Tone Mapping Techniques and Color Image Difference in Global Illumination

Krešimir Matković

February 18, 1998

Dissertation

Tone Mapping Techniques and Color Image Difference in Global Illumination

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften unter der Leitung von

Prof. Dr. Werner Purgathofer
E186
Institut für Computergraphik

eingereicht an der Technischen Universität Wien
Technisch-Naturwissenschaftliche Fakultät

von

Dipl.-Ing. Krešimir Matković
Matr.-Nr. 9426659
Rötzergasse 20/5, A-1170 Wien

Wien, im December 1997

Kurzfassung

Tone Mapping (Farbtransformation) ist der letzte Schritt eines jeden photorealistischen Bilderzeugungsverfahrens. Aufgrund der Nichtlinearitäten und der vorhandenen Einschränkungen des Farbraums und des dynamischen Verhaltens des zur Darstellung verwendeten Geräts ist es nötig, eine Farbtransformation auf die berechneten Farbwerte anzuwenden.

Wir beschreiben den Stand der Forschung im Bereich Transformationsmethoden und einige neue Methoden. Der Hauptbeitrag dieser Arbeit liegt in der interaktiven Abgleichung von Kontrast und Blende sowie in Methoden mit minimalem Informationsverlust und Messung des einfallenden Lichts.

Die interaktive Abgleichung ermöglicht die Darstellung der Szene mit einer gewünschten Beleuchtungs-Stimmung, selbst wenn die Beleuchtungswerte in fiktiven Einheiten berechnet wurden.

Die Methoden mit minimalem Informationsverlust basieren in gewisser Weise auf dem Ansatz des Photographen. Die Farbtransformation wird nur auf ein bestimmtes Farbintervall angewandt, welches automatisch gewählt wird. Der ursprüngliche Kontrast aller Pixel in diesem Intervall bleibt erhalten. Darüberhinaus ist die auf Fehlerbeschrnkungen basierende Methode eine Erweiterung von Schlicks Verfahren.

Die Methode zur Messung des einfallenden Lichts basiert ebenfalls auf einer in der Photographie üblichen Vorgangsweise. Diese Methode ermöglicht die genaue Reproduktion von Farben. Selbst wenn die durchschnittliche Reflexion einer Szene sehr klein oder groß ist, wird diese Methode die ursprünglichen Farben reproduzieren können, eine Eigenschaft, die konkurrierenden Methoden fehlt. Die Grundidee ist, die einfallende Beleuchtung durch simulierte Lichtmessung in der Szene zu messen und die daraus resultierende Skalierung auf die berechneten Farbwerte anzuwenden.

Neben diesen eigenen Beiträgen werden andere relevante Ansätze besprochen. Wir beschreiben die Transformation von Tumblin und Rushmeier, die Kontrastbasierte Methode von Ward, das weitverbreitete Verfahren mit Durchschnittsbildung, den exponentiellen Ansatz von Ferschin et al., Schlicks Abbildung, ein auf Sichtbarkeit beruhendes Verfahren zur Farbtransformation von Larson et al. und einen visuelle Adaption berücksichtigenden Ansatz von Ferwerda et al.

Leider gibt es keine letztgültige Lösung zur Farbtransformation. Jede Methode hat Stärken und Schwächen, und der Benutzer sollte die geignete Methode wählen können.

Die Arbeit endet mit der Präsentation eines Algorithmus zur Berechnung der Farbbilddifferenz. Eine gute Metrik zur Bewertung des Farbabstandes zweier Bilder wird in der Computergraphik oft benötigt, ist aber nicht leicht zu konstruieren. Die in dieser Arbeit beschriebene Metrik beruht auf der men-

schlichen Wahrnehmung und arbeitet im Bildbereich. Eine Fourier- oder Wavelet-Transformation ist daher nicht nötig, was das Verfahren schnell und intuitiv macht. Diese Methode ist die einzige, die explizit den Abstand des Beobachters zum Bild in Betracht zieht.

Abstract

Tone mapping is the final step of every rendering process. Due to display devices' nonlinearities, reduced color gamuts and moderate dynamic ranges it is necessary to apply some mapping technique on the computed radiances.

We described mapping methods that are considered to be state of the art today, and some newly developed techniques. The main contributions of this thesis in tone mapping techniques are interactive calibration of contrast and aperture, minimum information loss methods and incident light metering.

The interactive calibration technique makes it possible to display a desired scene lighting atmosphere if the radiance values are rendered in fictitious units.

Minimum information loss techniques are based, in a way, on the photographers' approach. The mapping function is applied only on a certain radiance interval, which is chosen automatically. The original contrast of all pixels inside the interval is preserved. Furthermore, the bounded error version of the minimum loss method is an extension of Schlick's method.

The incident light metering method was inspired by the photographers' approach, too. This method makes it possible to reproduce original colors faithfully. Even if the average reflectance of a scene is very low, or very high, this method will reproduce original colors, which is not the case with other methods. The idea is to measure the incident light using diffusors in the scene, and then to compute a scale factor based on the incident light and apply this scale factor on the computed radiances.

Beside these, other tone mapping techniques are described in this work. We described Tumblin and Rushmeier's mapping, Ward's contrast based scale factor, the widely used mean value mapping, an exponential mapping introduced by Ferschin et al., Schlick's mapping, a visibility matching tone operator introduced by Larson et al., and a model of visual adaptation proposed by Ferwerda et al.

Unfortunately there is no ultimative solution to the tone mapping problem. Every method has its strengths and weaknesses, and the user should choose a method according to his or her needs.

Finally, this thesis ends with a color image difference algorithm. A good image metric is often needed in computer graphics. The method described here is a perception based metric that operates in the original image space (there is no need for Fourier or wavelet transform), what makes the whole method fast and intuitive. This is the only method that stresses distance dependency explicitly.

Acknowledgements

This thesis would have never been written without the help from the following people:

Very, very special thanks go to Dr. László Neumann for sharing his ideas with me, and helping me in all ways. This work is the result of our cooperation in the last three years, which was a wonderful time for me. I also thank Attila Neumann for all his remarks and algorithms. This thesis is a result of numerous discussions in Budapest and Vienna, and it will remind us all on this time.

Very special thanks go to Prof. Werner Purgathofer who let me do my dissertation studies at his Institute. He always had time for me, and he has spent a lot of hours discussing my ideas and correcting my written work. I also thank Prof. Wilhelm Barth for acting as second reviewer of the thesis.

Special thanks go to my wife and daughter whom I missed very often in these last three years.

Special thanks go to my parents and sister for their material and moral support.

Finally, I would like to thank all people at the Institute of Computer Graphics, and especially Jan Přikryl for his friendly advises and help concerning LaTeX, Dieter Schmalstieg for helping me with the translation of the abstract, and Robert F. Tobler and Alexander Wilkie for rendering float images for me.

Contents

Chapter 1

Introduction

Computer graphics is one of the newest visual media. It has become established as an important tool in design, entertainment, advertisements, fine arts, and many other applications, where images are needed. One of the many fields of computer graphics is image synthesis, often called rendering. Photorealistic rendering turns the rules of geometry and physics into pictures that could hardly be distinguished from photographs. Local illumination methods can render images ignoring the impact of objects on the illumination of other objects in the scenes. Therefore, if local illumination methods are used, shadows, penumbras, specular reflections and refractions, diffuse interreflections, etc., cannot be taken into account. On the other hand, global illumination models, ray tracing and radiosity (the two most popular), try to model light in an environment. Of course, such methods take much more time (local illumination methods are implemented in modern graphics hardware), but, as stated before, the results can not be distinguished from photographs.

Every rendering process consists of two steps. The first is the computing of luminance values, and the second is the mapping of the computed values to the values appropriate for displaying on common display devices. There is a lot of research dealing with the first step, but the second step is surprisingly often neglected, although it is far from trivial. Actually there are just a few authors dealing with this problem, in contrast to hundreds of researchers who are improving the first rendering step. Our work is primarily concerned with the final step of the rendering process. It is assumed that image is rendered, and floating point values of pixels' color components are known. We will not deal with methods that are used to compute these values. The floating point image will be called the "raw image".

In the ideal case the raw image should be mapped to the display device so that the displayed image creates the same sensation in the viewer as would have been experienced by seeing the real environment. Unfortunately, there are many

obstacles to realizing this ideal. These include the display devices' nonlinearities, limited color gamut, limited display device contrast, changing lighting conditions in the viewing environment, human vision rules, the basic limitations of representing 3D reality as a 2D projection, etc. Some of these obstacles will be explained later.

Various mapping methods are described in this work. Some methods take into account the above mentioned problems, or at least some of them, while other, more simple methods, do not. Some familiarity with color science, radiometry and photometry is necessary to understand this work, therefore chapter 2 deals with color science basics, radiometry and photometry and some aspects of human vision are described in chapter 2 as well.

Chapter 3 describes various display devices. Actually CRT as the most used display device in computer graphics, slides, and some printers are described. Data measured by the authors are also given in this chapter.

In chapter 4 linear scale factor methods are introduced. Probably the most widely used mapping is the use of a single scale factor which maps the average luminance to 0.5 input value, assuming that the display device's input range is [0,1], and that the device has linear response. Unfortunately, such scale factor can not reproduce the original atmosphere of the scene. Actually it will display the scene lit by a very weak light source, and the same scene lit by a very strong light source as being the same image, because of linearity of the integral operator in the rendering equation [ArKi90].

An interactive mapping technique introduced by Matković and Neumann in [MaNe96] makes it possible to display images with the proper atmosphere if this is known. The method uses two parameters called contrast and aperture, and maps the raw image according to subjective user settings. The interactive calibration mapping method is one of the contributions of this thesis. At the end of the fourth chapter a contrast based scale factor proposed by Greg Ward [Ward94] s described. Ward's mapping makes differences just visible in the real world become just visible in the displayed image. If the visibility analysis is crucial (e.g. the design of emergency lighting) this could be the right mapping method. Improvements of this method are introduced by Ferwerda et al. [FPSG96] and Ward et al. [LaRP97] and they are described in the next chapter.

In chapter 5 non-linear scale factors are introduced. A mapping technique proposed by Schlick [Schl94] is described first. Schlick's method is actually a computational improvement of the logarithmic mapping based on Weber's law. This is an automatic method that yields good results if the overall raw image contrast is not too high. Further, a non-linear mapping technique as suggested by Ferschin et. al. in [FeTP94] is described. Ferschin et al. introduced a method which suppresses the influence of a few very bright pixels influencing the average too much. If luminances in the raw image are computed in absolute units,

the appropriate atmosphere based on preserving the original brightness, could be reproduced using Tumblin and Rushmeier's mapping technique. This method is introduced in [TuRu93] and [TuRu91], and this is still one of the most comprehensive solutions of the raw image mapping problem. Unfortunately it is solved only for gray scale pictures. The method is described in section 5.3, and section 5.4 describes a model of visual adaptation introduced by Ferwerda et al. [FPSG96]. The model of visual adaptation is based on the Ward's model. Here the rules of human adaptation are taken into account. Even temporal effects well known from real life (e.g. the inability to see when entering a cinema until the eyes have adapted) can be simulated in computer graphics using this mapping method. Chapter 5 finishes with an overview of the visibility matching tone operator [LaRP97]. This is a further improvement of Ward's original operator.

Chapters 6 and 7 describe the main contribution of this thesis. Methods described in this chapters were introduced together with László Neumann [NeMP96], [NMNP97]. The whole family of methods called minimum information loss mapping is described in chapter 6. The main idea is to find the clipping interval so that minimum amount of information is lost, thereby preserving the original contrast of all correctly displayed pixels. Two variants are described, in the first the color component is assumed to be essential information, and in the second the pixel is treated as essential information. The second variant is called the minimum area loss. The method works especially well in back light scenes, which are often displayed as too dark if average value mapping is used. The methods are not conditioned by knowledge of absolute units. Another possibility is to limit allowed information loss, and find the smallest contrast interval which still satisfies limited error condition. Of course, in this case the original contrast is not always preserved.

Chapter 7 describes incident light metering in computer graphics. Incident light metering is a well known method in professional photography and the movie industry. In fact, it was used at the beginning of the photography era by portrait photographers. Although it is not practical for amateur photographers (light should be measured at the subject position, not at the camera), it can be implemented in computer graphics. It overcomes the problem of average mapping, where a very bright scene (e.g. a snow covered mountain) and a very dark one (e.g. a heap of coal) are displayed as medium gray (or close to it), which makes the bright scene too dark and the dark scene too bright. When incident light metering is used, raw images are mapped correctly, and the absolute units should not be known. We recommend using this method when absolute units are not known (which is most often the case due to difficulties in getting appropriate data for light sources and materials) and the scene settings are not usual (e.g. very bright, or very dark scenes, scenes with back light etc.). Note that bright or dark scene here does not mean well or poorly lit scene, but rather the scene with low

or high object's reflectances. Actually, this is the only method that will reproduce selected colors even for the scenes with very low or very high average reflectance. The tone mapping part of this thesis ends with chapter 7.

The next chapter, perception based color image difference, presents a new algorithm, developed together with László Neumann, for computing the difference between two images. A good image metric is often needed in computer graphics. All progrssive rendering methods should check convergance somehow, lossy compression algorithms should be evaluated, sometimes the resulting images of various rendering or tone mapping techiques are compared, etc. The most often used metric in computer graphics is the mean squared error. Unfortunatelly it does not corespond to human perception, and sometimes images that look similar can have larger difference than obviously different images, when the mean square metric is used.

Two recent papers by Rushmeier et al. [RWPSR95] and Gaddipati et al. [GaMY97], deal with perception based image metrics. They compute the image distance either in Fourier or in the wavelet space, which makes them computationally expensive and not intuitive. The color is not handled completely correctly in these two approaches.

We introduce a new method that operates in the original space and handles the color more accurately.

This thesis ends with results and conclusion chapters.

Chapter 2

Color Science Basics

Since a familiarity with radiometry, photometry, color science and human vision is necessary to understand this work, this chapter will describe some color science and human vision basics. It would be impossible to cover the whole area of color science or human vision in a single chapter. Many books have been written on the above subjects [WySt82], [Boyn92], [Hunt92], and they are still not completely understood. Therefore, all we want here is to give a brief overview that will help to follow this work successfully.

The way we see objects around us depends on three factors. The first one is light. It is clear that we can not see if there is no light. Most of us have experienced, also, different perception under different lighting conditions (remember ladies checking the clothes colors in front of the shop in the daylight). The second one is the object characteristics itself. Some objects are red, some are blue, etc. The third subject involved in color vision is the human observer. It is impossible to describe the color sensation in our mind. Actually, it is impossible to describe any sensation. All we can say is that some color looks like some other, but this is actually not a description of color. Nevertheless, there is the way of measuring colors, and light, and these methods will be described next. We will start with light measuring (radiometry and photometry), then proceed with colorimetry and finally describe some human vision characteristics that are interesting for tone mapping techniques.

2.1 Photometry and Radiometry

From our experience with black and white photography, we know that we can see and recognize an object without knowing its hue. On the other hand we know that color photos conwey much more additional information. In order to understand color perception we have to be concerned with spatial as well as with chromatic

vision. Since there will be no vision without light let's describe light itself, first.

2.1.1 Light

Humans have wondered throughout history how we are able, through the sense of vision, to discern the nature and color of objects far removed from our bodies - objects with which we are obviously not in contact. The ancient Greek philosophers correctly reasoned that something must pass between our eyes and the objects we see. Plato developed an emanation theory of vision, which says that an inner fire gives rise to visual rays shooting outward from the eye. Other philosophers either accepted the emanation theory or used some other approach to explain human vision.

During the middle ages, Arab natural philosopher Alhazen rejected the emanation theory of vision. He was convinced that an optical image similar to the one produced by a pin-hole camera is produced in the eye.

During the Italian Renaissance, the great scientist-painter Leonardo da Vinci developed perspective drawing, and speculated about human vision. He was convinced that there is some kind of image inside the eye. As the laws of light refraction and the nature of light were unknown then, da Vinci could not develop his theory successfully.

The seventeenth century marks the start of the modern era for the study of light and vision. Spectacle lenses had been discovered by 1285, and positive lenses have been used to improve the performance of the pin-hole camera. Kepler understood how the lenses worked in the telescope he was using. He correctly believed that there is also some kind of retinal image in our eye, but the vision in its totalty was still not clear to him.

Throughout the period of history described so far, nothing was known about the physical nature of light. Isaac Newton was the first who realized that white light is composed of the whole spectrum.

Nowadays, it is well known that light is electromagnetic radiation. The electromagnetic spectrum extends from very low frequency radio waves, through microwaves, infrared, visible and ultraviolet light to x-rays and gamma rays. Our eyes respond to the visible light. If we want to detect the rest of the electromagnetic spectrum, special instruments ranging from radio receivers to scintillation counters are required. An exact description of electromagnetic radiation requires a thorough knowledge of quantum electrodynamics and Maxwell's electromagnetic field equations which is beyond the scope of this work. The visible spectrum is considered to have the wavelengths between 380 and 770 nm. It means if electromagnetic radiation of such wavelength hits our eye, we will see it. The perceived color depends on the wavelength of the radiation. We will deal with measuring light independently from the wavelength first.

2.1.2 Radiometry

Radiometry is the science of measuring light in any portion of the spectrum. Therefore, the color is not important to the radiometry.

Light is radiant energy. Electromagnetic radiation transports energy through space. A broadband source, like the Sun, emits the energy throughout most of the spectrum, while, on the other hand, single-wavelength laser emits radiation only at one specific wavelength.

We can define spectral radiant energy, which is the amount of radiant energy per unit wavelength interval at wavelength λ, as:

$$Q_\lambda = dQ/d\lambda \tag{2.1}$$

Radiant flux is then defined as:

$$\Phi = dQ/dt \tag{2.2}$$

where Q is radiant energy, and t is time. Spectral radiant flux is defined as $\Phi_\lambda = d\Phi/d\lambda$. Radiant flux density is the radiant flux per unit area at a point on the surface. There are two possibilities. The flux can be arriving at the surface (radiant flux density is then called irradiance):

$$E = d\Phi/dA \tag{2.3}$$

And the flux can be leaving the surface (radiant flux density is then referred to as radiant exitance):

$$M = d\Phi/dA \tag{2.4}$$

There are also spectral forms of radiant flux densities, E_λ and M_λ.

If we think of a ray of light arriving at or leaving a point on a surface in a given direction, then radiance is simply an infinitesimal amount of radiant flux contained in this ray. Actually the ray should be an infinitesimally narrow cone with its apex at a point on a surface. The cone has a differential solid angle $d\omega$ that is measured in steradians. Of course, a ray intersecs the surface at angle θ. Therefore a projected area $dAcos\theta$ instead of the area dA should be used. The definition of radiance is then:

$$L = \frac{d^2\Phi}{dA(d\omega cos\theta)} \tag{2.5}$$

Unlike radiant flux density, the definition of radiance does not distinguish between flux arriving at or leaving the surface. Spectral radiance, as radiance per unit wavelength interval at wavelength λ is also defined.

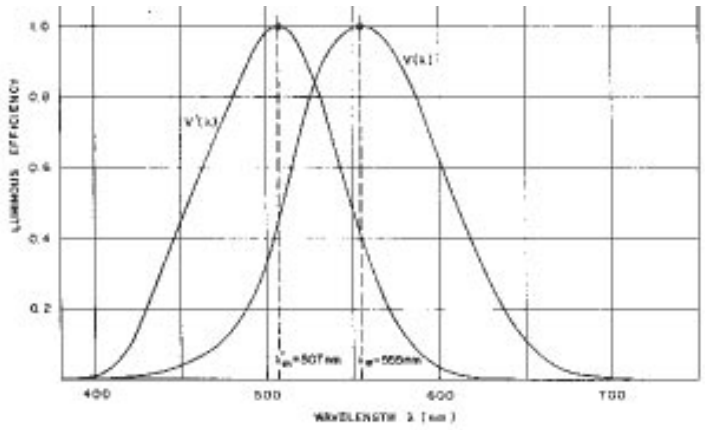


Figure 2.1: CIE photometric curve for photopic, V(λ), and scotopic V'(λ) [WySt82]

2.1.3 Photometry

Photometry measures visible light in units that are weighted according to the sensitivity of the human eye. Our eye is a complex, nonlinear, detector of electromagnetic radiation with wavelengths between 380 and 770 nm. The sensitivity of the human eye varies with the wavelength. Figure 2.1 shows the CIE photometric curve (CIE stands for *Commission Internationale d'Eclairage* - International Commission on Illumination). This curve tells us that a light source of strength 1 $W/m^2 steradian$ will appear brighter if it emits light of wavelength 550 nm, than the same strength light source that emits light of 440 nm wavelength. Actually, all that photometry does is the weighting of radiometric units using the CIE photometric curve. The only difference between radiometry and photometry is in their units of measurement. All radiometric units have their photometric counterparts. We will mention only luminance as the counterpart of radiance. In fact it is just photometrically weighted radiance. Note that digital image synthesis simulates the light in an enviroment, and as a result radiances (or luminances) of particular wavelengths are computed. This radiances are stored in the raw-image, which is then mapped to the display device, using one of the mapping techiques.

Up to now color nas not been taken into account. The next section, colorimetry, will give us a brief overview of color.

2.2 Colorimetry

Colorimetry is the science of measuring colors. Although each of us can perceive colors slightly differently, the CIE has defined a standard observer. A set of standard conditions for performing color measuring experiments has also been proposed by CIE. A number of color matching experiments have been performed under these standardized conditions. Color matching experiments consists of choosing three particular light sources, that emit light on the white screen, where three projections overlap and form an additive mixture. On the other side of the screen a target color is projected, and an observer tries to match the target light by altering the intensities of the three light sources. The weights of light sources are in the range $[-1, 1]$. Negative weights are allowed, as it is not possible to match all colors using only positive weights. A negative weight does not mean subtracting color from the additive mixture, but rather adding this color to the target color. After many experiments using light sources of the wavelengths red=700 nm, green=546.1 nm and blue is 435.8 nm [WySt82] color matching curves as shown in figure 2.2 were proposed by CIE.

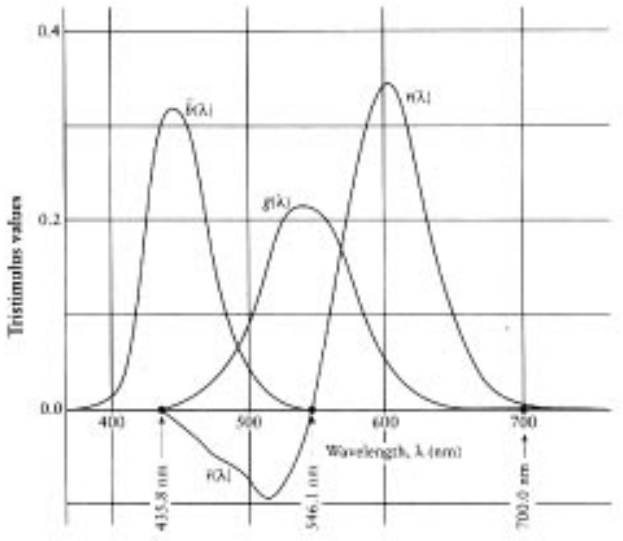


Figure 2.2: The r, g, and b color-matching curves [WySt82]

As it was inconvenient to have negative values in the matching functions CIE proposed a linear transformation of matching functions resulting in CIE $\overline{x}$, $\overline{y}$ and $\overline{z}$ matching functions, as shown in figure 2.3. Note that there are no negative values in these matching functions.

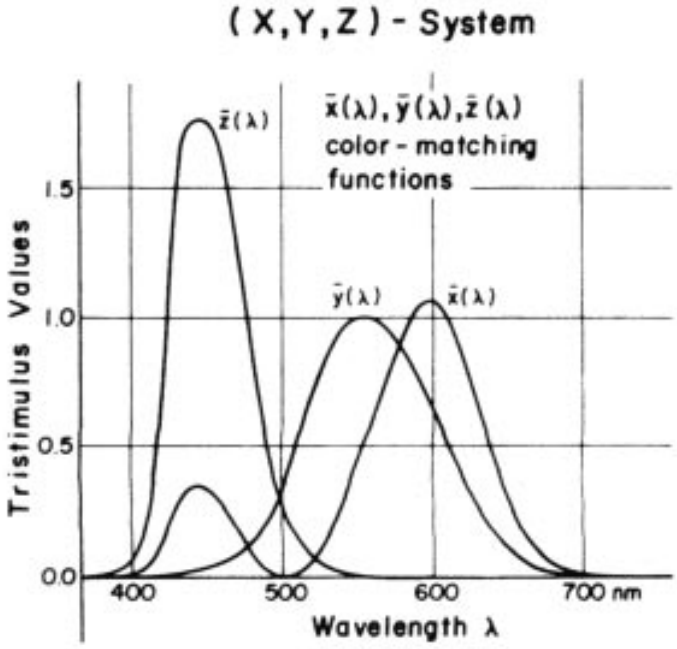


Figure 2.3: The $\overline{x}$, $\overline{y}$, and $\overline{z}$ color-matching curves [WySt82]

Now if the surface reflectance, and the light source distribution are known, their product defines color as $C(\lambda)$, and the weights XYZ can be found using the following equations:

$$X = \int_{380}^{780} C(\lambda)\overline{x}(\lambda)d\lambda \tag{2.6}$$

$$Y = \int_{380}^{780} C(\lambda)\overline{y}(\lambda)d\lambda \tag{2.7}$$

$$Z = \int_{380}^{780} C(\lambda)\overline{z}(\lambda)d\lambda \tag{2.8}$$

The weights, X, Y and Z define a color in the CIE XYZ space. Note that it is possible that two objects with different spectral reflectance, under certain illuminantion, appear the same, i.e. have the same CIE XYZ values. The CIE XYZ is a 3D linear color space, and it is quite awkward to work in it directly. It is common to project this space to the X+Y+Z=1 plane. The result is a 2D space known as the CIE chromaticity diagram. The coordinates in this space are usually called x and y and they are derived from XYZ using the following equations:

$$x = \frac{X}{X + Y + Z} \tag{2.9}$$

$$y = \frac{Y}{X+Y+Z} \tag{2.10}$$

$$z = \frac{Z}{X+Y+Z} = 1 - x - y \tag{2.11}$$

As the z component bears no additional information, it is often omitted. Note that since xy space is just a projection of the 3D XYZ space, each point in xy corresponds to many points in the original space. Actually the missing information is luminance Y. Color is usually described by xyY coordinates, where x and y determine the chromaticity and Y the lightness component of color. Figure 2.4 shows the CIE xy chromaticity diagram.

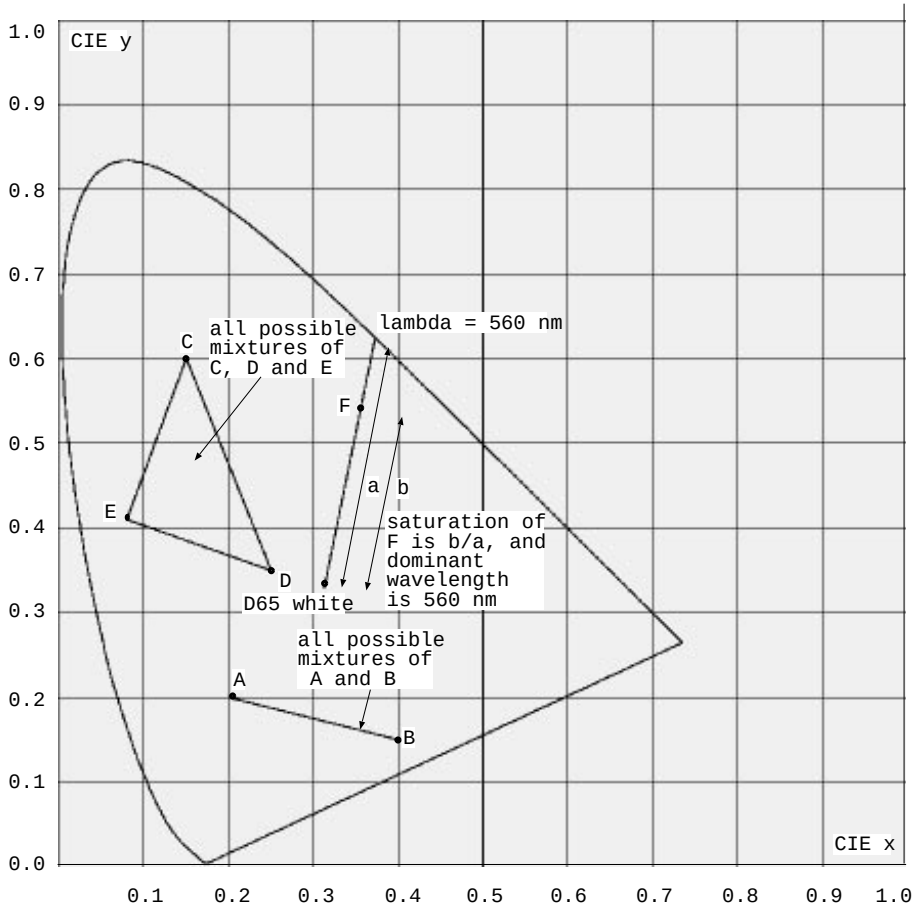


Figure 2.4: CIE xy chromaticity diagram

Chromaticity diagrams can give us a lot of useful information on a particular color. The horseshoe shaped curve represents the whole spectrum. The straight

line connecting the lowest wavelength blue and the highest wavelength red is called the "purple line" and does not represent spectral colors. The white point lies somewhere inside the diagram, depending on the light source used (e.g. D65 light source is defined to simulate day-light and has $x = 0.312727$ and $y = 0.329024$). If a line is drawn through the white point and a particular xy color, then the ratio between this point to white point distance, and the spectral line to the white point distance gives us the color saturation. If the color is close to the spectral line its saturation is high. The dominant wavelength which determines a color's hue is determined by the intersection of the line with the spectral line. An interesting property of the xy chromaticity diagram is that all possible mixtures of colors x_1y_1 and x_2y_2 are given by the straight line connecting these two points. It is clear that all the possible mixtures of the three colors x_1y_1, x_2y_2 and x_3y_3 then lie inside the triangle determined with those three points. Now, it is clear that the color gamut of any display device using three primaries (like a standard CRT monitor) is only a subset of all visible colors.

In spite of all the useful characteristics of the CIE xy chromaticity diagram, it lacks one very important characteristic. Namely, if the distance between any arbitrary two points is the same as the distance between an other point pair, the perceived distance will not be the same. In the worst case, if the perceived distances are the same, actual distances can differ as much as 20 times. In order to correct this, researchers are trying to find a perceptually uniform color space. It has, unfortunately, still not been found. CIE proposed two alternatives as improvements compared with CIE xyY space. These are CIE LUV and CIE LAB. Although they are referred to as perceptually uniform color spaces by some authors, they are not. Just for comparison, two perceptually equally distant color pairs, can differ in the CIE LUV distance as much as 4 times. This is a significant improvement compared to 20 times by original space, but it is still not perfect. Conversions between CIE XYZ and CIE LUV are defined with the formulas:

$$L^* = 116 \cdot \sqrt[3]{\frac{Y}{Y_{White}}} - 16 \tag{2.12}$$

$$u^* = 13 \cdot L^* \cdot (u' - u'_{White}) \tag{2.13}$$

$$v^* = 13 \cdot L^* \cdot (v' - v'_{White}) \tag{2.14}$$

where

$$u' = \frac{4 \cdot X}{X + 15 \cdot Y + 3 \cdot Z}$$

$$v' = \frac{9 \cdot Y}{X + 15 \cdot Y + 3 \cdot Z}$$

The distance between two colors in the CIE LUV space can be computed using

CIE LUV color difference formula:

$$\Delta E^* = \sqrt{\Delta L^* + \Delta u^* + \Delta v^*} \tag{2.15}$$

2.3 Human Vision

Up to now we have seen how light and colors can be measured. But what happens when light reaches our eye? It hits the photoreceptors on the retina, and they send the signal through nerves to the brain, where an image is formed. As stated before, it is possible that each one of us creates different image from the same stimuli. As we can not describe our sensation, it can not be quantified in any way, either. The complete vision system is not completely understood yet, but there are many known human vision characteristics. A lot of research is done in the laboratories, under special conditions. E.g. the observer has adapted to a certain light level (it can even take half an hour or more to fully adapt to some conditions), he or she sees only a small portion of the whole visual angle and so on. Sometimes, results of such experiments are applied on computer generated images, viewed in a complex environment, that is far from the ideal laboratory settings. That is the fact, that should always be in our mind, when evaluating various tone mapping methods. In this section we are going to describe various characteristics used in various tone mapping techniques. There are a lot more known human vision characteristics, but they are beyond the scope of this work.

The light intensity range that we experience every day is huge. The ratio of light at noon on a sunny day and the moonlight can be as much as 10 million. As stated before this light hits photoreceptors in our retina, namely rods and cones. Rods are extremely sensitive to light and provide achromatic vision at scotopic levels of illumination (10^{-6} to $10\ cd/m^2$). They provide achromatic vision, and that is the reason why we can not see colors in dark surroundings. The cones (there are three types of them) are less sensitive, but provide color vision at photopic levels of illumination ($0.01\ to\ 10^8\ cd/m^2$). Note that both systems are active at light levels between 0.01 and $10\ cd/m^2$. This range is called the mesoptic range. Unfortunately the mesoptic range is the poorest researched, and this is the range that is exercised by computer-based office environments with CRT monitors and subdued lighting.

What happens after the light hits photoreceptors? The signal travels by neural units to the brain where an image is formed. It is interesting, that despite the fact that incoming light can have a dynamic range of nearly 14 log units, the neural units can transfer the signal having the dynamic range of only about 1.5 log units. It is obvious that there is some adaptation mechanism involved in our vision. It means that we adapt to some luminance value, and then we can perceive data

in a certain dynamic range near the adaptation level. One of the most important characteristics that changes with different adaptation levels is the just noticeable difference.

2.3.1 Just Noticeable Difference

According to Weber's law, from the beginning of the century, the ratio $\Delta L/L$ of the just noticeable difference ΔL and the luminance L is constant, and equals 0.02 for a wide range of luminances. Nowadays there are better descriptions of just noticeable difference, and it is clear that it is not constant but depends on the adaptation level, and can be approximated using Weber's law just at certain adaptation levels.

The mapping function proposed by Greg Ward in [Ward94] relies on the work of Blackwell conducted in the early 1970s. Using a briefly flashing dot on a uniform background Blackwell established the relationship between adaptation luminance, L_a, and just noticeable difference in luminance $\Delta L(L_a)$ as:

$$\Delta L(L_a) = 0.0594 \cdot (1.219 + L_a^{0.4})^{2.5} \tag{2.16}$$

That means that if there is a patch of luminance $L_a + \Delta L_a$ on the background of luminance L_a it will be discernible, but the patch of luminance $L_a + \epsilon$, where $\epsilon < \Delta L_a$ will not.

A more complex function for the whole range of human vision is used by Ferwerda et al. [FPSG96], and later by Larson et al. in [LaRP97]. It accounts for both rod and cone, response, and is given in equation 2.17.

$$log(\Delta L(L_a)) = \begin{cases} -2.86 & \text{if } log(L_a) < -3.94 \\ (0.405 log(L_a) + 1.6)^{2.18} - 2.86 & \text{if } -3.94 \leq log(L_a) < -1.44 \\ log(L_a) - 0.395 & \text{if } -1.44 \leq log(L_a) < -0.0184 \\ (0.249 log(L_a) + 0.65)^{2.7} - 0.72 & \text{if } -0.0184 \leq log(L_a) < 1.9 \\ log(L_a) - 1.255 & \text{if } log(L_a) \geq 1.9 \end{cases} \tag{2.17}$$

Ferwerda et al. [FPSG96] and Larson et al. [LaRP97] also exploit the changes in visual acuity. Visual acuity is the measure of the visual system's ability to resolve spatial details. It drops off significantly for low illumination levels. Actually it is about $50\ cycles/degree$ at $3\ log\ cd/m^2$ and drops off to about $2.2\ cycles/degree$ at $-3.3\ log\ cd/m^2$.

Ferwerda et al. also used the time aspect of adaptation. We are all familiar with the fact that we can not see immediately after entering the cinema if the film has already begun. After some period of time we can see the details again. Using Ferwerda's model it is possible to simulate such time changes of adaptation in computer graphics.

2.3.2 Brightness as a Function of Luminance

Let us now consider the brightness perception. Brightness is the magnitude of the subjective sensation produced by visible light. The light intensity can easily be measured, but brightness as a subjective phenomena cannot be exactly measured. Nevertheless, brightness is often approximated as log luminance, or luminance powered to 1/2 to 1/3 depending on the authors. More precise studies showed that there is no one single formula, but rather the brightness-luminance relation depends on the adaptation level and the surrounding light. We will describe the work of Stevens et al. [StSt63] extensively used by Tumblin and Rushmeier in developing their tone mapping operator in this section.

Stevens et al. [StSt63] devised the "brils" units to measure the subjective value of brightness. According to Stevens 1 bril equals the sensation of brightness induced in a fully dark-adapted eye by a brief exposure to a 5 degree white target of $10^{-6}lambert$ ($1\mu lambert$) luminance.

Note that two images with different luminance values can have the same brightness values, and appear to be the same. The reason lies in the adaptation mechanism, and the inability of neural units to transfer high dynamic range signals from the retina to the brain. Actually we are very poor judges of absolute luminances, all that we can judge is the change in luminance, i.e. the brightness.

What did Stevens do? He measured brightness as a function of luminance and adaptation by using "haploskopic matching". That means he tried to match the brightness when one eye is dark adapted (standard condition for brightness measuring) and the other eye is adapted to a test value. Brightness comparison between two eyes was made quickly, before either could change adaptation level significantly. Measured brightness is then:

$$B = K \cdot (R_{targ} - R_{thresh})^n \tag{2.18}$$

where B is brightness in brils, R_{targ} is radiance of target in $millilamberts$, R_{thresh} is threshold of detectable radiance in millilamberts (this depends on the adaptation radiance), and n and K are constants, dependent on the strength of the adapting field. For full dark-adaptation $R_{thresh} = 0$, $n = 0.33$, and $K = 10$.

Stevens proposed the next equation from his measurements:

$$log_{10}(B) = 0.004 \cdot [(S - 27) \cdot (8.4 - R) - 108] \tag{2.19}$$

where, assuming L_w is adapting, white background luminance in lamberts and L_{targ} is target luminance in lamberts, $S = L_w$ in dB, where $0\ dB = 10^{10}\ lamberts$

$$S = 10 \cdot log_{10}(\frac{L_w}{10^{-10}} lamberts) = 100 + 10 \cdot log_{10}(L_w)$$

and R is the target luminance difference in dB

$$R = S - 10 \cdot log_{10}(\frac{L_{targ}}{10^{-10}} lamberts) = (10 \cdot log_{10}(L_w) - 10 \cdot log_{10}(L_{targ})$$

After substituting S and R expressions in equation 2.19 we can write the final equation:

$$log_{10}(B) = \alpha \cdot log_{10}(L) + \beta \tag{2.20}$$

where B is brightness in brils, L is viewed (target) radiance in lamberts, L_w is luminance of white surround and

$$\alpha = 0.4 \cdot log_{10}(L_w) + 2.92$$

$$\beta = -0.4 \cdot (log_{10}(L_w))^2 + (-2.584 \cdot log_{10}(L_w)) + 2.0208$$

These complex formulas provided by Stevens are, unfortunately, neither valid nor accurate when applied to more complex images. They are valid for laboratory settings only. Bartelson and Breneman [BaBr67] have measured many test photographs in order to find appropriate brightness versus luminance function for more complex images. They have proposed an extended formula for complex scenes:

$$log_{10}(B) = \alpha + \beta \cdot log_{10}(R_{rw}) - \gamma \cdot R_{rw}^{\delta} \tag{2.21}$$

where $\alpha,\ \beta,\ \gamma$, and δ are parameters dependent on viewing conditions and are given graphically.

2.3.3 Brightness as a Function of Reflectance

Up to now brightness was considered only as function of luminance. As luminance is the product of incoming illumination intensity and reflectance (for non-emitting surfaces) the same luminance can be obtained from surfaces having different reflectances by changing the illumination. Franck [Fran94] showed that reflectance influences perceived brightness as well. This is valid only for complex scenes, and his conclusion is based on an experiment. There is still no quantitative data or suggested formulas, but he claims that a white surface will always appear brighter then a black surface of the same luminance when cues to the reflectances are present, as is frequently true in digital image synthesis. This fact is often neglected, but obviously should be taken into account by tone mapping functions.

2.3.4 Adaptation and Veiling Luminance

Up to now we have been talking about adaptation luminance without saying to which luminance a human observer would adapt. It is assumed that we will adapt

to the luminance of our fixation point which approximately covers one visual degree (1.5°, by some authors). Just to make things a little bit more complicated, the adaptation luminance depends on the surrounding luminances as well. The influence of the surrounding is not large, but if there are some glare sources in the periphery, the veiling luminance should be taken into consideration.

Bright glare sources in the periphery reduces contrast visibility because light scattered in the lens obscures the fovea. The influence of the veiling luminance to the adaptation luminance is well documented in the literature, and we will present here a model introduced by Moon and Spencer [MoSp45], because this is the model used by Larson et al. in [LaRP97].

Moon and Spencer proposed the next formula for the corrected adaptation luminance L_a:

$$L_a = 0.913 \cdot L_f + \frac{K}{\pi} \int_{\theta > \theta_f} \int \frac{L(\theta, \phi)}{\theta^2} \cdot cos(\theta) \cdot sin(\theta) d\theta d\phi \tag{2.22}$$

where L_a is the corrected adaptation luminance in cd/m^2, L_f is the average foveal luminance in cd/m^2, $L(\theta, \phi)$ is the luminance in the direction (θ, ϕ), θ_f is foveal half angle, $\approx 0.00873\ rad(0.5^\circ)$, and K is the constant measured by Holloday [Holl26], 0.0096.

It is obvious from the above equation that the periphery contributes less than 9% to the adaptation luminance. If there are no bright sources in the periphery this influence can be neglected.

2.3.5 Contrast Sensitivity Function

The contrast sensitivity function described here will be used to develop the color image metric described in chapter 8. Contrast sensitivity is sometimes called visual accuity [LaRP97], [FPSG96]. We will use the term contrast sensitivity here, since we have used this terminology throughout chapter 8. Mannos and Sakrison [MaSa74] proposed a model of the human contrast sensitivity function. The contrast sensitivity function tells us how sensitive we are to the various frequencies of visual stimuli. If the frequency of visual stimuli is too high we will not be able to recognize the stimuli pattern any more. Imagine an image consisting of vertical black and white stripes. If the stripes are very thin (i.e. a few thousand per millimeter) we will be unable to see individual stripes. All that we will see is a gray image. If the stripes then become wider and wider, there is a threshold width, from which on we are able to distinguish the stripes. The contrast sensitivity function proposed by Manos and Sakrison is

$$A(f) = 2.6 \cdot (0.0192 + 0.114 \cdot f) \cdot e^{-(0.114 \cdot f)^{1.1}} \tag{2.23}$$

f in equation 2.23 is the spatial frequency of the visual stimuli given in cycles/degree. The function has a peak of value 1 aproximately at $f = 8.0$ cycles/degree, and is meaningless for frequencies above 60 cycles/degree. Figure 2.5 shows the contrast sensitivity function $A(f)$.

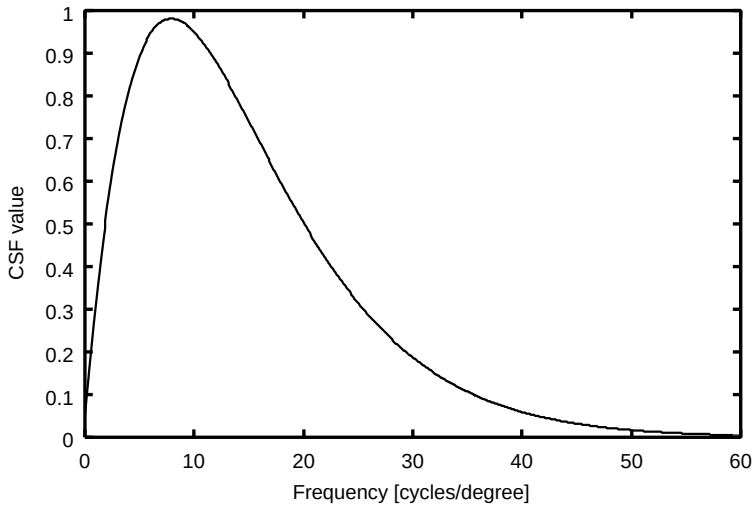


Figure 2.5: Contrast sensitivity function

The reason why we can not distinguish patterns with high frequncies is the limited number of photoreceptors in our eye. There are several other functions proposed by other authors, but we choose the above function [MaSa74] because it can be simply analitically described. The same function is also used by Rushmeier et al. [RWPSR95] and Gaddipati et al. [GaMY97], which was another motivating factor in using this function.

Chapter 3

Display Devices

Since the main goal of most rendering processes is to display the image for human observation, we should examine some display media characteristics in order to use this media properly. Rendered images can have a dynamic range of several thousands, and even more. As stated in the previous chapter our visual system operates in an impressive dynamic range. Unfortunately, display media dynamic ranges are quite small. For example, a CRT monitor, which is the most widely used display media in computer graphics has a dynamic range of up to 100! Obviously, huge dynamic range raw images should be mapped somehow to the relatively small dynamic range of display devices. An ideal display media would have a dynamic range that equals that of human vision capabilities, and would have the possibility of displaying luminances as low as the threshold of human vision, and as high as the maximum still perceivable luminance. It should be capable of reproducing visible colors, as well. Although such devices have long been in existance in acoustics, they will not be available in the video media for a long time.

There are two kinds of display media, light-emitting like CRT or, in a way, projected slides, and light-propagating, like photos or prints, which do not emit light themselves. Light propagating media is suitable for displaying solid colors by means of an external light source, while the other group has gamuts exceeding the solid colors and has the capability to display more saturated colors. E.g. the saturation of CRT blue can never be achieved with photo paper.

There are three major problems concerning display devices. The first is the display devices' non-linear response, the second is the limited dynamic range and the third is the limited color gamut.

Practically all display media have nonlinear characteristics. Fortunately, that is not such a big problem, as long as the user is aware of it. If the characteristic of the device is known, some correction can be done, and the device will act as a linear device. The situation is more complex for display chains. Let us take a chain of CRT $\rightarrow$ negative color film $\rightarrow$ photo paper, or a chain film-writer $\rightarrow$

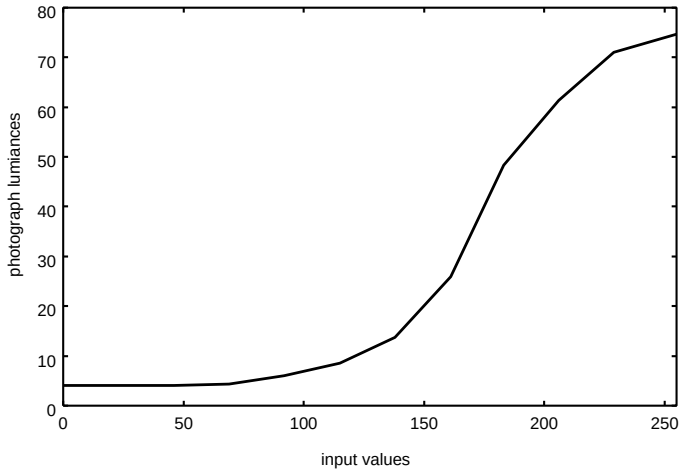


Figure 3.1: Measured characteristic of a photograph print made from slide

color slide $\rightarrow$ photo paper. It means, an image is displayed on the monitor, then photographed, and finally a print is made from color negative film (this is valid for the first chain). For such chains the last link is important, and inputs to the first link should provide the desired results at the chain's end. In other words, if the photo is made from a slide, which is made by a film recorder and the input to the film recorder is a TIFF file, it is not important how this TIFF file looks on the CRT, or when the slide is projected. All that matters is that the final print looks satisfactory. Of course the final print depends on many variables. They include the type of film used, the quality and temperature of the developing chemicals (in the above case there are two developments), the length of time the film is in the chemicals, the type of photo paper used etc. There are also some influences from one color channel to others in all color media, some additivity failures [ToHe89] and so on. It is obvious that it would be quite difficult, if not almost impossible for the common user, to take into account all the above mentioned difficulties. What the common user should do, is to check the linearity of the device (chain) and the available device contrast.

The second, and more complex problem is the limited device contrast. As mentioned earlier, huge contrast raw images should be mapped to relatively small contrast display devices. This is, actually the most challenging part of most tone mapping techniques. We have measured CIE Y values of the photography made from the slide from the film recorder. Figure 3.1 shows the results. Input values to the film recorder are on the x axis, and the CIE Y values of the last chain link - photo, are on the y axis. Achieved contrast was $74.7/4.0 = 18.675$.

Typical device contrasts are given in table 3.1, the ideal values and values that

we have measured on common available devices are given in the table. Note quite a big difference in photo contrast. We have measured the contrast of photos that are automatically processed by Kodak. Some professional laboratories in Vienna are offering better service, but at, approximately, six times the cost of the usual Kodak service. We assume that the contrast in this case would be greater, and closer to the theoretical maximum.

Display media	Typical contrast	Measured contrast
CRT	50-200	50
Photographic prints	100	18.675
Photographic slides	1000	
Newsprint printed in B/W	10	

Table 3.1: Typical display device contrasts

The third above mentioned problem is the display devices' limited color gamut. There is a lot of research done in gamut mapping. Throughout this work, we will apply simple color components clipping, in case they exceed the device gamut. This approach can lead to hue changes in some cases, but more advanced gamut mapping techniques are out of the scope of this work, and can be found in [GeAl89], [HoBe93], [WoAB94].

The most widely used media for computer graphics are the CRT monitor and color printer, for sure. Figure 3.2 shows the color gamut of a typical monitor, and a printer. The gamut of a printer using highly saturated inks is shown as well. It is obvious from this figure why is it impossible to reproduce monitor images on the color printer perfectly.

We find the slides and CRT the most interesting media, therefore they will be explained in a simplified, yet for us sufficient way next.

3.1 Slides and Goldberg Rule

An image is created on a slide as the result of a chemical process. The light arriving on the film through the lenses causes some chemical reactions on the emulsion that is on the film, and an image is formed. The film is then processed, and eventually processed once more to obtain photographic prints.

Various films are often described using an "H-D" plot (H-D stands for Hurter and Driffield who devised them in 1890 [JaMe66]). An H-D plot describes a density as a response to a given exposure. Let us define exposure and density next.

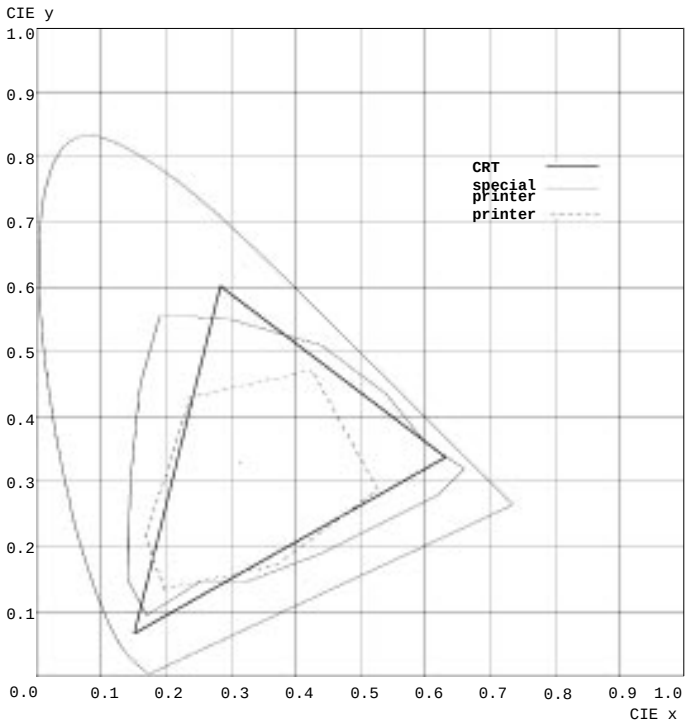


Figure 3.2: Color gamut of printer and CRT

Exposure is defined as the product of the irradiance incident upon the photosensitive surface (I) and the time during which the surface is exposed (t):

$$E = It \tag{3.1}$$

Photographic exposure is usually given in photometric rather than radiometric units. In equation 3.1, I is given in $lumens/m^2$ or lux and exposure is then in $lux - sec$. In the cases where the incoming flux is a continuous spectral distribution, the exposure is given by the integral:

$$E = \int E_\lambda d\lambda = \int I_\lambda t d\lambda \tag{3.2}$$

The measured response of a photographic material is given in density. Density is a unitless, logarithmic measure that indicates the opacity of an emulsion that results from processing.

Transmission density is used for describing the response of photographic film. It is defined as:

$$D_T = log_{10} 1/T \tag{3.3}$$

where T is a transmittance value, $T \in [0, 1]$, that gives the ratio of light transmitted through an emulsion to the quantity of light incident to it. There is also the reflective density D_R which is used in describing the response of photographic papers.

Let us consider now, a color slide with characteristic curves as in fig. 3.3. The horizontal axis shows the logarithm of the exposure, the vertical axis shows the densities in the r,g,b channels.

The density ranges of the three color channels are somewhat different from another, hence an achromatic gray can only be produced in the coincident part of the density ranges. It can be seen in fig. 3.3 that the density ranges exceed 3. In linear terms: the contrast exceeds 1000! The corresponding illumination range on the horizontal axis is $[-2.5, -0.1]$. Values less then -2.5 and greater then -0.1 cause no change in densities (so called "fog" and "flare" regions). That means, roughly, that the logarithm of the range of exposition is 2.4, a value of about 250 on a linear scale! This is much higher than the displayable effective contrast on any other medium. Dynamic ranges of black and white glossy photo papers can be up to 60, but for all other media including color photos and prints it is significantly smaller, in a range of 10 to 40. In photography, only the straight curve sections are convenient. Moreover, the standard is a paper print made from the slide, perceptionally correct for a contrast rendering of 1:1. This means that we want to reproduce the original contrast of 50 as 50 on the final print. Therefore, the really applicable range of contrast for exposures does not exceed the value of 45, that is about 1.65 in log_{10} (actually the recommended contrast value in photography is 32). For instance, on the horizontal axis of fig. 3.3 the section $[-0.25, -1.9]$ may be applied. This corresponds to the density range $[0.2, 2.7]$ on the vertical axis. That means that the original scene contrast of ≈ 45 is mapped to a contrast of ≈ 316 on the slide.

Slide gamma is defined as the slope of the straight part of the characteristic seen in fig. 3.3. Slide gamma values vary from 1.5 to 1.7, and this is the reason why the scene contrast is enlarged on the slide. The question emerges why the gamma value is not 1. Obviously, on a paper print (enlargement) gamma must be 1, otherwise the print would not be equivalent to the original appearance. Remember that we want to reproduce original contrast exactly. This is possible since photo papers have gamma values as well. Enlargements from slides are made on so called positive papers, with gamma values less than 1. The resultant gamma is the product of the gammas of the slide and of the positive paper, in fact, about 1. Actually the original scene contrast will be enlarged on slide, and then again decreased by the use of a positive photo paper. Similarly, soft color negatives, with gamma values less than 1, and enlargement papers of hard gradation (with gamma values over 1) have resultant gamma values of about 1. The original contrast is changed on slides due to a γ value of 1.5 to 1.7. Even the color components are

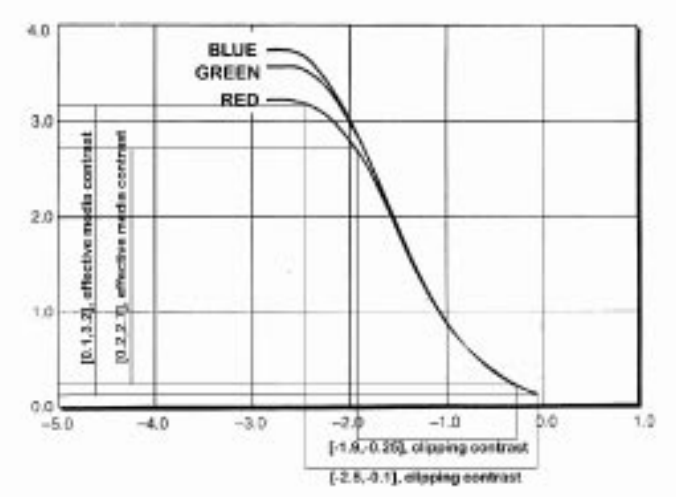


Figure 3.3: The characteristic curves of Kodak Ektachrome Panther 100x Professional

changed from r, g, b to r^γ, g^γ, b^γ. The question emerges how it is possible that slides are, despite all these facts, perceived correctly when projected. A projected slide enlarges the original contrast, but the perceived image looks correct. The Goldberg rule gives us the answer [Schr81]. The Goldberg rule states that the way we see something depends in an exponential way (Goldberg-Gamma) on the surrounding lighting. If the human eye is adapted to a very dark surrounding, a Goldberg-Gamma value of 1.5 to 1.7 gives perceptually correct results. For eyes adapted to brightness the Goldberg-Gamma decreases down to 1. That is one of the reasons why slides viewed in a well lit room do not look satisfactory, but when they are projected in a dark room they are just perfect. In other words, our visual system reduces the contrast in the dark surrounding, and high contrast projection is perceived as normal contrast image. On the other hand if a slide is viewed in a bright surrounding, its contrast is too high, and it looks somehow contrastless. This paradox that high contrast image seems to have low contrast, and a low contrast image looks as full of contrast will be described in the section 4.2.

Note, that it would be possible to exploit slide characteristic in another way. We could compute a raw image, and compress the original contrast of let's say 300 to 50, and use it as input to the film writer. The resulting slide will then increase input contrast to the original 300 value. In this case the slide should be viewed in a well lit room, placed on a diffuse light source. If there were a big enough slide, it would outperform all common display media this way.

3.2 CRTs

CRT monitors are the most widely used display devices in computer graphics. A CRT (cathode ray tube) is the essential part of all CRT monitors. It contains three electron guns which emit narrow streams of electrons. The streams pass through deflection coils that band the beam up, down, right and left. The inside of the front face of the CRT is coated with three different phosphors, each of them emits light of certain wavelength when struck by electrons. The intensity of light depends on the number of incident electrons. There are various phosphors, but red, green and blue emitting type phosphors are used in common monitors. As all displayable colors are formed from an additive mixture of red, green and blue, it is clear that the color gamut of a CRT monitor is a subset of all visible colors.

We will examine some CRT monitor characteristics next. Just as before, we are interested in the monitor response to various input values. Unfortunately this response is not linear. For a CRT with N displayable levels, the luminance of grade i, L_i can usually be approximated by:

$$L_i = \left(\frac{i}{N}\right)^{\gamma_{display}} \tag{3.4}$$

where $i = 0, 1, 2, \ldots, N$. The value of $\gamma_{display}$, of course, is slightly different for each color channel.

We have measured L_i for the red, green and blue color channels of an SGI color monitor. The results are shown in figure 3.4. It can be seen that default gamma correction is not perfect, but it is close to linear response. For most applications it will be sufficient.

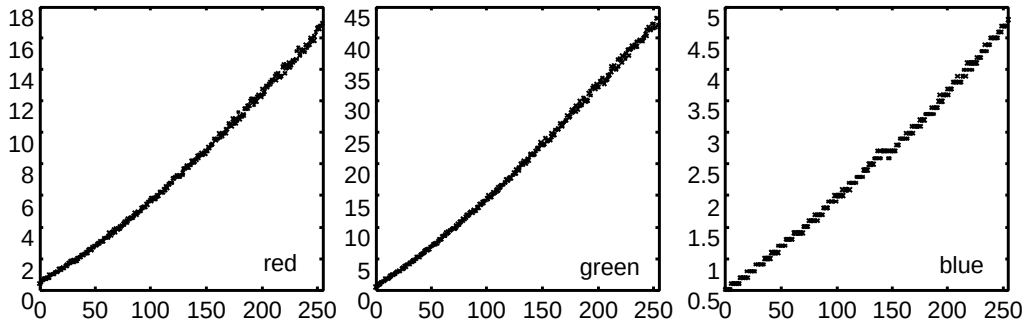


Figure 3.4: Red, green and blue gun response after γ-correction

Although it seems in (3.4) that the CRT contrast is infinite (the lowest luminance value is 0), there is no real 0 luminance, i.e. the CRT emits some radiance

even for i=0, (if the measured characteristics in the figure 3.4 are observed carefully it can be seen that there is some emittance even for 0 input level (best seen for red and blue guns), and there is always some reflection from surrounding light on the screen [ToHe89]. If you look at the switched off display in a lit room you will see it is far from totally black. This unwanted luminance level that is proportional to the ambient light is added to the image. It means that in (3.4) a constant proportional to the ambient luminance level should be added. The ambient luminance decreases the CRT contrast and the gamut as well. Figure 3.5 illustrates the problem. It shows how gamut changes with the increase of ambient light.

There is a lot of research done on the calibration of the CRT monitor [Durr87], [Hall89]. Actually the characteristic changes as time passes, it depends on room temperature and even on the orientation of the monitor (influence of the Earth magnetic poles [Fitz89]). All of these influences are relatively small, and they are not so important for the common user.

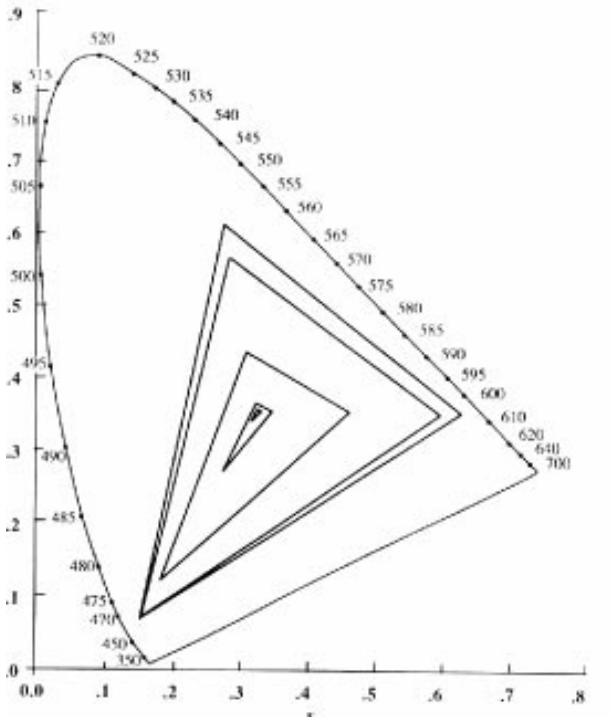


Figure 3.5: Chromatic gamut of CRT for ambient illumination covering a range of five log units. [Layc83]

The CRT contrast depends on the ambient illumination level, on user settings, and on particular device characteristics. For the same CRT the contrast can vary

from 20 to 100. The best would be to measure the particular CRT and to work in a room with a constant illumination. Of course, in this case the device settings once set, should not be changed any more. The Goldberg-Gamma should be taken into account for CRTs as well. In dark rooms the Goldberg-Gamma should be set to 1.5, in dim surroundings to 1.2 (default setting for TV). When a CRT is viewed in very bright surroundings the Goldberg-Gamma decreases to 1. Goldberg-Gamma correction can be done simultaneously with display gamma correction. Instead of $\gamma_{display}$ in (3.4), $\gamma_{display}/\gamma_{Goldberg}$ should be used. More exact relationships than the Goldberg rule are described in the RLAB color space for luminance and chromatic adaptation [Fair94].

Figure 3.4 shows the measured CIE Y of the red, green and blue color channel. It can be seen on the y axis that the maximum values of Y are not the same. The blue gun emits the lowest maximum intensity, and the green gun the highest. If we are interested in computing an equivalent luminance image from our rgb image, then each of the color channels should be properly weighted. For this particular monitor, the weights can be computed from maximum intensities as:

$$r_{weight} = \frac{L_{red\_max}}{L_{red\_max} + L_{green\_max} + L_{blue\_max}} \tag{3.5}$$

$$g_{weight} = \frac{L_{green\_max}}{L_{red\_max} + L_{green\_max} + L_{blue\_max}} \tag{3.6}$$

$$b_{weight} = \frac{L_{blue\_max}}{L_{red\_max} + L_{green\_max} + L_{blue\_max}} \tag{3.7}$$

For our monitor, the weights are $r_{weight} = 0.268$, $g_{weight} = 0.667$ and $b_{weight} = 0.065$.

We have measured chromaticies CIE x and y values for our monitor as well. It is interesting that chromaticies for a particular phosphor were not constant as expected. The measured CIE x,y chromaticies are shown in figure 3.6. We suppose the reason lies in the fact that there are some cross effects among phosphors. There can be also some error caused by the measuring instrument at low intensities. Figure 3.6 shows gamuts for intensities 10, 30, 50, 70,..., 250. The chromaticies change decreases with higher intensities.

Note that the weighting functions computed in eq. 3.5 using maximum intensities, can be computed from chromaticies as well. Each phosphor has a dominant wavelength, and the CIE photometric curve gives a weighting factor for each of the three phosphors. Assuming our phosphor chromaticies have CIE x,y coordinates and dominant wavelength values as given in table 3.2, corresponding weights are: $r_{weight} = 0.324$, $g_{weight} = 0.631$ and $b_{weight} = 0.045$. These results are similar to the results obtained using the maximum intensities. Errors are due to non-perfect measuring conditions.

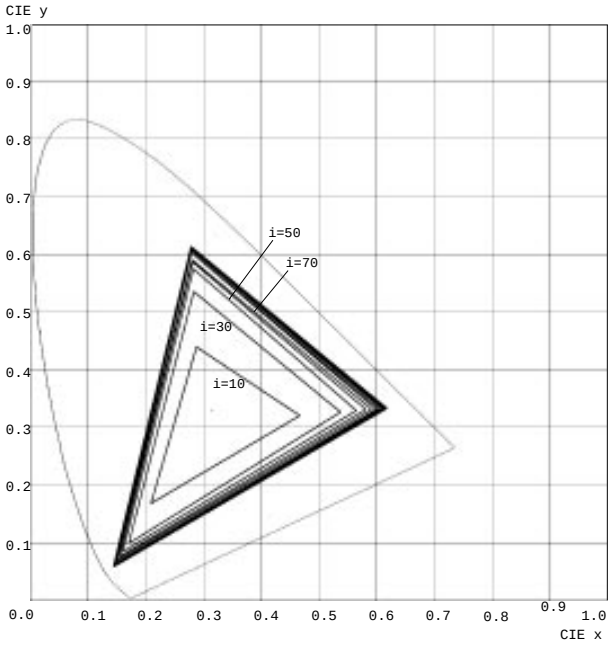


Figure 3.6: Change of phosphor chromaticies with increasing intensities

	x	y	λ
red	0.617	0.334	610
green	0.278	0.614	545
blue	0.154	0.058	463

Table 3.2: CIE xy coordinates and dominant wavelength of three phosphor primaries

Chapter 4

Linear Scale-Factor Methods

From now on, we will assume that the display device is calibrated, such that it has linear response and input range [0,1]. This input range corresponds to [0,255] for R, G and B color channels for today's standard devices. Throughout this work "n" will be used for the device input value, $n \in [0, 1]$, and "L" for the computed luminance value, $L \in [0, \infty]$. Our intention is to describe the various functions $n = f(L)$.

Most rendering software is still not able to render a raw image in absolute units. Rendering in absolute units can be quite a tricky job. When absolute units are required the exact data for light sources and materials are needed. This data is hard to get or measure, and sometimes the tolerance range of given data is too high. Data for artificial light sources can be obtained from the manufacturers and if natural light is used then the time of day, latitude and sky conditions should be taken into account. The data needed to define BRDFs for materials used in the scene is far more difficult to find than light source data is. The BRDF depends both on the chemical composition of the material and on the condition of the surface (smooth, rough, oxidized, etc.). Furthermore many common materials do not have spatially uniform BRDFs. Because of all these reasons most rendering tools still work with fictitious units. Some methods take absolute units into account and can not be used with raw images rendered in fictitious units.

Although human vision certainly does not use a linear scaling function, this group of mapping methods renders acceptable results for a wide range of applications. Its strengths are its simplicity and speed, and if the right method is chosen, the results can be acceptable for almost all applications if the raw image dynamic range is not too high.

The reason why linear mapping renders acceptable results, if the right scale factor is chosen, lies in the adaptation mechanism. If the right "adaptation" level is chosen the error introduced by using a linear mapping function is acceptable. The problem is only to find the right scale-factor.

As we want to get various images from computed luminances it is obvious that these computed values have to be stored. Usually these values are floating point numbers, one for each color channel (red, green and blue in most cases). Saving such a float image is very memory demanding (12 bytes per pixel). Greg Ward in [Ward92] suggests an elegant way of handling this problem and introduces a way how a float image can be stored using only 4 bytes per pixel and achieving almost the same result. There are also two other raw image formats (one supported by TIFF file type, and other introduced by Pixar). The raw image formats are described in the Appendix, as they are a necessity for this kind of image manipulation.

Another possibility of handling the float-image size problem is to use a low-resolution preview picture to find the parameters which will be used in the final high-resolution picture. The combination of these two methods should solve the memory problem even in computers with a very low amount of memory.

The first intuitive solution to the mapping problem is the use of a linear scale-factor such that the maximum radiance L_{max} is mapped to 1,

$$n = \frac{L}{L_{max}} \tag{4.1}$$

This mapping is useless if the light source is visible or the image contrast is too high. In these cases the final image will be too dark. The results of this mapping method are shown and discussed in the Results chapter. An improvement of this method, especially popular in the radiosity community, is the mapping of the largest non self-emitting pixel to 1. Unfortunately, in the case of strong secondary light sources this method still renders very dark images. The second drawback is that pixel self-emittance is known only in the first rendering phase, therefore it is not possible to estimate which pixel is and which is not self emitting from the raw image.

4.1 Mean Value Mapping

Probably the most widely used mapping method today is the mean value mapping technique. The idea is to map the average radiance to 0.5 input value, and then clip the values larger than 1 to 1:

$$n = 0.5 \cdot \frac{L}{L_{ave}} \tag{4.2}$$

where L_{ave} is the average radiance value, and n is set to 1 if $n > 1$.

According to the above equation, the value $2 \cdot L_{ave}$ will be mapped to 1 and all values larger than $2 \cdot L_{ave}$ will be clipped to 1. Obviously the information in

the range $[2 \cdot L_{ave}, L_{max}]$ is lost, although there can be some interesting details. Another problem is the case when few very high radiance values increase the average too much, making the final image too dark. There is also a problem that arises from the fact that the global illumination solution is linear in source radiance [ArKi90], that means results for any two light source strengths are directly proportional. Therefore the mean value mapping will produce the same images for various light sources' strengths. This problem cannot be overcome unless absolute units are known. Using fictitious units makes it impossible to know whether the scene is supposed to be well lit or dark. Another drawback of the mean value mapping technique arises when the average scene reflectance is very low or very high. Imagine an image representing a heap of coal. If the average value (which is low) is mapped to 0.5 the whole image will be too bright. On the other hand, an image of snow covered mountains will be too dark. In spite of all these drawbacks this is still the most widely used mapping method today. Most of today's rendering software can not produce absolute unit raw images, and most scenes are not very dark or very bright. These two facts make it possible to use mean value mapping successfully for most renderings. Of course, when an appropriate lighting atmosphere or the correct objects' colors are important, some other mapping should be used. Result images rendered using the mean value mapping method are shown in results chapter, color plates 1a, 5a, 9a, 9b, 11a, 14a, and 14b.

4.2 Interactive Calibration

The mean value mapping technique corresponds to the simplest photographers' approach, where the average light is measured and the aperture is automatically set such that the average is mapped to the medium gray. More advanced photographers know that they have to adjust the aperture manually if the scene is not a typical one. By setting the aperture manually the final photo can be darker or lighter. Photographers also know how to enhance the overall contrast of the image. Sometimes it is necessary to use a flashlight even on a sunny day if portraits are shot. Experienced photographers will choose an appropriate film speed as well.

Just as in real life, we did not want to use just the mean value mapping method, so we have proposed to use interactive calibration. Together with chapters 6, 7, and 8 this is the main contribution of this thesis.

Up to now we have been describing the mapping functions that are applied to the whole raw image. We introduce another approach here, namely that a clipping interval is selected first, and the mapping function is applied to this interval only. Values outside the interval are simply clipped to the interval borders.

Our intention is to find a clipping interval [s,e]. The start point of the interval will be mapped to the minimum displayable value and the end point to the max-

imum displayable value. Just as in photography the center of the interval is not always the average value, actually in professional photo this is almost never true. The interval is found by varying aperture and contrast, as explained in the next chapters.

4.2.1 Logarithmic Histogram

Logarithmic histograms will be used to explain the idea of this method. In the real implementation a histogram is never used. It just helps us to understand the method. Let's explain what the logarithmic histogram is and what its advantages are. As the log_2 (log_2 is used in analogy to photography) is not defined for 0, first all values that are smaller than a certain minimum value should be set to this minimum value, and as the size of the histogram array is limited, the values above the maximum allowed value should be set to this maximum value. Now, for the log_2 values of each of the r, g, b components, the appropriate histogram array element is increased. The total number of histogram entries is three times the number of pixels as each color component is entered separately in the histogram array. Using a logarithmic instead of a linear histogram has several advantages. A wide range of luminance values can be represented with a relatively small array, and the contrast manipulation (introduced later) is easier to explain.

4.2.2 Varying aperture

The interactive calibration method was inspired by photography, but our intention is not to apply exact photographic methods. We will adopt some terminology from photography and use it in a simplified way. In this paper the aperture value is defined as 0, if and only if, the average luminance value, x_{mean}, is in the center of the clipping window on the linear scale. When displayed on the log scale it is in the upper half of the interval. Setting the aperture to +1 means shifting the x_{mean} value by 1 on the log_2 scale, which doubles it. In our implementation the aperture can be set to ± 0.5, ± 1, ± 1.5,...,± 3. Of course, the aperture can be set to larger values or a finer step can be applied, yet, from the hundred years of photographic experience and from our experience, this is almost never necessary.

4.2.3 Varying contrast

In the classical mean value approach the simple scale factor $0.5/x_{mean}$ is applied to all values. The next step is to clip all values larger than 1 to 1 and those less than 0 (usually there are no such values) to 0. This approach clips the values far from the average. Actually, all values that are more than twice as large as x_{mean} are clipped to 1, which means that some interesting details, may not be displayed. By

varying the contrast and the aperture this problem can be successfully solved. Let us define the contrast as the ratio between the largest and the smallest luminance value in a float image. The contrast window on the log scale always has the same size for a given contrast and this is one more reason for using a logarithmic histogram. The original contrast of a float image can be anything from as low as 10 or less (although such low-contrast images are rare) to as high as 1000 or more. Using a larger contrast interval, less pixels will be clipped, but when they are mapped to the input values, a low-contrast image will be generated. On the other hand, the use of a small contrast clipping interval will produce an image which is considered as a high-contrast image. (See results chapter for visualisation of this confusing fact). In figure 4.1 various contrast windows and aperture settings are displayed, with the logarithmic histogram of the image shown in results section, color plates 3 and 4.

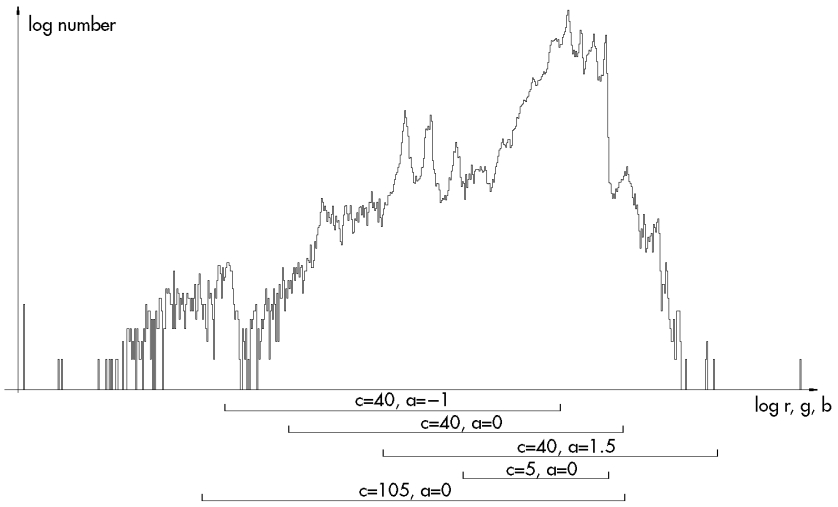


Figure 4.1: Various contrast and aperture settings

4.2.4 Mapping of the Interval

Our intention is to find the $[s, e]$ interval such that the final image (produced by mapping $[s, e]$ to the display device input values) is satisfactory. This means that it shows the atmosphere supposed to be in the scene, or that it shows lots of details in a selected area of the image. The user varies contrast and aperture settings until the image looks satisfactory. We assume that the display device is calibrated, has linear response and the ratio between the maximum and minimum

displayable lightness is $C_{display}$. We want to map $[s, e]$ to $[1/C_{display}, 1]$. We will call $1/C_{display}$ value $''\kappa''$. For most CRTs $C_{display}$ is about 50, which makes κ equal to 0.02. Given the x_{mean} value, the aperture value a and contrast c, the start point s of the interval is

$$s = \frac{2 \cdot x_{mean} \cdot 2^a}{1 + c} = \frac{2^{1+a} \cdot x_{mean}}{1 + c} \tag{4.3}$$

and the end point e can be computed as

$$e = s \cdot c \tag{4.4}$$

Once the interval is chosen, the mapping for float red, green, and blue values can be done. The values outside the $[s, e]$ interval should be clipped to s and e prior to the mapping. This color clipping produces visible color distortions in some cases. There are numerous other clipping approaches but they are out of the scope of this thesis. There are many ways of mapping $[s, e]$ to $[\kappa, 1]$. If the chosen contrast c is the same as the available device contrast $C_{display}$, a linear mapping gives the best results. The linear mapping is done by simply dividing the color components by e. The interval is mapped linearly to $[0, 1]$ using the following formulas:

$$R = red/e \tag{4.5}$$
$$G = green/e \tag{4.6}$$
$$B = blue/e \tag{4.7}$$

In the case when c differs from $C_{display}$ a linear mapping does not give satisfactory results. According to human perception rules [StSt63], [Hunt92] a mapping function of type $f(x) = x^n$ can be chosen for appropriate results. The value of n lies between 1/3 (e.g. in CIE LUV and CIE LAB lightness formulas) and 1/2 (e.g. in Hunter LUV lightness formulae) [Hunt92]. We suggest taking $n = 0.4$ as default value. Two auxiliary values u and v for the mapping from $x \in [s, e]$ to $y \in [\kappa, 1]$ are defined as :

$$u = \frac{f(1) - f(\kappa)}{f(e) - f(s)} \tag{4.8}$$
$$v = f(1) - u \cdot f(e) \tag{4.9}$$

The mapping functions for mapping the red, green and blue values (all in the $[s, e]$ range) to R,G,B (in the $[\kappa, 1]$ range) are:

$$R = f^{-1}(u \cdot f(red) + v) \tag{4.10}$$
$$G = f^{-1}(u \cdot f(green) + v) \tag{4.11}$$
$$B = f^{-1}(u \cdot f(blue) + v) \tag{4.12}$$

Images mapped using this type of mapping function will be brighter than those mapped with the linear function, but they will show more details in the dark part of the image and they will be perceptionally more correct. Now R, G and B can be used as input to the display device look up table.

4.2.5 Conclusion

We have described a method for mapping calculated luminance values to values appropriate for displaying. Varying the contrast and aperture, images with the appropriate atmosphere can be displayed without knowing the luminances in absolute units. Also, for research purposes, some interesting parts of the images can be shown with lots of details by setting a small contrast and an appropriate aperture value. The introduced methods can save us additional renderings in most cases, and they reduce the time needed to produce the final image that way. Results strongly depend on the user settings, so this is a completely subjective method. Images mapped using this method are shown in the results chapter, color plates 2, 3, and 4.

4.3 Ward's Contrast Based Scale-factor

The idea is to display bright scenes as bright and poorly lit scenes as dark, making the differences just visible in the real world just visible on the display. In other words the visibility in the scene is preserved. The scale-factor is derived from the contrast sensitivity studies conducted by Blackwell in the early 1970s (see Human vision section). Blackwell established the following relationship between the adaptation luminance L_a and the minimum discernible difference in luminance:

$$\Delta L(L_a) = 0.0594 \cdot (1.219 + L_a^{0.4})^{2.5} \tag{4.13}$$

Ward used Blackwell's studies, but we should be aware that all models described here and based on human perception are only an approximation of the human vision system, since Blackwell's experiments were conducted in perfect laboratory conditions that are far from the usual complex viewing conditions of a typical work place. Ward wanted to find a proportionality constant between display luminance and world luminance that yields a display with roughly the same contrast visibility as the actual scene. He wanted to find a multiplier m such that:

$$L_d = m \cdot L_w \tag{4.14}$$

where L_d is display luminance at an image point, and L_w is world luminance at an image point.

The key assumption that enables the calculation of m is that:

$$\Delta L(L_{da}) = m \cdot \Delta L(L_{wa}) \tag{4.15}$$

where $\Delta L(L_{da})$ is the minimum discernible luminance change at L_{da}, L_{da} is the display adaptation luminance and L_{wa} is the world adaptation luminance.

This assumption makes differences just visible in the real world just visible on the display, and this was the main goal of the whole mapping. Solving the equation 4.15 for m gives:

$$m = \left[\frac{1.219 + L_{da}^{0.4}}{1.219 + L_{wa}^{0.4}}\right]^{2.5} \tag{4.16}$$

In order to get the scale-factor which converts a raw image to a display device input interval [0,1], the maximum display luminance, L_{dmax}, should be known. The display adaptation luminance of the viewer, L_{da}, should be known as well. Ward suggests to take $L_{da} = L_{dmax}/2$, as he has found that this is close enough for most applications. The final scale-factor n is then:

$$n = \frac{1}{L_{dmax}} \cdot \left[\frac{1.219 + (L_{dmax}/2)^{0.4}}{1.219 + L_{wa}^{0.4}}\right]^{2.5} \tag{4.17}$$

Two unknowns are the maximum display luminance which can be measured, and the world adaptation level. If no other information is available, the average value of the raw image can be taken as the world adaptation level although this is not the completely correct way. If the area of interest in the scene is known, this can be used as the adaptation level. In this way results can be interpreted as telling us how well a person would see in an environment while looking at this point. Note that the raw image should be rendered in absolute units to use this method. This is still not easy for a common user, but this is the only way the lighting atmosphere can be automatically taken into account. This simple linear scale-factor can be used for a wide range of applications where lighting simulation is important (architecture visualisation) and it renders great results. Color plates 9c, 9d, 13a, 13b, 17a, and 17b in results chapter illustrate this method.

Chapter 5

Non-Linear Scale-Factor Methods

In this chapter non linear scale factors will be described. The motivation for using non linear scale factors is to be found in Weber's law. This law derived in the last century states that the ratio of the brightness discrimination threshold ΔL and the corresponding brightness L, $\Delta L/L$ is constant over a wide range of luminances. Logarithmic mapping represents this law. According to some more recent experiments, exponential mapping would be more appropriate. These mappings are expensive to compute and always have some free parameters that should be set for each raw image separately. Schlick in [Schl94] introduced uniform rational quantization achieving comparable or better results than exponential or logarithmic mapping with much less computational effort.

5.1 Schlick's Mapping

As stated above Schlick wanted to introduce a mapping function which would act similar to a logarithmic function but which would be much simpler to compute. He proposed the so-called rational mapping function:

$$n = \frac{p \cdot L}{p \cdot L - L + L_{max}} \tag{5.1}$$

This function is intended to account for the non-linearities of both the display device and human perception. The biggest advantages of Schlick's mapping are its speed and automatic selection of the parameter p. The mapping is fast as it needs only one division, one multiplication, one subtraction, and one addition.

A free parameter in logarithmic and exponential mapping depends on many factors such as the adaptation level of the observer, the display device characteristics, human vision rules, etc. It would be ideal to involve all of these factors, but it would be computationally very expensive and some of the human vision

mechanisms are not understood yet. There are methods that include more human vision characteristics, and they will be described in the next sections.

Schlick has made the assumption that the level of the darkest non-black gray is what really changes under different viewing conditions and device settings. The user should find the input level of the darkest non-black gray level M and compute the parameter p according to the following formula:

$$p = \frac{M \cdot L_{max} - M \cdot L_{min}}{N \cdot L_{min} - M \cdot L_{min}} \approx \frac{M \cdot L_{max}}{N \cdot L_{min}} \tag{5.2}$$

The idea is then to map the smallest non-zero pixel value of a raw image to the darkest non-black gray of a display device. In order to find M, Schlick proposes to display squares of different grays randomly on a black background and to select the darkest still recognizable square.

Although this mapping is very fast we find its use limited to raw images with a moderate overall contrast value. Using this mapping with some high contrast raw images led us to unusable results. When we set the parameter p manually the resulting images were great. The reason why results are unsatisfactory for very high contrast images, lies in the fact that the mapping is applied to the whole raw image range. Suppose that there is only one pixel in the raw image that has a very high luminance value L_{max}. This pixel will cause p to explode, and the whole mapping technique will become useless.

This method is ilustrated in color plates 1c, 6a, 10a, 10b and 10f.

A possible solution for high contrast raw images could be to combine Schlick's mapping with the minimum information loss method as described in chapter 6.

Schlick speculated about spatially non uniform mapping in his work as well, but ended up with just another spatially uniform solution, similar to the original one.

5.2 Exponential Mapping

This section will describe an exponential mapping method introduced by Ferschin et al. in [FeTP94]. The main idea of this mapping method is to use an exponential function that will be not affected by a few very dark pixels. This function has actually been developed from the idea of using a stepwise linear mapping function. An exponential function overcomes the problem of discontinuities in the shadows when a stepwise linear function is used. The exponential mapping function is:

$$n = n_{max} \cdot (1 - e^{\frac{-L}{L_{average}}}) \tag{5.3}$$

Figure 5.1 shows the exponential mapping function for a hypothetic raw image with luminances in range $[0, 5000]$, and an average value of 1000. The resulting

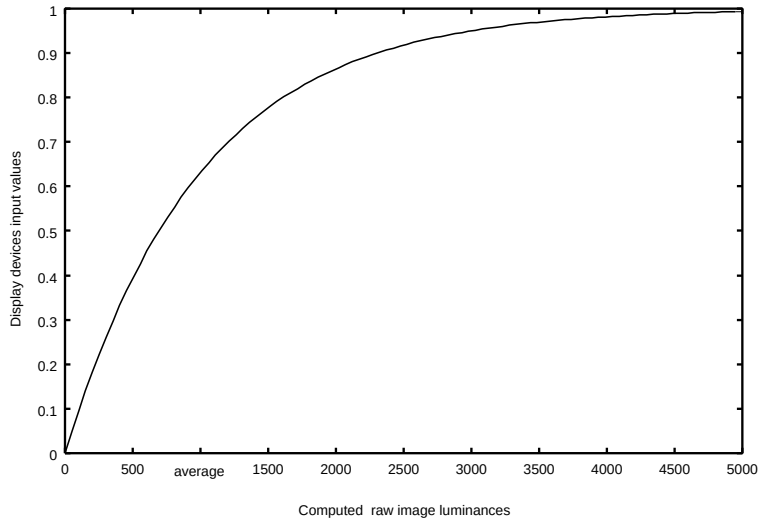


Figure 5.1: Exponential mapping

images provide a smooth transition between all luminance gradients. The authors also introduced an alternative form of compression through the alignment of a reference white point of the radiosity scene with the white-point of the monitor actually used. This really interesting approach is realized by using the radiosity white point in the conversion of the radiosity results to the CIE XYZ, and then using the display device's white point by converting CIE XYZ values to the monitor values.

Authors have also experimented with clipping in CIE LUV and CIE LAB color spaces, and reported no visual differences between all three clippings. Therefore, they suggest using RGB clipping because of its reduced computational demands.

Note that this method is also an average based method. This means that the average value will always be mapped to the 0.632 input value, displaying dark scenes too bright and bright scenes too dark.

The resulting images rendered using this mapping are shown in the results chapter, color plates 1b and 5b.

5.3 Tumblin and Rushmeier's Mapping

This mapping method was first introduced as a technical report in 1991 [TuRu91] and then in 1993 [TuRu93], and is still considered as state of the art in tone mapping. Unfortunately, it is developed for gray scale images only. It is very comprehensive but still not complete. It includes seven different parameters which should

be set by the user, and some of them are not always available to common users. Just as in Ward's mapping, absolute units are required here. Note that it cannot be considered as a drawback, it is just a necessity if the lighting atmosphere is important. If a raw image is rendered in fictitious units, it is just not possible to say if the lighting in the scene is strong or weak.

Tumblin and Rushmeier find the inspiration for their work in the fact that most image synthesis algorithms do not know the difference between night and day - differences that are so obvious to the human eye. They wanted to display the original lighting atmosphere of the raw image in the final image. If the firefly illumination is simulated it should produce a very dark image, and on the other hand if the scene is lit with a very strong anti aircraft search light the final image should be almost completely white. The process of tone reproduction is well known in photography, where the original atmosphere of the scene should be displayed in the photo. Fortunately, computer graphics have unlimited possibilities in choosing the tone reproduction function. It can be very complex, and it is still quite easy to implement. Photographers, on the other hand, are limited with just a few chemicals and photo papers, and can not develop such sophisticated tone operators.

Tumblin and Rushmeier's (TR) mapping technique uses results obtained by Stevens et al. [StSt63] regarding the brightness associated with a luminance at a particular adaptation level. A power law that relates luminance L measured in $lamberts$ to brightness B in $brils$ at an adaptation level of L_a is:

$$B = 10^{\beta_a} \cdot L^{\alpha_a} \tag{5.4}$$

where α_a and β_a are:

$$\alpha_a = 0.4 \cdot log_{10}(L_a) + 2.92 \tag{5.5}$$

$$\beta_a = -0.4 \cdot (log_{10}(L_a))^2 - 2.58 \cdot log_{10}(L_a) + 2.02 \tag{5.6}$$

In the first technical report [TuRu91] the authors used the average value as the adaptation luminance, but in the final version [TuRu93] they used another approach. They used the assumption that the eye adapts in an attempt to keep most brightness near the "brightness constancy" contour of $8.4\ dB'$ [StSt63] below L_a, therefore L_a is:

$$log_{10}(L_a) = E\{log_{10}(L)\} \tag{5.7}$$

where $E\{log_{10}(L)\}$ is the expected value of $log_{10}(L)$.

Although Tumblin and Rushmeier are aware that Stevens's data is not valid for complex viewing conditions and complex scenes (see section Human Vision), they still used this model, and not the improvement suggested by Bartelson et al. [BaBr67], because of lower computational cost, and problems with the root finding of the extended model.

The TR tone mapping operator attempts to match the brightness of the real-world luminance with the brightness of the display luminance. By setting $B_w = B_d$ where B_w represents the brightness of a real-world luminance and B_d the brightness of a display luminance, and using equations 5.4, 5.5, 5.6 as follows:

$$L_d = L_w^{\frac{\alpha_{a(w)}}{\alpha_{a(d)}}} \cdot 10^{\frac{\beta_{a(w)} - \beta_{a(d)}}{\alpha_{a(d)}}} \tag{5.8}$$

Note that $\alpha_{a(d)}$ and $\beta_{a(d)}$ are functions of the adapting luminance of the display $L_{a(d)}$. According to eq. 5.7, L_d should be known in order to compute $L_{a(d)}$. To solve this problem, the authors suggest taking $L_{a(d)}$ as a constant, since it has little influence on $\alpha_{a(d)}$ and $\beta_{a(d)}$. The suggested value is:

$$L_{a(d)} = \frac{L_{d\ max}}{\sqrt{C_{max}}} \tag{5.9}$$

where C_{max} is the maximum available display contrast, depending on the display gamma, ambient illumination and maximum available display luminance $L_{d\ max}$.

Finally the complete tone mapping operator can be written as:

$$n = \left[\frac{L^{\frac{\alpha_{a(w)}}{\alpha_{a(d)}}}}{L_{d\ max}} \cdot 10^{\frac{\beta_{a(w)} - \beta_{a(d)}}{\alpha_{a(d)}}} - \frac{1}{C_{max}} \right]^{\frac{1}{\gamma_{display}}} \tag{5.10}$$

This operator is designed to reproduce overall brightness appearance, and not to reproduce visibility (in contrast to Ward's, Ferwerda's and visibility matching mapping described later). We tested the TR operator and the resulting images look a little bit too dark. We suppose the reason lies in the fact that this mapping operates within the whole range of human vision. The model functions up to luminances of $3.18 \cdot 10^6\ cd/m^2$, and just for comparison, snow covered ground in full sunlight emits $1.6 \cdot 10^4\ cd/m^2$ or horizon sky emits $3.0 \cdot 10^4\ cd/m^2$ on a day with sunlit clouds. As the bright images are reserved for such extreme lighting conditions (search lights), normal light images tend too be a little bit too dark in our opinion. We have applied the TR operator on each color channel separately, just as Ward did in the RADIANCE [Ward94a] package. We are aware that this is not the right way, but finding a better way would certainly exceed the scope of this work.

Nevertheless, if the analysis of extreme lighting conditions is important this could be just the right model. It is complicated to implement this model exactly, but it is worth trying. Resulting images are shown in the results chapter, color plates 9e, 9f, 13c, 13d, 17c, and 17d.

5.4 Model of Visual Adaptation

A model of visual adaptation has been introduced by Ferwerda et al. in [FPSG96]. This model is based on Ward's contrast based scale factor. The difference is in the functions used for just noticeable differences. Ferwerda et al. use the whole set of different functions for scoptic and photopic vision. In the mesotopic range they use a linear combination of scotopic and photopic functions.

Visual acuity is also taken into account by this model, and dark image parts are simply blurred using a Gaussian convolution filter, since the human visual system is not capable of resolving details in dark areas (see Human Vision section).

The temporal aspect of adaptation is also taken into account. It is possible to simulate an adaptation process over time using this model.

As it is necessary to determine adaptation luminances for using this model, the authors suggest to assign display adaptation to half the maximum screen luminance (same as Ward in his contrast based scale-factor), and the world adaptation to half the maximum of the raw image luminance (different approach than used by Ward). This is probably the weakest point of the whole mapping technique. Imagine, for example, one extremely bright pixel at the edge of the image. It certainly will not influence adaptation level that much, as it will seem from this approach.

This is the first, and still the only mapping method that accounts for temporal change in adaptation. Therefore, if such visualisation is needed, this is the only possible choice of tone mapping. Ward et al. developed a new visibility matching operator, that borrows heavily from Ferwerda's model, but the adaptation level determination is much improved. This improved model will be described in more detail next.

5.5 Visibility Matching Tone Reproduction Operator

The tone mapping operator described in this section has not been published yet. Actually, only a short sketch was published at SIGGRAPH '97, and the whole paper is available electronically from the RADIANCE WWW site [RADI97]. We decided to include this mapping in this work as it seems that this mapping will be widely spread in the future. It is based on Ferwerda's mapping, but adaptation level determination is improved. In contrast to all previous methods, that used only one adaptation level for the whole raw image, this mapping uses many adaptation levels, just as our visual system does.

The tone mapping process starts with computing a small image where each pixel corresponds to one visual degree. Actually only center pixels correspond to the 1 degree angle, but the error introduced in the border pixels is neglectable.

The 1 degree area corresponds to the adaptation area of our visual system. This small image will be used to determine adaptation levels at each fixation point. Luminance is converted to brightness simply by taking the logarithmic values of luminances. Once the auxiliary image is generated, a logarithmic histogram is built.

A logarithmic histogram is built in the luminance range from the threshold of human vision ($10^{-4}\ cd/m^2$) or the minimum auxiliary image luminance, whichever is larger, to the maximum auxiliary image luminance.

The next step is histogram equalization. Actually this is not the next step in the fully implemented method, but this is the base of the whole algorithm, and therefore it is described first.

Histogram equalization is a well known technique from image processing. The cumulative frequency distribution $P(b)$ is needed to equalize a histogram. $P(b)$ is defined as:

$$P(b) = \frac{\sum_{b_i<b} f(b_i)}{T} \tag{5.11}$$

where T is the total number of histogram entries, and $P(b_i)$ is the frequency count for the histogram bin at b_i. Later on, the derivative of the above function will be needed as well. The derivative is:

$$\frac{dP(b)}{db} = \frac{f(b)}{T\Delta b} \tag{5.12}$$

where

$$\Delta b = \frac{log(L_{max}) - log(L_{min})}{N}$$

and N is the number of histogram bins, therefore Δb corresponds to the size of each bin. Note that since the cumulative frequency distribution is a numerical integration of the interval, the derivative of the cumulative distribution is the histogram with an appropriate normalization factor.

A common histogram equalization (but here applied on the raw-image) would produce an image where all brightness values have equal probability. The equalization formula can be written as:

$$B_d = log(L_{dmin}) + [log(L_{dmax}) - log(L_{dmin})] \cdot P(B) \tag{5.13}$$

where B_d stands for display brightness, and B for the raw image brightness. The problem with this approach is that the dynamic range will be compressed in the regions where there are only a few samples, and it will be expanded in highly populated regions. This results in an unnatural appearance of the final image, and certainly can not be used as useful tone mapping operator.

Suppose that the 60% of pixels have luminnace lower than $0.1 \cdot L_{dmax}$. In this case, small original luminnce range ($[0, 0.1 \cdot L_{dmax}$) will use 60% of available

display luminances. In order to prevent this unwanted effect, Larson et al. suggest limiting an allowed contrast in a particular region. Actually, although naive histogram equilasation will propose enlarging of the original contrast, it will not be allowed. At the beginning they used a linear ceiling, which can be written as:

$$\frac{dL_d}{dL} \leq \frac{L_d}{L} \tag{5.14}$$

In this way the contrast can not exceed the contrast obtained by using a linear scaling operator. Note that dL_d/dL is actually the slope of the mapping function $L_d = f(L)$. Using this type of ceiling the slope of the mapping function can not exceed that of appropriate linear mapping. From the above equations it is easy to derive following inequality:

$$f(b) \leq \frac{T\Delta b}{log(L_{d\ max}) - log(L_{d\ min})} \tag{5.15}$$

where $f(b)$ is the frequency count for the histogram bin b.

Using this inequality we can be sure that as long as no frequency count exceeds this ceiling, the resulting histogram will not exaggerate contrast. If some histogram bin is too populated, it is simply cut off and the ceiling procedure is repeated iteratively, until there are no more overpopulated histogram bins. This process is called *histogram adjustment* rather than histogram equalization by the authors.

Up to now, there were no human vision characteristics taken into account in this process. The authors say that if the raw image is not rendered in absolute units, a histogram adjustment, as just described, can be applied to the raw image. If absolute units are known, the method can be improved.

In the case of absolute unit raw image, the ceiling function is derived from the just noticeable difference functions (see section Human Vision). The just noticeable difference for adaptation level L_a will be called $\Delta L_t(L_a)$. To guarantee that the display representation does not exhibit contrast that is more noticeable than it would be in the actual scene, the slope of the operator is constrained to the ratio of the two adaptation thresholds for the display and actual scene. Note that this is the same principle introduced by Ward in his contrast based scale-factor [Ward94] and used by Ferwerda et al. in [FPSG96]. In contrast to the earlier approaches, where only one adaptation level was chosen, this constraint will be met at all potentional adaptation levels, making this operator more accurate. The new ceiling is then:

$$\frac{dL_d}{dL} \leq \frac{\Delta L_t(L_d)}{\Delta L_t(L)} \tag{5.16}$$

After a similar derivation as with the linear ceiling a ceiling inequation for $f(b)$ is

defined as:

$$f(b) \leq \frac{\Delta L_t(L_d)}{\Delta L_t(L_w)} \cdot \frac{T \Delta bL}{[log(L_{d\,max}) - log(L_{d\,min})] \cdot L_d} \tag{5.17}$$

Just as before, an iterative process leads to the solution. Applying this sort of ceiling function will produce images that reproduce original visibility on the final image. This operator is further improved by taking the veiling luminance into account (see section Human Vision).

The formulas described in the Human Vision section are used to incorporate veiling luminance. This is the computationally most expensive part of the algorithm, as the sum (integral in the original formulas) has to be computed over all samples for each fixation point using the following equation

$$L_{vi} = 0.087 \cdot \frac{\sum_{j \neq i} \frac{L_j cos(\Theta_{i,j})}{\Theta^2_{i,j}}}{\sum_{j \neq i} \frac{cos(\Theta_{i,j})}{\Theta^2_{i,j}}} \tag{5.18}$$

where L_{vi} is the veiling luminance for fixation point i, L_j is original fixation luminance for fixation point j, and $\Theta_{i,j}$ is the angle between sample i and j (in radians, that is why 0.087 stands at the begining of the eq. 5.18).

Once the veiling luminances L_{vi} are computed, the original luminances L_i are replaced by L_{ai} using the formula derived from eq. 2.22:

$$L_{ai} = 0.913 \cdot L_i + L_{vi} \tag{5.19}$$

Visual acuity is taken into account, as well. The data used was the same used by Ferwerda et al., it is described in the human vision section. The dark image parts are blurred, as the human vision acuity drops with luminance. Ward et al. used the mip-maps approach well known from texture mapping to realize various acuities depending on the absolute luminance levels.

The visibility matching tone operator described in this section, tries to match the visibilities in the original scenes with the display visibilities. Various aspects of human vision were taken into account, and the produced images match the visibilities. On the other hand, the final images look a little bit unusual. Maybe that is because we are not used to images and photos that match the visibility in high range images. This means that if there is a scene with a window on a sunny day, the new operator will display the interior and exterior part, and will try to match the visibility in both parts. It may look unusual to us, as we are used to seeing either the interior or the exterior in such a case (although we can see both in real life).

If the analysis of visibility is crucial in a rendering process (e.g. simulation of emergency lighting) this operator should be used. On the other hand, in everyday use it can sometimes render unusual (that does not mean incorrect) images.

Chapter 6

Minimum Information Loss Methods

This chapter describes the whole family of mapping methods called minimum information loss methods. These methods are one of the contributions of this thesis. We developed these methods in order to determine exposure automatically. The methods follow, in a way, a photographers' approach. The main idea is to place the clipping interval so that a minimum amount of information is lost, thereby preserving the original contrast of all correctly displayed pixels.

Just as in interactive calibration (see section 4.2) a raw image histogram is used in the optimization process.

6.1 Search for the Optimum Contrast Interval

6.1.1 How is it done in Photography?

The method of mean light intensity is the most commonly used method in non-professional photography since in most cases it gives acceptable results. According to [Morv84] the method gave good results in 80% of amateur motifs (the sample was ten thousand motifs). In the remaining 20% the aperture should be shifted up or down to obtain optimal results. E.g., if the main subject is in shadow, an average measure causes the main subject to appear only as a silhouette. For the sake of a display rich in detail, the aperture often has to be opened by 2 or 3 units, compared to the measured mean value. This means that the desired result would be obtained from a mean illumination twice or three times higher on the log_2 scale - i.e. the value 4 or 8 times higher than the average value is displayed as "medium gray". Thus, the clipping window has to be shifted on the log_2 scale up to 3 units compared to values resulting from the average measurement method. For strong

contrast subjects in professional photography, a spot metering is made for all the typical details, that may result in a subject contrast of even several hundreds or more. Then, a contrast interval of 32 has to be selected, which produces the least loss due to the forced clipping. The fundamental method presented in this work follows that idea.

6.1.2 Mean Light Method in Photography

In photography light measurement techniques work with the assumption that the scene contrast is 32. The value two times greater than the average measured lightness will be the white on the picture. The logarithmic middle gray, which is very near to the perceptual middle gray is $0.18 = 2^{-2.5}$ times the lightness of white or $2^{-1.5}$ times the measured average value. All values larger than the white value (two times average) will be clipped to white, and all values below white/32=average/16=black will be clipped to black. In other words, the clipping window on the logarithmic scale is logarithmic middle gray ± 2.5. Let us consider an example from computer graphics with classical mean value linear mapping. We assume a gamma corrected display device with linear response and input range [0,1]. 0 corresponds to the minimum displayable luminance level (in our case let it be 1/32) and 1 corresponds to the maximum displayable luminance value (let it be 1). The values larger than two times the average value will be clipped to 1, as in photography, but there is no clipping for low values. 0 absolute luminance will be displayed as 1/32 in our example. Figure 6.1 illustrates the two approaches. The differences in the final images are not significant because the main difference occurs in dark image parts where our perception in bright surroundings can not perceive them.

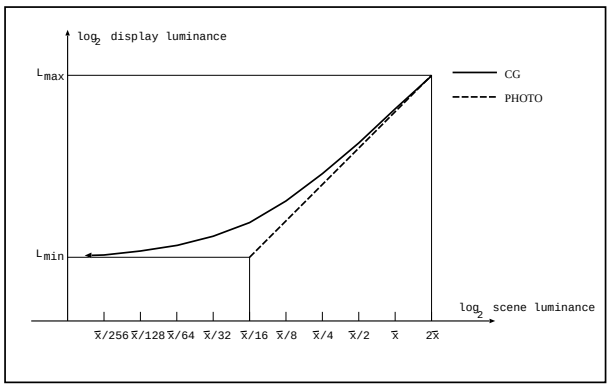


Figure 6.1: Photographic and CG mean value mapping

6.1.3 Main Goal of the Optimization

The main goal of our method is to find the interval $[A, B]$ such that $B = C \cdot A$, where the C is the given clipping contrast, and that a minimum amount of information is lost due to the applied clipping. The clipping contrast depends on the display media, and should equal the maximum displayable medium contrast. Clipping will be done in the most simple way, thus setting values larger than B to B, and those less than A to A. As we want the minimum amount of information to be lost, let us first define what information is. Two approaches will be introduced.

In the first approach each color component is considered as an equally important information unit. So minimizing the information loss means minimizing the total number of color components clipped. Note that the loss of e.g. 3% of color components could mean that 3% of the pixels are affected by clipping (if each of these 3% is affected in all 3 components) or up to 9% of the pixels are affected by clipping (if all of the clipped color components belong to different pixels). Usually there are not many pixels clipped only in one or two components. These are color highlights and saturated colors. As these pixels make up only a tiny part of an average image, the percentage of affected pixels tends to be close to 3%.

In the second approach the pixel is considered to be the essential information unit, so the number of affected pixels shall be minimized. Usually there are no big differences between the optimum intervals in these two approaches, but some difference almost always occurs. The first approach will be called minimum information loss, and the second minimum area loss.

6.1.4 Minimum Information Loss

Building a Logarithmic Histogram

The whole optimization process is done on a logarithmic image histogram. A logarithmic and not a linear histogram is used, because the clipping interval has a constant size for a given contrast on the log scale. Another reason is the possibility of building a logarithmic histogram from high contrast images with a relatively small array. Throughout this chapter log_2 is used, although any other base would yield the same results (the base 2 is chosen in analogy to photography). While forming the histogram, values below a certain value L_{min} $(log_2(L_{min}) = l_{min})$ and above a value $L_{max}(log_2(L_{max}) = l_{max})$ should be clipped to these values, as the log_2 is not defined for 0, and the histogram array is limited for huge luminance values. The histogram is an array, in which each array member holds the number of pixels with the corresponding luminance. The luminance values for each pixel in the raw image are stored in float format, so actually each histogram array member represents luminances in a certain interval. These equidistant histogram

intervals are arbitrarily long, but a finer histogram yields a more precise result. We will define the histogram array as $H[k]$, where $k \in [1, k_{max}]$, the value l_{min} corresponds to $H[1]$ and l_{max} to $H[k_{max}]$. The luminance corresponding to $H[k]$ will be called L_k and the $log_2 L_k = l_k$. The difference of the logarithms of two adjacent grades, the elementary histogram interval δ, is

$$\delta = \frac{l_2 - l_1}{k_{max}} \tag{6.1}$$

The size of the clipping interval for a given clipping contrast C is called $CLIP$ and

$$CLIP = int(log_2(C)/\delta) \tag{6.2}$$

A possible choice for the above values could be: $L_{min} = 2^{-20}$, $l_{min} = -20$, $l_{max} = 20$, $C = 2^5 = 32$, $k_{max} = 8000$, then $d = 0.005$ and $CLIP = 1000$. The histogram array is initialized to zero, and then for each color component of all raw image pixels, $H[k]$ is incremented, where

$$k = int((log_2 d - l_{min})/\delta) \tag{6.3}$$

and d stands for one r, g, or b color component. Note that the raw image does not need to be stored. The histogram can be built simultaneously with the calculation of pixel luminances. Note also that the number of histogram entries (the sum of all $H[k]$) is three times the number of pixels in the raw image. After the histogram is formed, the optimum clipping interval has to be found.

Search for the Optimum Clipping Interval

All we have to do is to apply a discrete algorithm on the histogram formed in the previous chapter. The length of the interval is CLIP, and we can find its position by shifting the interval along the complete histogram in discrete steps of one. In each step the error sum is increased by the outgoing histogram member and decreased by the incoming one. The pseudo code is given in algorithm 1.

```
error :=0;
for i := CLIP + 1 to kmax do
    error += H[i];
leasterror := error;
best := 1;
for j := 1 to kmax - CLIP do
    begin
       error +=H[j];
       error -= H[j + CLIP];
       if error <= leasterror then
           begin
              leasterror :=error;
              best := j + 1
           end
    end
```

Algorithm 1

The optimum clipping interval starts at $H[best]$, and ends at $H[best+CLIP-1]$. The algorithm runs in linear time $O(k_{max})$. The actual $[a, b]$ interval is then:

$$a = l_{min} + best \cdot \delta \tag{6.4}$$

$$b = a + c \tag{6.5}$$

or in linear scale:

$$A = 2^a \tag{6.6}$$

$$B = A \cdot C \tag{6.7}$$

Now clipping can be done at $[A, B]$ and the final image can be generated. Figure 6.2 shows a logarithmic histogram of the images shown in color plates 1 and 2, and the optimum clipping interval for contrast C=50.

Clipping Error

The simplest is to consider all clipped pixels as contributing to the error equally, as is done in algorithm 1. This means applying a 0-1 type discrete error function. Obviously it is not really the same if a pixel with a value just slightly above B, or one with a value very far from B is clipped to B. The problem can be solved by applying an error function linearly penalizing the distance from B (or A for

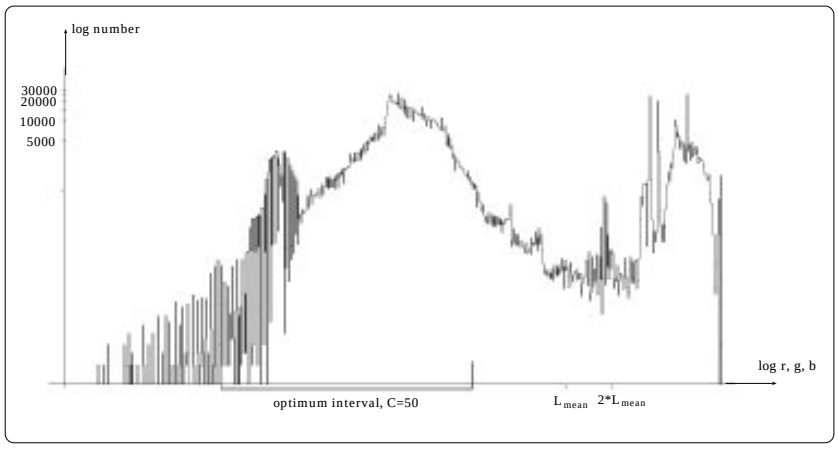


Figure 6.2: Logarithmic histogram with optimum interval of C=50

the other side of the interval). A combination of these two error functions, a value limited alternative of the latter can be used as well. All functions are shown in fig. 6.3. Although it seems that applying an error function which is not the discrete 0-1 function can increase the complexity and run time, we will show how an error function of type c in fig. 6.3 can be applied, and the process still runs in linear time (this algorithm was suggested by Attila Neumann).

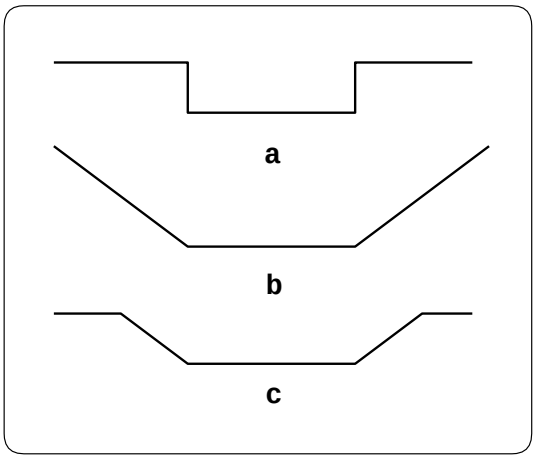


Figure 6.3: Various types of error functions

Let us define the error function as in figure 6.4. $error(a, k)$ gives the error

weight of position k for the clipping interval $[a, b]$, where $b = a + CLIP$.

$$error(a,k) = \begin{cases} 0 & \text{for } k \in [a, b-1] \\ G & \text{for } k \in [1, a-d-1] \cup [b+d, k_{max}] \\ (a-k) \cdot \frac{G}{d+1} & \text{for } k \in [b, b+d-1] \\ (k-b) \cdot \frac{G}{d+1} & \text{for } k \in [a-d, a-1] \end{cases} \tag{6.8}$$

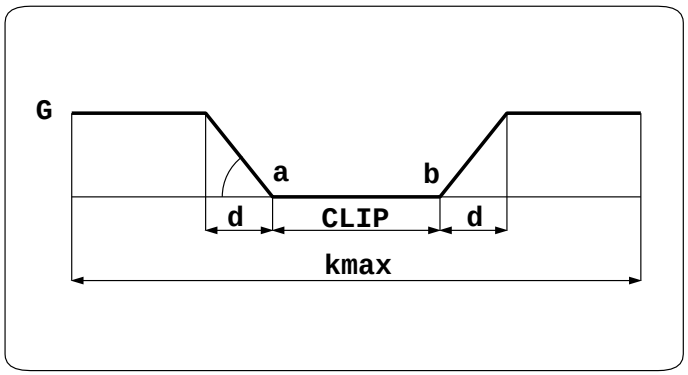


Figure 6.4: Error function

for given G and d:

$$g = \frac{G}{d+1} \tag{6.9}$$

We need one additional precomputed array, $e[a]$. The array e contains influences on the error for each step. It is defined recursively. First let us define $f(a)$ as:

$$f(a) = \begin{cases} 0 & \text{for } a \leq 0 \\ f(a-1) + H[a] & \text{for } 1 \leq a \leq k_{max} \\ f(k_{max}) & \text{for } a > k_{max} \end{cases} \tag{6.10}$$

and e[0] as:

$$e[0] = \sum_{i=1}^{k_{\max}} error(0,i) = \sum_{CLIP+1}^{CLIP+d} (i - CLIP) \cdot G/d \cdot H[i] + G \cdot [f(k_{max}) - f(CLIP+d)] \tag{6.11}$$

Note that $e[0]$ corresponds to the starting position as seen in fig. 6.5.
We will define $\Delta e[a+1]$ as:

$$\Delta e[a+1] = e[a+1] - e[a] \tag{6.12}$$

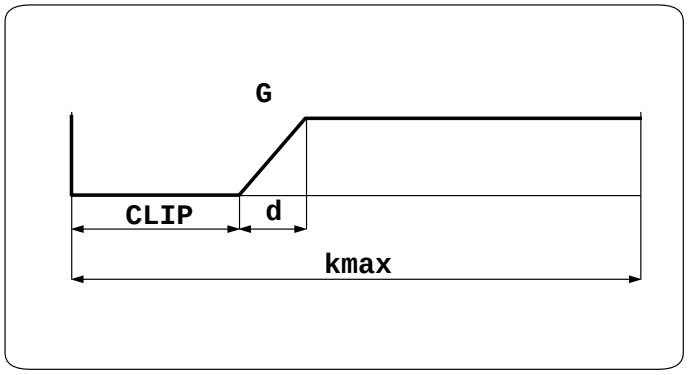


Figure 6.5: Starting position

$$e[a] = e[a-1] + g \cdot [f(a+1) - f(a-d)] - g \cdot [f(a+CLIP+d+1) - f(a+CLIP)] \tag{6.13}$$

Once the $e[a]$ has been computed, all that has to be done, is to find the minimal $e[a]$ for $a \in [0, k_{max} - CLIP]$, and a is then the starting point of the interval.

6.1.5 Minimum Area Loss

In this section we are going to explain how the clipped image area can be minimized. Our goal is again to find an $[A, B]$ interval, of known contrast C, and clipping all color components outside $[A, B]$ just as in the previous chapters. Instead of the logarithmic histogram, we should use a 3D representation, to preserve the information to which pixel each color component belongs. We will use a discrete space, just as before, and we will use a logarithmic axis for the same reason we used a logarithmic histogram. The problem of finding the interval now becomes the problem of finding the cube which contains the most pixels. Note that the cube can only be shifted along the line r=g=b, as we have to apply the same $[A, B]$ interval to all color components. The problem is illustrated in figure 6.6. Although it might seem to be a complex problem from computational geometry it can be solved in linear time. First note that all pixels that lie outside the sweep formed by moving the cube cannot ever be contained in the cube. Such pixels can immediately be considered as error-pixels.

The idea of the algorithm is to pretabulate the influence of every pixel on all steps. This is done quite simply with two lists. Let's call them inclusion and exclusion lists. The inclusion list contains the effect on the total error if the cube is moved forward, so that the pixel is just included in the cube, and the exclusion list contains the effect on the total error when the pixel just leaves the cube. The pseudo code is given in algorithm 2.

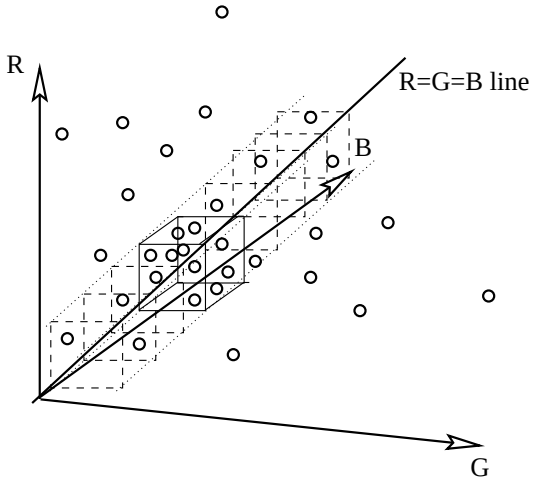


Figure 6.6: Minimum area loss idea

Pixels outside the cube are considered as equally weighted error-pixels whether they are near the cube or very far away. The inclusion of various error functions is straight forward, and although the complexity and the run time are increased, it can be still done in a linear time.

6.1.6 Mapping of the Interval

Once the optimum clipping interval is found it should be mapped to the display device's input values. If the clipping contrast equals the display device's contrast a linear mapping should be used. The implementation is straightforward. First all values less than A are set to be equal A, and all values greater than B, are set to be equal B. Now, the $[A, B]$ interval should be linearly mapped to the $[0, 1]$ input interval.

```
ext_error := 0;
int_error := 0;
error := 0;
for all pixels do
    begin
       if max(r,g,b) - min(r,g,b) <= CLIP then
           begin
              I[max(r,g,b)]++;
              E[min(r,g,b)]++;
           end
       else
           ext_error++;
    end
for i:= CLIP + 1 to kmax do
    int_error := int_error + I[i];
least_error := int_error;
best := 0;
for j := 1 to kmax - CLIP do
    begin
       int_error -=I[j + CLIP - 1];
       int_error += E[j -1];
       if int_error < least_error then
           begin
              least_error :=int_error;
              best :=j;
           end
    end
error :=least_error + ext_error;
```

Algorithm 2

6.1.7 Limited Information Loss

Up to now we have proposed a fixed contrast, and we have found a clipping interval such that the lost information is minimized. Sometimes, for high contrast images, the information loss will be too high. In this case, we propose to bound the information loss, and to find the smallest clipping interval that causes no more

than the allowed information loss. As the clipping contrast will be higher than the display device contrast in this case, a linear mapping will not function properly any more. In order to match the human vision characteristics, we propose to apply Schlick's mapping on the clipping interval. One of the biggest disadvantages of Schlick's mapping technique is the explosion of the parameter p if the total image contrast is too high. If there are only few very, very bright pixels in the image, they increase the overall contrast, and the parameter p explodes. If our limited error loss scheme is applied, for a proposed error (i.e. 10%[1]) the clipping contrast will be smaller than the overall contrast, and Schlick's mapping will produce good results. If there are few very bright or very dark pixels in the image they will be clipped and will not contribute to the computation of the parameter p. Note that almost the same results would be obtained if a linear mapping on the log scale would be applied. We recommend Schlick's mapping due to its lower computational cost, and its ability to adjust the final image according to the display media characteristic (the least non-black input level). The following equation describes Schlick's mapping:

$$n = \frac{p \cdot L}{p \cdot L - L + L_{max}} \tag{6.14}$$

where n is the input level, $n \in [0, 1]$, L is the luminance value and p is:

$$p = \frac{M \cdot L_{max} - M \cdot L_{min}}{N \cdot L_{min} - M \cdot L_{min}} \approx \frac{M \cdot L_{max}}{N \cdot L_{min}} \tag{6.15}$$

where M is the smallest non-black input level, and N is the number of input levels. In order to find M, Schlick proposes to display squares of different grays randomly on a black background and to select the darkest still recognizable square. Of course values L_{min} and L_{max} in eq. 6.14 should be replaced with A and B in our case.

Images mapped using these methods are shown in the results chapter, color plates 1d, 1e, 1f, 6b, 7a, 7b, 8a, and 8b.

[1]The allowed error is an arbitrary chosen small percentage. From our experience, results will be satisfactory in most cases if the error is less than 10%.

Chapter 7

Incident Light Metering

All mapping methods described so far were based on raw image radiances only. As there is no information on objects' reflectances in the raw image, bright and dark objects can not be distinguished. The method described in this chapter, incident light metering, makes it possible to reproduce original objects' colors. That means if, e.g., a dark blue color is assigned to an object in the modelling phase, it will be displayed as dark blue in the final image, even if the whole scene is composed of dark objects (that is the case when other methods would fail, as described later). The inspiration for this method came from the incident light metering used in professional photography and the movie industry.

7.1 Light Metering in Photography

As stated before, the method to be introduced relies on photography analogies. There are two ways how light can be measured in photography:

The first is incident light metering. It was used at the beginning of the photography era by portrait photographers. The main idea is to measure the light falling on the subject that is photographed. The incident light meter is placed at the subject position and aimed to the camera (see fig. 7.1). Portrait photographers had the possibility of walking up to the subject, measuring the light, and then walking back to the camera to adjust the exposure. Actually, this is still done in professional photography, and in the movie industry, where the incident light is measured even outdoors via fill panels, and stand-ins are paid to stand around the scene and be metered for light adjustment purposes (that is the reason why they are called stand-ins).

For landscape photographers it is not possible to walk to the subject to measure the light, so they use reflected light metering only, where they measure the light coming to the camera from the subject direction (fig. 7.1). When people

started to take photos of moving targets, it was also more convenient to measure the light from the camera position. Unfortunately, in contrast to incident light metering, reflected light metering is not independent of subject reflectivity or BRDFs of the scene, and the relative preponderance of light or dark areas in the scene. This is still the biggest drawback of reflected light metering. Nowadays, however, in spite of all its drawbacks, it is used more often than incident light metering, since a lot of improvements in reflected light metering have been introduced. All these improvements try to compensate for the subject variations over the imaginary average gray scene.

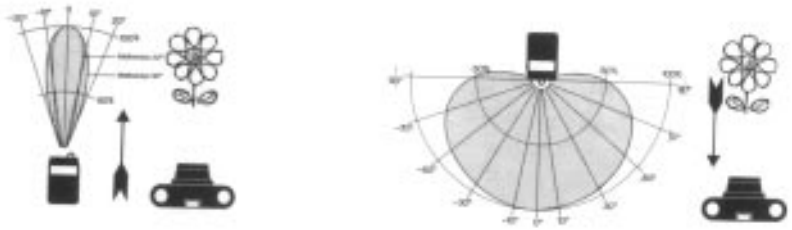


Figure 7.1: Reflected and incident light metering

Figure 7.2 shows a light meter that can be used for reflected and incident light metering. Note the white hemisphere that acts as a half-space integral.

7.2 Incident Light Metering in Computer Graphics

In spite of its inability to properly display bright or dark scenes, reflected light metering has been used in computer graphics exclusively up to now. Maybe the computer graphics community has been influenced by the fact that this is the dominant method in photography. It is certainly not convenient for a photographer to measure the incident light in all cases (and sometimes it is simply impossible). However, in computer generated images it can be achieved quite simply, as will be explained now. We will explain the method for the monochrome (or black and white) case in detail first. The extension to the color case is straight forward and described in the appendix.

The main idea is to place a limited number of diffusors into the scene. A diffusor is a half-space integrator used to measure irradiance. In computer graphics

Figure 7.2: Incident and reflected light meter

a perfectly diffuse surface can be used as the diffusor. Actually, the outgoing radiance of a surface with perfectly Lambertian BRDF is proportional to the surface's irradiance.

Usually only one diffusor is used in photography, as the photographer knows the subject of interest in the scene. For computer generated images more than one diffusor will be used, but the number of diffusors is still low compared to the number of patches or pixels in the final image. We suggest using e.g. $8 \times 8 \ldots 32 \times 32$ diffusors in the scene. The number of samples required for incident light measuring is lower than the number for measuring reflected light, because the variance of incident light is much lower. Due to the lower variance, the contrast of irradiance values is also lower than the contrast of radiance values.

Of course it is possible to allow the user to place diffusors interactively into the scene, but our intention was to create an automatic method. Once the diffusors have been placed, the irradiance E of each diffusor should be computed as:

$$E = \int_{\Omega} L(d^{in}) \cdot cos\Theta^{in} d\omega \tag{7.1}$$

where Θ^{in} is the incident angle,$[\Theta^{\text{in}}]$ = rad, ω is a solid angle, $[\omega]$ = sr, Ω is the hemisphere, $L(d^{\text{in}})$ is the radiance in the d incoming direction, $[L] = Wm^{-2}sr^{-1}$

In general, for an arbitrary BRDF $\rho(d^{\text{in}}, d^{\text{out}})$ the outgoing radiance for distributed light sources is:

$$L(d^{out}) = \int_{\Omega} L(d^{in}) \cdot \rho(d^{in}, d^{out}) \cdot cos\Theta^{in} d\omega \tag{7.2}$$

For an ideal Lambertian BRDF

$$\rho(L, V) = const. = \frac{a}{\pi}[sr^{-1}] \tag{7.3}$$

where a is the dimensionless albedo or reflectivity, $a \in [0, 1]$. That means that:

$$0 \leq \rho(L, V) \leq \frac{1}{\pi} \tag{7.4}$$

Inserting a Lambertian BRDF in (7.2) and using (7.1) gives:

$$L^{out} = a \cdot \frac{E}{\pi} \tag{7.5}$$

where $L^{out} = L(d^{out})$ and has the same value for each d^{out} direction of the half space.

In the implementation methods to be described later, diffusors will be realized as perfectly Lambertian, white, elementary metering surfaces, each with the same albedo value: $a = 1$. For the sake of simplicity, let us first assume that the lighting is totally homogeneous, i.e. each diffusor receives the same irradiance value E. Therefore, all of the metering surfaces are the same, corresponding to (7.5):

$$L^{out}_{meter} = \frac{E}{\pi} \tag{7.6}$$

The linear scale factor m used to map the original radiance image is then:

$$m = \frac{1}{L^{out}_{meter}} = \frac{\pi}{E} \tag{7.7}$$

Different surfaces in the scene can have different albedo values. How will they be displayed using the scale factor m? According to (7.5) each surface emits the radiance $L^{out} = a \cdot E/\pi$. Multiplying this radiance with m (7.7), results in exactly the albedo "a" being received for displaying. This is the exact, desired, and correct solution!

In the case of inhomogeneous lighting there is a set of positions with different E_{meter} (and L^{out}_{meter}) values. From this set, a representative value should be produced to be used as an appropriate scale factor. Our first idea was to use the average value of a truncated histogram of the metering radiances (e.g. the lowest x % and the highest x % of the histogram values are clipped, with an x between 3 and 10), because some very dark or very light areas in the image can greatly influence the average value. The value of x is completely arbitrarily chosen and can be increased in order to eliminate the excessive influence of a dark or light

part of the image. To eliminate this effect we use the median value, where $x = 50$ %. In this general case the scale factor m is:

$$m = \frac{1}{L^{out}_{median}} = \frac{\pi}{E_{median}} \tag{7.8}$$

where L^{out}_{median} is the median value of the L^{out}_{meter} values of the diffusor set.

In the case of inhomogeneous lighting, this method does not produce the exact albedo value "a" for diffuse surfaces. However, the method offers a good compromise, in general a value near to the original "a". Sometimes the resulting device input value will exceed 1, in this case we propose to simply clip such values to 1.

Note that this is a two pass method. In the first pass a small irradiance image is rendered and the scale factor m is found. In the second pass the original image is rendered in the full resolution and this raw image is then mapped using the scale factor m from the first pass. Of course, there are no diffusors in the original scene used to render the final image.

The irradiance computation is included in some rendering packages (e.g. RADIANCE [Ward94a]). The irradiance calculation is often used for lighting engineering purposes. It is interesting that, in spite of this fact, up to now irradiances were not used to simulate incident light metering in computer graphics. In the next chapters we are going to explain how irradiance computation can be implemented in most rendering software that uses any kind of ray tracing or radiosity.

7.3 Irradiance Computation

7.3.1 Simple Ray Tracing without Interreflections

The method will first be explained for simple ray tracing without interreflections. This is certainly not a physically plausible case, but due to its low computational cost, it is often used in real time applications (e.g. virtual reality) and it is widely spread in commercially available software. Generally, the diffusors can be placed in the scene using a regular grid on the projection plane. This grid resolution can be significantly coarser than the image resolution. The first intersections with scene objects are computed from the view point, and at these intersection points the diffusors are placed, with the same normal as the intersected object's surface. Diffusors are white, perfectly diffuse surface elements, used only for metering purposes, and are not visible in the final image.

If no interreflections are taken into account, the irradiances are influenced only by direct illumination from the light sources (sometimes, when an "ambient term" is added, it should be added to irradiance values also, simply by adding a constant

to all irradiance values). According to (7.1) irradiance can be expressed as

$$E = C_{\text{amb}} + \frac{1}{r \cdot \pi} \cdot \sum_{k=1}^{M} \frac{P_k}{r_k^2} \cdot cos\Theta_k \tag{7.9}$$

where M is the number of point light sources, P_k is the power of the light source k, r_k is the distance between the light source k and the diffusor, Θ_k is the incident angle, C_{amb} can be estimated from the average reflectivity.

It can be shown that a view dependent ambient term can also be used for non diffuse BRDFs, this is the topic of forthcoming work.

The irradiances are determined for a very low resolution image, which is not displayed. Of course, it is possible to render the "irradiance image" in high resolution to illustrate the irradiance distribution in the scene. For illustration purposes, instead of using white metering elements, all scene surfaces are set to medium gray (albedo = 0.5) Lambertian surfaces instead of using their original BRDFs. We call the result of rendering such a scene a cement image (see Color plates 11b, 15a and 15b). Note that this cement image is completely independent of the original BRDFs, so it is the same for various scene attribute settings. It depends only on direct lighting, as interreflections are not taken into consideration, yet.

7.3.2 Distribution Ray Tracing

In distribution ray tracing [Glas95], interreflection effects are additionally taken into account. To compute the irradiances in this case, again a low resolution image will be used for the metering. The diffusors are again placed into the scene according to a coarse regular grid on the projection plane. Diffusors are, just as before, elementary, white, perfectly diffuse surfaces, not displayed in the final image.

Let us assume a distribution ray tracing method using multiple reflected single rays starting from the view point. Now we place a white diffusor metering element at each first intersection surface point (see fig. 7.3). Each of these individual diffusors gathers the irradiance from its half space caused by direct lighting and by interreflections. Interreflections are realized with single rays starting from the actual elementary metering surface according to a cosine distribution. More precisely, by this rendering calculation the correct values (see eq. 7.2) are estimated. From these values the median value is determined and used to compute the scale factor m (see eq. 7.8) which is then applied to the original radiance image.

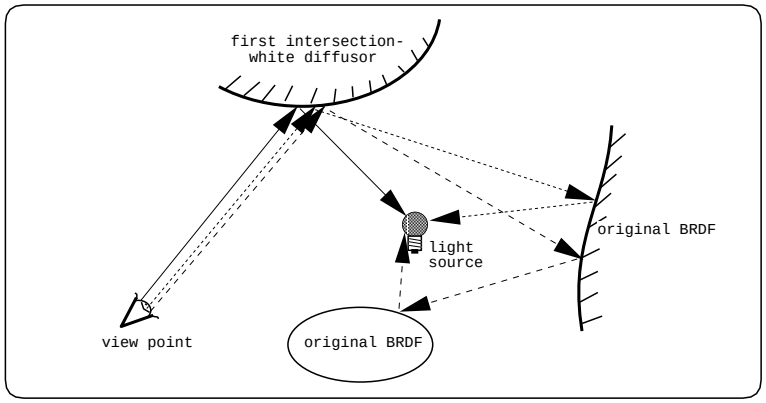


Figure 7.3: Irradiance metering with distribution ray tracing

7.3.3 Radiosity

The incident light metering method can be easily implemented using radiosity as well. During the final gathering step a limited number of scene patches should be replaced by auxiliary light metering patches. These auxiliary patches are perfectly diffuse white patches that are taken into account only during the final gathering step. In this way their radiosity is proportional to the irradiance and these radiosity values can be used for finding the scale factor. Of course, auxiliary patches are not displayed in the final image, they are just used to find the scale factor. Note that the predominantly used diffuse radiosity method does not use radiance values, but radiosity, B, which is π times radiance. In this case the eq. (7.5) will be:

$$B = a \cdot E \tag{7.10}$$

and the scale factor will be:

$$m = \frac{1}{B^{meter}_{median}} \tag{7.11}$$

7.4 Color Case

The incident light metering has so far been described for the monochrome case. In reality the diffusors gather colored light from half spaces. Color light can either be caused by color light sources or can be the result of interreflections at ”color” BRDFs.

The color case should include the monochrome case as a special case (here monochrome means that $r = g = b$, or in the general case the spectral reflectivity $\rho(\lambda_i) = const.$), as neutral color and lighting. The results should then be the same as in the monochrome case.

It is possible to introduce more color models for irradiance computation which satisfy these conditions. These methods are very similar under average lighting conditions, but they can differ a lot when used in highly saturated color lighting (e.g. disco, color neon, theater, fireworks lighting etc.).

Our intention for the color case is to find a scalar value f that represents the irradiance value of the full spectrum. When using a simple 3 channel model (for lighting and for BRDFs) for a given rgb triplet this value f can be: $f(r,g,b) = Y(r) + Y(g) + Y(b)$, where Y represents the CIE Y value which corresponds to the relative lightness and has a maximum value of 100. But also other functions like $f(r,g,b) = (r+g+b)/3$ or $f(r,g,b) = max(r,g,b)$ are appropriate for the neutral color. In the last two models the neutral color is represented by an rgb triplet where $r = g = b$. Which formula from these three should be used? The Y approach is widely used in photography, in spite of its drawbacks. As stated before this approach gives good results for average lighting color conditions. Figure 7.4 shows the ideal (human eye) and the real spectral sensitivity curve of silicon blue cells used in an incident light meter. To compare the three suggested scalar

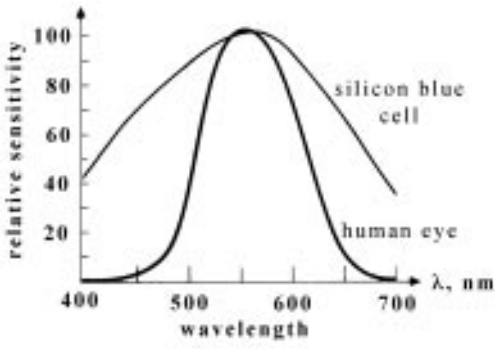


Figure 7.4: Spectral sensitivity of human eye and Si blue cell

functions $f(r,g,b)$ we take a sample scene that consists of a homogeneously lit square, which lies normal to the viewing direction. One quarter of this square is medium red, i.e. $Y(r) = 0.5 \cdot 20 = 10$, and $(r,g,b) = (0.5,0,0)$, the second quarter is medium green, i.e. $Y(g) = 0.5 \cdot 70 = 35$, $(r,g,b) = (0,0.5,0)$, the third quarter is medium blue, i.e. $Y(b) = 0.5 \cdot 10 = 5$, $(r,g,b) = (0,0,0.5)$, and finally the last quarter is neutral, medium gray, i.e. $Y(n) = 0.5 \cdot 100 = 50$, $(r,g,b) = (0.5,0.5,05)$.

For neutral homogeneous lighting $(r,g,b) = (1,1,1)$ irradiance metering results in a neutral color which should only be scaled with $Y = 100$ or $(r,g,b) = (1,1,1)$ (for the sake of simplicity, the irradiance to radiance conversion will not

be explained, we only want to illustrate the original color shift). The resulting colors will be the same as the original, medium red, medium green, medium blue and medium gray.

Let us see what happens for a blue lighting $(0, 0, 1)$. With light metering we will measure an irradiance of Y=10, so we should use a scaling factor of $100/10 = 10$, which will make all colors highly (10 times) over-exposed. More precisely, the red and green square quarters will be black as they have no blue component, but the blue and gray square quarters will result in $(r, g, b) = (0, 0, 5)$ which will be displayed as $(0, 0, 1)$ after the clipping instead of the correct $(0, 0, 0.5)$.

For green lighting, over-exposure will still occur, with $(r, g, b) = (0, 0.71, 0)$ for the green and gray square quarter, since green has the biggest Y value, and blue the smallest (see fig. 7.4).

When $f(r, g, b) = (r+g+b)/3$ is used as an irradiance measuring formula, for neutral (white) light the result is the same as above. For green and blue lighting the resulting values are the same (only the corresponding components are not). For this formula, green will be lighter than before, but blue will not be so highly over-exposed. The scale factor will be 3 for both lightings, so instead of the correct 0.5 the corresponding components will be 1.5, i.e. 1 after clipping. The zeros will remain zeros. It is obvious that the above two methods sometimes produce significant errors.

Finally, the third function $f(r, g, b) = max(r, g, b)$ will be examined. The idea is to compute the (r, g, b) irradiance value for each diffusor and to store the maximum value for later median determination. In the above example, with white, green and blue lighting the new (fictive) irradiance value will be 1, and the scale factor will be 1 as well. For white lighting the results are correct just as before. With blue lighting the red and green square quarters will be black (which is the correct result), and the medium blue and the medium gray square quarters will be correctly displayed as $(0, 0, 0.5)$ medium blue.

The results will also be correct for all square quarters with green and red lightings. We suggest the reader finds scale factors for various combinations of lighting (yellow, purple, etc.) and for various color samples. This method always gives correct or significantly better results than the first two approaches.

In the general case when there are K ($K > 3$) discrete wavelengths, the appropriate (r, g, b) triplet should be computed first. This can be done by computing CIE XYZ coordinates first, and then converting them to r, g, b.

Sometimes one of the r, g, or b values can become negative, which means that the color is outside of the device's gamut. At least one of the r, g, b values will always be positive, however, so the selected maximum $f(r, g, b) = max(r, g, b)$ (fictive) irradiance is always positive.

For neutral lighting the new maximum method gives the same result, for unsaturated lighting it gives a similar result and for saturated lighting it gives signif-

icantly better results than the ”Y approach” widely used in photography.

The incident light metering method is ilustrated in the results chapter, color plates 9, 10, 11, 12, 13, 14, 15, 16, and 17.

7.5 Conclusion and Future Work

A new approach of mapping luminance values to display devices has been introduced. The completely new idea of applying the incident light metering method in computer graphics has been described. Just as in photography, it gives good results for average scenes, and for complicated lighting condition scenes as well. It shows its strength even where usual methods based on reflected light metering fail. The new method displays bright objects as really bright, and dark objects as really dark, independent of the average reflectivity of the scene. All colors are reproduced close to the originally selected patterns using only a simple linear scale factor which is easy to compute. It works well for high contrast (e.g. back lit) scenes, too. The method can be used with or without absolute units. The lighting atmosphere of the whole scene can be changed if the scale factor m is multiplied with a constant c, where e.g. $c \in [1/2, 2]$. In this way, irrespective of real irradiances, it is possible to give an image a brighter or darker lighting atmosphere.

An important topic for future work is the use of absolute units in irradiance metering. It would be interesting to combine the incident light metering with the work done by Tumblin and Rushmeier [TuRu93], combining two important human vision characteristics this way.

Chapter 8

Color Image Difference

A good image metric is often needed in digital image synthesis. It can be used to check the convergence behavior in progressive methods, or to compare images rendered using various rendering methods. If images are rendered using various levels of detail (LOD) it is very important to evaluate if a certain LOD image is sufficient. Furthermore lossy compression methods should also be evaluated somehow. Reproductions on various media should be compared as well. Finally, an image query problem, when the most similar image to a target image is seeked in the large image data-base, happens to be another image metric application.

Of course, it is possible to compare two images by averaging pixel by pixel differences. Unfortunately, human vision does not compare images this way, therefore the results differ significantly from human comparison. As digital images are in most cases rendered in order to be observed by humans, a metric should correspond in some way to human vision. More precisely, if a human observer would state that the distance between images A and B is greater than that between images B and C, we expect a metric to give the same results. Comparing images with the popular mean square error metric (MSE) produces results that can differ greatly from human evaluation [Giro93]. Our intention is not to give the final solution for a perceptual metric, but rather to offer a simple, efficient way of how images can be compared in the original space with one single number.

More complex comparisons transform the image in the Fourier [RWPSR95] or wavelet [GaMY97] space and perform the comparison there. Rushmeier et al. [RWPSR95] introduce various techniques for comparing luminance images. They use some ideas from image compression and develop new metrics. All these metrics are computed after the images are transformed to Fourier space, and the whole method is designed for luminance images. As luminance images contain no information on color, these metrics obviously fail for color images, they are intended to be used for gray scale images only.

Jacobs et al. [JaFS95] have introduced a very fast multi-resolution querying

method, which is intended for a different purpose. Their work is based on the wavelet transform.

Gaddipati et al. [GaMY97] have introduced a wavelet based metric. In order to compute some coefficients used by this metric, images should be transformed to the Fourier space first, which makes this metric very computationally expensive. The whole metric operates on the separate CIE LUV components, actually only the L^* component of the CIE LUV space was used by the authors, and they did not suggest how other components could be used and combined with the L^* measure.

We are going to offer a solution for color images in original space using some human vision characteristics. Of course, it is possible to compute the luminance of each pixel and then to use a metric for luminance images, but as stated before, some color differences would be lost. Our idea is to find the weighted average of color differences of an appropriate set of "area pairs". The areas will be rectangles, which are quasi-randomly defined in the image. The results of various sizes and numbers of rectangles are combined.

The contrast sensitivity function (see Human vision section) as suggested by Manos and Sacrison [MaSa74] will be used as the weighting function. The same function is used by Rushmeier et al. [RWPSR95] and Gaddipati et al. [GaMY97].

8.1 Contrast Sensitivity Function

The contrast sensitivity function described in the Human vision section is given for frequencies in cycles/degree. As we want to compute our metric in image space, frequencies should be converted from cycles/degree to pixels/degree. This conversion is simple and follows straight forward from the viewing geometry for a given viewing distance. Actually the whole metric is viewing distance dependent. (The other above mentioned metrics are also viewing distance dependent, although it is not explicitly mentioned in the above papers). Consider, once more, our picture with the black and white stripes. Let the stripes be 1 cm wide now. You will agree, the stripes will be clearly and sharply distinguishable if the image is observed from 50 cm. But what happens if the viewing distance is, let's say, 100 meters. The image becomes gray again. Let us derive the cycles/degree to pixels/degree conversion now. Actually we are interested in how many pixels contains one visual degree. We will denote the viewing distance given in cm as d, the display size in cm as W, the display resolution in pixels as R, the width of the display portion covered by 1 visual degree given in cm as w, and the number of pixels in w as r (see fig. 8.1).

From simple geometry follows:

$$w = 2 \cdot d \cdot tan\frac{\alpha}{2} \tag{8.1}$$

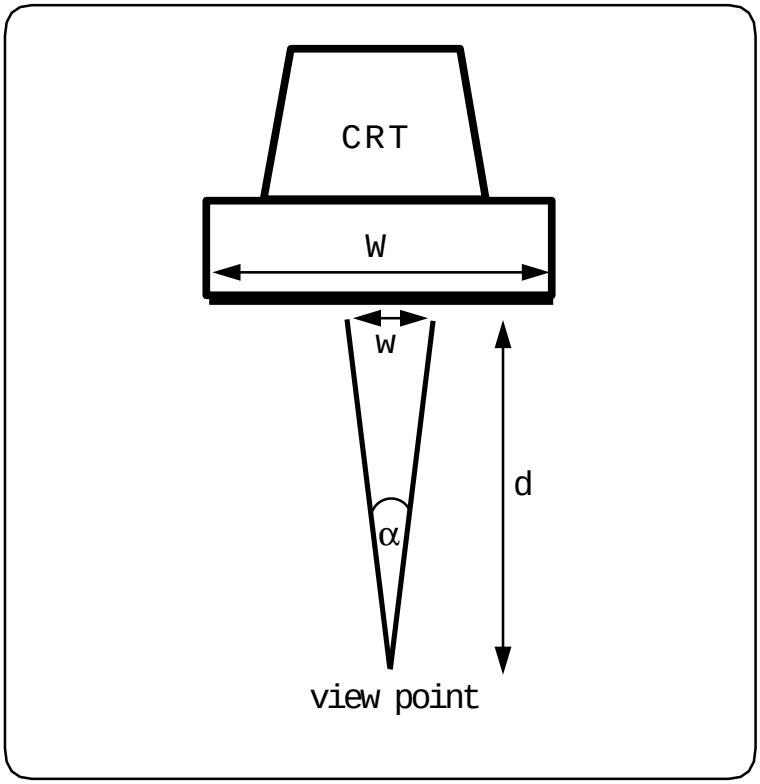


Figure 8.1: Viewing geometry

$$r = \frac{w}{W} \cdot R = \frac{2 \cdot R \cdot d \cdot tan\frac{\alpha}{2}}{W} \tag{8.2}$$

Now, according to the Shannon sampling theorem [Glas95], the maximum frequency that can be realized using r pixels is $r/2$ cycles.

Let us illustrate this conversion using the next example: Our display device is a CRT monitor. The width of the monitor W is 34 cm. The display resolution is 1280 pixels, and the viewing distance is 50 cm. We are interested in the number of pixels r contained in one ($\alpha = 1^\circ$) visual degree. According to (8.2) r is:

$$r = \frac{2 \cdot R \cdot d \cdot tan\frac{\alpha}{2}}{W} = \frac{2 \cdot 1280 \cdot 50 \cdot tan0.5^\circ}{34} = 32.854 \tag{8.3}$$

Since the maximum displayable frequency is $r/2$, frequencies above 16.427 cycles/degree cannot be displayed. They represent the sub-pixel range (see fig. 8.2). If we want to exploit our visual system to its maximum we should move the maximum frequency toward 60 (remember that the contrast sensitivity function is practically 0 above $60\ cycles/degree$, see section 2.3.5 Contrast Sensitivity Function),

either by increasing the viewing distance d, or resolution R (see equation 8.2). If the distance d is increased further, such that the maximum frequency goes above 60, we are not able to see the individual pixels any more. Larger pixel areas will become the essential image elements now. Note that if the changes on the pixel basis are important we can decrease the viewing distance d in order to move the maximum frequency near the peak of the contrast sensitivity function $A(f)$.

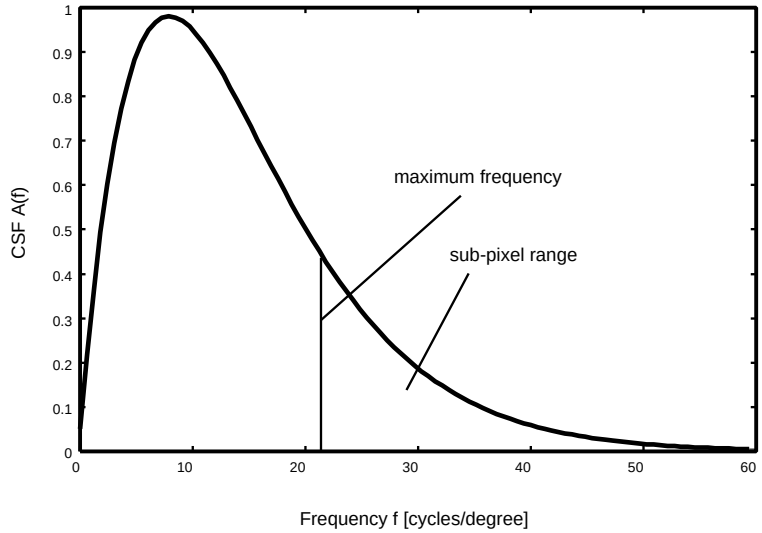


Figure 8.2: Subpixel range for maximum displayable frequency of 22 cycles/degre

8.2 The Main Idea

Our main idea is to place a limited number of rectangles of various sizes in an image (more precisely in each of the two images that will be compared), then to compute the average color of each rectangle in CIE XYZ color space, and finally to convert the average color to the CIE LUV space and compute the color difference using the CIE LUV color difference formula:

$$\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2} \tag{8.4}$$

Color differences will be weighted according to the rectangle size and the contrast sensitivity function. In this way the differences that are more visible to us will be weighted stronger, and they will contribute more to the final distance. CIE LUV space was chosen as it is perceptually more uniform than CIE XYZ. If there is some noise in the image, it will automatically be neglected by the contrast

sensitivity function, unless it is visible and significantly influences our vision. Actually, more visible differences will contribute to the error more significantly.

As the number of all possible rectangles of various sizes in an image is huge we are going to use only a subset of all rectangles. We will not allow very thin rectangles, as we do not think they are so important in the image comparison. Positions and orientation of rectangles will be chosen quasi randomly, which makes the metric deterministic.

8.2.1 Algorithm Details

As stated before, we will need a large number of rectangles, of which we need the average color. In order to compute the average colors of rectangles fast we use summed area tables [Crow84]. A separate table is built for each of the three CIE XYZ color components for both images. That makes six tables, each containing a number of entries equal to the total number of pixels in the image (assuming that the compared images have the same size). Element $T(i,j)$ of the table T (see fig. 8.3) contains the sum of values of all pixels $X(x,y)$ such that $x \leq i$ and $y \leq j$. The average of the rectangle defined with points (k,l) and (i,j) such that $k \leq i$ and $l \leq j$ is then:

$$A = \frac{T(i,j) - T(k-1,j) - T(i,l-1) + T(k-1,l-1)}{(i-k+1)\cdot(j-l+1)} \tag{8.5}$$

Now, only the position and size of the rectangle has to be given to compute the

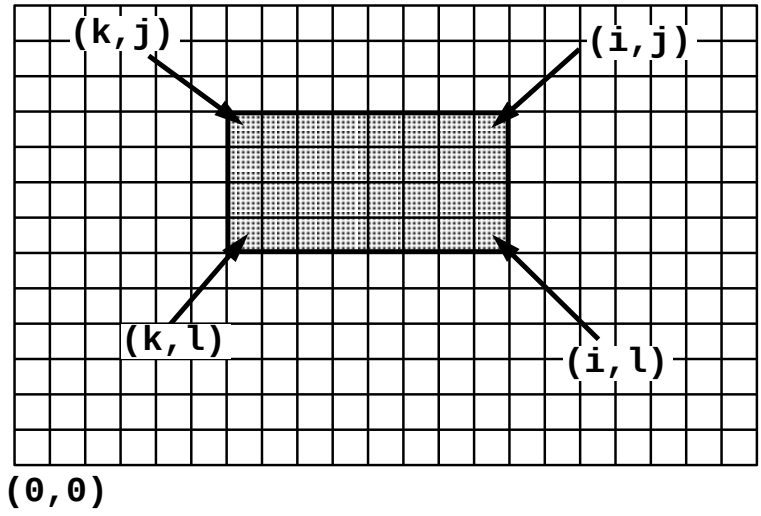


Figure 8.3: Summed area table

average with only a few operations.

The weighting of the particular rectangles according to the contrast sensitivity function is done implicitly using importance sampling. The integral of the contrast sensitivity function $g(f) = \int_{0^+}^{f} A(x)dx$ is precomputed and normalized by dividing all values with $g(60)$ (see fig. 8.4).

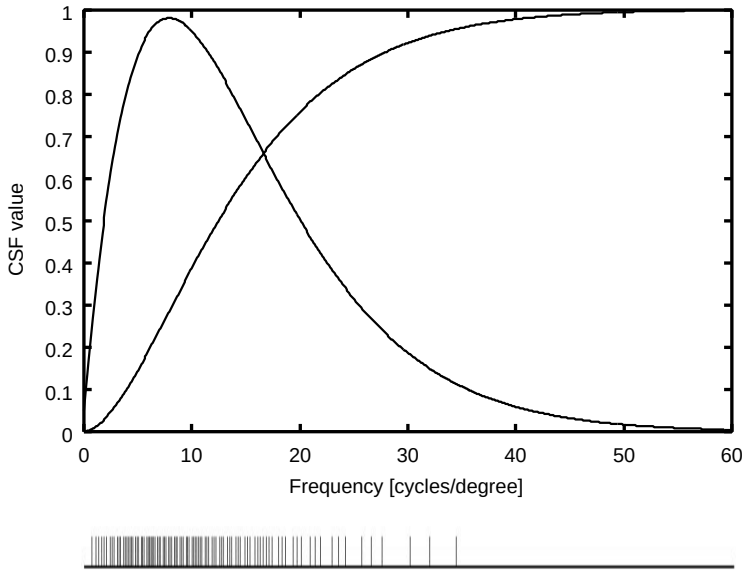


Figure 8.4: Contrast sensitivity function and its integral + random sampling

Randomly sampling over the domain of the inverse of $g(f)$ produces a frequency distribution as desired, so that no additional weights are necessary to get a distribution proportional to $A(f)$ (see bottom line in fig. 8.4).

Our idea is to select the position on the y axis quasi-randomly, by doing so, the metric has some meaning during the computation process as well. Once the size of a rectangle is determined, the orientation, i.e. longer and shorter side size, should be determined. The size of the rectangle corresponds to the rectangle's diagonal, and the maximum allowed ratio of the longer to the shorter side of the rectangle is the golden section ratio ($\phi = 1.618034\ldots$). We do not allow narrow rectangles as they are not so important in visual comparison. We speculated using various shapes, but since all shapes will be covered using a sufficient number of rectangles, we use only rectangles which make the method fast as well. The orientation of the rectangle is determined by choosing the angle between the diagonal and the horizontal axis. This angle is in the range $[\beta_{min}, \beta_{max}]$ (see fig. 8.5) and a particular angle is chosen quasi randomly. When the size and the orientation of

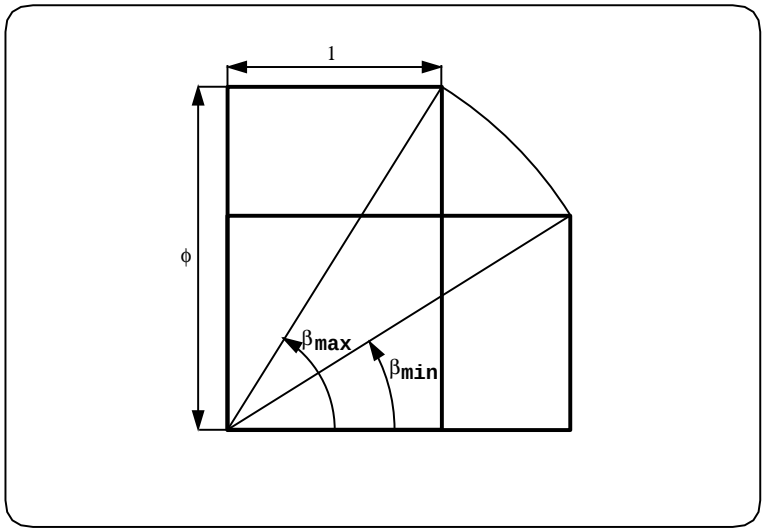


Figure 8.5: Possible orientations of the rectangles

the rectangle are known, the position of the rectangle can be determined. The rectangle position in the image is also determined by quasi random determination of its lower left corner. The position domain for this procedure is reduced, such that the complete rectangle lies within the image. Figure 8.6 shows the first 100, 250, 500 and 1000 rectangles placed in an 512×512 image, assuming the maximum displayable frequency is 60 cycles/degree. Although there are lot of uncovered image areas after 1000 rectangles are placed in the image, this result will rarely be significantly different from final solution. According to our experiments 10000 rectangles will be sufficient in most cases.

Therefore we have a $4-$dimensional quasi random problem. We have chosen the Halton sequence [Glas95] to compute these quasi-random numbers.

The Halton sequence for an N-dimensional point x_m is defined as:

$$x_m = (\phi_2(m), \phi_3(m), \ldots, \phi_{p_{N-1}}(m), \phi_{p_N}(m)) \tag{8.6}$$

where p_i refers to the ith prime number, and the function $\phi_r(m)$ is the radical-inverse function of m to the base r. The value of the radical-inverse function $\phi_r(m)$ is obtained by simply reflecting the digits of m written in base r around the decimal point. Therefore if m is:

$$m = a_0 \cdot r^0 + a_1 \cdot r^1 + \ldots + a_n \cdot r^n \tag{8.7}$$

the radical inverse function $\phi_r(m)$ is:

$$\phi_r(m) = a_0 \cdot r^{-1} + a_1 \cdot r^{-2} + \ldots + a_n \cdot r^{-(n+1)} \tag{8.8}$$

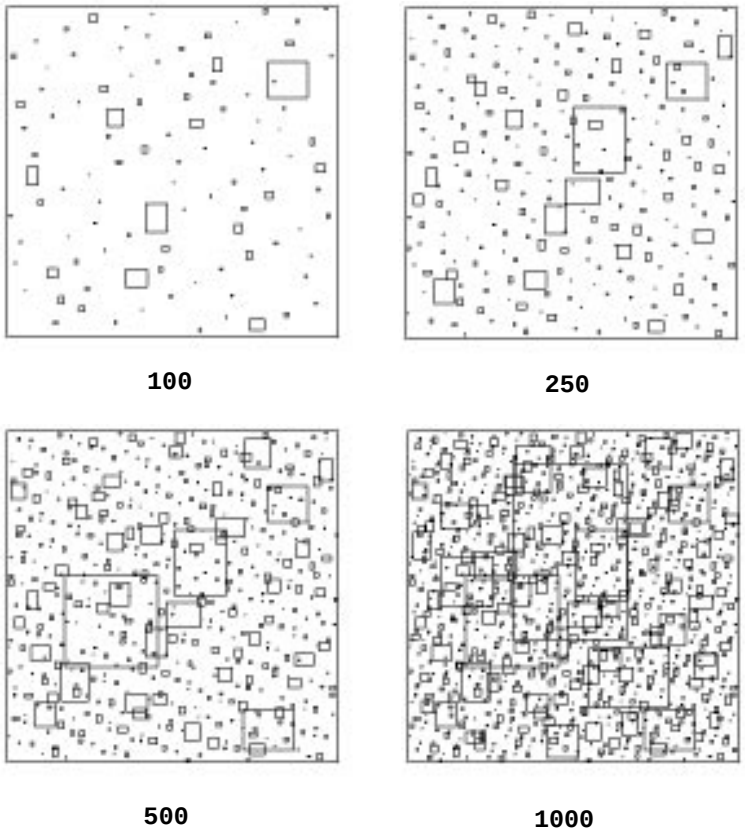


Figure 8.6: First rectangles

As our problem is $4-$dimensional we need the ϕ_2, ϕ_3, ϕ_5 and ϕ_7 functions. The size will be chosen using ϕ_2, the orientation using ϕ_3 and the position in the image using ϕ_5 and ϕ_7. Now, the user needs only to set the total number of rectangles and the corresponding metric can be computed. Note, that the metric produces (slightly) different values for different numbers of rectangles, i.e. for every number of rectangles a different metric is actually defined.

If the maximum displayable frequency is much lower than 60 cycles/degree it could be a good idea to subsample the image and to make the summed area tables correspond to the sub-pixel resolution, e.g. four entries per pixel. In this way sub-pixel areas can be computed, and a pixel size area can cover four quarters of the neighboring pixels. It seems to us that a finer subdivision than four sub-pixels per pixel would be rarely needed. If the subdivision is not done, the weights of the sub-pixel range should be somehow added to the highest, still displayable frequency.

The whole idea of placing various rectangles in the image space, overcomes

the drawback of wavelet transforms that various resolution sub-images have fixed positions. A quarter of an image sized square placed exactly in the center of the image will never be considered as an entity using wavelet transforms.

8.2.2 Modified CIE LUV Color Difference Formula

We stated before that we want to use the CIE LUV color difference formula. CIE LUV space was designed to approach a perceptually uniform color space, but it was designed in strictly set laboratory conditions, with adapted observers. In every day life, we are rarely fully adapted to any particular luminance level. Most of the time we are in complex environments, where our eyes do not have enough time to fully adapt to one specific luminance level. The perception in complex environments has still not been studied extensively. Hunt [Hunt92] and Nemcsics [Nemc93] studied perception in complex environments. Nemcsics developed a new color space called *coloroid* [Hunt92], [Nemc80], [Nemc87], which is designed for complex environments and solid colors (not for self-emitting media).

It is interesting that Nemcsics reports that just noticeable color difference can be up to 4 times larger in a complex environment, than for the fully adapted eye. After experiments with 2500 observers Nemcsics concluded that the brightness L is proportional to $\sqrt{Y}$, rather than to $\sqrt[3]{Y}$ as proposed in CIE LUV formula (for a complex viewing environment, of course). We made a compromise and use a modified CIE LUV formula such that L is proportional to $\sqrt{Y}$ and the chromacity component is computed using CIE uv, but as functions of the new L':

$$L' = 10 \cdot \sqrt{Y} \tag{8.9}$$

$$u' = f(L') \tag{8.10}$$

$$v' = f(L') \tag{8.11}$$

This method is certainly not one hundred percent correct, but finding a better one will exceed the scope of this thesis.

Furthermore, we modified the color difference formula as well. Namely it is known that colors that have a CIE LUV difference of less than 1 appear to be the same. This limit value changes with view conditions, frequency of stimuli etc. Therefore, if two colors are intended to be different, it is recommendable to choose colors with a CIE LUV difference of 6 or more.

Since we are not able to see color differences less than 1, we modified the difference formula so that all original differences of less than 1 are set to 0. Note that this modification makes it possible that images A and B are the same, images B and C are the same, but images A and C are different. This is not a metric in the mathematical sense any more, but it has some other advantages.

Of course the modified difference formula should be applied to each particular rectangle pair and not on the final difference. In this way, if the total average difference is some small number (e.g. 0.5) it means that there is a visible difference somewhere in the image.

8.2.3 Color Image Difference in a Distance Range

As stated before all perception based metrics are distance dependent. If someone asks us to see if there is any difference between two similar printed pictures, we will move the pictures back and forth in order to see some difference. This is true if the differences are not significant and obvious at first sight, of course. Actually, what we are doing is moving the maximum frequency on the contrast sensitivity curve, and trying to match it with a frequency where difference appears. There are lots of situations where the viewing range is limited and known in advance. For example, if some rendered images are to be presented in a classroom, the viewing distance range is known, it is between the viewing distance of the first row, and the viewing distance of the last row. If images are rendered with some progressive method, the rendering can be stopped much earlier than if images are intended to be observed from 50 cm. On the other hand, if some blurring operator is shown, the lecturer should apply quite a large blurring matrix if he or she wants all those present at the lecture to see the blurring effect. Even in conferences and university lectures, images that can not be distinguished are frequently presented as examples for differences. If the perceptual metric is taken into account such situations can be avoided. The next example could be a display device in a shopping window. Sometimes a monitor is placed in a shopping window and some animation is running. Obviously, the distance between the monitor and the window-glass is the minimum viewing distance, and the distance between the sidewalk border and the monitor is the maximum distance. If this were taken into account during the rendering of animation frames, the rendering time could be shortened significantly. There are numerous other similar examples, that can save us a lot of extra rendering time. If the image difference in a distance range should be computed, the algorithm would be a little bit different. It is then possible to compute the same number of rectangles of various sizes, and to apply weighting functions according to the particular viewing distances.

8.2.4 Image Query

The color image difference introduced here can be used for fast image query as well. We propose to make a low resolution (e.g. 60×60) version of each data-base image. Furthermore, the method will be not view distance dependent any more.

The maximum frequency will be set so that the most important frequency corresponds to the rectangles of approximately $1/2$ to $1/3$ size of the reduced images. Then, the first 200 rectangles are found, and average L', u' and v' values are stored in an array. This whole process is done once, and the L', u', and v' values (200 values for each image) are stored. Now, the target image is drawn by the user, or submitted somehow else, and the query begins. The target image is reduced to the low resolution, 200 rectangles are found (note that they will correspond to the pretabulated data-base rectangles as the whole method is deterministic), and the image differences are computed. The whole data-base does not have to be sorted, we only want to find for example the top 10 images. When the highest difference limit value of the *top ten club* is known, the current difference evaluation can stop as soon as the sum of rectangle differences exceeds this top limit. In this way, the difference computation time will decrease as *the top 10 club* will have better and better limits.

8.3 Algorithm Summary

Let us summarize the image difference algorithm now. The integration of the contrast sensitivity function can be precomputed and stored in an array. There are two images (let us assume they are of equal size), both contain r, g, and b values for each pixel. First the r, g, and b values are transformed to CIE XYZ values, and summed area tables are built for X, Y, and Z values for both images (6 tables are built). The Halton series can be precomputed or easily computed on the fly. The size of a rectangle is determined, and its orientation and position in the images using Halton series. Average X, Y, and Z of the rectangles are computed using (8.5) and colors are converted to the CIE LUV color space. Now the color difference is computed using (8.11) and this difference is added to the total distance. At the end, the total distance is divided by the number of rectangles and gives us the difference between two images. Algorithm 3 shows the pseudo code of the whole process.

```
convert rgb values to XYZ values for both images;
build summed area tables for X, Y and Z for both images;
compute cycle/degree→pixel/degree scale factor;
for total number of rectangles do:
    begin
       select size using precomputed importance sampling;
       select orientation using Halton3 series;
       select x position of rectangle using Halton5 series;
       select y position of rectangle using Halton7 series;
       compute average XYZ color of rectangle in both images;
       convert XYZ to LUV values;
       compute CIE LUV color difference;
       add difference to total difference;
    end
devide total difference with number of rectangles;
```

Algorithm 3

8.4 Conclusion and Future Work

We have presented a perception based image metric which can be used whenever two images should be compared. Human vision is taken into account using the contrast sensitivity function given by Mannos and Sakrison [MaSa74] as well as the CIE LUV color space which is almost perceptually uniform. The results are similar to those reported by Gaddipati et al. [GaMY97]. We find our metric more intuitive as it is computed in the original image space. The newly proposed approach overcomes some wavelet's drawbacks such as the fixed position of certain frequency levels, as well. There is also no need to transform the image in order to compute the metric and the color is treated more accurately due to the use of the CIE LUV color difference formula. The new method can be used for steering the rendering process when progressive methods are used, for evaluating sufficiency of appropriate levels in various LOD algorithms, or for an image query from large data bases. It can also show us if some differences will be visible from a certain distance range or not.

This is certainly not the ultimate metric, but rather one more solution that works fine, and can be easily understood. We will try to take into account some additional human vision properties in the future.

Chapter 9

Results

This chapter will describe results of various tone mapping and image difference techniques. Tone mapping techniques will be described first.

9.1 Tone Mapping Techniques

We will start with fictitious unit methods, as most rendering packages are still not able to render raw images in absolute units. Test image 1 was rendered using the RADIANCE package [Ward94a], without taking its advantage of absolute units rendering.

Test scene 1 is a simple, back lit scene consisting of few objects. Strong back light (window in our case) causes problems for most tone mapping techniques. Color plate 1a shows the image obtained by the mean value mapping technique. Since the average value is high, the whole image is too dark. The exponential mapping technique (Color plate 1b) produces a little bit brighter image, since the average is mapped to 0.632, and not to 0.5 as by the mean value mapping method. Schlick's mapping technique tries to display the exterior and the interior part of the scene simultaneously. Although, this is the only mapping that displays exterior automatically, we find the final result still a little bit too dark. The factor M in the computation of parameter p (see eq. 5.1, 5.2) was set to 2, in Color plate 1c. Note that this is the best case, and increasing M (which is often needed for various media) would cause an increase of parameter p.

The minimum information loss methods were tested using the same raw image. Color plate 1d illustrates the minimum information loss technique. We have added an error function of type c (see Error Function section). Color plate 1e shows that there is no visible difference. Applying the minimum area loss method does not show a significant difference, either (Color plate 1f). Actually, the differences between these three methods will rarely be significant. The interior of the

scene is displayed with a lot of details. The results of interreflections can clearly be seen (color leak on the walls), and the overall impression corresponds, in a way, to a photo of the scene.

Test scene 1 has the mean value $L_{mean} = 3.882625$, and Schlick's parameter $p = 54.428318$, assuming $M = 2$ (see section Schlick's mapping). Note that the mean value mapping method will actually display all pixels in the range $[0, 2 \cdot L_{mean}]$. Table 9.1 shows the data for the minimum loss methods. Starting and ending points of the intervals are shown in the table. Error function was of type c, with $\alpha = 45^\circ$ and $d = 1/3$ of the clipping interval.

method	a	b	contrast	loss
min. info loss	0.017948	0.897421	50	17.604%
with error fun.	0,020447	1.022327	50	18.867%
min. area loss	0.017998	0.899913	50	18.839%

Table 9.1: Scene 1 minimum loss data

Color plate 2 illustrates the interactive calibration. All images have contrast $c = 50$, and aperture values are -3, -1.5, 0, and 3. This mapping technique gives us the possibility to examine particular parts of the image. For example, the reflection of the red sphere in the window can be seen when the aperture is set to 3. On the other hand, interesting interreflection details can be examined if the aperture is shifted to -3 or to even more negative values.

The next scene is mapped using only the interactive calibration. The logarithmic histogram of this scene is shown in the interactive calibration section (fig. 4.1). Color plate 4 shows this scene mapped using contrast 5 and 105, and aperture 0. This is a very low overall contrast image, so mapping using contrast 5 did not cause significant information loss. Note that the image mapped using a small contrast window has much higher contrast, than the image generated with a large clipping contrast.

Scene 3 is a more complex scene rendered using the RADIANCE software. Color plates 5a, 5b and 6a show the third scene mapped using the mean value, exponential and Schlick's mapping technique. Note that the mean value and the exponential mapping methods do not display any details under the table. On the other hand Schlick's method shows details under the table, but the overall impression is very poor. The reason lies in the very high overall contrast of the raw image. This raw image has the mean value $L_{mean} = 1.292$ and Schlick's parameter $p = 9719.833$, assuming $M = 2$, which is the best case. The lost information by the minimum information loss methods here is very large for a contrast of 50. Table 9.2 shows the data for this scene. The error function was chosen the same as earlier. As the loss for contrast 50 was too large, we increased the contrast, so

long as the loss was under 10%, and the new contrast was $c = 2300$. Of course, it would be incorrect to linearly map such a large clipping interval, and Schlick's mapping was applied on this clipping interval. Results of the minimum loss techniques are shown in Color plates 6b, 7a, 7b, 8c, and 8d. Due to very similar mapping intervals of the minimum information loss and the mean value mapping, these images will look almost the same in this case.

method	a	b	contrast	loss
min. info loss	0.040442	2.022103	50	28.329%
with error fun.	0,041810	2.090512	50	30.134
+Schlick's mapp.	0.002300	5.290552	2300	9.065%
min. area loss	0.045311	2.265544	50	38.811%

Table 9.2: Scene 3 minimum loss data

All so far shown images illustrate how various final images can be generated from the same raw image. Note that this type of image manipulation is not possible without the raw image. Once the mapping is done, pixel values are discretised in the range $0 \ldots 255$, and every brightening or darkening of the image will cause a large information loss.

Up to now we have not mentioned incident light metering. As stated before, incident light metering was designed in order to reproduce original objects' colors. It will be illustrated using 3 scenes. Comparisons with mean value, Schlick's, Ward's and Tumblin and Rushmeier's (TR) mapping techniques will be made. In order to apply Ward's and TR mapping methods, we have used default values suggested by the RADIANCE rendering package. Note that TR images are a little bit darker, but the settings were the same as for Ward's mapping technique. Increasing the light sources' strength drastically, would brighten TR images, but we wanted to examine the difference between dark and bright scenes here, and the relative difference would remain the same even for extremely strong light sources.

The results are shown for three scenes.

The idea of color reproduction is illustrated in color plates 9 and 10. Two simple scenes were rendered using no interreflection. The only difference between the two scenes is the color of the room. In the first case (left) the walls are almost white and in the second (right) the walls are dark gray. The cubes' colors are the same and the light settings are equal as well. The color bar under each image shows the originally selected cubes' and walls' colors. Color plates 10c, 10d and 10e show results of the new incident light mapping. It can be seen that all colors are reproduced faithfully.

Due to the different average, mean mapping produces different colors for the two scenes (although the cubes have the same colors and the light settings are the

same). Ward's mapping fails to reproduce the colors as well as the TR-mapping. Note that all these three alternative mapping techniques are based on the average scene luminance, and therefore the colors can not be reproduced faithfully for both cases as the average values are different for these two scenes. The mapping method proposed by Schlick [Schl94] produces acceptable images for the first two cases. Schlick's mapping depends on a parameter p, which depends on the total raw image contrast. As the contrasts of the two scenes are similar, the parameter p does not differ much. But if the orange box in the bright scene is substituted by a black box the overall contrast is drastically increased and Schlick's method fails. The new, incident light metering method reproduces original colors correctly in all cases. Note that it would be possible to manually calibrate Schlick's parameter p, but then the method would not be automatic any more. There is the possibility to manually adjust the exposure in the RADIANCE package as well, but again this is no automatic method. Table 9.3 gives data for this scene. The mean values differ a lot. The parameter p required for Schlick's mapping is almost the same for the first two settings, but increases for the third. Data for the median value of radiances are also given in the table. Note that these values do not differ much from the mean values. We have used mean mapping for comparison throughout this paper since this is the most widely used method. Using the median value of radiances would not affect the final results significantly.

	dark room	bright room	black cube
mean radiance	0.002511	0.01341	0.01339
median radiance	0.001796	0.01432	0.01429
median irradiance	0.009160	0.009160	0.009160
parameter p	1.415	1.267	17.652

Table 9.3: Data for the cubes scene.

The next scene (color plates 11, 12 and 13) is also rendered without inter-reflections. Again, two different object color settings were used, one with bright colors, the other with dark colors. Almost all objects are differently defined in the two settings, only the Christmas balls are equal. On the other hand the light definition is the same for both settings. The color bar displays a selection of three of these colors: box lid, chair and wall.

The results of the new mapping are shown in color plates 12a (scene with bright surfaces) and 12b (scene with dark surfaces). Color plate 11a shows the dark scene using the mean value mapping. The bright scene mapped with the mean value approach looks almost the same, so it is not displayed. Color plate 11b shows the irradiance image of scene 2 (all objects are diffuse, medium gray).

Color plate 13 shows the same images generated using Ward's and the TR-mapping. It is obvious that these mappings cancel out the differences which should be visible. Data for this scene is given in tables 9.4 and 9.5.

settings	bright	dark
images	12a, 13a, 13c	11a, 12b, 13b, 13d
mean value	0.9842	0.2533
$L_{\text{meter}}^{\text{out}}$	0.7401	0.7401

Table 9.4: Data for the fifth scene (without interreflection)

	box lid	chair	wall
bright	0.9 0.2 0.2	0.5 0.1 0.1	0.8 0.8 0.8
dark	0.18 0.04 0.04	0.1 0.02 0.02	0.08 0.08 0.08

Table 9.5: RGB values for color bar 5

Finally, the sixth scene (color plates 14, 15, 16, and 17) is rendered with interreflections. Tables 9.6 and 9.7 show the data for this scene. Color plates 15a and 15b are irradiance images that display the irradiance distribution in the scene. The intensity was extracted from the r,g,b irradiances using the max(r,g,b) principle (see Incident Light Metering chapter). Note the differences in the irradiance distribution for dark (color plate 15b) and bright (color plate 15a) scenes due to interreflections (shadows are much brighter in the bright irradiance image). The reference color bar in plate 16 shows selected colors for the sofa and the wall in the bright and dark scene. All other objects change too, except for the tree which remains unchanged. The results of the new mapping technique are displayed in color plates 16a and 16b.

Mean value mapping was used to map images in color plates 14a and 14b. Again, the mean value method fails to reproduce the originally set colors. Color plate 17 show the same scene mapped using Ward's and TR-mapping. Note how the tree becomes significantly brighter with almost all methods, only the newly proposed algorithm leaves the tree unchanged (that it looks a little bit lighter is a perceptual sensation caused by the darker background).

Note that the TR-mapping is intended to be used for a different purpose. Scenes with various light settings would be displayed different. Images look a little bit too dark as the method functions in a huge range of luminances (the method works fine up to a luminance level of $3.18 \times 10^6 \text{cd}/\text{m}^2$ and just for comparison a snow covered ground in full sunlight emits $1.6 \times 10^4 \text{cd}/\text{m}^2$ according

settings	bright	dark
images	14a, 15a, 16a, 17a, 17c	14b, 15b, 16b, 17b, 17d
mean value	0.01816	0.04813
$L^{\text{out}}_{\text{meter}}$	0.04282	0.03183

Table 9.6: Data for the sixth scene (with interreflections).

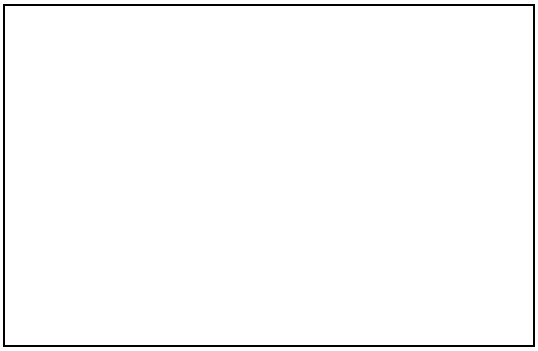
	wall	sofa
bright	0.3 0.8 0.8	0.87 0.5 0.25
dark	0.1 0.3 0.3	0.18 0.1 0.05

Table 9.7: RGB values for color bar 6

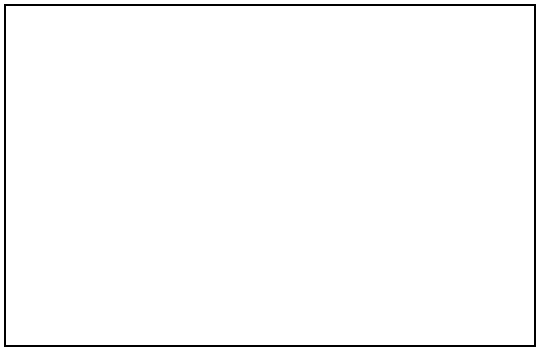
to [Glas95]). Note that light settings are not extremely low in our settings, and Ward's images were mapped using the same light levels and the same software. When using stronger light sources the TR-mapping will produce brighter images for both scenes, but again corresponding objects will appear similar although their color definition is very different.

The tone mapping techniques results end here. It can easily be seen from the above examples that there is no mapping technique that can be recommended for general use, but usually there is at least one technique that gives satisfactory results for each purpose.

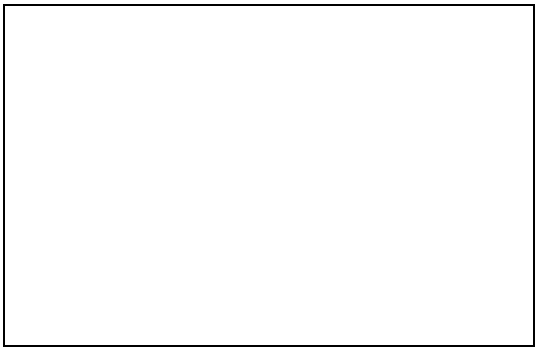
Color Plate 1



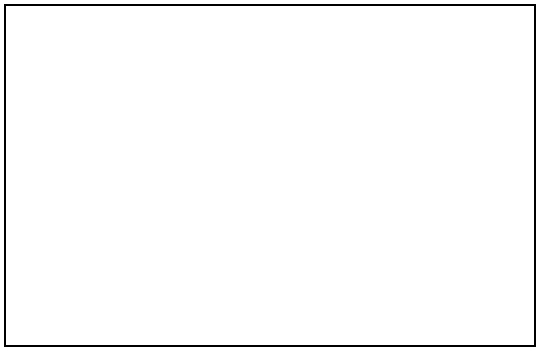
a. Mean value mapping



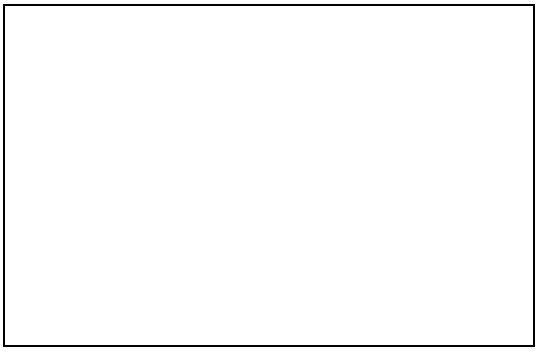
b. Exponential mapping



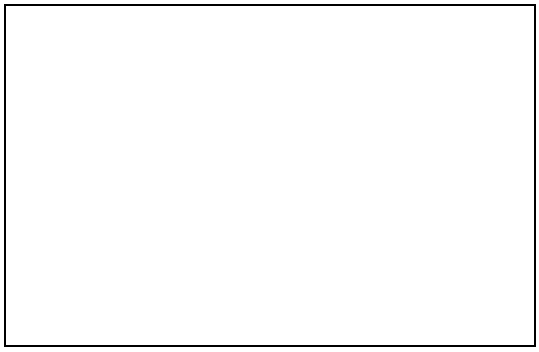
c. Schlick's mapping



d. Minimum information loss, C=50

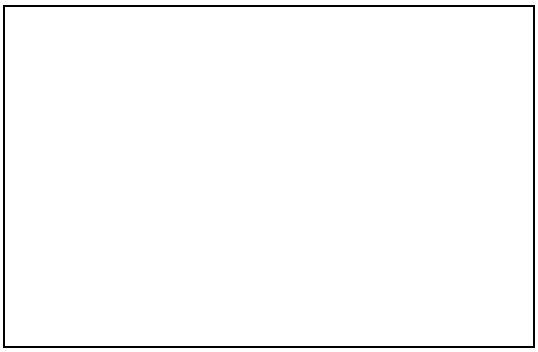


e. Minimum information loss with error function

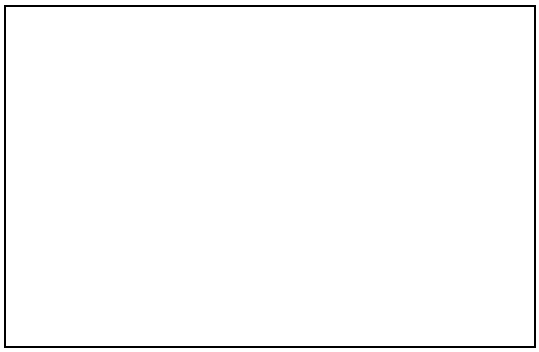


f. Minimum area loss

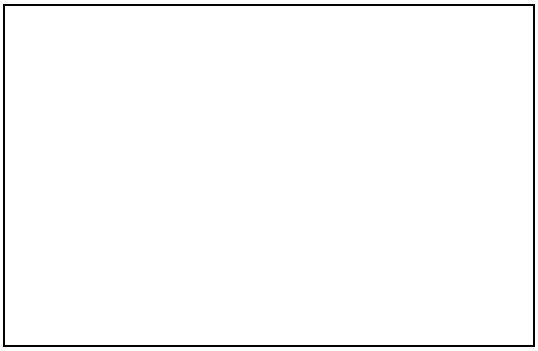
Color Plate 2



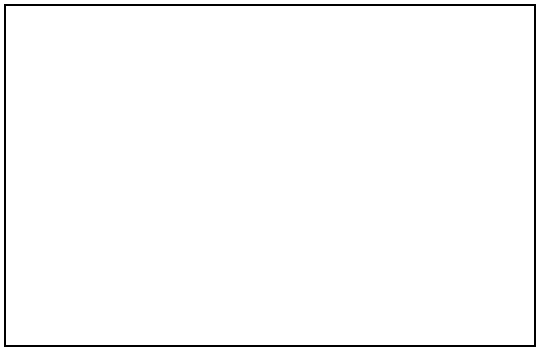
a. Interactive calibration
a=-3, c=50



b. Interactive calibration
a=-1.5, c=50

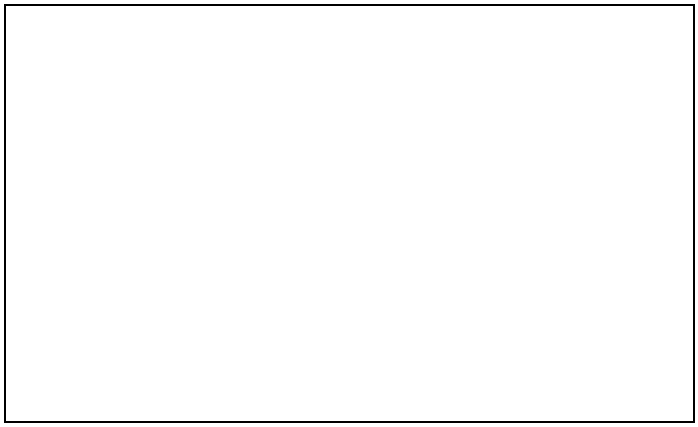


c. Interactive calibration
a=0, c=50

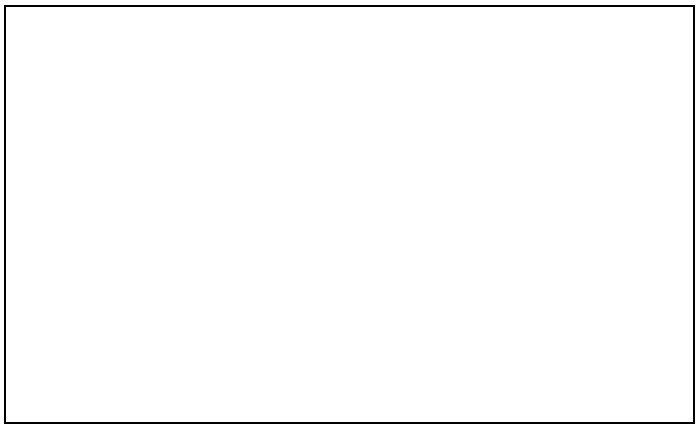


d. Interactive calibration
a=3, c=50

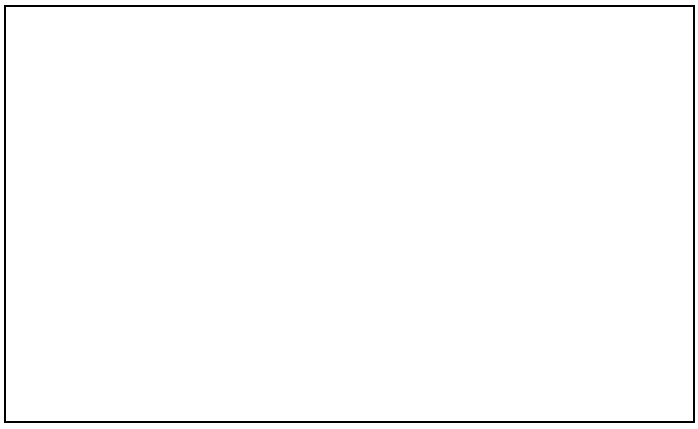
Color Plate 3



a. Interactive calibration a=1.5, c=40

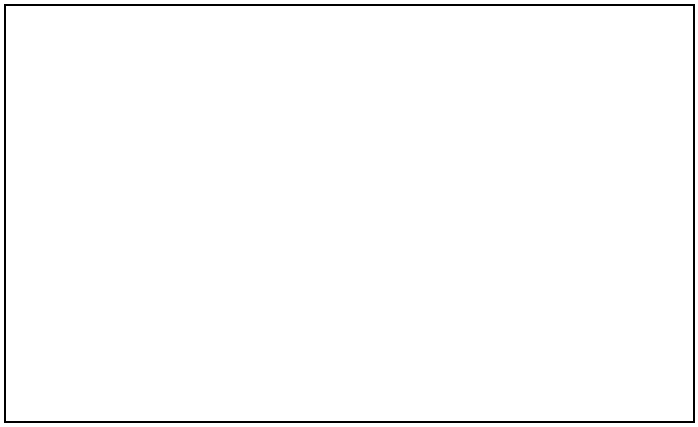


b. Interactive calibration, a=0, c=40

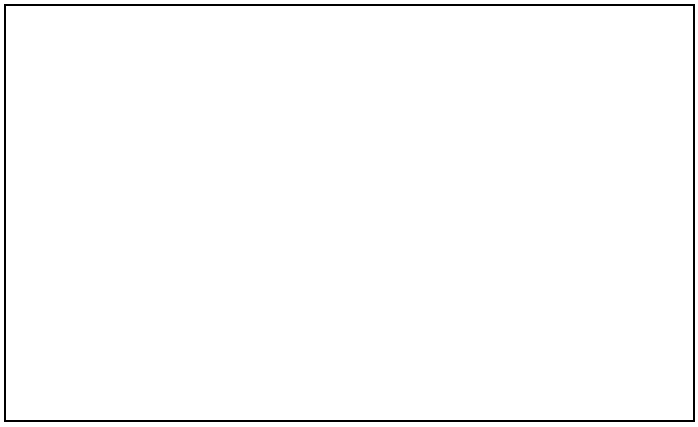


c. Interactive calibration, a=-1, c=40

Color Plate 4

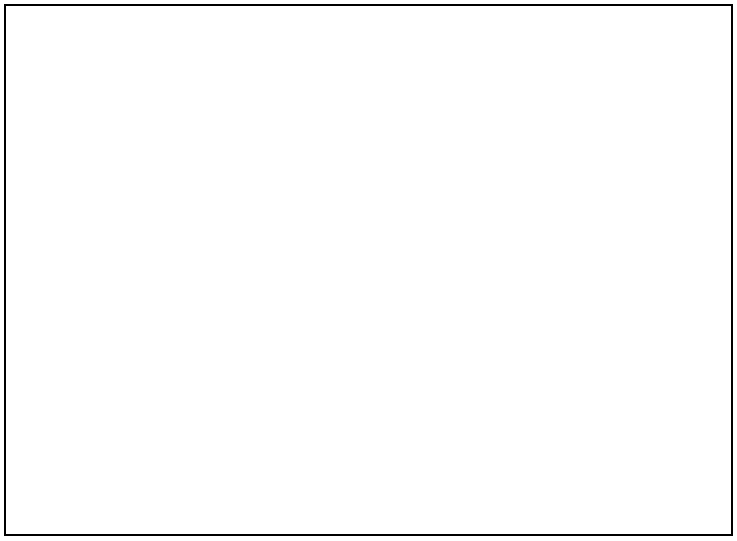


a. Interactive calibration, a=0, c=5

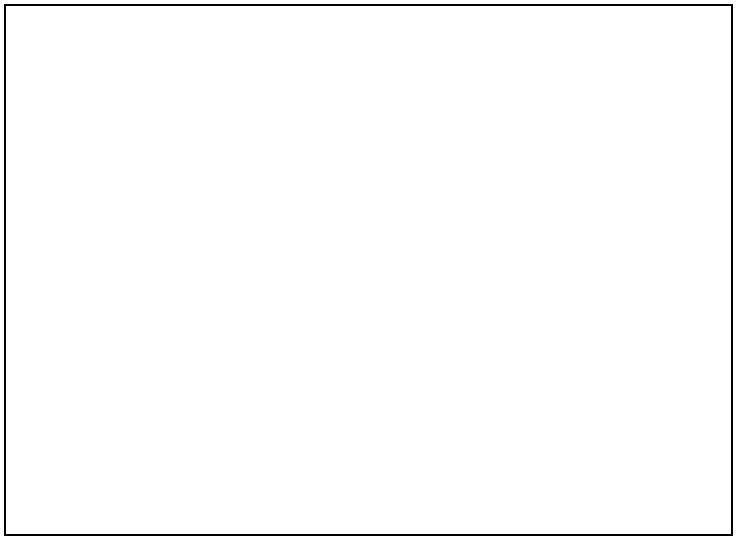


b. Interactive calibration, a=0, c=105

Color Plate 5

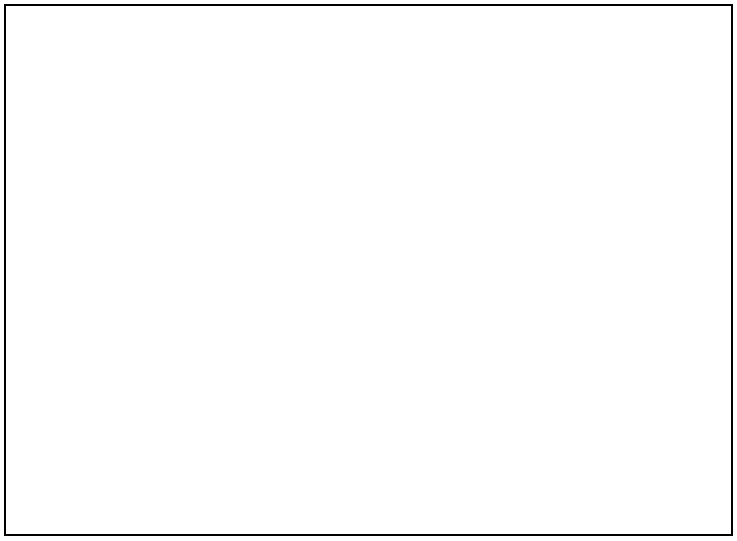


a. Mean value mapping

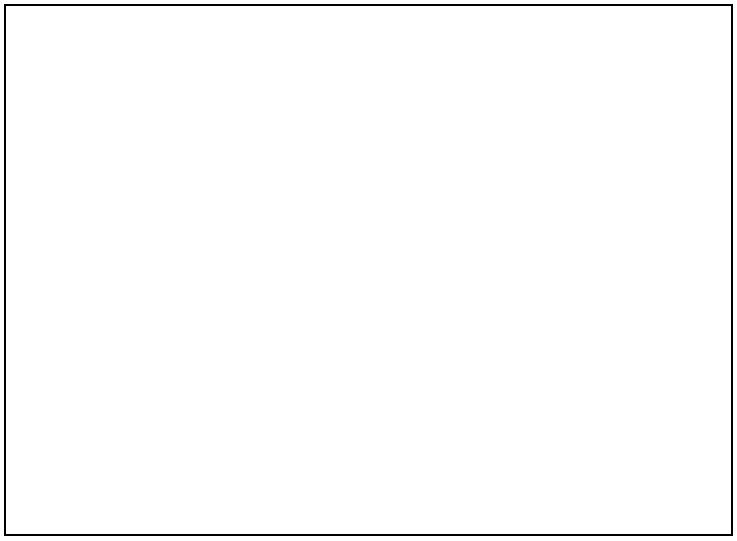


b. Exponential mapping

Color Plate 6

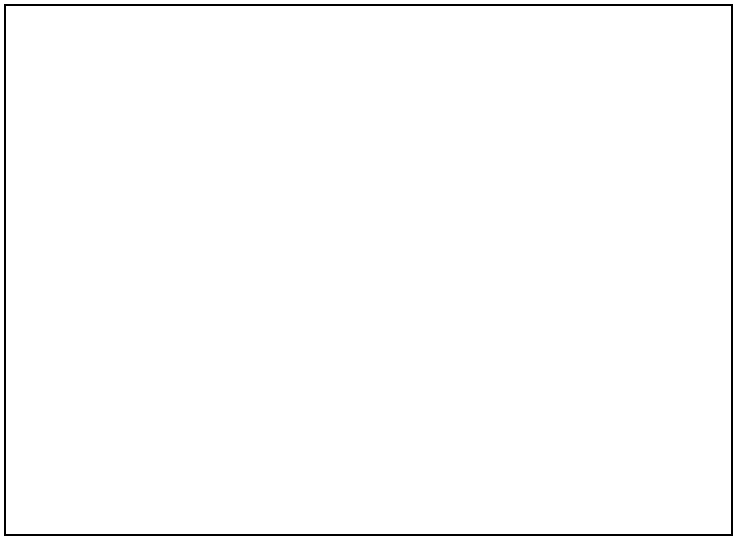


a. Schlick's mapping

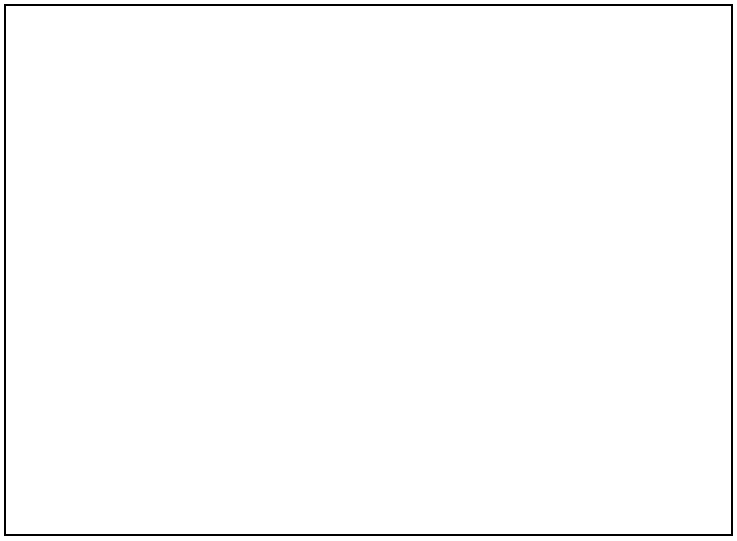


b. Minimum information loss, C=50

Color Plate 7

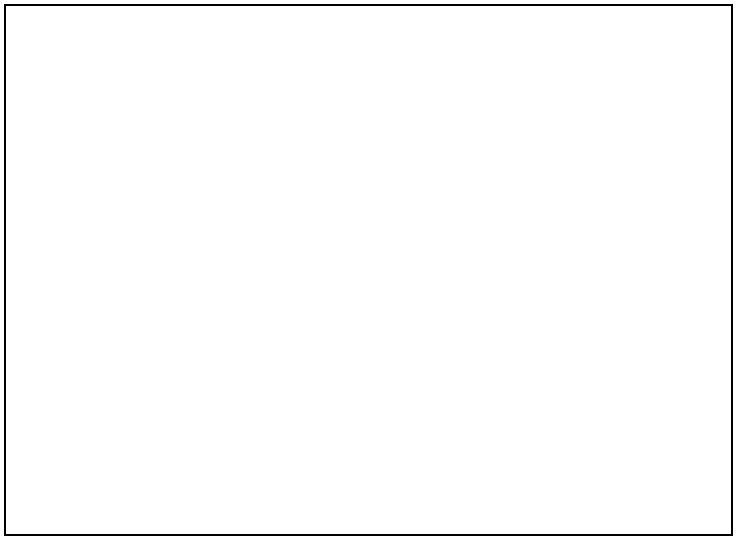


a. Minimum information loss, C=50
with error function

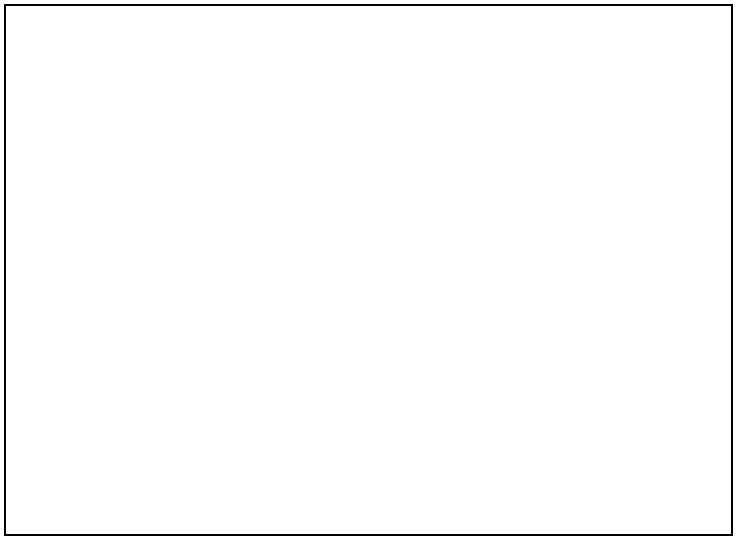


b. Minimum information loss, C=2300, Schlick's

Color Plate 8

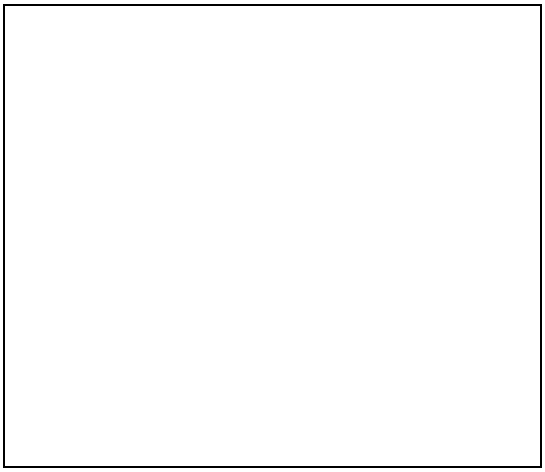


a. Minimum information loss, C=2300
Schlick's, with error function

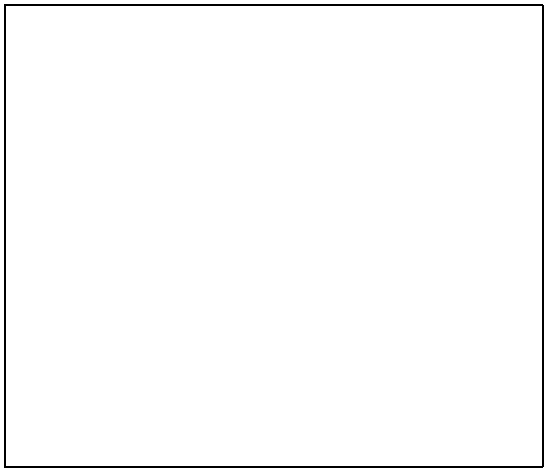


b. Minimum information loss, C=2300
linear on log scale

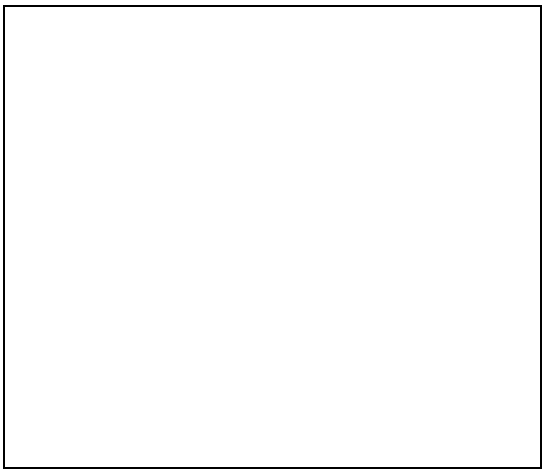
Color Plate 9



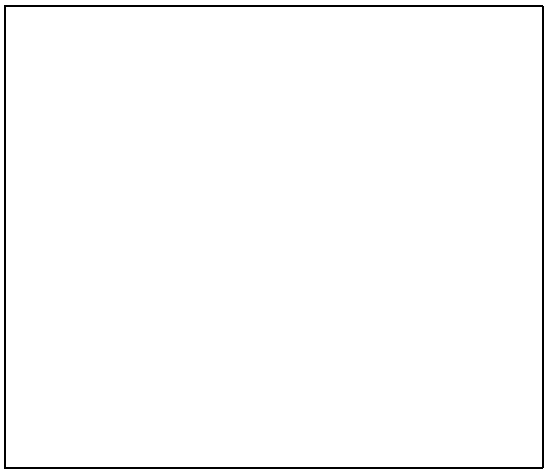
a. Bright scene, mean mapping



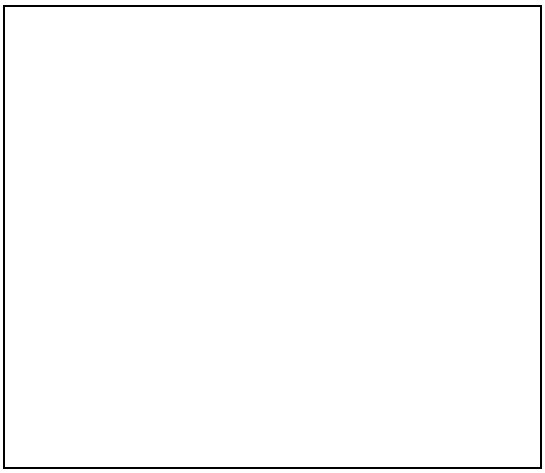
b. Dark scene, mean mapping



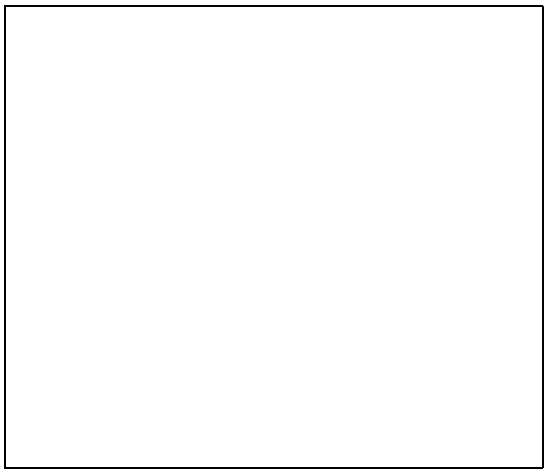
c. Bright scene Ward's mapping



d. Dark scene, Ward's mapping

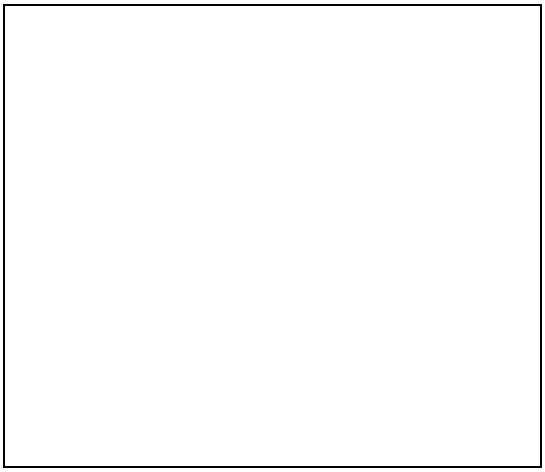


e. Bright scene, TR mapping

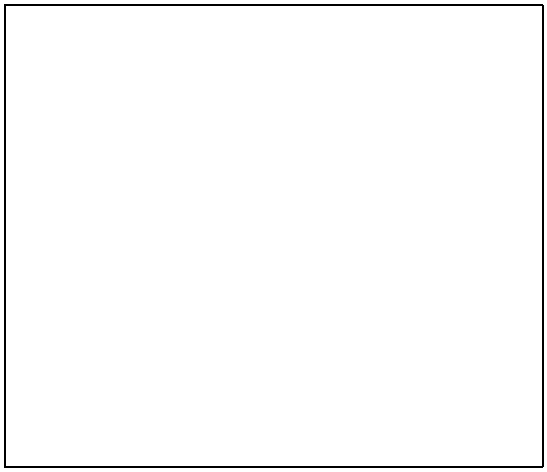


f. Dark scene, TR mapping

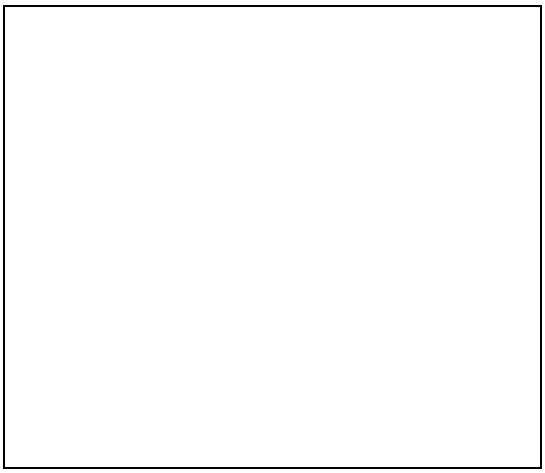
Color Plate 10



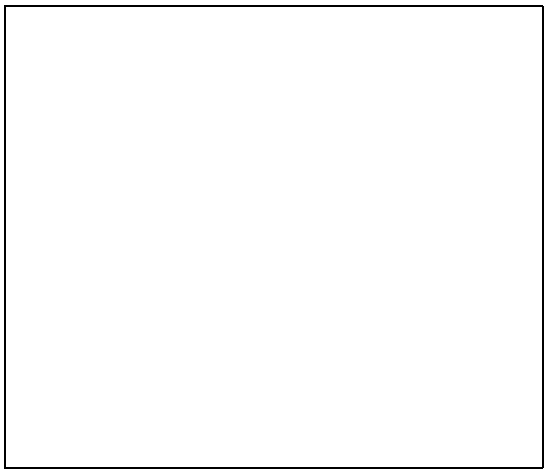
a. Bright scene, Schlick's mapping



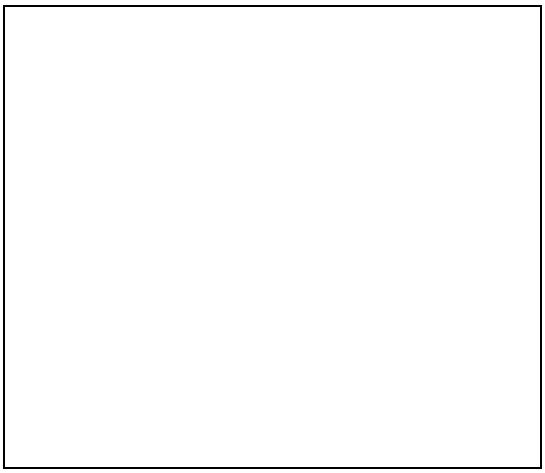
b. Dark scene, Schlick's mapping



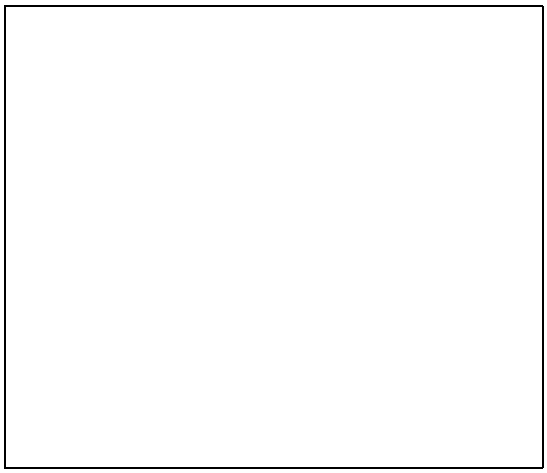
c. Bright scene, incident light



d. Dark scene, incident light

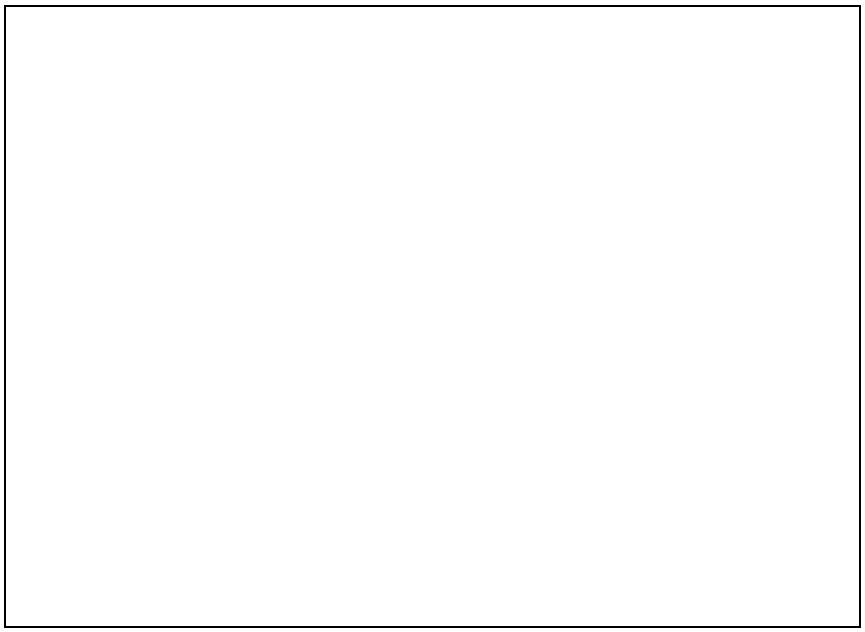


e. Bright, black cube, incident

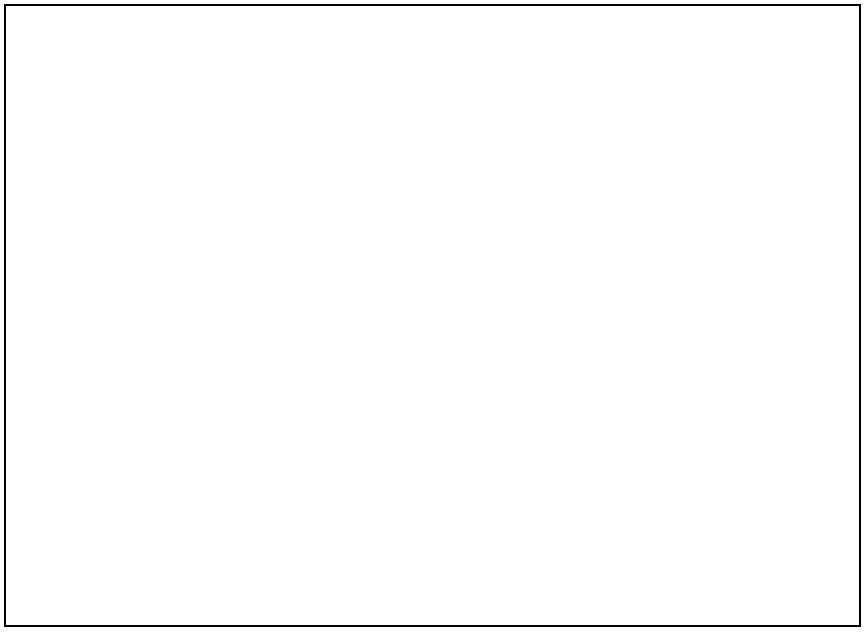


f. Bright, black cube, Schlick's

Color Plate 11

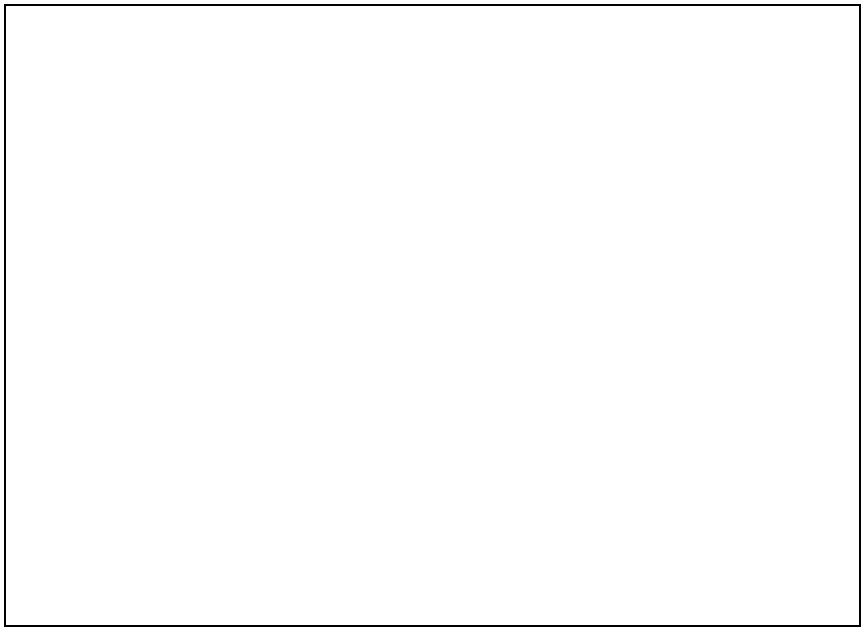


a. Dark scene, mean value mapping

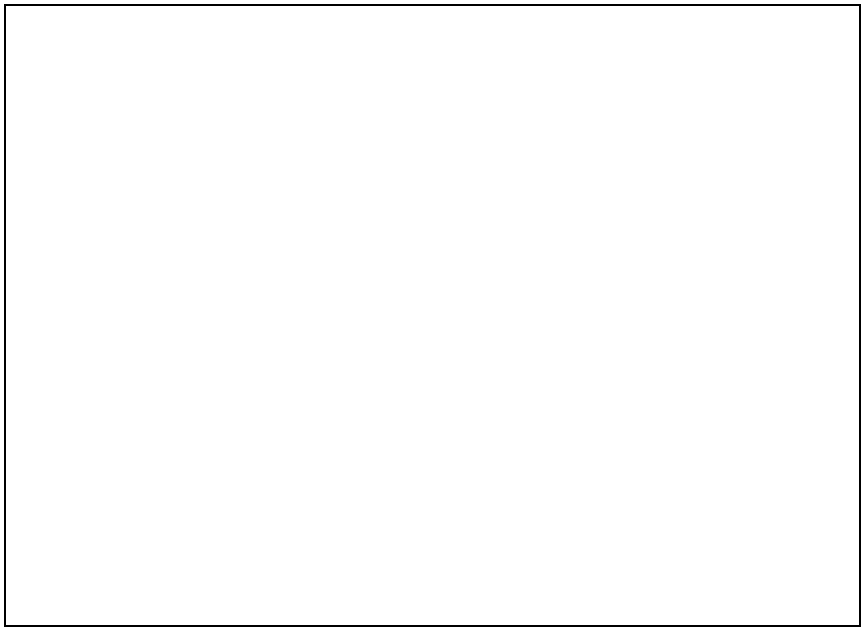


b. Irradiance image

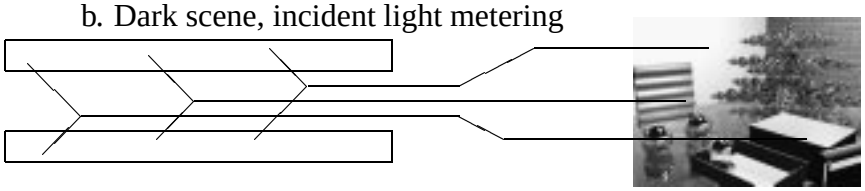
Color Plate 12



a. Bright scene, incident light metering

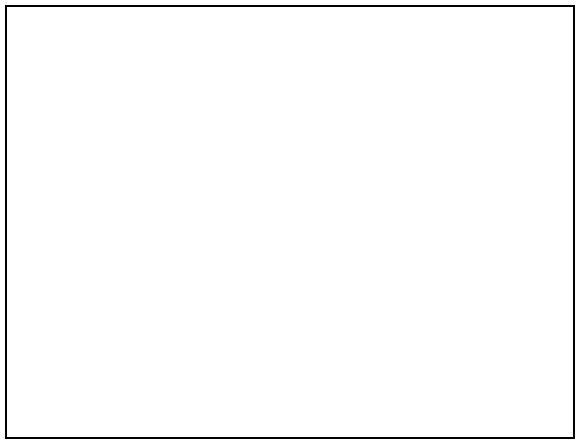


b. Dark scene, incident light metering

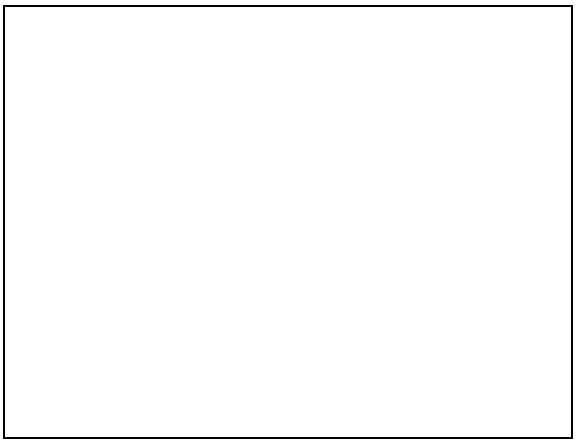


Box lid, chair and wall selected colors

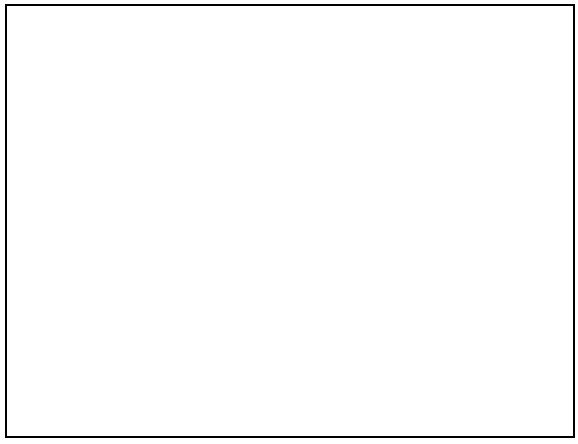
Color Plate 13



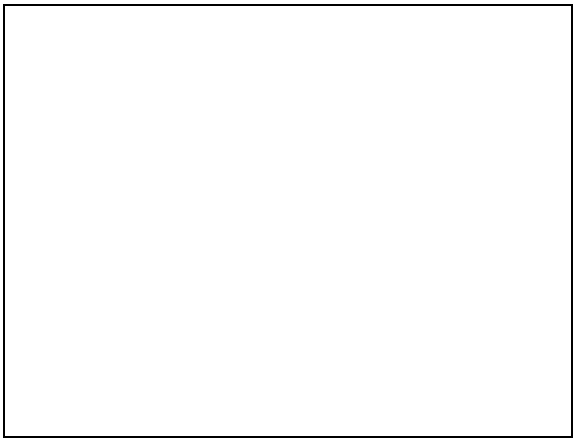
a. Bright scene, Ward's mapping



b. Dark scene, Ward's mapping

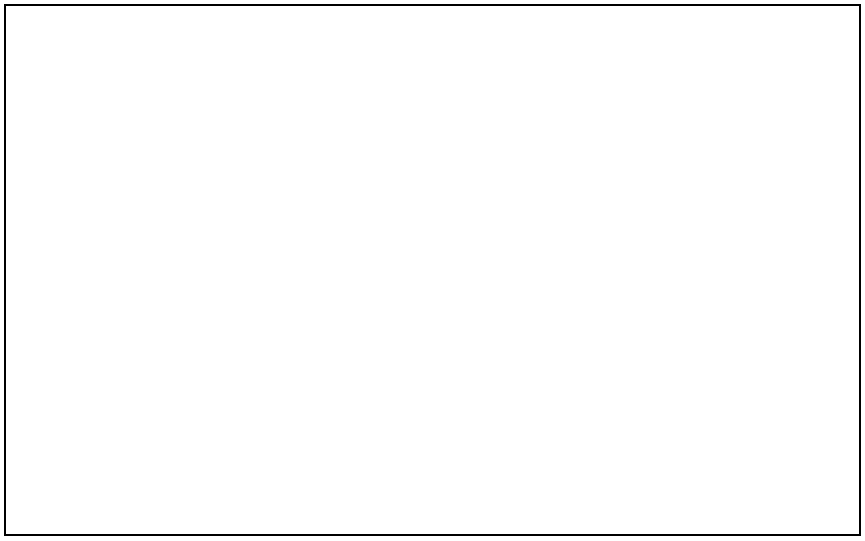


c. Bright scene, TR mapping

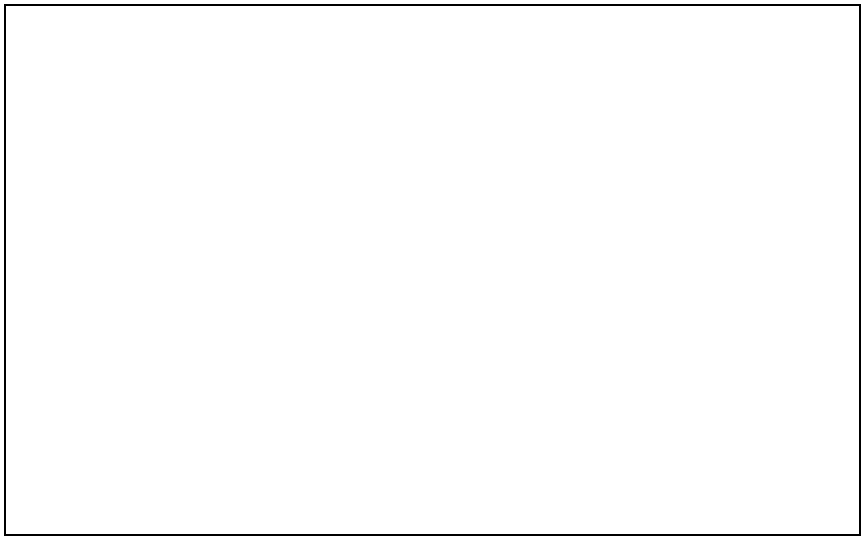


d. Dark scene, TR mapping

Color Plate 14

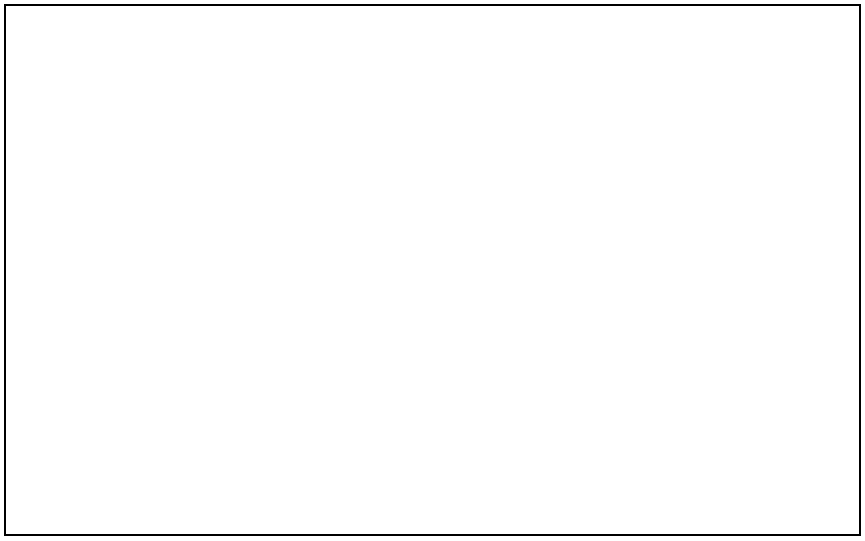


a. Bright scene, mean value mapping

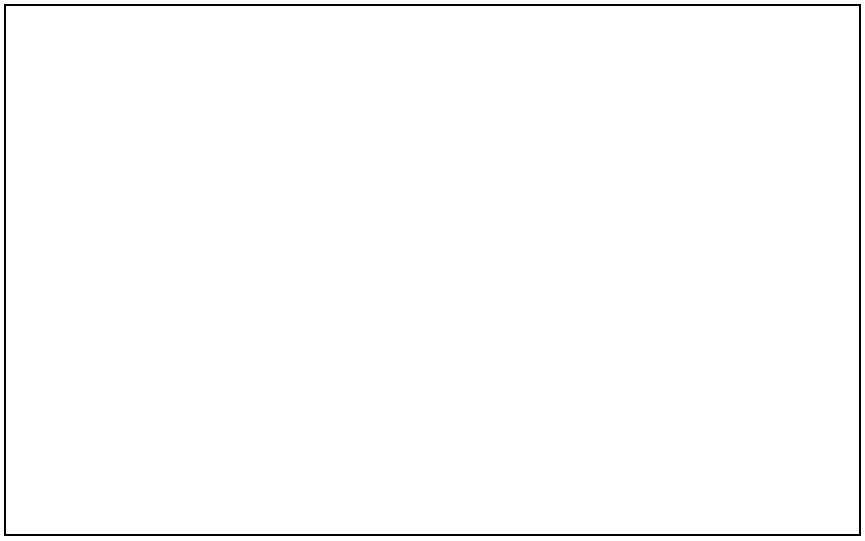


b. Dark scene, mean value mapping

Color Plate 15

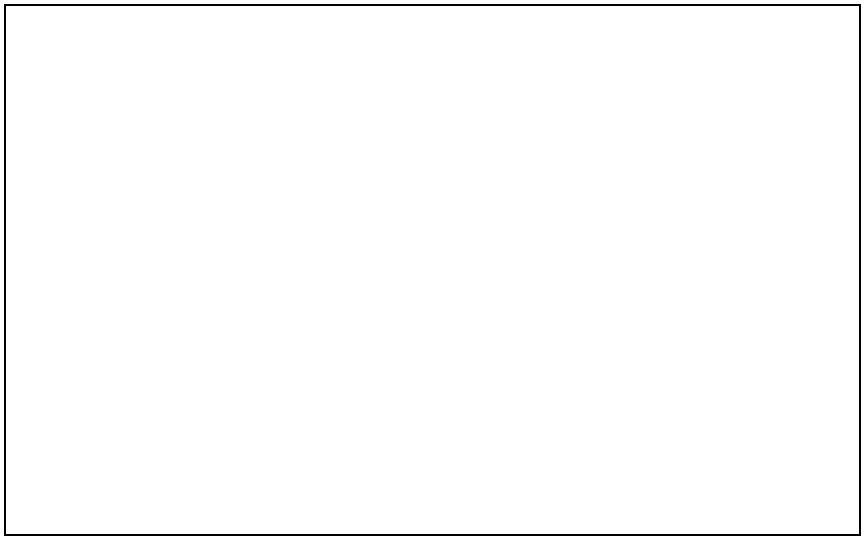


a. Bright scene, irradiance image

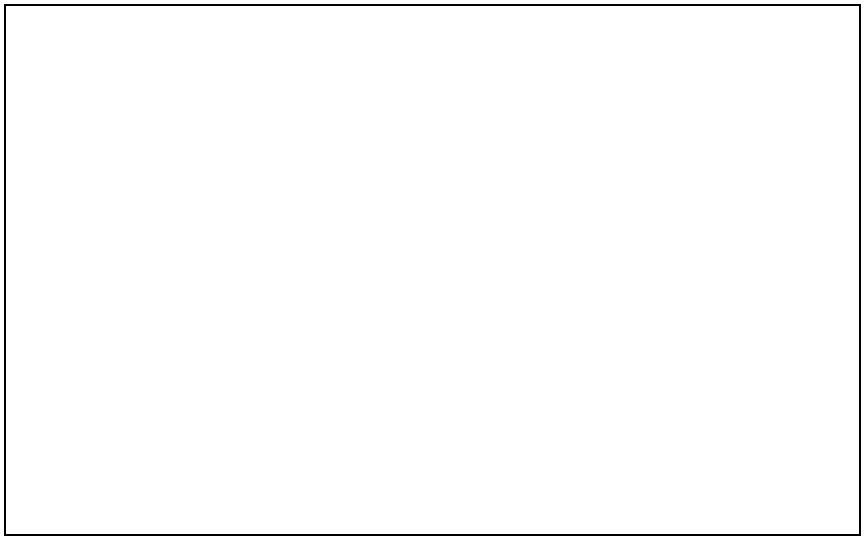


b. Dark scene, irradiance image

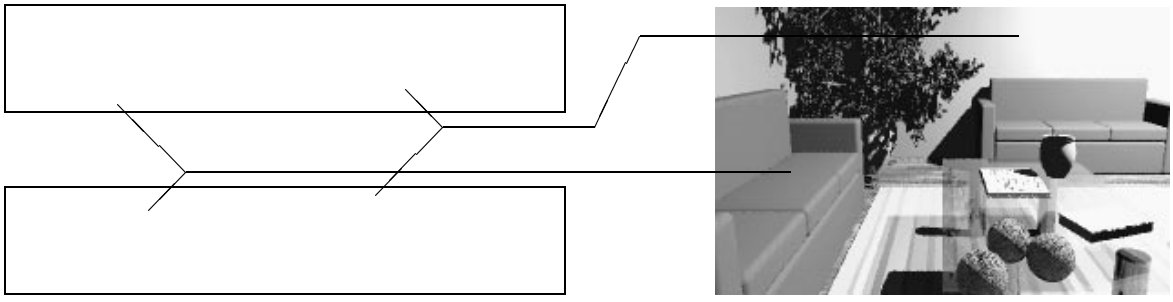
Color Plate 16



a. Bright scene, incident light metering

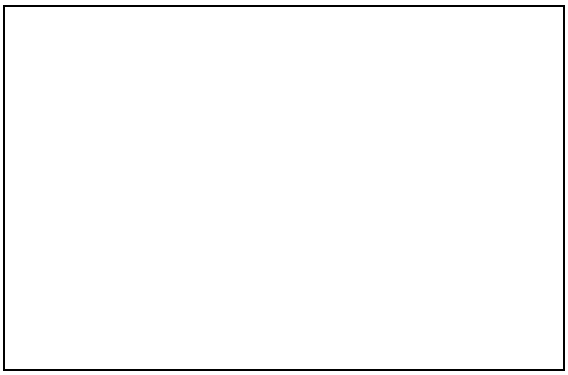


b. Dark scene, incident light metering

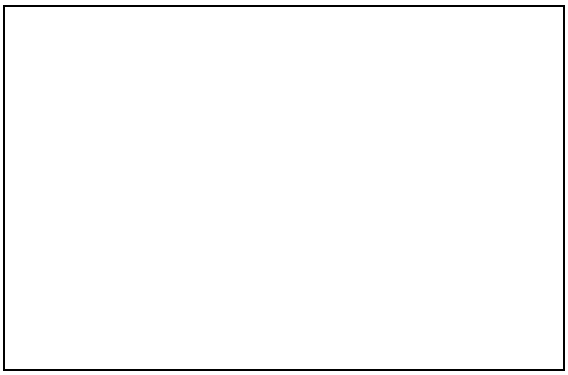


Sofa and wall selected colors

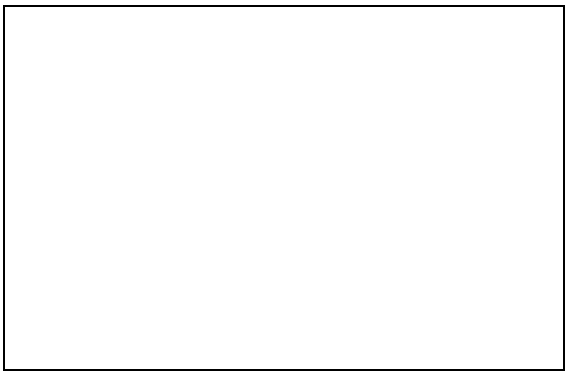
Color Plate 17



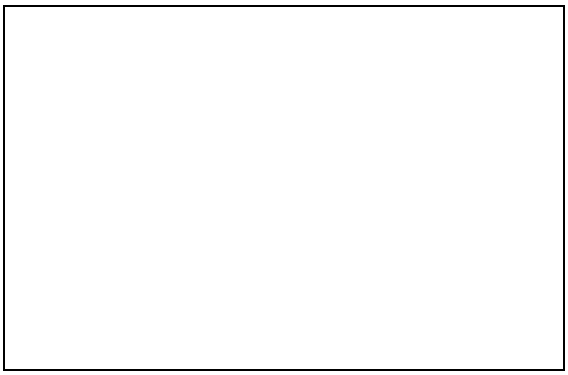
a. Bright scene, Ward's mapping



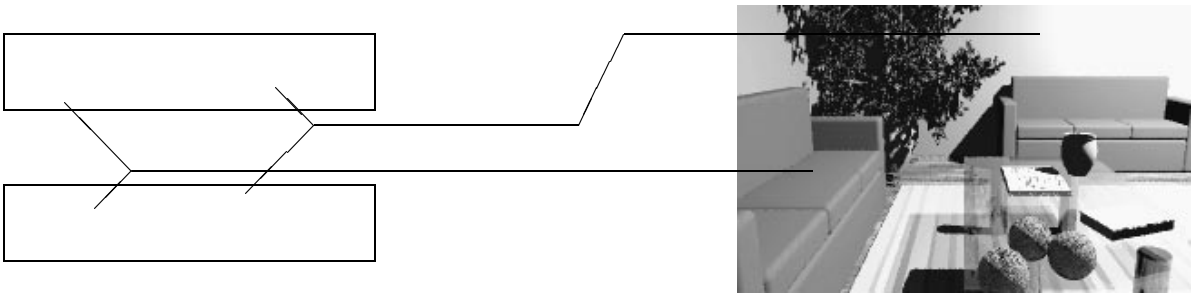
b. Dark scene, Ward's mapping



c. Bright scene, TR mapping



d. Dark scene, TR mapping



Sofa and wall selected colors

9.2 Color Image Difference

In this section the color image difference described in chapter 8 will be illustrated. First, the image series used by Gaddipati et al. [GaMY97] will be used to test the color image difference (CID), the metric introduced by Rushmeier et al., root mean squared error, and Gaddipati's metric. We have implemented Rushmeier's metric, although it is not 100% clear which viewing distance should be taken into account. We have chosen, in this example, the viewing distance so that the maximum displayable frequency is 60 cycles per visual degree for our metric, and the number of rectangles was 5000. The distance chosen by Gaddipati et al. is also not clear from their paper. Results are given in table 9.8, and images are shown in color plate 18a. Table 9.8 shows the differences between images rendered using various number of slices. This are rendering from a volume visualization done using various numbers of slices. Root mean squared error method (RMSE) is given for each component and for luminance image. The results for Gaddipati's metric are given only graphically in [GaMY97], and therefore they are not so precise.

method	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
RMSE									
red	4.162	2.320	1.688	1.276	1.051	0.869	0.770	0.710	0.648
green	1.492	0.938	0.551	0.429	0.327	0.289	0.236	0.218	0.194
blue	4.815	2.655	1.191	1.564	1.282	1.095	0.883	1.027	0.809
Y	0.665	0.386	0.262	0.202	0.163	0.137	0.114	0.111	0.092
Rushm.	16.763	1.108	0.534	0.399	0.206	0.104	0.093	0.060	0.070
Gaddi.	20.0	21.0	4.5	7.1	3.0	3.0	0.8	2.1	1.0
CID	1.490	0.636	0.428	0.346	0.279	0.243	0.211	0.290	0.225

Table 9.8: Results of different metrics for image series 1

Note the increasing difference at the end of the series which occurs by CID, Gaddipati's and Rushmeier's metric. This differences are not visible, but it is interesting that they occur in all three metrics.

Image series 2 in color plate 18b, shows images rendered using progressive radiosity. Numbers of rays shot is given under each image. We measured the CID, in order to estimate the sufficient number of rays. Results are given in table 9.9. Up to now we were always assuming the maximum displayable frequency is 60 cycles per degree. We have measured the differences in the third image series for increased viewing distances as well. The distance between the 500K rays and the 1M rays images was measured. Results are given in table 9.10. The results are valid for a 17 inch monitor, with 1280 pixels resolution. If it is known that

the image will be displayed on such a monitor, and it will be observed from a distance of 3 meters, it is sufficient to render the image using only 500K rays. This information can save us a lot of rendering time.

images	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10
CID max. f. 60	3.413	2.710	1.617	1.142	0.797	0.588	0.437	0.271	0.222

Table 9.9: CID for image series 2

viewing distance [cm]	CID
50	1.345
100	1.268
150	1.184
200	1.129
300	1.020
400	0.935
500	0.861
1000	0.630

Table 9.10: CID for various viewing distances, image series 2

We have measured the distance between the 50K and the 100K image using 950000 rectangles. Results are shown in table 9.11. It is clear that the difference converge, and 5000 or 10000 rectangles are sufficient in almost all cases.

The next example is shown in color plate 19a. This color plate shows an image series rendered using ray tracing and various numbers of samples per pixel. We have measured the difference between the 40, 80 and 160 samples per pixel images. Results are given in table 9.12. These results show us that the 40 and the 160 samples per pixel images can not be distinguished from a distance of 200 cm. Note that the rendering time was 4 times longer for 160 samples per pixel image.

The final example is illustrated in color plate 20. Color plate 20a shows original flower image, and color plate 20b original crow image. We have applied 3×3 blurring filter on the flower image (20c), and we have added some color patches to the original flower image (20d). There is also an image which has the same luminance, but colors are quite different (20e). Note that Rushmeier's metric would report no difference for this two images. There is also a combination of the blurred flower image and the crow image shown in color plate 20f. Resulting differences are given in table 9.13. All CID differences are computed assuming the maximum displayable frequency is 60 cycles/degree, and using 5000 rectangles. Note that

number of rectangles	CID
500	3.375642
1000	3.421180
2000	3.440896
5000	3.412503
10000	3.409281
20000	3.422541
40000	3.419777
80000	3.415657
160000	3.417164
320000	3.417722
640000	3.417656
900000	3.417328
910000	3.417486
920000	3.417535
930000	3.417455
940000	3.417468
950000	3.417440

Table 9.11: CID for various numbers of rectangles, image series 2

viewing dist.	40-160	80-160
50cm	1.878	1.476
100cm	1.409	1.112
200cm	0.789	0.597
400cm	0.380	0.276

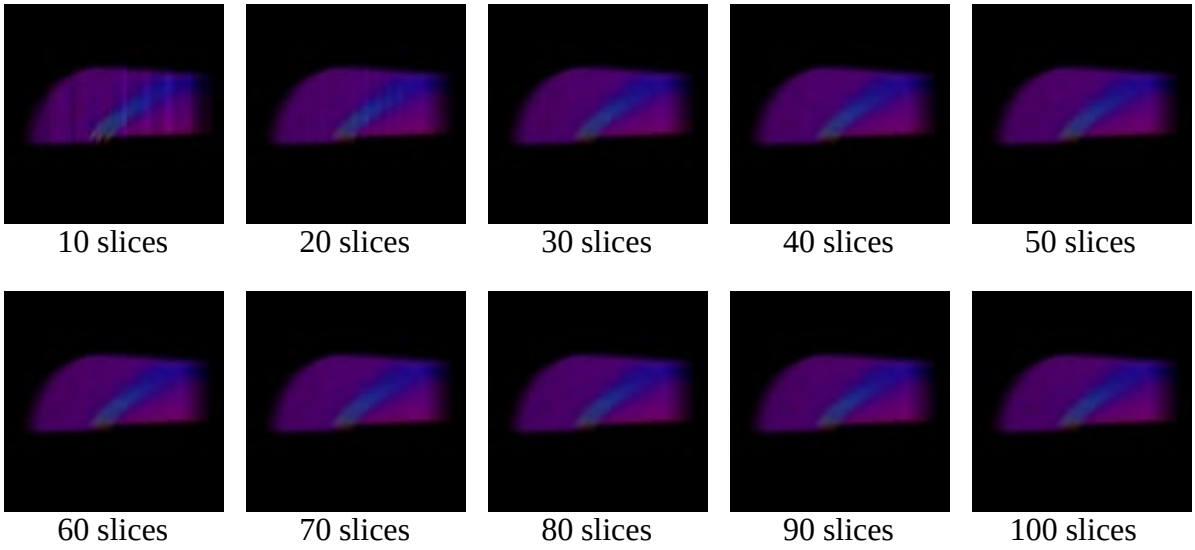
Table 9.12: CID for various viewing distances, image series 3

the often used root mean square error method reports a lower difference between the flower image with some color patches added (20d) and the original flower image (20a), than between the blurred (20c) and the original image (20a). A human observer would always report a different result.

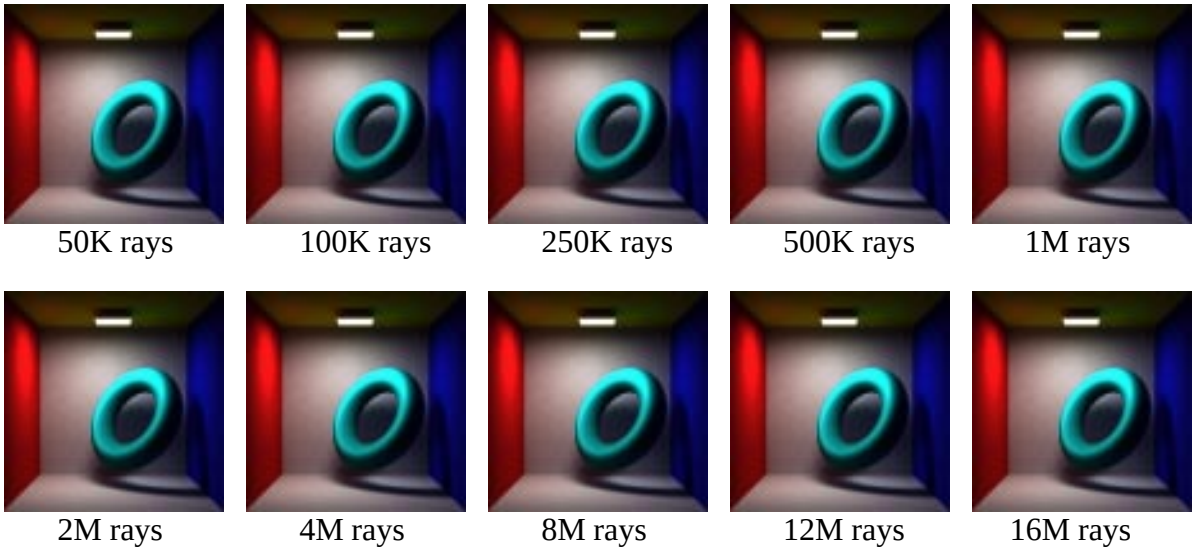
	b crow	c blur 3x3	d corrupted	e constant Y	f combination b+c
a	CID 52.547	CID 3.771	CID 5.098	CID 25.732	CID 30.582
	RMSE 38.948	RMSE 9.549	RMSE 5.285	RMSE 0	RMSE 28.562
b	CID 0	CID 51.595	CID 54.732	CID 50.143	CID 23.489
	RMSE 0	RMSE 36.930	RMSE 38.986	RMSE 38.949	RMSE 25.770
c	CID	CID 0	CID 7.759	CID 27.275	CID 28.110
	RMSE	RMSE 0	RMSE 10.584	RMSE 9.549	RMSE 26.452
d	CID	CID	CID 0	CID 28.406	CID 29.599
	RMSE	RMSE	RMSE 0	RMSE 13.097	RMSE 26.284
e	CID	CID	CID	CID 0	CID 30.579
	RMSE	RMSE	RMSE	RMSE 0	RMSE 28.725

Table 9.13: Differences for image series 3

Color Plate 18

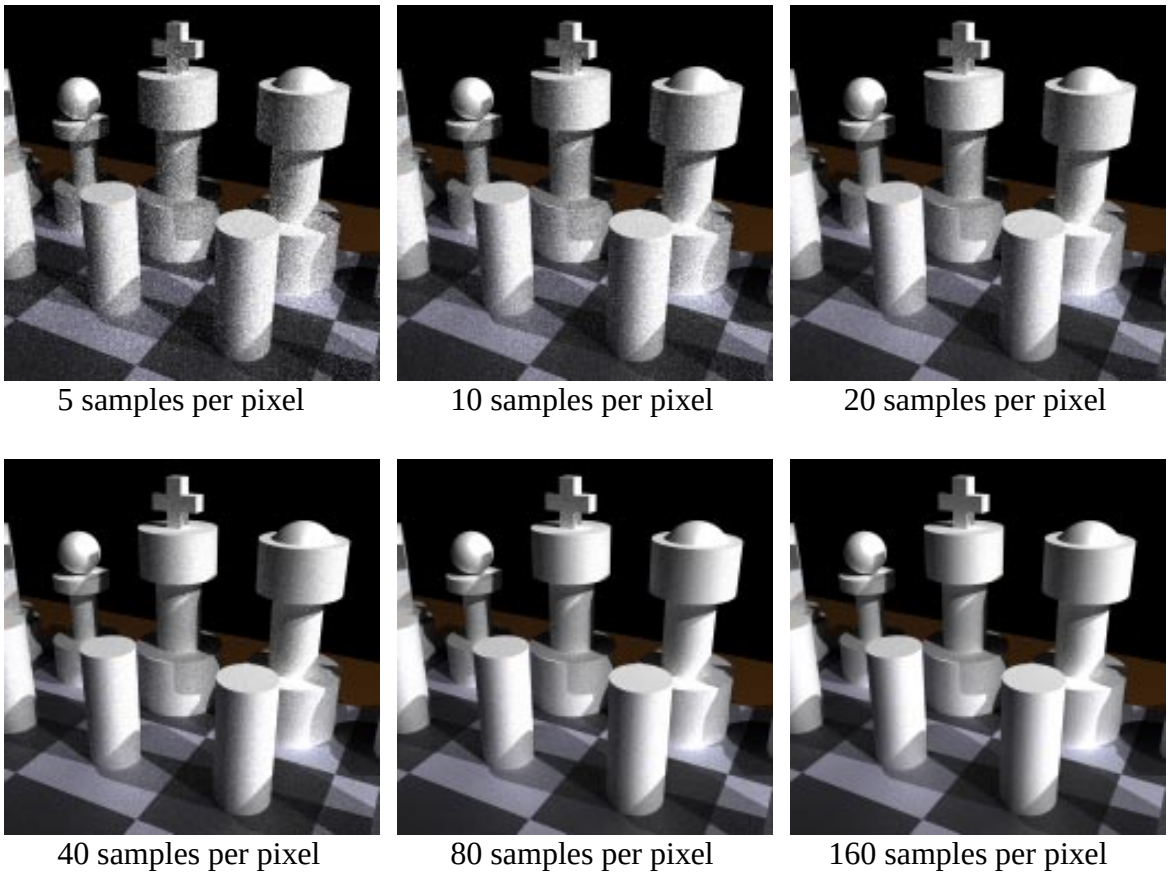


a. Image series 1



b. Image series 2

Color Plate 19



5 samples per pixel | 10 samples per pixel | 20 samples per pixel

40 samples per pixel | 80 samples per pixel | 160 samples per pixel

a. Image series 3

Color Plate 20

a. Original flower image

b. Original crow image

c. Flower image blured using 3x3 matrix

d. Corrupted flower image

e. Flower image of the same luminance as a.

f. Combination of b and c images

Chapter 10

Conclusion

We have presented various tone mapping techniques and image metrics used in computer graphics. This is still an open area for further research. There will be better and better display devices in the future, and tone mapping and perception based difference techniques will become more important.

Unfortunately, we can not say which tone mapping technique is the best. There are lots of good techniques for various purposes, but there is no one generally apply-able method. Everyone should choose the appropriate tone mapping method for a particular goal. If visibility and lighting design is crucial in some application, the visibility matching or TR mapping method should be used. Of course absolute units rendering is a necessity in this case. On the other hand, if the lighting conditions in the scene are unusual (e.g. back light) minimum loss or interactive calibration techniques could be just the right choice.

If the correct reproduction of the true colors is important, incident light metering is the clear winner. This method reproduces selected object colors faithfully, even if the average scene reflectance is very low, or very high (and this is not the case with other mapping methods).

Contributions of this thesis to tone mapping techniques are interactive calibration, minimum information and area loss techniques, and incident light metering.

The last part of this work presented a color image difference technique that operates in the original image space. It is more intuitive than metrics that operate in transform spaces, and the color is taken into account more properly. This technique describes also how image difference depends on the viewing distance. This metric is based on the contrast sensitivity function and CIE LUV space, and it corresponds to human perception.

Currently we are trying to find some results evaluation mechanism, that will correspond to human vision.

We are also trying to find a global contrast factor which shall describe the overall impression of an image, based on human vision. Such a factor can be used

to evaluate tone mapping techniques, and images in general.

Appendix A

Raw Image File Formats

All tone mapping techniques are applied to a raw image. Unfortunately, only a few rendering packages today offer the possibility for storing a raw image. Most rendering packages apply a tone mapping, and only the final image can be stored. In this way, a lot of information is lost. We have seen how different display media have different characteristics. An image mapped once for a particular monitor, does not have to be the optimum solution even for another monitor, and especially not for a printer or some other device. If a raw image is stored, mapping can be done for each particular device.

Furthermore, when better display devices become common in the future, raw images can be mapped optimally to the new media.

The reason why raw images are rarely stored is the large storage space requirement. It is true that storing a raw image as the array of float triplets will demand a huge amount of storage place (96 bits per pixel - four times the usually necessary 24 bits). Compression techniques applied on such files do not help a lot due to the very poor entropy characteristics of such files. Fortunately there are currently three freely available raw image formats that overcome the size problem, at the cost of reduced accuracy. They are RADIANCE [Ward94a], the log format proposed by Pixar, and finally a newly proposed extension to the TIFF file format by SGI, logLuv. We will describe each of these three formats next.

A.1 Radiance RGBE Format

This raw image file format was first described by Greg Ward [Ward92]. It uses only 32 bits (4 bytes) per pixel, which makes it comparable to the usual integer file formats. Furthermore this file format is much better to compress than a naive four floats format, which brings additional space savings. The idea is to use an 8-bit mantissa for each primary and follow it with a single 8-bit exponent. The

mantissa is usually normalized in floating point formats to lie between 0.5 and 1. In this raw image format, only the largest mantissa will certainly lie in this range. Actually, this format favors the largest primary value, and has a limited dynamic range. The error introduced is negligible, since in the case of primary values differencing in few orders of magnitude, the largest primary will determine the pixel's color anyhow. The following, simple example will illustrate this idea. If color is

$$[0.3\ 0.02\ 0.1]$$

it will be converted to:

$$[0.6\ 0.04\ 0.2] \cdot 2^{-1}$$

or, in 32 bit floating point format

$$[153\ 10\ 51\ 127]$$

Note that in the above example, the exponent -1 was translated to 127. In order to cover negative exponents as well, some offset should be added to the unsigned values. In this case 128 was chosen, which reserves the same range for values greater than 1 and less than 1. It is possible to adjust this offset value if necessary, but since $2^{127} \approx 10^{38}$ it will rarely be the necessary. This file format covers about 76 orders of magnitude with 1% relative accuracy.

The complete code for encoding the images using the RADIANCE file format can be found in the *Real Pixels* article [Ward92] or on the RADIANCE WWW page [RADI97].

A.2 Pixar's Log Format

Pixar recognized the need for a raw image file format in film recording. They developed a log encoding for RGB values, such that they need 33 bits per pixel. The code is implemented as part of Sam Leffler's TIFF Library [RADI97], and it is freely available electronically. The code covers about 3.5 orders of magnitude (which is sufficient for films), with 0.4% relative accuracy.

A.3 SGI's LogLuv Format

This is the newest of the three raw image formats mentioned here. It has been developed at SGI, and is included in Sam Leffler's TIFF Library just as Pixar Log format. It uses 32 bits/pixel and it is based on human perception, which may have a big advantage in the future, when more sophisticated devices will be available.

The new proposed format uses log encoding of luminance and $CIE(u^*, v^*)$ encoding of chroma. In this way the luminance range covers 38 orders of magnitude, and chroma covers all visible colors in imperceptible steps. The relative luminance accuracy is 0.3% and chroma errors are always under the visible threshold. This format could be the one that will be extensively used in the future.

Bibliography

[ArKi90] J. Arwo, D. Kirk, “Particle Transport and Image Synthesis”, Computer Graphics, Vol. 24, num. 4, 1990.

[BaBr67] C. J. Bartelson, E. J. Breneman, “Brightness Perception In Complex Fields”, Journal of the Optical Society of america, vol. 57, No. 7, pp. 953, 1967

[Blin89] J. F. Blinn, “Dirty Pixels”, IEEE Computer Graphics and Applications, pp. 100-105, July 1989.

[Boyn92] R. M. Boynton, “Human Color Vision”, Optical Society of America, 1992.

[Crow84] F. C. Crow, “Summed-Area Tables for Texture Mapping”, Computer Graphics (Proceedings of Siggraph ’84), pp. 207-212, (1984)

[Daly93] S. Daly, “The Visible Difference Predictor: An Algorithm for the Assessment of Image Fidelity.”, in A. B. Watson, editor, Digital Images and Human Vision, pp. 179-206, MIT Press, 1993.

[Digi86] “Digitart ’86”, Catalog of Hungarian Computer Graphics Exibition, ed. ‘Delta- Impulzus, Budapest, Hungary Oct. 1986.

[Durr87] H. J. Durret, “Color and the computer”, Academic Press, San Diego, 1987

[Fair94] M. D. Fairchild, “Visual Evaluation and Evolution of the RLAB Color Space”, IS&T/SID’s 2nd Color Imaging Conference: Color Science, System and Applications, pp. 69-73, 1994.

[FeTP94] P. Ferschin, I. Tastl, W. Purgathofer, “A Comparison of techniques for the Transformation of Radiosity Values to Monitor Colors”, Proceedings of 1st IEEE International Conference on Image Processing, Austin, Texas, Nov. 1994.

[FPSG96] J. A. Ferwerda, S. N. Pattaniak, P. Shirley, D. P. Greenberg, “A Model of Visual Adaptation for Realistic Image Synthesis”, Computer Graphics, pp. 249-258, 1996.

[Fitz89] K. Fitzgerald, “Don’t move that CRT!”, IEEE Spectrum, p. 16, Dec. 1989

[Fran94] P. Franck, “The Role of Reflectance on the Perception of Brightness”, IESNA Annual Conference, Miami, Florida, 1994

[GaMY97] A. Gaddipati, R. Machiraju, R. Yagel, “Steering Image Generation with Wawelet Based Perceptual Metric”, Eurographics 1997

[GeAl89] R. S. Gentile, J. P. Allebach, “A comparison of techniques for color gamut mismatch compensation”, Human Vision, Visual Processing and Digital Display, SPIE vol. 1077, 1989.

[Giro93] B. Girod, “What’s Wrong with Mean-squared Error?”, in A. B. Watson, editor, Digital Images and Human Vision, pp. 207-220, MIT Press, 1993.

[Glas95] A.S. Glassner, “Principles of Digital Image Synthesis”, Morgan Kaufmann Publishers, Inc., (1995)

[Glas95a] A. S. Glassner, K. P. Fishkin, D. H. Marimont, M. C. Stone, “Device Directed Rendering”, *ACM Transactions on Graphics*, Vol. 14, No. 1, pp. 58-76, (Jan 1995)

[Hall89] R. Hall, “Illumination and Color in Computer Generated Imagery”, Springer-Verlag, New York, 1989

[Holl26] L. L. Holladay, Journal of the Optical Society of America, 12, 271, 1926

[HoBe93] T. Hoshino, R. S. Berns R., “Color gamut mapping techniques for color hard copy devices”, Device Independent Color Imaging and Color Imaging Systems Integration, SPIE vol. 1909, 1993.

[Hunt92] R. W. G. Hunt, “Measuring Color”, 2nd Ed., Ellis Horwood, (1992)

[JaFS95] C. E. Jacobs, A. Finkelstein, D. H. Salesin, “Fast Multiresolution Image Querying”, Computer Graphics (Proceedings of Siggraph ’95), pp. 277-286, (1995)

[JaMe66] T. H. James, E. Kenneth Mess, The theory of The Photographic Process, 3rd Ed., The Macmillan Company, New York, 1996

[KaPC93] J. K. Kawai, J. S. Painter, M. F. Cohen, “Radioptimization - Goal Based Rendering”, *Computer Graphics (Proc. SIGGRAPH)*, pp. 147-154, (1993)

[LaRP97] G. W. Larson, H. Rushmeier, C. Piatko, “A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes”,Technical Report LBNL-39882, Lawrence Berkeley National Laboratory, March 1997

[Layc83] J. Laycock, “Color Contrast Calculations for Displays Viewed in Illumination”, RAE Technical Report No. 83089, 1983.

[MaSa74] J. L. Mannos, D. J. Sakrison, “The Effects of a Visual Fidelity Criterion on the Encoding of Images”, IEEE Transactions on Information Theory, pp. 525-535, Vol. 20, No 4, (1974)

[MaNe96] K. Matković, L. Neumann, “Interactive Calibration of the Mapping of Global Illumination Values to Display Devices”, Spring Conference on Computer Graphics, Budmerice, Slovakia, 1996.

[MaNP97] K. Matković, L. Neumann, W. Purgathofer, “A Survey of Tone Mapping Techniques”, Spring Conference on Computer Graphics, Budmerice, Slovakia, 1997

[McCa94] J. J. McCann, “Scene Normalization Mechanisms in Humans”, IS&T/SID’s 2nd Color Imaging Conference: Color Science, System and Applications, pp. 5-8, 1994.

[MoSp45] P. Moon, D. Spencer, “The Visual Effects of Non-Uniform Surrounds”, Journal of the Optical Society of America, vol. 35, No. 3, pp. 233-248, 1945

[Morv84] G. Morvay, “New Photolexicon”, Mszaki Knyvkiad, Budapest, Hungary, 1984.

[Nemc80] A. Nemcsics, “COLOROID Color System”, Color Research and Application, 5, pp. 113-120, 1980

[Nemc87] A. Nemcsics, “Color Space of the Coloroid Color System”, Color Research and Application, 12, pp 135-146, 1987

[Nemc93] A. Nemcsics, “Color Dynamics”, Publisher Akadmiai Kiad, Budapest, 1993.

[NeNe88] L. Neumann, A. Neumann, “Color Dynamics Applications of Separable Element Formula”, Proceedings of Conference on Color Dynamics ’88, Budapest, Hungary, Nov. 1988.

[NeMP96] L. Neumann, K. Matković, W. Purgathofer, “Automatic Exposure in Computer Graphics Based on the Minimum Information Loss Principle”, Technical Report, TR-186-2-96-08, Institute of Computer Graphics, Vienna University of Technology, 1996.

[NMNP97] L. Neumann, K. Matković, A. Neumann, W. Purgathofer, “Incident Light Metering in Computer Graphics”, Technical Report, TR-186-2-97-11, Institute of Computer Graphics, Vienna University of Technology, 1997.

[NeMP97] L. Neumann, K. Matković, W. Purgathofer, “Perception Based Color Image Difference”, Technical Report, TR-186-2-97-21, Institute of Computer Graphics, Vienna University of Technology, 1997.

[PhPr95] B. Pham, G. Pringle, “Color Correction for an Image Sequence”, IEEE Computer Graphics and Applications, pp. 38-42, May 1995.

[PoFo95] P. Poulin, A. Fournier, “Painting Surface Characteristics”, 6th Eurographics Workshop on Rendering, pp. 119-129, Dublin, June 1995.

[RADI97] RADIANCE WWW site:
http://radsite.lbl.gov/radiance/HOME.html, 1997

[RWPSR95] H. Rushmeier, G. Ward, C. Piatko, P. Sanders, B. Rust, “Comparing Real and Synthetic Images: Some Ideas About Metrics”, Sixth Eurographics Workshop on Rendering, Dublin, Ireland, pp. 82-91, (1995)

[Schl94] C. Schlick, “Quantization Techniques for Visualization of High Dynamic Range Pictures”, 5th Eurographics Workshop on Rendering, Darmstadt, Germany, 1994.

[SDSAG93] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, D. Greenberg, “Painting with Light”, *Computer Graphics (Proc. SIGGRAPH)*, pp. 143-146, (1993)

[Schr81] G. Schroeder, “Technische Fotografie”, Vogel Verlag, Wuerzburg, Germany, 1981.

[StSt63] S. S. Stevens, J. C. Stevens, “Brightness Function: Effects on Adaptation”, Journal of Optical Society of Amerika, vol. 53, No. 3, pp. 375-385, Mar. 1963.

[ToHe89] G. Tonnquist, L. Heng, “Application of Color Video Displays to Color Research”, Displays, pp. 171-180, July 1989.

[TuRu93] J. Tumblin, H. Rushmeier, “Tone Reproduction for Realistic Images”, Computer Graphics and Applications, pp 42-48, Nov. 1993.

[TuRu91] J. Tumblin, H. Rushmeier, “Tone Reproduction for Realistic Computer Generated Images”, Technical Report GIT-GVU-91-13, Georgia Institute of Technology, 1991.

[Ward92] G. Ward, “Real Pixels”, Graphics Gems II, Ed. by J. Arvo, Academic Press, 1992

[Ward94] G. Ward, “A Contrast-Based Scalefactor for Luminance Display”, Graphics Gems IV, Ed. by P. S. Heckbert, pp. 415-421, 1994.

[Ward94a] G. Ward, “The RADIANCE Lighting Simulation and Rendering System”, Computer Graphics, Vol. 28, 1994.

[WiPo92] H. Widdel, D. L. Post, “Color in Electronic Display”, Plenum Press, 1992.

[WoAB94] M. Wolski, J. P. Allebach, C. A. Bouman, “Gamut Mapping Squeezing the Most out of Your Color System”, IS&T/SID’s 2nd Color Imaging Conference: Color Science, System and Applications, 1994.

[WySt82] G. Wyszecki, W. S. Stiles, “Color Science, Concept and Methods, Quantitive Data and Formulae”, 2nd ed., John Wiley and Sons, (1982)

[Zuid94] K. Zuiderveld, “Contrast Limited Adaptive Histogram Equalization”, Graphics Gems IV, Ed. by P. S. Heckbert, pp. 474-485, 1994.

Curiculum Vitae