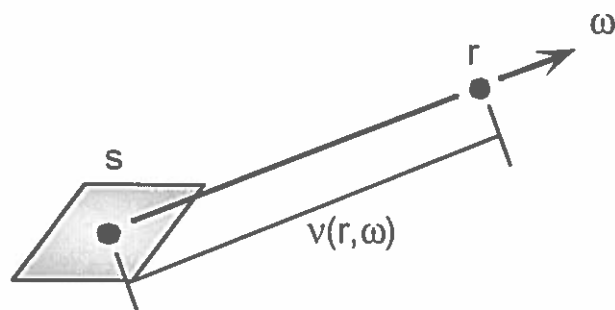


FIGURE 73. The visible surface function (after Glassner [56]).

we can re-express Equation 69 as

$$\frac{\partial}{\partial \alpha} [\mu(r - \alpha\omega, \omega)\Phi(r - \alpha\omega, \omega)] = -\mu(r - \alpha\omega, \omega)G(r - \alpha\omega, \omega). \quad (71)$$

Integrating both sides and solving for $\Phi(r - \alpha\omega, \omega)$ yields

$$\Phi(r - \alpha\omega, \omega) = \mu(r - \alpha\omega, \omega)\Phi(r - \alpha\omega, \omega) + \int_0^\alpha \mu(r - \tau\omega, \omega)G(r - \tau\omega, \omega)d\tau. \quad (72)$$

This expression tells us how to compute the flux at a point r in terms of the flux at some point $r - \alpha\omega$ backwards from r along the direction ω . These points at $r - \alpha\omega$ are simply our surface points s and the flux at these points are described from our boundary conditions of Equation 65. We define a visible surface function $\nu(r, \omega)$ to aid us in finding these surface points. This function takes as input a point r and returns the distance to the nearest surface point in the direction $-\omega$. This function is illustrated in Figure 73.

The integral form of the transport equation can now be found by substituting the surface point s for $r - \nu(r, \omega)\omega$ and expanding the terms of G . This yields the

final equation that we seek:

$$\Phi(r, \omega) = \mu(r, s)\Phi(s, \omega) + \int_0^h \left[\mu(r, a) \left(\epsilon(a, \omega) + \int_{S^2} \kappa(r, \omega' \rightarrow \omega) \Phi(a, \omega') d\omega' \right) \right] d\alpha, \quad (73)$$

where

$$\begin{aligned} a &= r - \alpha\omega \\ h &= \nu(r, \omega) \\ s &= r - h\omega \\ \mu(r, s) &= \exp \left[\int_0^{\|r-s\|} -\sigma_c(r - \tau\omega, \omega) d\tau \right] \\ \sigma_c(r, \omega) &= \int_{S^2} \kappa(r, \omega \rightarrow \omega') d\omega' + \sigma_a(r, \omega). \end{aligned}$$

This equation could probably stand explanation at this point. We will proceed by first analyzing the sub-expressions found within this equation. The term σ_c is simply the flux percentage lost due to outscattering and volumetric absorption at a specific point along the direction of interest ω . The term μ is then the sum of these losses between two points in space. The gain term embedded within the integral on the right side of Equation 73 is just the flux gains due to volumetric emission and inscattering. This term is similarly described at a specific point and along the direction of interest ω .

The term $\mu(r, s)\Phi(s, \omega)$ is the flux contribution to the point of interest r from the surface point s . The flux at the surface point s is found from our boundary conditions of Equation 65. The flux at this surface point is weighted by μ which is the percentage of the surface flux lost due to volumetric absorption and outscattering

as it travels from s to r .

The integral on the right-hand side of Equation 73 accounts for the flux contributions due to volumetric effects. The terms within the integral represent the flux gains from a point in space weighted by the percentage of that flux lost as it travels from that point to the point of interest r . The integral is then the sum of these gains across all points from the surface to the point of interest.

The integral form of the transport equation therefore describes the flux at a particular point in space as the net contribution of these final two sub-expressions. The first accounts for the flux contributions from the boundary surface and the second accounts for the volumetric flux contributions between the surface and the specified point.

Derivation of the Full Radiance Equation

This section describes the derivation of the full radiance equation as well as a number of common approximations to it. The radiance equation is based upon the integral form of the transport equation but is defined in terms of radiance instead of the abstract notion of flux. It additionally incorporates the effects of blackbody radiation, phosphorescence and fluorescence.

In order to generate the radiance equation we must first convert from flux to radiance. This is accomplished by noting that since the transport equation defines a relationship between the flux at all points and directions in space, it must therefore define the same relationship for the flux densities at those points. Described another way, this says that if a relationship holds for a function (Φ^e) , it must also hold for the derivatives of that function $(\frac{d^2\Phi^e}{dA^2d\omega})$.

Since the relationship will be stated in terms of radiance we must also convert the scattering term (κ) to a BRDF term ($f_r \cos\theta$). The cosine is necessary to account for differences in orientation of source and receiving patches.

The transport equation expresses emittance with the single term ϵ . A more specific expression should state that light emittance can be expressed in two forms; blackbody ϵ_b and phosphorescence ϵ_p :

$$\epsilon(p, t, \lambda) = \epsilon_b(p, \omega, t, \lambda) + \epsilon_p(p, \omega, t, \lambda). \quad (74)$$

Blackbody emission can be thought of as direct or immediate emission. It is given by

$$\epsilon_b(\nu, \omega, p, t) = m_b(\omega) \frac{2\pi h\nu^3}{c^2} \frac{1}{\exp[h\nu/kT(p, t)] - 1}. \quad (75)$$

This expresses the emission in terms of basic quantities such as the frequency, wavelength and speed of light. For convenience we will express this value in terms of its radiance L^e .

Phosphorescent emission is the result of a material trapping incident energy for a period of time before re-emitting it. This term is given by

$$\epsilon_p(p, \omega, t, \lambda) = m_p(\omega) \int_{-\infty}^t d(t - \tau) P_p(p, \lambda) \int_{\Theta_i} L(p, \omega', \lambda, \tau) \cos\theta' d\omega' d\tau. \quad (76)$$

This says that the phosphorescent emission is the result of a directional function times the integral from the start of time to the present of the decay times the phosphorescent efficiency function times the incident radiance from all directions.

Fluorescence is the effect of a material trapping energy at one wavelength λ' and re-radiating it at another wavelength λ . This can be thought of as a scattering that occurs across wavelengths. To account for this effect we change the incident radiance in the scattering term to account for radiance scattered from other frequencies. The scattering term then becomes

$$\int_{\Theta_i^i} f(p, \omega' \rightarrow \omega) \int_{\mathcal{R}_\nu} P_f(p, \lambda' \rightarrow \lambda) L(p, \omega', \lambda') d\lambda' d\omega', \quad (77)$$

where we have generated the incident radiance by integrating across all wavelengths, a fluorescent efficiency function (P_f) times the radiance at that wavelength.

The full radiance equation can be given by combining all of the pieces above. This yields the complete description of the radiance at a point in space:

$$\begin{aligned} L(r, \omega, \lambda, e, t) = & \mu(r, s) [L^e(s, \omega, t, \lambda) \\ & + m_p(\omega) \int_{-\infty}^t d(t - \tau) P_p(s, \lambda) \int_{\Theta_i^i} L(s, \omega', \lambda, e, \tau) \cos\theta' d\omega' d\tau \\ & + \int_{\Theta_i^i} f(s, \omega' \rightarrow \omega, \lambda) \int_{\mathcal{R}_\nu} P_f(s, \lambda' \rightarrow \lambda) L(s, \omega', \lambda', e, t) d\lambda' \cos\theta' d\omega' \\ & + \int_0^{h(r, \omega)} \mu(r, a) [L^e(a, \omega, t, \lambda) \\ & + m_p(\omega) \int_{-\infty}^t d(t - \tau) P_p(a, \lambda) \int_{\Theta_i^i} L(s, \omega', \lambda, e, \tau) \cos\theta' d\omega' d\tau \\ & + \int_{\Theta_i^i} f(a, \omega' \rightarrow \omega, \lambda) \int_{\mathcal{R}_\nu} P_f(a, \lambda' \rightarrow \lambda) L(a, \omega', \lambda', e, t) d\lambda' \cos\theta' d\omega'] d\alpha. \quad (78) \end{aligned}$$

To understand this expression we must analyze it a piece at a time. The portion

of the expression within the two sets of square brackets is just the radiance gains from a point in space. It includes blackbody emission, phosphorescent emission and inscattering (including radiance scattered from other frequencies). In the first half of the radiance equation this quantity is evaluated at a surface point and multiplied by μ which is the attenuation from the surface point to the point of interest. The second half of the radiance equation accounts for volumetric emissions by evaluating the radiance gains between the surface point and the point of interest. These gains are then multiplied by the attenuation from the point of emission to the point of interest.

The full radiance equation is obviously very complex, and it would take an exorbitant amount of time to compute. We therefore seek to reduce it to a more manageable scale. The first common simplification is to eliminate the effects of polarization, phosphorescence and fluorescence.

This means that we are assuming that light is unpolarized, that the different wavelengths of light do not interact and that the illumination does not change over time. These are somewhat uncommon effects and can therefore be eliminated without a large loss of generality. The simplified equation is then

$$L(r, \omega) = \mu(r, s) \left[L^e(s, \omega) + \int_{\Theta_i^i} f(s, \omega' \rightarrow \omega, \lambda) L(s, \omega') \cos\theta' d\omega' \right] + \int_0^{h(r, \omega)} \mu(r, a) \left[L^e(a, \omega) + \int_{\Theta_i^i} f(a, \omega' \rightarrow \omega, \lambda) L(a, \omega') \cos\theta' d\omega' \right] d\alpha. \quad (79)$$

The final simplification assumes that all synthesis occurs in a vacuum. This means that there are no volumetric effects due to emission, scattering or absorption.

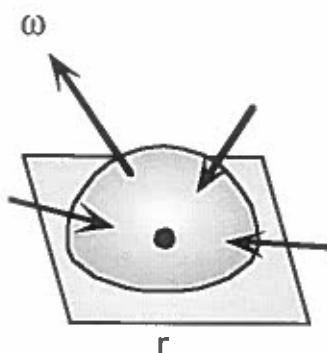


FIGURE 74. The geometry of the outgoing form of the radiance equation (after Glassner [56]).

We are therefore only concerned with the effect of surfaces on the radiance, and the radiance equation reduces to

$$L(r, \omega) = L^e(s, \omega) + \int_{\Theta_i} f(s, \omega' \rightarrow \omega, \lambda) L(s, \omega') \cos\theta' d\omega'. \quad (80)$$

This equation determines the radiance at a point in space by looking backward from that point to determine radiance incident from the nearest surface in the direction $-\omega$. It is often useful to instead express the radiance in the outgoing direction from a surface point r . This situation is illustrated in Figure 74. A slight modification to notation is all that is required to yield the outgoing form of Equation 80:

$$L(r, \omega^o) = L^e(s, \omega^o) + \int_{\Theta_i} f(s, \omega' \rightarrow \omega^o, \lambda) L(r, \omega') \cos\theta_r d\omega'. \quad (81)$$

This expression says that the outgoing radiance from a point r is just the sum of the emitted radiance from that surface point and the reflected radiance from all incident directions. This limited form of the full radiance equation is what most current global illumination algorithms are dedicated to solving.

APPENDIX B

PROOF OF OPTIMUM SPAWNING FORMULA

This appendix describes the proof of Equation 27. It is an extension of a proof given by Mikhailov [97]. The validity of this expression for the base case of two variables is easily seen by minimizing $cost \times V[f]$ with respect to N_1 and solving for N_1 . For an arbitrary number of variables the proof proceeds by induction. The variance and cost of a function of $k + 1$ variables can be expressed as

$$V[f] = D_0 + \frac{1}{N_1} \left(D_1 + \frac{1}{N_2} D_2 + \cdots + \frac{1}{N_2 \cdots N_k} D_k \right) = D_0 + \frac{1}{N_1} D_1^* \quad (82)$$

and

$$cost = 1 + N_1 (1 + N_2 + \cdots + N_2 \cdots N_k) = 1 + N_1 C_1^*. \quad (83)$$

The product of Equations 82 and 83 yields Q (Equation 25). Minimizing Q with respect to N_1 yields

$$N_1 = \sqrt{\frac{D_1^*}{C_1^* D_0}}. \quad (84)$$

Substituting this result into the expression for Q , we see that

$$Q = \left(\sqrt{D_0} + \sqrt{C_1^* D_1^*} \right)^2 \quad (85)$$

This equation is minimized when $C_1^* D_1^*$ is minimized. Assuming that Equation 27 minimizes Q for k variables (i.e. yields $N_2 \cdots N_k$ that minimizes $C_1^* D_1^*$), then to complete the proof we must show that Equation 27 also holds for N_1 . By inspection of Equation 84 we see that this reduces to showing $\frac{D_1^*}{C_1^*} = D_1$, or in general that

$$\frac{V[f]}{\text{cost}} = D_0. \quad (86)$$

For $k = 0$ this result is obvious. For arbitrary k this can be seen by substituting Equation 84 into the ratio of Equations 82 and 83.

BIBLIOGRAPHY

- [1] Ahumada Jr., A., "Computational Image Quality Metrics: A Review," *SID Digest*, Vol. 24, pp. 305-308, 1993.
- [2] Akimoto, T., Mase, K., Hashimoto, A., and Suenga, Y., "Pixel Selected Ray Tracing," *Proc. Eurographics '89*, W. Hansmann, F. R. A. Hopgood, and W. Straber, Editors, North-Holland, Amsterdam, pp. 39-50, 1989.
- [3] Albert, A. E. and Gardner Jr., L. A., *Stochastic Approximation and Nonlinear Regression*, M.I.T. Press: Massachusetts, 1967.
- [4] Appel, A., "Some Techniques for Shading Machine Renderings of Solids," *AFIPS 1968 Spring Joint Computer Conference*, Vol. 32, pp. 37-45, 1968.
- [5] Argence, J., "Antialiasing for Ray Tracing Using CSG Modeling," *New Trends in Computer Graphics (Proc. CG International '88)*, Tosiyasu Kunii, Editor, Springer-Verlag, New York pp. 199-208, 1988.
- [6] Arvo, J., "Backward Ray Tracing," *Developments in Ray Tracing*, ACM SIGGRAPH, Course Notes, 1986.
- [7] Arvo, J. and Kirk, D., "Fast Ray Tracing by Ray Classification," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 55-64, 1987.
- [8] Arvo, J. and Kirk D., "Particle Transport and Image Synthesis," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 63-66, 1990.
- [9] Arvo, J., Torrance, K. and Smits, B., "A Framework for the Analysis of Error in Global Illumination Algorithms," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 75-84, 1994.
- [10] Barten, P. G. J., "The Square Root Integral (SQRI): A New Metric to Describe the Effect of Various Display Parameters on Perceived Image Quality," *Human Vision, Visual Processing, and Digital Display*, Proc. SPIE, Vol. 1077, pp. 73-82, 1989.
- [11] Bedford, R. E. and Wyszecki, G. "Axial Chromatic Aberration of the Human Eye," *Journal of the Optical Society of America*, Vol. 47, No. 6, pp. 564-565, 1957.

- [12] Blackmore, C. and Campbell, F. W., "On the Existence of Neurones in the Human Visual System Selectively Sensitive to the Orientation and Size of Retinal Images," *J. Physiol. (Lond.)*, Vol. 203, pp. 237-260, 1969.
- [13] Bodewig, E., *Matrix Calculus*, North Holland Publishing Company: Amsterdam, 1956.
- [14] Bolin, M. R., "Application of Image Compression Algorithms to Realistic Image Synthesis," *Technical Report CIS-TR-95-19, University of Oregon*, June 1995.
- [15] Bolin, M. R. and Meyer G. W., "A Frequency Based Ray Tracer," *Computer Graphics, Annual Conference Series, ACM SIGGRAPH*, pp. 409-418, 1995.
- [16] Bolin, M. R. and Meyer G. W., "An Error Metric for Monte Carlo Ray Tracing," *Rendering Techniques '97*, J. Dorsey and P. Slusallek, Editors, Springer-Verlag, New York, pp. 57-68, 1997.
- [17] Bolin, M. R. and Meyer G. W., "A Perceptually Based Adaptive Sampling Algorithm," *Computer Graphics, Annual Conference Series, ACM SIGGRAPH*, pp. 299-309, 1998.
- [18] Bouville, C., Tellier, P., and Bouatouch, K., "Low Sampling Densities Using a Psychovisual Approach," *Proc. Eurographics '91*, Frits H. Post and Wilhelm Barth, Editors, North-Holland, Amsterdam, pp. 167-182 1991.
- [19] Bradley, A., Switkes, E., and DeValois, K., "Orientation and Spatial Frequency Selectivity of Adaptation to Color and Luminance Gratings," *Vision Research*, Vol. 28, No. 7, pp. 841-856, 1988.
- [20] Buchsbaum, G., "Color Signal Coding: Color Vision and Color Television," *Color Research and Application*, Vol. 12, No. 5, pp. 266-269, 1987.
- [21] Buchsbaum, G. and Gottschalk, A., "Trichromacy, Opponent Colours Coding and Optimum Colour Information Transmission in the Retina," *Proceedings of the Royal Society of London, Series B*, Vol. 220, pp. 89-113, 1983.
- [22] Burt, P. J. and Adelson, E. H., "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, Vol. 31, No. 4, pp. 532-540, 1983.
- [23] Campbell, F. W. and Kulikowski, J. J., "Orientational Selectivity of the Human Visual System," *J. Physiol. (Lond.)*, Vol. 187, pp. 437-445, 1966.

- [24] Campbell, F. W. and Maffei, L., "Electrophysiological Evidence for the Existence of Orientation and Size Detectors in the Human Visual System," *J. Physiol. (Lond.)*, Vol. 207, pp. 635-652, 1970.
- [25] Campbell, F. W. and Robson, J. G., "Application of Fourier Analysis to the Visibility of Gratings," *J. Physiol. (Lond.)*, Vol. 197, pp. 551-566, 1968.
- [26] Carlson, C. R. and Cohen, R. W., "A simple psychological model for predicting the visibility of displayed information," *Proceedings of the SID*, Vol. 21, No. 3, pp. 229-246, 1980.
- [27] Chiu, K., Herf, M., Shirley, P., Swamy, S., Wang, C., and Zimmerman, K., "Spatially Non-Uniform Scaling Functions for High Contrast Images," *Proceedings of Graphics Interface 1993*, pp. 245-253, 1993.
- [28] Cicerone, C. M. and Nerger, J. L., "The Relative Numbers of Long-Wavelength-Sensitive to Middle-Wavelength-Sensitive Cones in the Human Fovea Centralis," *Vision Research*, Vol. 29, No. 1, pp. 115-128, 1989.
- [29] CIE, "An Analytic Model for Describing the Influence of Lighting Parameters Upon Visual Performance," *Vol 1. Technical Foundations*, CIE 19/2.1, Technical Committee 3.1 1981.
- [30] Cohen, A., Daubechies, I., and Feauveau, J. C., "Biorthogonal Bases of Compactly Supported Wavelets," *Communications on Pure and Applied Mathematics*, Vol. 45, No. 5, pp. 485-500, 1992.
- [31] Cohen, M. and Greenberg, D. P., "The Hemi-Cube: A Radiosity Solution for Complex Environments," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 31-40, 1985.
- [32] Cohen, M. F. and Wallace, J. R., *Radiosity and Realistic Image Synthesis*, Academic Press: San Diego, 1993.
- [33] Cook, R. L., "Stochastic Sampling in Computer Graphics," *ACM Transactions on Graphics*, Vol. 5, No. 1, pp. 51-72, 1986.
- [34] Cook, R. L., Porter, T. and Carpenter, L., "Distributed Ray Tracing," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 137-145, 1984.
- [35] Daly, S., "Application of a Noise Adaptive Contrast Sensitivity Function to Image Data Compression," *SPIE Human Vision, Visual Processing and Digital Display*, Vol 1077, pp. 217-227, 1989.

- [36] Daly, S., "The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity," *Digital Images and Human Vision*, A. B. Watson, Editor, MIT Press, Cambridge, MA, pp. 179-206, 1993.
- [37] Daubechies, I., "Orthonormal Bases of Compactly Supported Wavelets," *Communications on Pure and Applied Mathematics*, Vol. 41, No. 7, pp. 909-996, 1988.
- [38] DeValois, R. L. and DeValois, K. K., *Spatial Vision*, Oxford University Press: New York 1990.
- [39] DeValois, R. L., Smith, C. J., Karoly, A. J., and Kitai, S. T., "Electrical Responses of Primate Visual System, I. Different Layers of Macaque Lateral Geniculate Nucleus," *Journal of Comparative and Physiological Psychology*, Vol. 51, pp. 662-668, 1958.
- [40] Dippé, M. A. and Wold, E. H., "Antialiasing Through Stochastic Sampling," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 69-78, 1985.
- [41] Estevez, O., "On the Fundamental Data-base of Normal and Dichromatic Color Vision," *Ph.D. Thesis, University of Amsterdam*, Krips Repro Meppel, Amsterdam 1979.
- [42] Faugeras, O. D., "Digital Color Image Processing Within the Framework of a Human Visual Model," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. 27, No. 4, pp. 380-393, 1979.
- [43] Ferwerda, J. A., "Fundamentals of Spatial Vision," *Principles of Visual Perception and its Applications in Computer Graphics*, ACM SIGGRAPH, Course Notes, pp. 1-27, 1997.
- [44] Ferwerda, J. A., Pattanaik, S. N., Shirley, P., and Greenberg, D. P., "A Model of Visual Adaptation for Realistic Image Synthesis," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 249-258, 1996.
- [45] Ferwerda, J. A., Shirley, P., Pattanaik, S. N., and Greenberg, D. P., "A Model of Visual Masking for Computer Graphics," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 143-152, 1997.
- [46] Field, D. J., "Relations Between the Statistics of Natural Images and the Response Properties of Cortical Cells," *J. Opt. Soc. Am. A*, Vol. 4, pp. 2379-2394, 1987.

- [47] Field, D. J., "What the Statistics of Natural Images Tell Us About Visual Coding," *Human Vision, Visual Processing, and Digital Display*, Proc. SPIE, Vol. 1077, pp. 269-276, 1989.
- [48] Freeman, W. T. and Adelson, E. H., "The Design and Use of Steerable Filters," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 9, pp. 891-906, 1991.
- [49] Geisler, W. S. and Davila, K. D., "Ideal Discriminators in Spatial Vision: Two-Point Stimuli," *Journal of the Optical Society of America A*, Vol. 2, pp. 1483-1497, 1985.
- [50] Gergenfurther, K. R. and Kipper, D. C., "Contrast Detection in Luminance and Chromatic noise," *Journal of the Optical Society of America A*, Vol. 9, No. 11, pp. 1880-1888. 1992.
- [51] Gervais, M. J., Harvey, Jr., L. O., and Roberts, J. O., "Identification Confusions Among Letters of the Alphabet," *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 10, No. 5, pp. 655-666. 1984.
- [52] Gibson, S. and Hubbold, R. J., "Perceptually-Driven Radiosity," *Computer Graphics Forum*, Vol. 16, pp. 129-140, 1997.
- [53] Glassner, A. S., "Space Subdivision for Fast Ray Tracing," *IEEE Computer Graphics and Applications*, Vol. 4, No. 10, pp. 15-22, 1984.
- [54] Glassner, A. S., *An Introduction to Ray Tracing*, Academic Press: San Diego, 1989.
- [55] Glassner, A. S., Editor, *Advanced Topics in Ray Tracing*, ACM SIGGRAPH, Course Notes, 1990.
- [56] Glassner, A. S., *Principles of Digital Image Synthesis*, Morgan Kaufmann: San Francisco, 1995.
- [57] Goldstein, R. A. and Nagel, R., "Three-Dimensional Visual Simulation," *Visual Simulation*, Vol. 16, No. 1, pp. 25-31, 1971.
- [58] Gondek, J. S., Meyer, G. W., and Newman, J. G., "Wavelength Dependent Reflectance Functions," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 213-220, 1994.
- [59] Goral, C. M., Torrance, K. E., Greenberg, D. P., and Battaile, B., "Modeling the Interaction of Light Between Diffuse Surfaces," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 213-222, 1984.

- [60] Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F., "The Lumigraph," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 43-54, 1996.
- [61] Graham, C. H., Editor, *Vision and Visual Perception*, John Wiley & Sons, Inc.: New York, 1965.
- [62] Graham, N., Robson, J. G., and Nachmias, J., "Grating Summation in Fovea and Periphery," *Vision Research*, Vol. 18, pp. 815-825, 1978.
- [63] Granger, E. M. and Cupery, K. N., "An Optical Merit Function (SQF), which correlates with subjective image judgements," *Photographic Science and Engineering*, Vol. 16, No. 3, pp. 221-230, 1972.
- [64] Greene, N. and Kass, M., "Error-Bounded Antialiased Rendering of Complex Environments," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 59-66, 1994.
- [65] Guenter, B. K., Yun, H. C., and Mersereau, R. M., "Motion Compensated Compression of Computer Animation Frames," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 297-304, 1994.
- [66] Guth, S. L., Massof, R. W., and Benzscharow, T., "Vector Model for Normal and Dichromatic Color Vision," *J. Opt. Soc. Am.*, Vol. 70, pp. 197-211, 1980.
- [67] Hall, C. F. and Hall, E. L., "A Nonlinear Model for the Spatial Characteristics of the Human Visual System," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 7, No. 3, pp. 161-170, 1977.
- [68] Hamming, R. W., *Digital Filters*, Prentice-Hall: New Jersey, 1977.
- [69] Hashimoto, A., Akimoto, T., Mase, K., and Suenga, Y., "Vista Ray-Tracing: High Speed Ray Tracing Using Perspective Projection Image," *New Advances in Computer Graphics (Proc. CG International '89)*, Rae A. Earnshaw and Brian Wyvill, Editors, Springer-Verlag, New York pp. 549-561, 1989.
- [70] Heckbert, P., Editor, *Global Illumination*, ACM SIGGRAPH, Course Notes, 1993.
- [71] Hurvich, L. M. and Jameson, D., "Some Quantitative Aspects of an Opponent-Colors Theory, II. Brightness, Saturation, and Hue in Normal and Dichromatic Vision," *Journal of the Optical Society of America*, Vol. 45, pp. 602-616, 1955.
- [72] Ingling, C. R., "The Spectral Sensitivity of the Opponent-Colors Channels," *Vision Research*, Vol. 17 pp. 1083-1090, 1977.

- [73] Jameson, D. and Hurvich, L. M., "Some Quantitative Aspects of an Opponent-Colors Theory, I. Chromatic Responses and Spectral Saturation," *Journal of the Optical Society of America*, Vol. 45, pp. 546-552, 1955.
- [74] Jansen, F. W. and van Wijk, J. J., "Fast Previewing Techniques in Raster Graphics," *Proc. Eurographics '83*, P. J. W. ten Hagen, Editor, North-Holland, Amsterdam, pp. 195-202, 1983.
- [75] Kahn, H., "Use of Different Monte Carlo Sampling Techniques," *Symposium on Monte Carlo Methods*, Wiley: New York pp. 146-190, 1956.
- [76] Kajiya, J. R., "The Rendering Equation," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 143-150, 1986.
- [77] Karasaridis, A. and Simoncelli, E., "A Filter Design Technique for Steerable Pyramid Image Transforms," *In Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, 1996.
- [78] Kim, S. P. and Bose, N. K., "Reconstruction of 2-D bandlimited discrete signals from nonuniform samples," *IEEE Proceedings*, Vol. 137, Pt. F(3), pp. 197-204, 1990.
- [79] Kirk, D. and Arvo, J., "Unbiased Sampling Techniques for Image Synthesis," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 153-156, 1991.
- [80] Lee, M. E. and Redner, R. A., "A Note on the Use of Nonlinear Filtering in Computer Graphics," *IEEE Computer Graphics and Applications*, Vol. 10, No. 5, pp. 23-29, 1990.
- [81] Lee, M. E., Redner, R. A. and Uselton, S. P., "Statistically Optimized Sampling for Distributed Ray Tracing," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 61-67, 1985.
- [82] Legge, G. E. and Foley, J. M., "Contrast Masking in human vision," *Journal of the Optical Society of America*, Vol. 70, pp. 1458-1470, 1980.
- [83] Li, B., "An Analysis and Comparison of Two Visual Discrimination Models," *Master's Thesis, University of Oregon*, June 1997.
- [84] Li, B., Meyer, G. W., and Klassen, R. V., "A Comparison of Two Image Quality Models," to appear in *Human Vision and Electronic Imaging III*, B. E. Rogowitz and T. N. Pappas, Editors, Proc. SPIE, Vol. 3299, 1998.

- [85] Lischinski, D., Smits, B. and Greenberg, D. P., "Bounds and Error Estimates for Radiosity," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 67-74, 1994.
- [86] Lloyd, C. J. C. and Beaton, R. J., "Design of a Spatial-Chromatic Human Vision Model for Evaluating Full-color Display Systems," *Human Vision and Electronic Imaging: Models, Methods and Applications*, Proc. SPIE, Vol. 1249, pp. 23-37, 1990.
- [87] Losada, M. A. and Mullen, K. T., "The Spatial Tuning of Chromatic Mechanisms Identified by Simultaneous Masking," *Vision Research*, Vol. 34, No. 3, pp. 331-341. 1994.
- [88] Lubin, J., "A Visual Discrimination Model for Imaging System Design and Evaluation," *Vision Models for Target Detection and Recognition*, Eli Peli, Editor, World Scientific, New Jersey, pp. 245-283, 1995.
- [89] Maloney, L. T., "Evaluation of linear models of surface spectral reflectance with small numbers of parameters," *J. Opt. Soc. Am. A*, Vol. 3, pp. 1673-1683. 1986.
- [90] Mannos, J. L. and Sakrison, D. J., "The Effects of a Visual Fidelity Criterion on the Encoding of Images," *IEEE Transactions on Information Theory*, Vol. 20, No. 4, pp. 525-536, 1974.
- [91] Marimont, D. H. and Wandell, B. A., "Matching Color Images: The Impact of Axial Chromatic Aberration," *J. Opt. Soc. Am. A*, Vol. 12, pp. 3113-3122, 1993.
- [92] Martin, R. A., Ahumada Jr., A. J., and Larimer, J. O., "Color Matrix Display Simulation Based Upon Luminance and Chromatic Contrast Sensitivity of Early Vision," *Human Vision and Electronic Imaging III*, Proc. SPIE, Vol. 1666, pp. 336-342, 1992.
- [93] McCool, M. and Fiume, E., "Hierarchical Poisson Disk Sampling Distributions," *Proceedings of Graphics Interface 1992*, pp. 94-105, 1992.
- [94] Meyer, G. W., "Wavelength Selection for Synthetic Image Generation," *Computer Vision, Graphics, and Image Processing*, Vol. 41, pp. 57-79, 1988.
- [95] Meyer, G. W. and Liu, A., "Color Spatial Acuity Control of a Screen Subdivision Image Synthesis Algorithm," *Human Vision, Visual Processing, and Digital Display III*, Bernice E. Rogowitz, Editor, Proc. SPIE, Vol. 1666, pp. 387-399, 1992.

- [96] Meyer, G. W., Rushmeier, H. E., Cohen, M. F., Greenberg, D. P., and Torrance, K. E., "An experimental evaluation of computer graphics imagery," *ACM Transactions on Graphics*, Vol. 5, No. 1 pp. 30-50, 1986.
- [97] Mikhailov, G. A., *Monte Carlo Methods*, Springer-Verlag: Berlin Heidelberg 1992.
- [98] Mitchell, D. P., "Generating Antialiased Images at Low Sampling Densities," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 65-72, 1987.
- [99] Mitchell, D. P., "Spectrally Optimal Sampling for Distributed Ray Tracing," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 157-164, 1991.
- [100] Mitchell, D. P. and Netravali, A. N., "Reconstruction Filters in Computer Graphics," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 221-228, 1988.
- [101] Mullen, K. T., "The Contrast Sensitivity of Human Colour Vision to Red-Green and Blue-Yellow Chromatic Gratings," *J. Physiol. (Lond.)*, Vol. 359, pp. 381-400, 1985.
- [102] Myszkowski, K., "The Visible Differences Predictor: Applications to Global Illumination Problems," to appear in *Rendering Techniques '98*, G. Drettakis and N. Max, Editors, Springer-Verlag, Wien, 1998.
- [103] Neumann, L. and Neumann, A., "Radiosity and Hybrid Methods," *ACM Transactions on Graphics*, Vol. 14, No. 3, pp. 233-265, 1995.
- [104] Nielson, K. R. K., Watson, A. B., and Ahumada Jr., A. J., "Application of a Computable Model of Human Spatial Vision to Phase Discrimination," *Journal of the Optical Society of America A*, Vol. 2, No. 9, pp. 1600-1606, 1985.
- [105] Nishita, T. and Nakamae, E., "Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 23-30, 1985.
- [106] Oppenheim, A. V., Willsky, A. S., and Young, I. T., *Signals and Systems*, Prentice-Hall: New Jersey, 1983.
- [107] Painter, J. and Sloan, K., "Antialiased Ray Tracing by Adaptive Progressive Refinement," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 281-288, 1989.

- [108] Pattanaik, S. N. and Mudur, S. P., "Computation of Global Illumination by Monte Carlo Simulation of the Particle Model of Light," *Third Eurographics Workshop on Rendering*, pp. 71-83, 1992.
- [109] Pattanaik, S. N. and Mudur, S. P., "The Potential Equation and Importance in Illumination Computations," *Computer Graphics Forum*, Vol. 12, No. 2, pp. 131-136, 1993.
- [110] Pattanaik, S. N. and Mudur, S. P., "Adjoint Equations and Random Walks for Illumination Computation," *ACM Transactions on Graphics*, Vol. 14, No. 1, pp. 77-102, 1995.
- [111] Pattanaik, S. N., Ferwerda, J. A., Fairchild, M. D., and Greenberg, D. P., "A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display," *Computer Graphics, Annual Conference Series, ACM SIGGRAPH*, pp. 287-298, 1998.
- [112] Pearson, D. E., *Transmission and Display of Pictorial Information*, John Wiley and Sons, 1975.
- [113] Pelli, D. G., "Uncertainty Explains Many Aspects of Visual Contrast Detection and Discrimination," *Journal of the Optical Society of America A*, Vol. 2, No. 9, pp. 1508-1511, 1985.
- [114] Phillips, G. C. and Wilson, H. R., "Orientation Bandwidths of Spatial Mechanisms Measured by Masking," *Journal of the Optical Society of America A*, Vol. 1, No. 2, pp. 226-232, 1984.
- [115] Poirson, A. B. and Wandell, B. A., "Pattern-Color Separable Pathways Predict Sensitivity to Simple Colored Patterns," *Vision Research*, Vol. 35, No. 2, pp. 239-254, 1996.
- [116] Powell, M. J. D., "A Theorem on Rank One Modifications to a Matrix and its Inverse," *Computer Journal*, Vol. 12, pp. 288-290, 1969.
- [117] Pratt, W. K., *Digital Image Processing*, Second Edition, John Wiley and Sons, 1991.
- [118] Purgathofer, W., "A Statistical Method for Adaptive Stochastic Sampling," *Proc. Eurographics '86*, Aristides A. G. Requicha, Editor, North-Holland, Amsterdam, pp. 145-152, 1986.
- [119] Quick, R. F., "A Vector-Magnitude model for Contrast Detection," *Kybernetik*, Vol. 16, pp. 65-67, 1974.

- [120] Rao, K. R. and Yip, P., *Discrete Cosine Transform*, Academic Press: Boston, 1990.
- [121] Redner, R. A., Lee, M. E. and Uselton, S. P., "Smooth B-Spline Illumination Maps for Bidirectional Ray Tracing," *ACM Transactions on Graphics*, Vol. 14, No. 4, pp. 337-362, 1995.
- [122] Roth, S. D., "Ray Casting for Modeling Solids," *Computer Graphics and Image Processing*, Vol. 18, No. 2, pp. 109-144, 1982.
- [123] Ruderman, D. L., "Origins of Scaling in Natural Images," *Human Vision, Visual Processing, and Digital Display*, Proc. SPIE, Vol. 2657, pp. 120-131, 1996.
- [124] Rushmeier, H. E. and Ward, G. J., "Energy Preserving Non-Linear Filters," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 131-138, 1994.
- [125] Rushmeier, H., Ward, G., Piatko, C., Sanders, P., and Rust, B., "Comparing Real and Synthetic Images: Some Ideas About Metrics," *Rendering Techniques '95*, P. M. Hanrahan and W. Purgathofer, Editors Springer-Verlag, New York, pp. 82-91, 1995.
- [126] Sauer, K. D. and Allebach, J. P., "Iterative Reconstruction of Band-Limited Images from Nonuniformly Spaced Samples," *IEEE Transactions on Circuits and Systems*, Vol. 34, No. 12, pp. 1497-1506, 1987.
- [127] Schade Sr., O. H., "Optical and Photoelectric Analog of the Eye," *Journal of the Optical Society of America*, Vol. 46, No. 9, pp. 721-739, 1957.
- [128] Schreiber, W. F., *Fundamentals of Electronic Imaging Systems*, Springer-Verlag: Berlin Heidelberg, 1993.
- [129] Shannon, C. E., "Communication in the Presence of Noise," *Proc. IRE*, Vol. 37, pp. 10-21 1949.
- [130] Shirley, P., "Discrepancy as a Quality Measure for Sample Distributions," *Proc. Eurographics '91*, Frits H. Post and Wilhelm Barth, Editors, North-Holland, Amsterdam, pp. 183-194, 1991.
- [131] Shirley, P. and Sakas, G., "Results of the 1994 Survey on Image Synthesis," *Photorealistic Rendering Techniques*, Springer-Verlag, New York pp. 3-6, 1995.

- [132] Shirley, P., Wang, C. and Zimmerman, K., "Monte Carlo Techniques for Direct Lighting Calculations," *ACM Transactions on Graphics*, Vol. 15, No. 1, pp. 1-36, 1996.
- [133] Shu, R. and Liu, A., "A Fast Ray Casting Algorithm Using Adaptive Isotriangular Subdivision," *Proc. Visualization '91*, Gregory M. Nielson and Larry Rosenblum, Editors IEEE Computer Society, Los Alamitos, CA, 1991.
- [134] Spencer, G., Shirley, P., Zimmerman, K., and Greenberg, D. P., "Physically-Based Glare Effects for Digital Images," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 325-334, 1995.
- [135] Stark, P. A., *Introduction to Numerical Methods*, Macmillan Publishing Co.: New York, 1970.
- [136] Stevens, J. C. and Stevens, S. S., "Brightness Function: Effects of Adaptation," *Journal of the Optical Society of America*, Vol. 53, No. 3, pp. 375-385, 1963.
- [137] Stevens, S. S., "To Honor Fechner and Repeal His Law," *Science*, Vol. 133, No. 13, pp. 80-86, 1961.
- [138] Stockham, T. G., "Image Processing in the Context of a Visual Model," *Proceedings of the IEEE*, Vol. 60, No. 7, pp. 828-842, 1972.
- [139] Tamstorf, R. and Jensen, H. W., "Adaptive Sampling and Bias Estimation in Path Tracing," *Rendering Techniques '97*, J. Dorsey and P. Slusallek, Editors, Springer-Verlag, New York, pp. 285-295, 1997.
- [140] Thomas, D., Netravali, A. N., and Fox, D. S., "Anti-Aliased Ray Tracing with Covers," *Computer Graphics Forum*, Vol. 8, No. 4, pp. 325-336, 1989.
- [141] Tumblin J. and Rushmeier, H., "Tone Reproduction for Realistic Images," *IEEE Computer Graphics and Applications*, pp. 42-48, 1993.
- [142] Tumblin J., Hodgins, J., and Guenter, B., "Display of High Contrast Images Using Models of Visual Adaptation," *Visual Proceedings, Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 154, 1997.
- [143] van Walsum, T., van Nieuwenhuizen, P. R., and Jansen, F. W., "Refinement Criteria for Adaptive Stochastic Ray Tracing of Textures," *Proc. Eurographics '91*, Frits H. Post and Wilhelm Barth, Editors, North-Holland, Amsterdam, pp. 155-166, 1991.

- [144] Veach, E. and Guibas, L. J., "Optimally Combining Sampling Techniques for Monte Carlo Rendering," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 419-428, 1995.
- [145] Wallace, G. K., "The JPEG Still Picture Compression Standard," *Communications of the ACM*, Vol. 34, No. 4, pp. 30-44, 1991.
- [146] Wallach, D. S., Kunapalli, S., and Cohen, M. F., "Accelerated MPEG Compression of Dynamic Polygonal Scenes," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 193-196, 1994.
- [147] Wandell, B. A., *Foundations of Vision*, Sinauer Associates, Inc.: Sunderland, MA, 1990.
- [148] Ward, G., "A Contrast-based Scalefactor for Luminance Display," *Graphics Gems IV*, P. S. Heckbert, Editor, Academic Press Professional: Boston, pp. 415-421, 1994.
- [149] Ward, G. J., Rubinstein, F. M. and Clear, R. D., "A Ray Tracing Solution for Diffuse Interreflection," *Computer Graphics, Annual Conference Series*, ACM SIGGRAPH, pp. 85-92, 1988.
- [150] Ward Larson, G., Rushmeier, H., and Piatko, C., "A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3, No. 4, pp. 291-306, 1997.
- [151] Watson, A. B., "Detection and Recognition of Simple Spatial Forms," *Physical and Biological Processing of Images*, O. J. Braddick and A. C. Sleight, Editors, Springer-Verlag: New York, pp. 100-114, 1983.
- [152] Watson, A. B., "Efficiency of a Model Human Image Code," *Journal of the Optical Society of America A*, Vol. 4, No. 12, pp. 2401-2417, 1987.
- [153] Watson, A. B., "The Cortex Transform: Rapid Computation of Simulated Neural Images," *Computer Vision, Graphics, and Image Processing*, Vol. 39, pp. 311-327, 1987.
- [154] Watson, A. B., Solomon, J. A., Ahumada, A., and Gale, A., "DCT Basis Function Visibility: Effects of Viewing Distance and Contrast Masking," *Human Vision, Visual Processing, and Digital Display IV*, Proc. SPIE, Vol. 2179, pp. 99-108, 1994.
- [155] Watt, A. and Watt, M., *Advanced Animation and Rendering Techniques*, Addison-Wesley: Wokingham, 1992.

- [156] Whitted, T., "An Improved Illumination Model for Shaded Display," *Communications of the ACM*, Vol. 23, No. 6, pp. 343-349, 1980.
- [157] Williams, D. R., MacLeod, D. I. A., and Hayhoe, M. M., "Punctuate Sensitivity of the Blue-Sensitive Mechanism," *Vision Research*, Vol. 21, pp. 1357-1375, 1981.
- [158] Wilson, H. R., "Psychophysical Models of Spatial Vision and Hyperacuity," *Vision and Visual Dysfunction: Vol. 10: Spatial Vision*, D. Regan, Editor, CRC Press Inc., Boston, MA, pp. 64-86, 1991.
- [159] Wilson, H. R. and Bergen, J. R., "A Four Mechanism Model for Threshold Spatial Vision," *Vision Research*, Vol. 19, pp. 19-32, 1979.
- [160] Wilson, H. R. and Gelb, D. J., "Modified Line-Element Theory for Spatial-Frequency and Width Discrimination," *Journal of the Optical Society of America A*, Vol. 1, No. 1, pp. 124-131, 1984.
- [161] Wilson, H. R. and Regan, D., "Spatial-Frequency Adaptation and Grating Discrimination: Predictions of a Line-element Model," *Journal of the Optical Society of America A*, Vol. 1, No. 11, pp. 1091-1096, 1984.
- [162] Wilson, H. R., McFarlane, D. K., and Phillips, G. C., "Spatial Frequency Tuning of Orientation Selective Units Estimated by Oblique Masking," *Vision Research*, Vol. 23, No. 9, pp. 873-882, 1983.
- [163] Wyvill, G. and Sharp, P., "Fast Antialiasing of Ray Traced Images," *New Trends in Computer Graphics (Proc. CG International '88)*, Tosiyasu Kunii, Editor, Springer-Verlag, New York, pp. 579-587, 1988.
- [164] Zetsche, C. and Hauske, G., "Multiple Channel Model for the Prediction of Subjective Image Quality," *Human Vision, Visual Processing, and Digital Display*, Proc. SPIE, Vol. 1077, pp. 209-216, 1989.
- [165] Zhang, X. M. and Wandell, B. A., "A Spatial Extension of CIELAB for Digital Color Image Reproduction," *Society for Information Display Symposium Technical Digest*, Vol. 27, pp. 731-734, 1996.
- [166] Zhang, X. M., Farrell, J. E., and Wandell, B. A., "Applications of a Spatial Extension to CIELAB," *Very High Resolution and Quality Imaging II*, V. R. Algazi, S. Ono, and A. G. Tescher, Editors, Proc. SPIE, Vol. 3025, pp. 154-157, 1997.
- [167] Zhang, X. M., Setiawan, E. and Wandell, B. A., "Image Distortion Maps," *Final Program and Proceedings of the Fifth IS&T Color Imaging Conference. Color Science, Systems and Applications*, pp. 120-125, 1997.

- [168] Zhang, X. M., Silverstein, D. A., Farrell, J. E., and Wandell, B. A., "Color Image Quality Metric S-CIELAB and Its Application on Halftone Texture Visibility," *IEEE Compton97 Digest of Papers*, pp. 44-48, 1997.