

TECHNISCHE UNIVERSITÄT WIEN

DISSERTATION

Real Time Tone Mapping

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften/der Naturwissenschaften unter der Leitung von

Univ. – Prof. Dipl. – Ing. Dr. Techn. Werner Purgathofer

E 186

Institut für Computergraphik und Algorithmen

eingereicht an der Technischen Universität Wien

Fakultät für Informatik

Von

Alessandro Artusi

Matrikelnummer 0027231

Dißberggasse 10/16 1030 Wien

Wien, am

eigenhändige Unterschrift

Table of Contents

Abstract	p. 5
Zusammenfassung	p. 6
Introduction	p. 7
Chapter 1 Human Visual System (HVS)	
Introduction	p. 9
1.1 Eye	p. 9
1.2 Formation Image Process	p. 11
1.2.1 Rods and Cones Vision	p. 12
1.3 Visual Adaptation	p. 12
1.3.1 Dark Adaptation	p. 16
1.3.2 Light Adaptation	p. 16
1.3.3 Chromatic Adaptation	p. 17
1.3.4 Retinal Process	p. 17
1.4 Human Visual Effects	p. 17
1.4.1 Glare	p. 18
1.4.2 Visual Acuity	p. 20
1.4.3 Colour Sensitivity	p. 21
Chapter 2 Colour	
Introduction	p. 22
2.1 What is Colour?	p. 22
2.1.1 Additive and Subtractive Mixture	p. 23
2.1.2 The colorimetry	p. 23
2.2 Colour Spaces	p. 26
2.2.1 Device Dependent Colour Spaces	p. 26
2.2.2 Device Independent Colour spaces	p. 29
2.3 Error Colour Metrics	p. 31
Chapter 3 Output Devices	
Introduction	p. 34
3.1 Display System	p. 34
3.1.1 Cathode-Ray Tube (CRT)	p. 34
3.1.2 Liquid Crystal Display (LCD)	p. 36
3.1.3 Plasma	p. 37
3.1.4 Gamma Correction	p. 38
3.2 Printer System	p. 40
3.3 Colorimetric Characterization of Device	p. 52

Chapter 4 Tone Mapping

Introduction	p. 54
4.1 Image Synthesis	p. 54
4.2 Tone Mapping	p. 56
4.2.1 Global Operators	p. 58
4.2.2 Local Operators	p. 66
4.3 Time-Dependent Tone Mapping	p. 72
4.3.1 Pattanaik et al. 2000	p. 72
4.3.2 Durand and Dorsey 2000	p. 73
4.4 Real-Time Tone Mapping	p. 74
4.5 Tone Mapping for Printers	p. 75

Chapter 5 Novel Time-Dependent Tone Mapping Model

Introduction	p. 77
5.1 Chromatic Adaptation	p. 78
5.2 Model Development	p. 80
5.2.1 Pattanaik et al. Model 2000	p. 80
5.2.2 Novel Time-Dependent Tone Mapping Model	p. 85
5.3 Human Visual System Limitations	p. 90
5.3.1 Integration of the Human Visual System Limitations in the Time-Dependent Model	p. 90
5.3.2 Glare	p. 91
5.3.3 Visual Acuity	p. 92
5.3.4 Colour Sensitivity	p. 93

Chapter 6 Real-Time Tone Mapping Model

Introduction	p. 94
6.1 Overview	p. 95
6.2 Framework	p. 96
6.2.1 Sampling	p. 96
6.2.2 Tone Mapping	p. 98
6.2.3 Fitting	p. 98
6.2.4 Reconstruction	p. 100
6.3 Hardware	p. 101
6.3.1 Sampling	p. 101
6.3.2 Tone Mapping and Fitting	p. 102
6.3.3 Reconstruction	p. 102
6.4 Discussion	p. 103

Chapter 7 Tone Mapping for Printers

Introduction	p. 105
7.1 Framework	p. 105
7.2 Colorimetric Characterization of a Colour Printer	p. 107

7.2.1 Background	p. 109
7.2.2 Proposed Model	p. 115
7.3 Discussion	p. 120

Chapter 8 Experimental Results

Introduction	p. 122
8.1 Time-Dependent Tone Mapping Operator	p. 122
8.2 Real Time Tone Mapping Framework	p. 126
8.2.1 Summary	p. 130
8.3 Colorimetric Characterization of a Colour Printer	p. 131
8.3.1 Training and Test sets	p. 132
8.3.2 Experimental Results	p. 132
8.3.3 Summary	p. 139

Chapter 9 Conclusions and Future Work

p. 141

Bibliography

p. 144

Abstract

The dynamic range limitations of current image output devices such as monitors and printers, do not allow to visualize High Dynamic Range images correctly. Tone mapping helps to solve this problem, but real-time performances are not yet achieved. In this thesis we analyse the acceleration question about tone mapping in the context of monitors and printers. Two different solutions are presented. For monitors, we describe a framework that accelerates any complex pre-existing global tone mapping operator and its implementation on the hardware. For printers, an integration of tone mapping with a fast colorimetric characterization model is proposed. Furthermore the time-dependency characteristic of the Human Visual System is analysed, and a model improving the performances of an existing one is presented.

Zusammenfassung

Der eingeschränkte Dynamikbereich derzeitiger Bildausgabegeräte, z.B. Bildschirme und Drucker, gestattet nicht die korrekte Darstellung von Bildern mit hohem Dynamikumfang (High Dynamic Range Images). Tone Mapping hilft bei der Lösung des Problems, jedoch gestattet der hohe Rechenaufwand keine Echtzeitanwendung. In dieser Arbeit wird die Frage nach Beschleunigungsmöglichkeiten von Tone Mapping im Zusammenhang mit Bildschirmen und Druckern analysiert. Zwei verschiedene Lösungen werden präsentiert. Für Bildschirme wird ein Framework-System beschrieben, das beliebige komplexe vorhandene globale Tone Mapping-Operatoren beschleunigen kann, sowie dessen hardwarebeschleunigte Implementierung. Für Drucker wird die Integration von Tone Mapping und einem schnellen colorimetrischen Charakterisierungsmodell vorgeschlagen. Darüberhinaus wird die Zeitabhängigkeit des menschlichen Sehsinns analysiert und ein verbessertes Modell desselben vorgestellt.

Introduction

The accurate simulation of the distribution of light energy in scenes does not guarantee that the displayed (or printed) images will have a realistic visual appearance. This is due to the following reasons: first, the *range of the light energy* in the scene may be vastly different from the range that can be produced by the monitor (Low dynamic Range LDR). Second, the *visual state of the two observers* (*i.e.* scene and monitor) are completely different. Third, to capture the correct appearance of the original scene, one needs to simulate the complex behaviour of the *Human Visual System* (HVS).

Those reasons have to be considered to produce realistic images from *High Dynamic Range* (HDR) data. The process of mapping HDR data to LDR is called tone mapping (TM).

In many applications such as *real-time rendering, games, simulation etc.*, time constraints reduce the possibility to use high quality TM facilities. To have usable TM operators we need to speed up their computation also using available graphic hardware.

Numerous TM operators have been presented in the literature e.g. [Ar+03][As02][Ch+94][Co+01][Dr+03][DuDo00][DuDo02][Fa+02][Fe+96][Go+0][Na+90][Pa+98][Pa+00][Re+02][Sc+00][Sc94][TuRu93][Tu+99][TuTu99][Wa94][Wa+97]. However these operators solve just parts of the TM problem, and are usable only in some applications.

Furthermore, the current graphic hardware is more flexible and suitable for programming; however several limitations still reduce the possibility to implement on it complex algorithms. In fact, as discussed in Goodnight et al. [Go+03], troublesome aspect of GPU programming is that it requires exceedingly careful optimization in order to extract the performance we would expect. Several factors, such as: limited memory bandwidth, driver overhead (especially context-switching overhead), etc. contribute to this problem. Some of these problems can be reduced but not avoided [Go+03].

The innovation introduced by NVIDIA [In03], with a high level programming language (CG toolkit), helps the programmers to speed up the programming phase. However, the limited number of assembly instructions (1024) reduces the possibility to implement a sophisticated algorithm without any modification. Nevertheless, if we are able to overcome these limitations of the graphic hardware, the implementation on GPU of a TM operator will be a stand-alone application that considers only some aspects of the TM problem. A general and flexible system that allows us to play with any pre-existing TM operator and use it in real-time applications is yet not available.

The thesis is organized as follows: in Chapter 1 we introduce the background to understand the mechanisms of the Human Visual System. In Chapter 2 we present the definition of colour, colour spaces, and colour error metrics. An introduction on the output devices and on their colorimetric characterization is given in Chapter 3. A survey on TM is introduced in Chapter 4. A novel Time-Dependent tone mapping model is presented in Chapter 5. In Chapter 6 a general TM framework for printers is presented. Chapter 7 is devoted to the introduction of a novel real-time framework to accelerate any complex pre-existing global TM operator. Experimental results are discussed in Chapter 8. Chapter 9 is devoted to conclusions and future work.

Chapter 1

Human Visual System

Introduction

The range of light energy in the real world is vast. The Human Visual System (HVS) copes with it by adapting to the prevailing condition of illumination.

The adaptation process is complex and yet not completely known. However if we want to simulate and reproduce what a human sees, we have to take into account this process. In this chapter we present a description of the HVS and its adaptation process. In particular in Section 1, the eye structure is presented. In Section 2, we describe its processes. Visual adaptation has an important role, and it is discussed in Section 3. Finally, in Section 4 important visual effects of the HVS are discussed.

1.1 Eye

The fundamental part of the HVS is the eye. It is an organ with approximately spherical shape [WySt82], see Figure 1.1. The sphere has a radius of about 12 mm. In the protective envelope there are three pairs of extrinsic muscles that move the eye in its bony orbit. The eye consists of: *cornea*, *pupil*, *lens*, *aqueous humour*, *vitreous body* and *retina*.

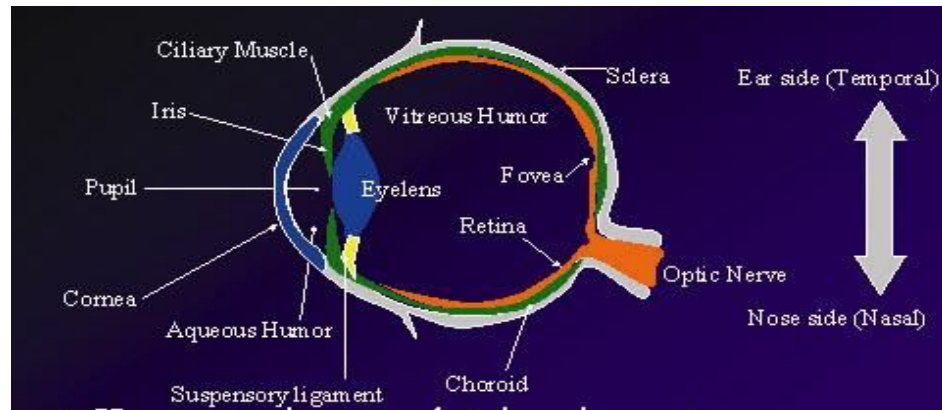


Figure 1.1: Human eye structure.

The cornea is the outermost layer through which the external stimulus enters the eye. Although of a complex lamellar structure, the cornea is transparent and void of blood vessels [WySt82]. The pupil is the hole in the iris diaphragm through which the light passes. The lens is a biconvex multilayered structure. The shape of the lens change during the accommodation and these changes occur mostly at its anterior surface, which touches the iris [WySt82]. The aqueous humour is located between cornea and lens. This liquid, continuously generated and absorbed, controls the intraocular pressure, which is greater than the atmospheric pressure, to maintain the structural integrity of the eye [WySt82]. Between the lens and the retina, there is the vitreous body, which consists of a transparent jelly interlaced with fibers [WySt82]. The retina is a complex and multilayered structure lining most *choroids*, the vascular and pigmented layer attached to the sclera, the protecting envelope of the eye [WySt82]. There are two elements of particular importance in the retina: *fovea and photoreceptor*. The fovea is a particular area of the retina where vision is most acute. There are two types of photo receptors in the retina: *cones* and *rods*. Also in the fovea the cones photo receptors are presents, but they have a different structure than the cones in the retinal region. In the fovea there is the high density of cones, this gives to the fovea the exceptional capacity to resolve the fine details in an optical image focused there. The Most central cones and each group of rods have a direct line to the brain through the inner surface of the retina and the optic nerve. In addition to these direct connections there are myriads of local cross connections in the retina [JuWy75]. Some questions about the retina are still open. Indeed, the various processes of signal generation, transmission and coding that occur in the retina when a visual stimulus enters are not yet well understood [WySt82]. The eye pigments are also important in the vision process. The eye contain different pigments: *macula, lens, blood, choroids coat, rods and cone pigments*. The nerve layer of the retina inside and near the

fovea, between the vitreous humour and the cone layer, is coloured with a yellow or brownish pigment. This brownish spot is known as macula lutea, and the pigment is called macula pigment [JuWy75]. The lens pigment is developing by the lenses and it is yellowish or brownish. This pigment prevents some of incident energy, particularly the short-wave energy, from reaching the retina [JuWy75]. The capillaries of the retina are almost opaque because of the blood pigment. In the choroids coat there is a pigment that serves to absorb the radiant energy after this passed through the rod-cone layer of the retina. Persons that are free from choroids pigment have low visual acuity and fear of light [JuWy75]. Pigments are also presents in the rods and cones photo receptors. These pigments absorb part of the radiant flux coming into the retina and the image in the retina is formed such pattern of radiant flux of varying density [JuWy75].

1.2 Formation Image Process

The formation image process in the HVS pipeline, is shown in Figure 1.2. The process can be divided in four stages: *image formation*, *exposure control*, *detection* and *processing*. The image formation is located in the cornea and the lens. Indeed the rays from an object strike the cornea and are thereby nearly focused on the retina. The lens completes the focusing. If there is a plenty of light, the *iris* diaphragm contracts so that the pupil become smaller and only the centre of the lens is used. This gives the sharpest image [JuWy75]. The radiant energy penetrating the cornea is not only brought to a sharp focus by the cornea and lens but is also modified in spectral composition by pigment, macula and lens, and transformed to nerve activity by pigments, rods and cones. Most of radiant energy is finally absorbed by pigment in the choroids coat [JuWy75]. These patterns of nerve activity are propagated from the receptors to the brain. There this information is elaborated and gives to the humans the capacity to see the image.

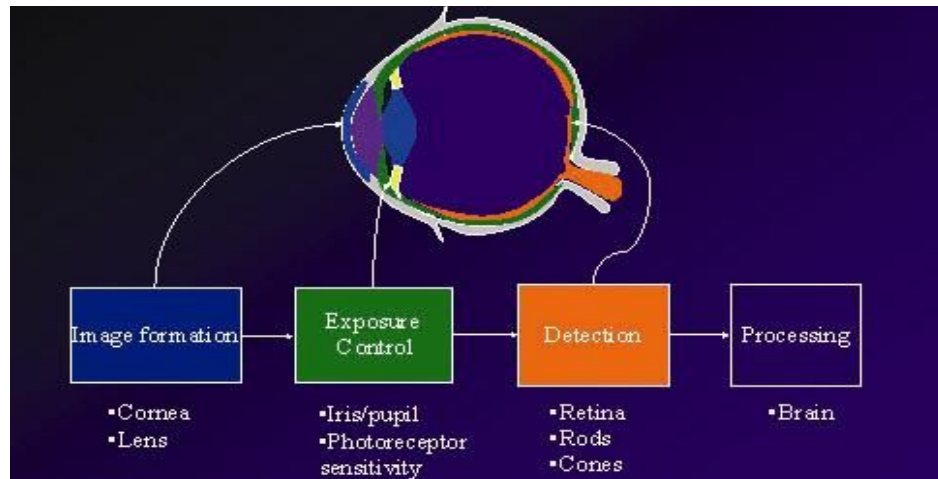


Figure 1.2: How the HVS reproduce the real image.

1.2.1 Rods and Cones Vision

The world range luminance is divided in three levels: *scotopic*, *mesopic* and *photopic*, Figure 1.3.

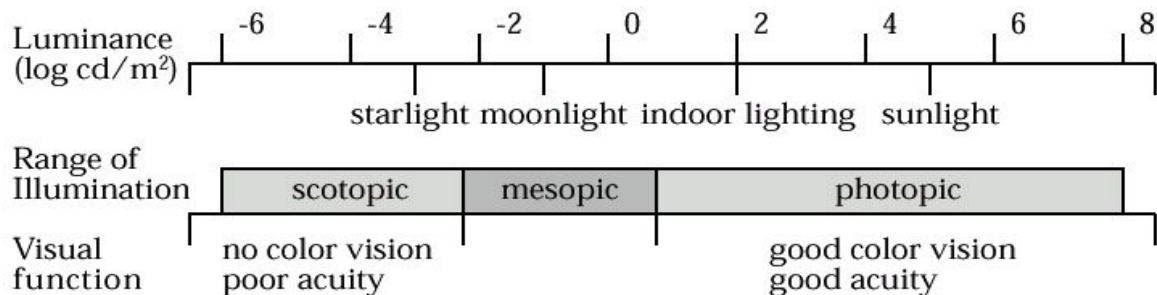


Figure 1.3: The range world luminance, after [Fe+96].

The rods are extremely sensitive to light and provide achromatic vision at scotopic levels of illumination ranging from 10^{-6} to 10 cd/m^2 . The cones are less sensitive than the rods, but provides colour vision at photopic levels of illumination in the range of 0.01 to 10^8 cd/m^2 . At light levels from 0.01 to 10 cd/m^2 both the rod and cone systems are active. This is known as the mesopic range [Fe+96].

1.3 Visual Adaptation

The process by which the HVS adjusts itself to the conditions, under which the eyes are exposed to radiant energy, is called *adaptation* [JuWy75]. Adaptation is achieved through the coordinated action of mechanical, photochemical, and neural processes in the visual system. Adaptation can also be described through the physiological phenomenon's happening inside the HVS. In fact several parts of the HVS such as pupil, the rods and cones systems,

bleaching and regeneration of receptor photo pigments, and changes in neural processing play a role in visual adaptation [Fe+96]. Through these four adaptation mechanisms, the HVS is able to work over a luminance range of nearly 14 log units; despite the fact that the individual neural units, which make up the system, have a response range of only about 1.5 log units [SpWe90]. With these mechanisms, the visual system moderates the effects of changing levels of illumination on visual response, providing sensitivity over a wide range of ambient light levels [Fe+96]. The pupil modifications in diameter according to the changes in luminance, as described in Section 1.1, produces about a log unit change in retinal luminance. This small change does not allow the pupil action to be sufficient to completely account for visual adaptation [SpWE90].

At high light intensities, the action of light depletes the photosensitive pigments in the rods and cones at a faster rate than chemical processes can restore them. This makes the receptors less sensitive to light. This process is known as *pigment bleaching* [Fe+96]. The neural response produced by a photo receptor cell depends on chemical reactions produced by the action of light on the cell's photo pigments [Fe+96]. The cell's response to light is limited by the maximum rate and intensity of these chemical reactions. If the reactions are occurring near their maximum levels, and the amount of light striking the photo pigments is increased, the cell may not be able to fully signal the increase. This situation is known as *saturation* process [Fe+96].

The result of saturation is a *response compression*: above a certain level incremental increases in light intensity will produce smaller and smaller changes in the cell's response rate [Fe+96]. As described in Section 1.2, the rods and cone photo receptors are connected through a network of neurons in the retina. Here an adaptive process that adjusts the base activity and gains the visual system to mitigate the effects of response compression in the photo receptors, take place.

These adaptive processes are subdivided in two classes: *multiplicative* and *subtractive*. The multiplicative process adjusts the gain of the system by effectively scaling the input by a constant related to the background luminance. This process acts very rapidly and accounts for changes in sensitivity over the first few seconds of adaptation [Fe+96].

A slower acting subtractive process reduces the base level of activity in the system caused by a constant background. This process accounts for the slow improvements in sensitivity measured over minutes of adaptation [Ad82]. The action of these mechanisms is reflected in the changes in visibility, colour appearance, visual acuity, and sensitivity over the time, and

can be observed in everyday experience and measured in psychophysical experiments [Fe+96]. A visual threshold is defined by the probability p of seeing a difference in brightness, hue, saturation, or a difference in some other attributes exhibited by given set of stimuli [WySt82]. The effects of adaptation have been measured in threshold experiments. Figure 1.4 shows the results of a threshold experiment that measured the changes in visibility that occur with changes in the level of illumination [Pa+98].

These curves are known as threshold-vs.-intensity (TVI) functions, and describe the dependence of the threshold from the background luminance. Over a wide range of background luminance, the size of the threshold increment increases in proportion to the background luminance making the functions linear on a log-log scale. This linear relationship $\Delta L = kL$ is known as *Weber's law* and indicates that the visual system has constant contrast sensitivity since the Weber contrast $\Delta L / L$ is constant over this range [Pa+98].

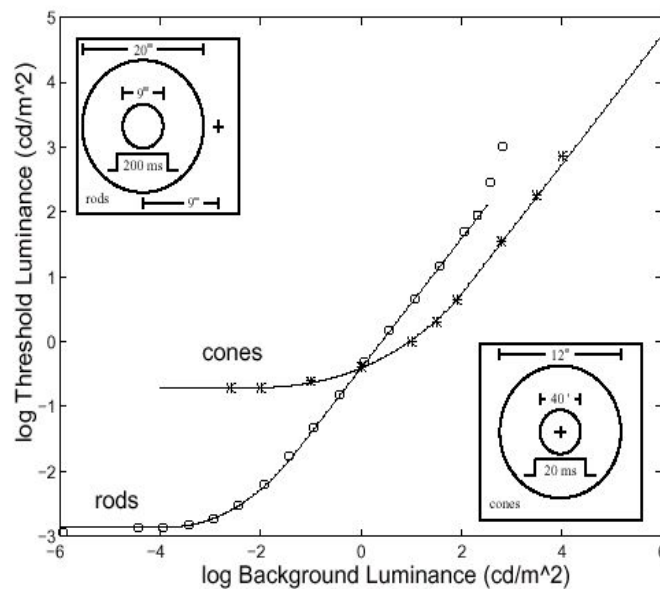


Figure 1.4: A psychophysical model of detection thresholds over the full range of vision (TVI), after [Fe+96].

In Figure 1.5, the spectral sensitivities of the rods and cones systems at the different luminance levels are shown. They are described by the scotopic, mesopic and photopic luminous efficiency functions. Figure 1.5 (a) shows the visual system spectral sensitivity at scotopic levels. At these levels detection is dominated by the rod system. Absolute sensitivity is quite high, but since the rod system is achromatic, colour will not be apparent [Fe+96]. At the mesopic levels the rods and cones systems are nearly equal in absolute sensitivity, Figure 1.5 (b). In Figure 1.5 (c), the behaviour of the visual system at photopic level is shown. At this level the cones system is dominant and the absolute sensitivity has dropped. In this case

the colour vision is active. Figure 1.6 shows the luminous efficiency functions as surface positioned with respect to the rod and cone system threshold sensitivities at different luminance levels [Fe+96]. These changes in spectral sensitivity with changing luminance level can account for a number of different colour appearance phenomenon's observed over the scotopic and photopic range. As the luminance level is raised into the mesopic range, the cone system will become active and the colours will begin to be see beginning with the long wavelength reds and progressing toward the middle wavelength greens. Only at relatively high luminances will short wavelength blue targets begin to appear coloured [Fe+96]. The changes in visual acuity are presented in Subsection 1.4.2.

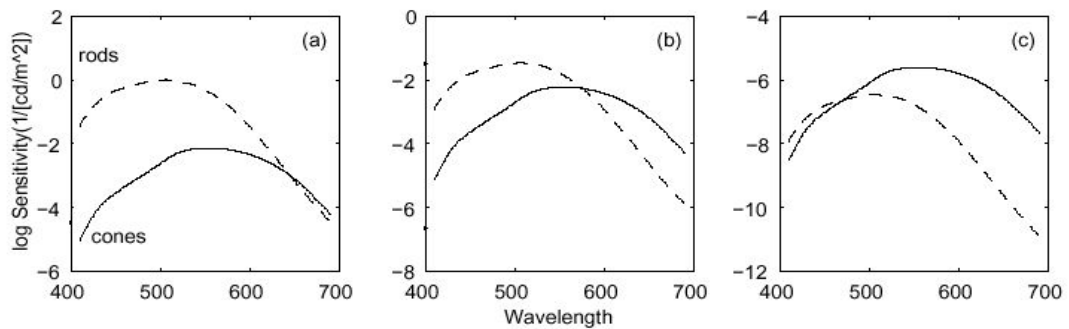


Figure 1.5: Changes in the spectral sensitivity of the visual system at (a) scotopic, (b) mesopic, and (c) photopic illumination, after [Fe+96].

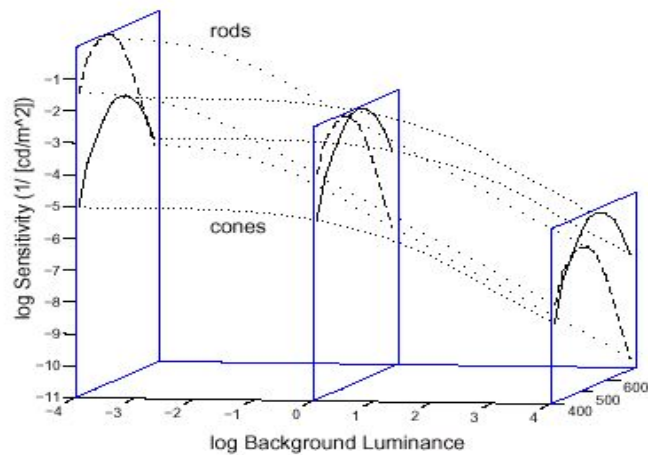


Figure 1.6: A model of threshold sensitivity as a function of wavelength and background luminance for the rods and cone systems, after [Fe+96].

The most important kinds of adaptation are: *dark*, *light* and *chromatic*. Dark and Light adaptations refer to the adjustment of the visual mechanism to changes in the rate at which radiant energy enters the eye. In contrast the chromatic adaptation refers primarily to the

adjustment of the visual mechanism to changes in its spectral distribution [JuWy75]. The adaptation does not happen instantaneously, in fact it takes time. The time is not the same for all kinds of adaptation. Indeed, the time for dark adaptation is different than that for light adaptation. These kinds of adaptation will be discussed in the next subsections.

1.3.1 Dark Adaptation

The dark adaptation increases the visual sensitivity experienced when luminance level decreases. The time occurring to realize the dark adaptation has been measured by Hecht [He34], see Figure 1.7. Also for the dark adaptation the compartment for cones and rods systems is different as shown in the graph. The visual system is completely adapted after 35 minutes.

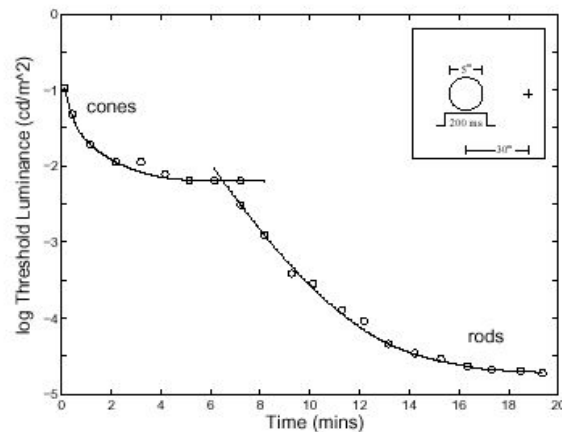


Figure 1.7: The time course of dark adaptation, after [Fe+96].

Visually, dark adaptation is experienced as the temporary blindness that occurs when we go rapidly from photopic to scotopic levels of illumination. The slow time adaptation occurs for the dark adaptation means that vision can be impaired for several minutes when we move quickly from high illumination levels to low ones [Fe+96].

1.3.2 Light Adaptation

Light adaptation is the decrease in visual sensitivity upon increases in the overall level of illumination [Fa98].

The compartment for light adaptation for the rods and cones systems is different as shown in Figure 1.8. The graph shows that the light adaptation of the rods system in the scotopic range is extremely rapid, it needs only 2 seconds. In contrast for the cone system the light adaptation is slower than the rods system.

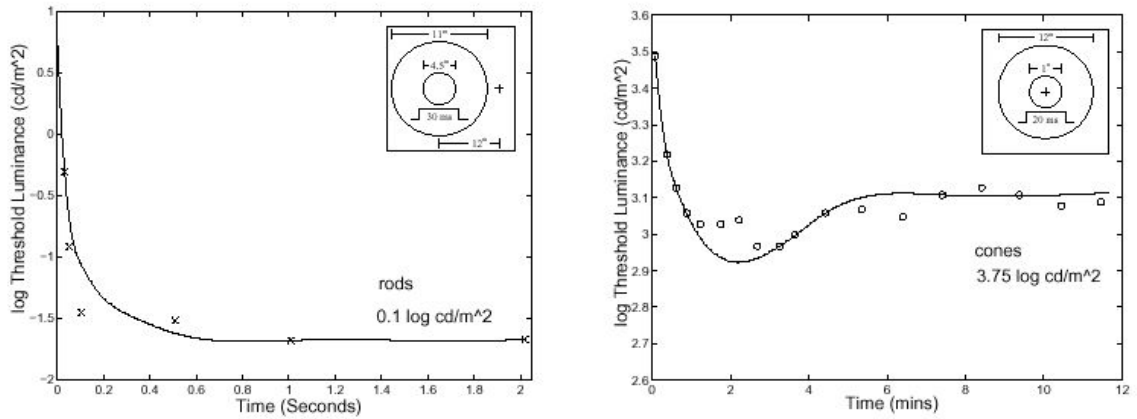


Figure 1.8: The time course of light adaptation in the cone and rods system, after [Fe+96].

Visually, light adaptation provides a distinctive experience [Fe+96]. When we go out quickly from low to high levels of illumination, at first everything is painfully glaring and we squint or close one eye to reduce the discomfort. However over time the overall brightness of the visual field diminishes to more comfortable levels and normal vision is restored [Fe+96].

1.3.3 Chromatic Adaptation

The last important kind of visual adaptation is called chromatic adaptation. It is defined as the HVS capability to adjust to widely varying colours of illumination in order to approximately preserve the appearance of object colours [Fa98].

1.3.4 Retinal Process

The mechanisms which control the time-dependent adaptation to varying luminance conditions occur inside the retina. The majority of the retinal cells can perceive only a small range of luminance values, compared to the entire luminance interval present in a scene. This range is adjusted continuously, to adapt to the light. Equation 1.1 below describes this process [Ar+01](refer also to [Pa+00]).

$$R(I) = R_{\max} \frac{I^n}{I^n + \sigma^n}; \quad 1.1$$

where I is the light intensity, R is the neural response ($0 < R < R_{\max}$), the constant σ is the value I that causes the half-maximum response, and n is a sensitivity control.

1.4 Human Visual Effects

The HVS present, besides at the visual adaptation capacity, some other effects: *glare*, *colour sensitivity*, and *visual acuity*. These effects are responsible of the loss capacity in some

circumstances, such as losing the ability to define fine details or distinguish colours in dark background, or to loss the ability to distinguish objects in particular luminance conditions.

1.4.1 Glare

The glare effect consists in the sensation that we proof when observe an incandescent bulb such as a lamp. In this case we have an impression of great brightness and interferences in the visibility of the object near the bulb. The glare effect can be subdivided in two components [Sp+95]: *flare* and *bloom*. The flare is composed of a *lenticular halo* and *ciliary corona*, and is primarily caused by the lens, Figure 1.10. Scattering causes the bloom from three parts of the visual system: the cornea, lens, and retina Figure 1.9.

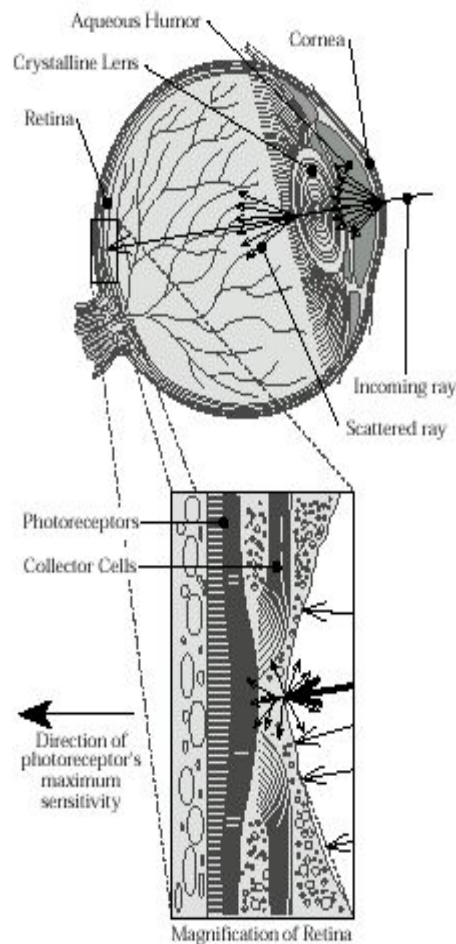


Figure 1.9: Scattering in the eye, after [Sp+95].

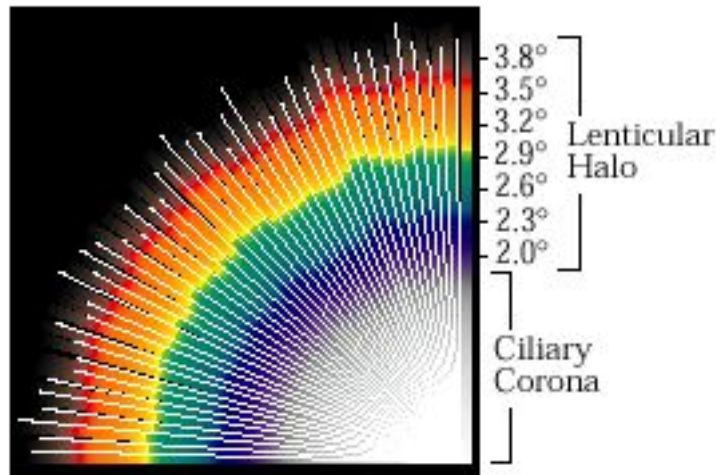


Figure 1.10: The Flare, after [Sp+95].

Lenticular Halo

The lenticular halo appears such a series of concentric coloured rings around the source [Sp+95], when we observe a point source of light in a dark surround. This phenomenon creates the illusion that the haloes around distant light sources appear larger than haloes around nearby sources Figure 1.11. The lenticular halo is caused by the circular optical grating formed by the radial fibers at the periphery of the crystalline lens [Si+53].

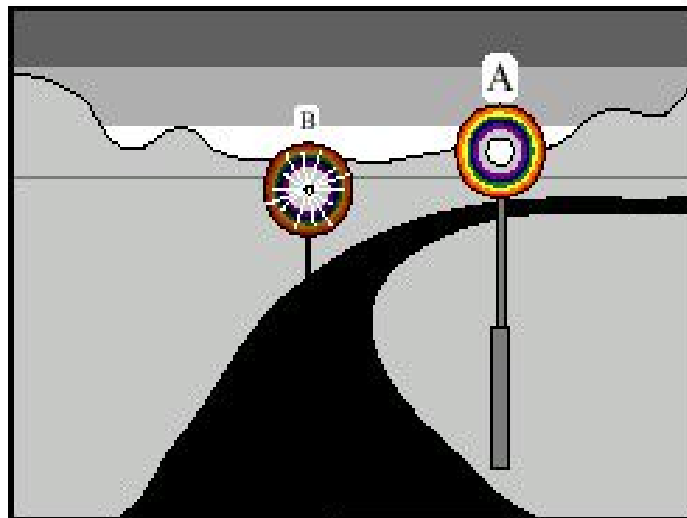


Figure 1.11: Example of lenticular halo, and his effects, after [Sp+95].

The Ciliary Corona

The ciliary corona consists of rays emanating from a point light source [Sp+95], Figure 1.10. These radial rays may extend beyond the lenticular halo, and are brighter and more pronounced as the angle subtended by the source decreases. The ciliary corona is caused by semi-random density fluctuation in the nucleus of the lens, which causes forward scattering that is independent of wavelength [He+92].

Bloom

The scattering of light in the ocular media causes the bloom, where the scatter contributions from the cornea, crystalline, lens, and retina occur in roughly equal portions [Sp+95]. In Figure 1.12 the bloom effect, also called *veiling luminance*, is illustrated. The light from the source A scatters inside the eye and is added to light coming from object B . This scattered light adds an effective luminance s that does not originate at B . Because the light is added to

both the light and dark parts of object B , the contrast ratio $\frac{L_2}{L_1}$ is reduced. The magnitude of glare is greater in the scotopic level [Sp+95]; this because the rods sensitivity does not has as high a directional sensitivity as the cone.

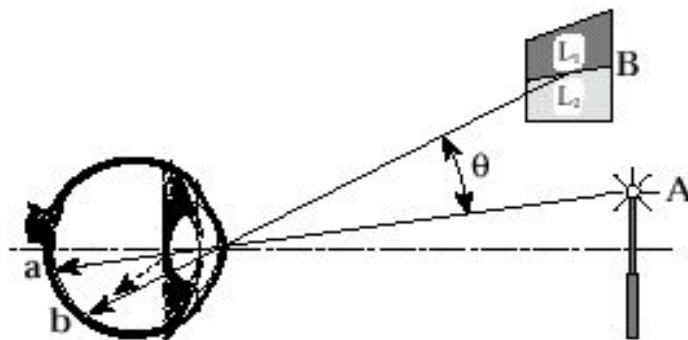


Figure 1.12: Reduction in contrast caused by bloom, after [Sp+95].

1.4.2 Visual acuity

The acuity is a measure of the HVS ability to resolve spatial details [Fe+96]. The human eye loses its ability to resolve fine detail in dark environments [Wa+97]. Shaler [Sh37] has studied the relationship between the adaptation level and the foveal acuity. The original Shaler's data is shown in Figure 1.13. This curve shows that for low adaptation luminance level, dark background, the visual acuity is low and the human eye is not able to define the

fine details. In contrast, with the increase of the adaptation luminance level increases also the visual acuity and the ability of the human eye to define the fine details. The equation 1.2 is the mathematical description of this relationship:

$$R_c(L_{rwa}) = 17.25 \tanh(1.4 \log(L_{rwa}) + 0.35) + 25.72 \quad , \quad 1.2$$

where L_{rwa} is the local adaptation luminance in cd/m^2 , for the real world, and $R_c(L_{rwa})$ is the visual acuity in *cycles/degree*. In this way it is possible to predict the visibility details, visual acuity, at different levels of illumination (local adaptation luminance).

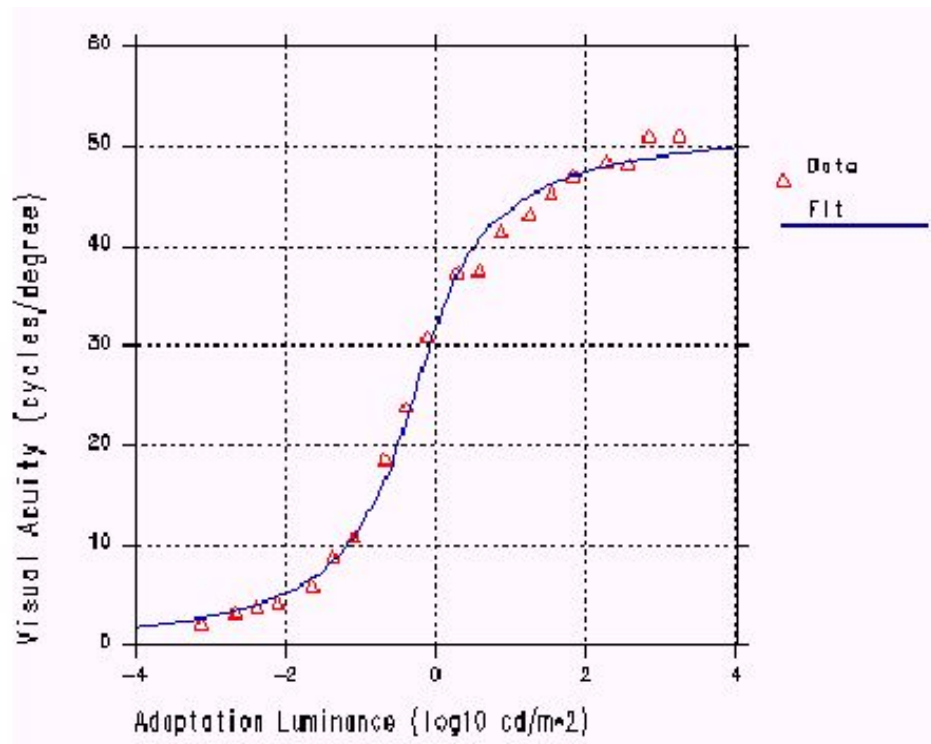


Figure 1.13: Visual acuity in function the background luminance from Shaler’s data, after [Fe+96].

1.4.3 Colour Sensitivity

Besides losing the ability to resolve fine details in dark environments, the human eye loses its capacity to see colour in dark environments. Indeed, in the scotopic range luminance, only the rods receptors are active. These receptors, as described in Section 1.2, are not responsible of colour vision but only of achromatic vision.

Chapter 2

Colour

Introduction

The colour is the physiological sensation that the HVS produces when the eye is stroked by a light reflected, emitted, or transmitted by an object. In this Chapter, we analyse the nature of colour, its attributes, and the factors that influence its appearance. Furthermore we describe the concepts of *colour model (colour space)* and *error metric*.

2.1 What is colour?

The colour is often identified as a characteristic of a given object. This is however not correct. The colour can be defined as a result of our physiological perception. This depends on several factors as material properties, conditions of the observer, characteristics of the visual system, and neural processes. In other words, the colour is the result of the interpretation of the data, in our brain, collected by the visual system. In this way the *stimulus* is defined as the visible radiation that strikes the eye of HVS. The answer of HVS to this stimulus is called *colour answer*. The source that generates the stimulus is called *stimulus object*.

In the colorimetry context the colour is associated with the stimulus instead with the answer. The perception of colour is defined as the *colour sensation*, and the term colour is used to identify a characteristic of the stimulus.

The attributes of colour, that are perceived when it is observed, are: *hue, saturation* and *lightness*. The hue is the attribute of colour by means of which a colour is perceived to be

red, yellow, green, blue, purple, etc. Pure white, black, and greys possess no hue. Colours that have hue are called *chromatic*, *achromatic* otherwise. Saturation is the colourfulness of an area judged in proportion to its brightness [Fa98]. Lightness is the brightness of an area judged relative to the brightness of a similarly illuminated area that appears to be white or highly transmitting [Fa98].

2.1.1 Additive and Subtractive

The colour can be produced combining or mixing three basic colours called *primary*. Two basic techniques are used to reproduce colour: *additive* and *subtractive*.

The additive technique is based on the concept that the colours can be added to get a new one see Figure 2.1 (a). This is the way in which the visual system mixes colours. Additive colour mixture is also the technique used to reproduce colours on a monitor.

The subtractive technique is based on the selective removal of wavelengths from light to produce a different colour. This is characteristic of a printer system, see Figure 2.1 (b).

The primaries colours are referred as additive primaries, for the additive mixture, and subtractive primaries, for the subtractive mixture techniques, respectively.

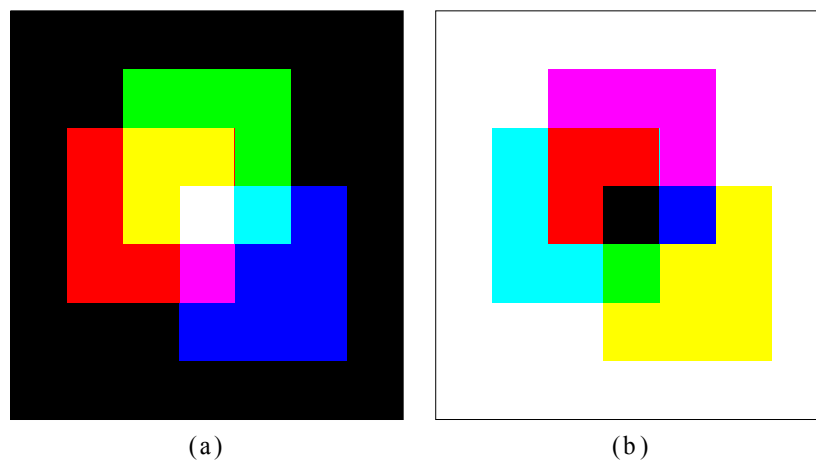


Figure 2.1: Additive mixture (a), subtractive mixture (b).

2.1.2 The colorimetry

In real applications in order to use the colour, one has to represent it in numerical terms. This is the main goal of a science called colorimetry. The guide authority in colorimetry, is the *Commission Internationale de l'Eclairage (CIE)* that is an international institute which is in charge to define the standards and the procedure for using colour in colorimetry applications. In colorimetry the sensation of colour, from a stimulus of arbitrary spectral composition, is described by three values called *tristimulus values* [Hu84]. They identify identical colours

perceived and derive the quantities correlated with the perceived attributes that describe the colour appearance.

In 1931 the CIE introduced a set of standard stimulus and the quantities of these stimuli to reproduce all colours of the visible spectrum by additive mixture. These data define the *Standard Colourimetric Observer 1931*. In other words they describe the characteristics of an average observer with a visual field between one and four degrees.

The primaries which are used are three achromatic lights: Red (700 nm), Green (546.1 nm), and Blue (435.8 nm). The *colour matching functions* $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ and $\bar{b}(\lambda)$ (see Figure 2.2), are used to obtain the tristimulus values of any colour stimulus starting from its power spectral distribution.

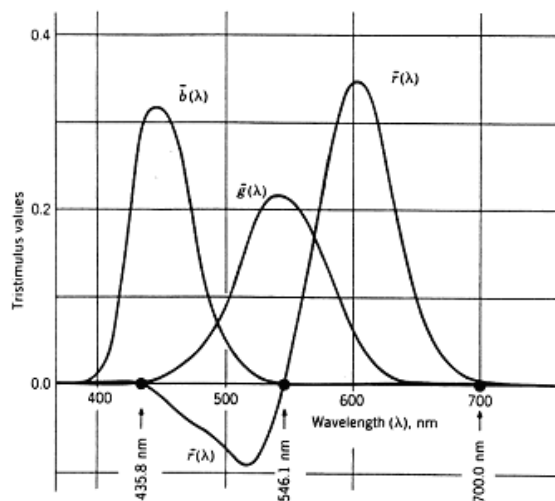


Figure 2.2: Colour matching functions of the primaries R, G, and B.

The *chromaticity coordinates* are derived from the tristimulus values. They are the quantities related to the three primary stimuli needed to reproduce any colour [Ag79].

For several reasons the CIE decided to adopt a model where any colour can be described by positive values [Hu84]. To this purpose, the CIE introduced the *imaginary primaries* X, Y, and Z.

In the same way, the colour matching functions have been introduced for the new primaries $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ (see Figure 2.3).

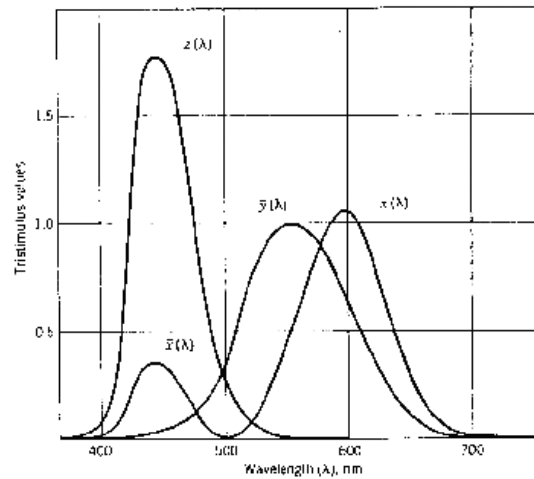


Figure 2.3: Colour matching functions of the new primaries X, Y, and Z.

An important diagram can be derived from the chromaticity coordinates, see Figure 2.4, and it is called *chromaticity diagram xy CIE 1931*.

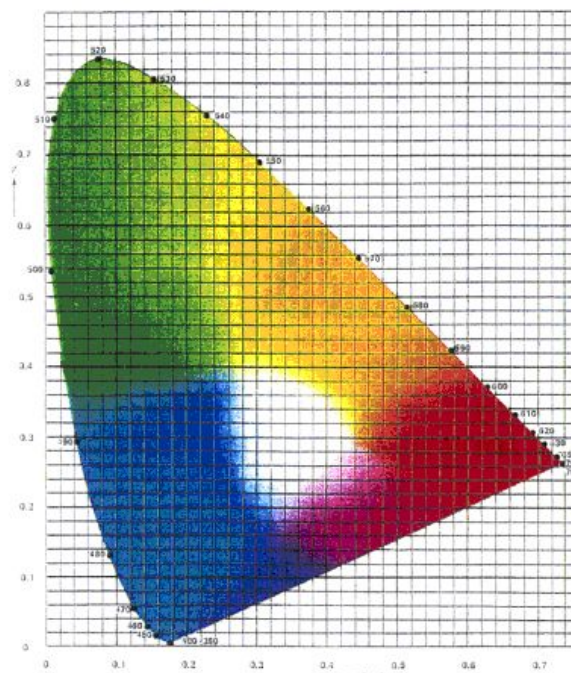


Figure 2.4: Chromaticity diagram xy CIE 1931.

A remarkable limitation of this diagram is its perceptual non-uniformity, which means that for colour pairs represented by the same Euclidean distance, the perceived differences can be rather different.

2.2 Colour Spaces

As mentioned in Section 2.1, three coordinates or parameters are used to describe the colour information. These parameters describe the position of a colour in a three dimensional space called *colour space*. The colour spaces can be divided in two main categories: *device dependent* and *device independent*. In device dependent colour spaces, the description of colour information is related to the characteristics of a particular device (input or output). For example, in a monitor it depends on the set of primaries phosphors, while in an ink-jet printer on the set of primary inks. This means that a colour (e.g. R=250, G=20, B=150) will appear different when represented on different monitors.

On the contrary a device independent colour space is not dependent from the characteristics of a particular device. This means that a colour represented in this colour space always corresponds to the same colour information.

2.2.1 Device Dependent Colour Spaces

RGB

The RGB colour space is a Cartesian cube represented by three additive primaries *Red*, *Green*, and *Blue*. The grey scale is the diagonal from black (0,0,0) to white (1,1,1), while the three colour channels RGB are located on the three Cartesian axes, see Figure 2.5. The colour space RGB is used for describing the colour space in the monitor, scanner and digital camera.

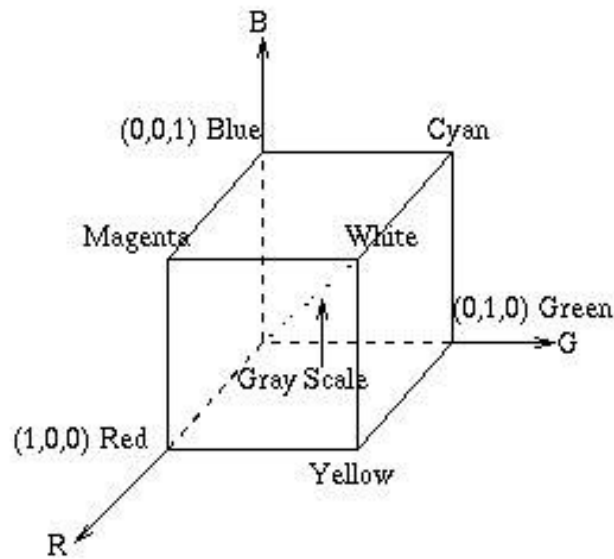


Figure 2.5: RGB colour space.

The RGB colour space is not a uniform colour space, and is not correlated with the human visual perception (is not linear with the visual perception [FoRo98]). Indeed, variation of the same degree in the range $[0, 1]$ does not produce always the same variation on the perceived colour. The RGB colour space is very common in television, video etc. [FoRo98].

CMY

The structure and the metric of the colour space CMY is similar to the structure and metric of the RGB colour space (see Figure 2.5). The colour is represented by a vector of three components, and these components are the subtractive primaries: *Cyan*, *Magenta* and *Yellow*. For this reason it is also called subtractive colour space.

In the interval $[0, 1]$, the colour specification in the colour space CMY can be obtained by its specification in RGB colour space following the linear transformations below:

$$C = 1 - R \quad M = 1 - G \quad Y = 1 - B . \quad 2.1$$

The CMY colour space has the same limitations of the RGB colour space.

HSL (Hue Saturation and Lightness)

This represents a wealth of similar colour spaces, alternative names include HIS (intensity), HSV (value), HCI (chroma/colourfulness), HVC, TSD (hue saturation and darkness) [FoRo98].

Most of these colour spaces are linear transform from RGB and are therefore device dependent and perceptually non-uniform [FoRo98].

YIQ, YUV, YCbCr, YCC (Luminance-Chrominance)

These are the television transmission colour spaces, sometime known as transmission primaries. YIQ and YUV are analogue spaces for NTSC and PAL system respectively while YcbYcr is a digital standard [FoRo98]. These colours spaces separate RGB into luminance and chrominance information and are useful in compression applications. These spaces are device dependent but are intended for use under strictly defined conditions within closed systems [FoRo98]. Kodak uses a derivative of YCC in its PhotoCD system, called PhotoYCC [FoRo98].

SRGB

Hewlett-Packard and Microsoft propose the addition of support for a standard colour space, sRGB, within the Microsoft operating systems, HP products, the Internet, and all other interested vendors. The aim of this colour space is to complement the current colour management strategies by enabling a third method of handling colour in the operating systems, device drivers, and the Internet that utilizes a simple and robust device independent colour definition. This will provide good quality and backward compatibility with minimum transmission and system overhead. Based on a calibrated colorimetric RGB colour space well suited to Cathode Ray Tube (CRT) monitors, television, scanners, digital cameras, and printing systems, such a space can be supported with minimum cost to software and hardware vendors [St+96].

The standard is divided in two parts: the viewing environment parameters with its dependencies on the HVS and the standard device space colorimetric definitions and transformations. The viewing environment descriptions contain all the necessary information, when combined with most colour appearance models, providing conversions between the standard and target viewing environments (see for more details [St+96]).

The colorimetric definitions provide the transforms necessary to convert between the sRGB colour space and the CIEXYZ two-degree observer colour space.

sRGB in combination with the reference viewing environments can be defined from standard CIE colorimetric values through simple mathematical transformations. CIE colorimetry provides the basis for sRGB encoding of colour. For the calculation of CIE colorimetric values, it is necessary to specify a viewing environment and a set of spectral sensitivities for a specific capture device [St+96]. An example of colour gamut of the sRGB colour space is showed in Figure 2.6.

CIE Diagram - 1931 2 Degree Standard Observer

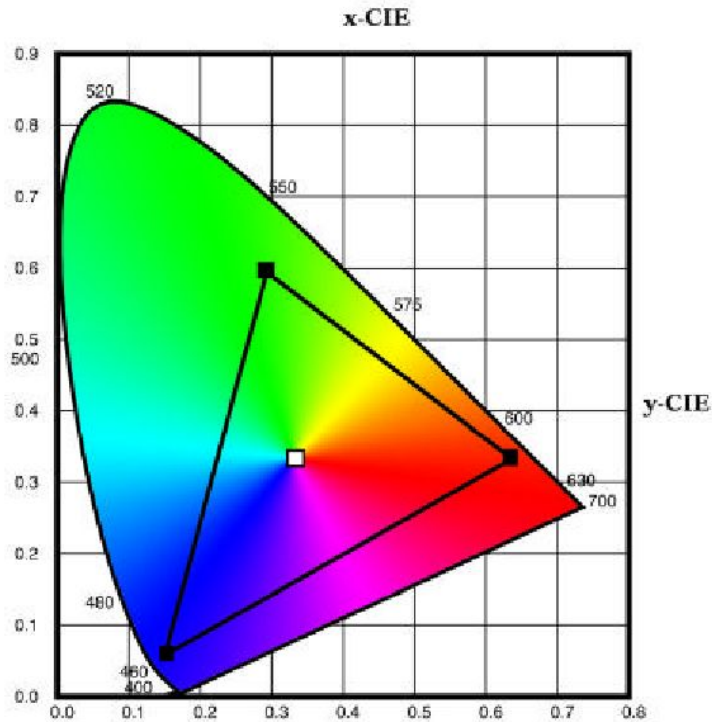


Figure 2.6: sRGB colour gamut, after [St98].

2.2.2 Device Independent Colour Spaces

CIE XYZ

The *CIE XYZ* colour space has been introduced by the *CIE* in 1931 and the all visible colours can be defined using only positive values. The *Y* value represents the luminance.

The chromaticity diagram is highly non-linear; for instance a vector of unit magnitude representing the difference between two chromaticities is not uniformly visible. A colour that is defined in this system is referred as Y_{xy} [FoRo98]. A third coordinate, z can also be defined but is redundant since $x + y + z = 1$ for all colours.

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

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

2.2

CIELab, CIELuv

The *CIELab* and *CIELuv* have been introduced by *CIE* in order to have linear colour spaces with visual perception. They are device independent colour spaces but suffer from being

quite unintuitive despite the L parameter having a good correlation with perceived lightness [FoRo98].

CIELab colour space is based on the CIE XYZ and is another attempt to linearise the perceptibility of unit vector colour differences. Colouring information is referred to the colour of white point of the system [FoRo98].

$$L = 116 \cdot \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 \quad \text{if } \frac{Y}{Y_n} > 0.008856, \quad 2.3$$

$$L = 903.3 \cdot \left(\frac{Y}{Y_n} \right) \quad \text{if } \frac{Y}{Y_n} \leq 0.008856,$$

$$a = 500 \cdot \left(f\left(\frac{X}{X_n} \right) - f\left(\frac{Y}{Y_n} \right) \right), \quad 2.4$$

$$b = 200 \cdot \left(f\left(\frac{Y}{Y_n} \right) - f\left(\frac{Z}{Z_n} \right) \right), \quad 2.5$$

$$f(t) = t^{\frac{1}{3}} \quad \text{if } t > 0.008856, \quad 2.6$$

$$f(t) = 7.787 \cdot t + 16 \frac{t^2}{116} \quad \text{if } t \leq 0.008856. \quad 2.7$$

The value L is scales from 0 to 100, and $X_n Y_n Z_n$ is the white point of the system. There are polar parameters (Chroma C , and Hue h) that more closely match the visual experience of colours [FoRo98].

$$C = (a^2 + b^2)^{0.5}, \quad 2.8$$

$$h_{ab} = \arctan\left(\frac{b}{a}\right). \quad 2.9$$

Hue is angle in four quadrants, and there is no saturation term in this system [FoRo98]. Also the colouring information, for the colour space CIELuv, is centred on the colour of the white point of the system.

$$L = 116 \cdot \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 \quad \text{if } \frac{Y}{Y_n} > 0.008856, \quad 2.10$$

$$L = 903.3 \cdot \left(\frac{Y}{Y_n} \right) \quad \text{if } \frac{Y}{Y_n} \leq 0.008856,$$

$$u = 13 \cdot L \cdot (u' - u'_n), \quad 2.11$$

$$v = 13 \cdot L \cdot (v' - v'_n). \quad 2.12$$

Also in this case L is scales from 0 to 100 for relative luminance $\left(\frac{Y}{Y_n}\right)$ scaling 0 to 1

[FoRo98]. The values of u' and v' are computed with the follow equations:

$$u' = \frac{2 \cdot x}{6 \cdot y - x + 15}, \quad 2.13$$

$$v' = \frac{4.5 \cdot x}{6 \cdot y - x + 15}. \quad 2.14$$

There are three polar parameters, which match more closely to the human visual experience than the Cartesian parameters [FoRo98].

$$C = (u'^2 + v'^2)^{0.5}, \quad 2.15$$

$$h_{uv} = \arctan\left(\frac{v'}{u'}\right), \quad 2.16$$

$$s_{uv} = \frac{C}{L}. \quad 2.17$$

The parameter s_{uv} represents the saturation.

2.3 Error Colour Metrics

The aim of the *error colour metric* is to define a strategy, in order to give some information about the colour differences between an original and a reproduction of it.

We can subdivide the *error colour metrics* in two categories: *single pair for colour patches* and *for digital imaging*. The first one is referred to a comparison between two single colour patches, without considering the appearance phenomena related to the spatial position and the context where two or more colours are positioned inside a digital image. The second one, takes into account these appearance phenomenons and is used to have an instrument for understanding the goodness colour reproduction of a digital colour image.

In this section we follow the guidelines of the CIE technical report [CIE00], and a set of average colour differences is described.

The CIE 1997 Colour appearance model (*CIECAM97s*) is capable to predict a number of visual phenomena including [CIE00]:

- The chromatic adaptation effect which refers to the prediction of the corresponding colours under a reference illuminant.
- The Hunt effect, which refers to an increase in colourfulness (or chroma) due to increase of luminance.

- The Stevens effect, which refers to an increase in brightness (or lightness) contrast with increasing luminance.
- The Contrast-media effect, which refers to an increase of the perceived contrast with increasing luminance level from dark (typical projection viewing) to dim (CRT viewing) and to light (reflection viewing).

The CIE propose that should be possible to use the CIECAM97s as the basis of a colour difference matrix for imaging applications [CIE00]. Also the LLAB model, derived by Luo and Guann [LuGu99], is able to predict various phenomena similar to the CIECAM97s. Several researchers tried to achieve an universally agreed colour difference formula, and these studies are mostly based on colour patches [CIE00]: *CIELAB* and *CIELUV*, *CMC(l:c)*, *BFD(l:c)*, *CIE94*, *Leeds Colour Difference Formula (LCD)*, *CIEDE2000*.

CIELab and CIELuv Colour Difference Formula

The characteristics of these two colour spaces are already described in Section 2.2.2, and the colour difference formula is defined as (for the CIELab colour space):

$$\Delta E_{ab}^* = (\Delta L^2 + \Delta a^2 + \Delta b^2)^{0.5}. \quad 2.18$$

Instead for the colour space CIELuv is:

$$\Delta E_{uv}^* = (\Delta L^2 + \Delta u^2 + \Delta v^2)^{0.5}. \quad 2.19$$

CMC(l:c)

The colour is denoted in CIElab colour space, and the formula is given below [CIE00]:

$$\Delta E = \left[\left(\frac{\Delta L_{ab}}{l \cdot S_L} \right)^2 + \left(\frac{\Delta C_{ab}}{c \cdot S_c} \right)^2 + \left(\frac{\Delta h_{ab}}{S_h} \right)^2 \right]^{0.5}. \quad 2.20$$

where $S_L = \frac{0.040975 \cdot L_{ab,1}}{(1 + 0.01765 \cdot L_{ab,1})}$, unless $L_{ab,1} < 16$ when $S_L = 0.511$. About the other

parameters:

$$S_c = \frac{0.0638 \cdot C_{ab,1}}{(1 + 0.0131 \cdot C_{ab,1})} + 0.638, \quad 2.21$$

$$S_h = S_c \cdot (T \cdot f + 1 - f). \quad 2.22$$

Where f and T are:

$$f = \left[\frac{C_{ab,1}^4}{(+1900)} \right]^{0.5}, \text{ and } T = 0.36 + \left| 0.4 \cdot \cos(h_{ab,1} + 35) \right|, \quad 2.23$$

unless $h_{ab,1}$ is between 164° and 345° when $T = 0.56 + |0.2 \cdot \cos(h_{ab,1} + 168)|$. The $L_{ab,1}$, $C_{ab,1}$ and $h_{ab,1}$ refer to the standard of a pair of samples. The weight l and c , lightness and chroma weight respectively, should equal two and one for predicting the acceptability of colour differences. For predicting the perceptibility of colour differences, l and c should be equal to one.

BFD(l:c)

The formula has been derived by Luo and Rigg [LuRi87] and has a similar structure to that of $CMC(l:c)$ [CIE00]. It is given below:

$$\Delta E = \left[\left(\frac{\Delta L_{BFD}}{l} \right)^2 + \left(\frac{\Delta C_{ab}}{c \cdot D_c} \right)^2 + \left(\frac{\Delta h_{ab}}{D_h} \right)^2 + R_T \cdot \left(\frac{\Delta C_{ab} \cdot \Delta h_{ab}}{D_c \cdot D_h} \right) \right]^{0.5} \quad 2.24$$

See the CIE specifications [CIE00] for the details of parameters involved in formula 2.24.

CIE94

MacDonald and Smith [McSm95] propose this new formula as a modification of the CIELab formula. It has similar structure to the $CMC(l:c)$, but with much simpler weighting functions:

$$\Delta E = \left[\left(\frac{\Delta L_{ab}}{K_L \cdot S_L} \right)^2 + \left(\frac{\Delta C_{ab}}{K_C \cdot S_C} \right)^2 + \left(\frac{\Delta h_{ab}}{K_h \cdot S_h} \right)^2 \right]^{0.5}, \quad 2.25$$

where $S_L = 1$, $S_C = 1 + 0.045 \cdot C_{ab,1}$, and $S_h = 1 + 0.015 \cdot C_{ab,1}$. The K_L , K_C , and K_h are parametric factors that for all the applications, except for the textile industry, a value of one is recommended [CIE00]. The $C_{ab,1}$ refers to the C_{ab} of the standard of a pair of samples

Simplicity is not always a good thing, and sometimes complexity is necessary to obtain high accuracy. For this reason two important but very complex formulas have been proposed, which are called LCD and $CIEDE2000$. These formulas introduced several modifications in order to improve the limitations of the CIE94 formula [CIE00].

Chapter 3

Output Devices

Introduction

A multimedia system consist of several input and output devices with different characteristics. These can be e.g. colour space, colour reproduction technique, hardware etc. In this chapter we describe the technology used in output devices as monitor and printer. In Section 3.1 the monitor system and the so called gamma correction problem are presented. The printer system is presented in Section 3.2. Finally, in Section 3.3, the colorimetric characterization problem is introduced.

3.1 Display system

To produce a monitor system different kinds of technology are used. These are: *Cathode-Ray Tube (CRT)*, *Liquid Crystal Display (LCD)* and *Plasma*. The characteristics of these technologies are described below.

3.1.1 Cathode-Ray Tube (CRT)

A CRT monitor is a quite simple device (see Figure 3.1). It consists of seven basic elements: *electron guns, control grids, accelerating plates, focusing structures, deflection structures, phosphor coating, and shadow mask*.

Electrons produced by a heated cathode are fired at a controlled rate through the control grid. The accelerating plates increase the velocity of the electron beams, and the focusing structures act to sharpen the fuzzy beam of electrons [Tr91]. The deflection structures guide the electron beam on a precise location on the phosphor coating, and exciting a phosphor the

light is emitted. The electron beam scans the monitor faceplate rapidly in a raster pattern (left to right, top to bottom), and the intensity of the beam is modulated during the scan so that the amount of light varies with the spatial position of the faceplate [Br+02]. The video voltage controlling the beam intensity is usually generated by a graphics card; which emits a new voltage on every tick of its pixel clock. The duration of each voltage sample determines the pixel's width [Br+02]. Colour monitors contain three interleaved phosphor types: *red*, *green*, and *blue*. There are three electron beams and a shadow mask arranged so that each beam illuminates only one of the three-phosphor types.

The process through the phosphor emits light is called *cathodo-luminescence*; this means that the kinetic energy in the fast stream of electrons is converted into light energy. If the image has to be maintained, then the phosphor needs to be continually fed with electrons; because the rate at which the phosphor fades, its persistence, is short [Tr91].

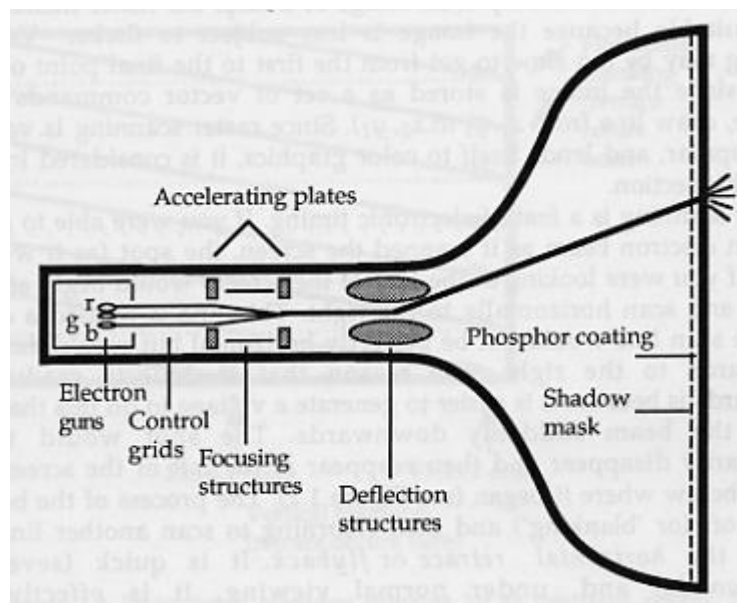


Figure 3.1: A cathode-ray tube display system, after [Tr91].

In Figure 3.2 is showed how the pixel is processed [Br+02]. The graphics card generates the video voltage based on the digital values stored in the on-board memory. These digital values can be written into two components of graphics card memory: *frame buffer* and *lookup table* [Br+02].

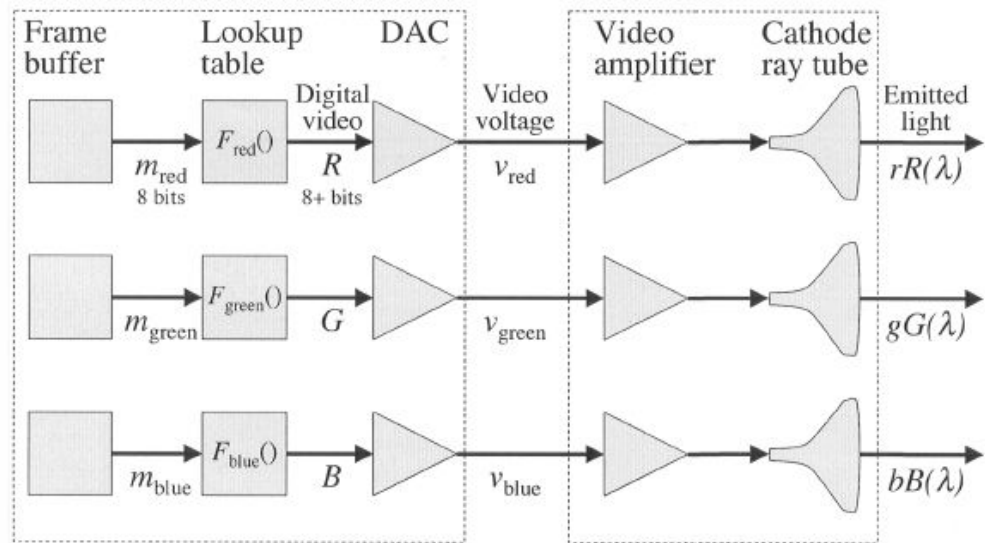


Figure 3.2: How the pixel is processed, after [Br+02].

3.1.2 Liquid Crystal Display (LCD)

The LCD monitors are becoming increasingly popular. Indeed are the dominant technology used for laptop computers. The most common LCDs for computer are backlit AMLCDs of the *twisted nematic* type. These are manufactured by deposition and patterning of (active) pixel electronics on a glass substrate. Each pixel element consists of a pair of linear polarizers with liquid-crystal (LC) material sandwiched in between [Sa02]. In Figure 3.3 is showed a pixel element. Colour displays are produced by laying a mosaic of red, green, and blue coloured filters on the substrate glass aligned with the pixel array [Sa02]. Frequently, the individual RGB pixels are rectangular and arranged so that three horizontally adjacent rectangular RGB pixels constitute a single square colour pixel. Thus the display appears to be composed of stripes of rectangular RGB pixels going vertically across the screen. The back-light is typically a fluorescent lamp with three prominent peaks in the red, green, and blue regions of the spectrum [Sa02].

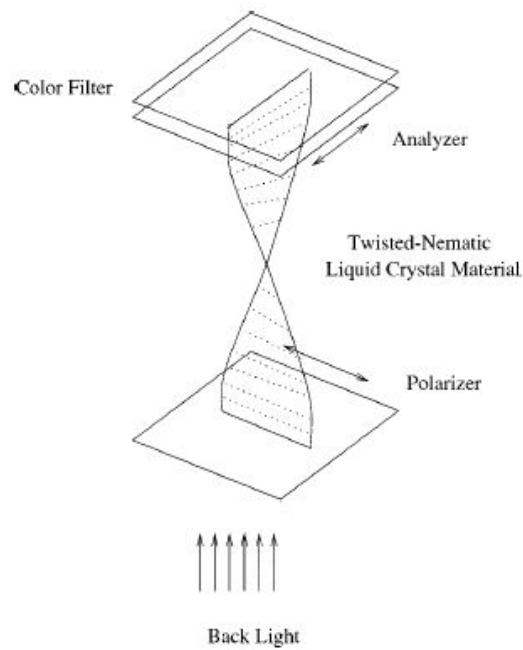


Figure 3.3: Structure of a LCD pixel, after [Sa02].

In most AMLCD colour monitors, the RGB pixels are driven and controlled independently. The emitted light is combined and averaged in the eye, just as in the CRT monitor [Sa02]. In Figure 3.4 is shown a comparison of the colour gamut reproducible by CRT and LCD monitors. Obviously, since the technology and the primaries adopted to reproduce the colours are different, so are the gamuts.

3.1.3 Plasma

The working principle of a plasma display is based on the emission of gas radiation. A gas emits radiations when is ionised. This is obtained applying on the gas high voltage. In practice this is achieved by bathing a pair of electrodes within the gas [Tr91]. A matrix of intersecting rows and columns of electrodes is used as display [Tr91].

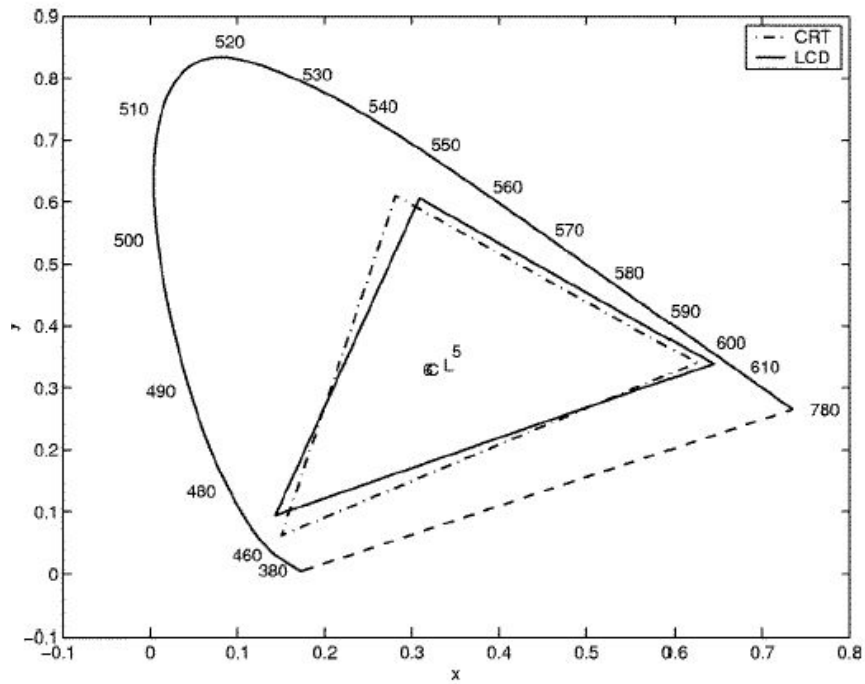


Figure 3.4: Colour Gamut of LCD and CRT monitor, after [Sa02].

3.1.4 Gamma correction

The symbol γ (gamma) represents a numerical parameter that describes the non-linearity of intensity reproduction. γ is used in several areas such as photography, video and computer graphics [Po93]. In this section we introduce the concept of gamma in the context of CRT and LCD monitors. The gamma value, for a CRT monitor, indicates a relationship between voltage input and light output that physicists call a *five-halves power law*. It says that the intensity of light produced at the face of the screen is proportional to the voltage input raised to the power $5/2$ [Po93]. In other words, the CRT monitor has voltage inputs that reflect this power function, as shown in Figure 3.5.

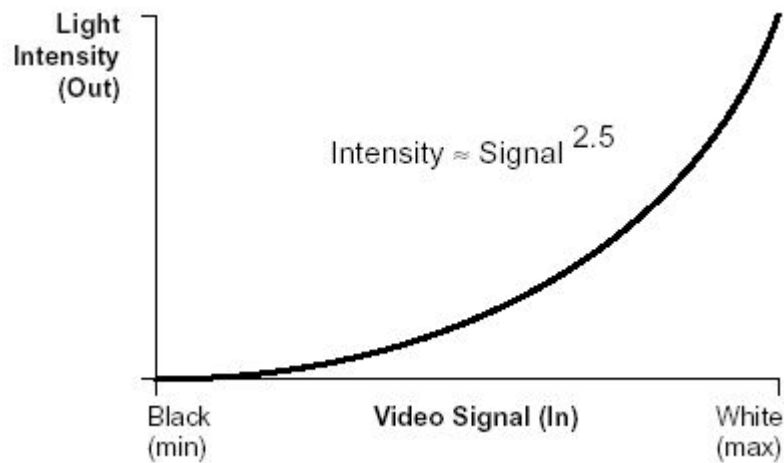


Figure 3.5: Power function applies to the single electron of a CRT monitor, after [Po93].

The functions associated with the three guns of a colour CRT monitor are not necessary identical [Po93]. The process used to correct this non-linearity, in order to achieve correct reproduction of the intensity, is called *gamma-correction* [Po93]. In the context of colour science, the gamma-correction process defines the right gamma value for the three guns of the CRT monitor and the parameters of the function that describes this non-linearity. This function is also called electro-optical *transfer function (TF)*. Several models have been proposed [Be96]. The TF describes the relationship between the signal used to drive a given monitor channel and the luminance produced by that channel [FaWy98]. In the case of a LCD monitor the TF has a different shape than the TF of a CRT monitor. Figure 3.6 shows the TF of a LCD monitor derived by Glasser [G197].

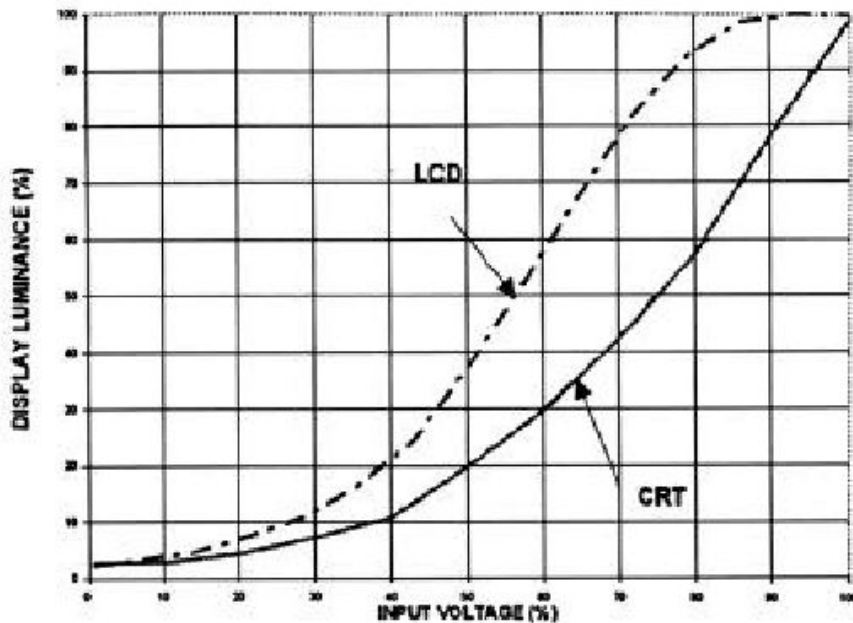


Figure 3.6: Comparison between the TF's of a LCD monitor and a CRT monitor, after [FaWy98].

Fairchild and Wyble [FaWy98] showed that while the raw physical performance of a LCD differs from that of a CRT, the digital drive circuitry for a general-purpose LCD monitor would be designed to mimic a CRT behaviour. Thus, the images presented to an user appear similar to those expected from a CRT monitor driven by a computer video output. In this way the TF of a LCD mimics the typical TF of a CRT display.

In the context of computer graphics the TF has the inverse form of the power function showed in Figure 3.5. In fact, we have as input the intensity values stored in the frame-buffer that are gamma-corrected by hardware lookup tables *on the fly* and a video signal is sent to the display (see Figure 3.7).

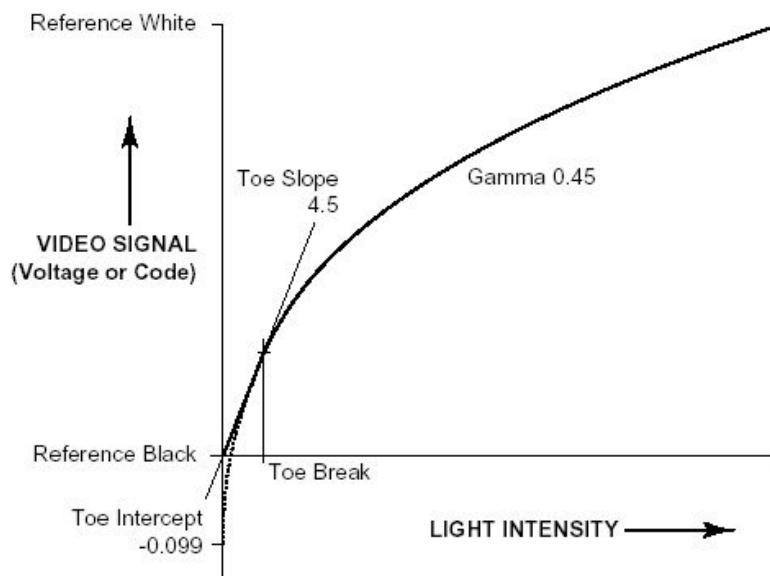


Figure 3.7: TF in the context of Computer Graphics, after [Po93].

3.2 Printer system

Creating colour accurately on paper has been one of the major areas of research in colour printing. Like monitors, printers closely position different amounts of key primary colours, which, from a distance, merge to form any colour; this process is known as dithering [Pc03]. Monitors and printers do this slightly differently however because the monitors are light sources, whereas the output from printers reflects light. So, monitors mix the light from phosphors made of the primary additive colours: red, green, and blue (RGB), while printers use inks made of the primary subtractive colours: cyan, magenta, and yellow (CMY). The coloured inks, reflecting the desired colour, absorb white light. In each case, the basic

primary colours are dithered to form the entire spectrum. Dithering breaks a colour pixel into an array of dots so that each dot is made up of one of the basic colours or left blank [Pc03].

Modern ink-jets introduced supplementary black ink in order to improve the black reproduction. In fact in the originally CMY printers, the black on a colour page, was made up from the three colours, which tends to result in an unsatisfactory dark green or grey colour usually referred to as composite black.

The most popular printer technology is the non-impact method. This technology is used by laser, ink-jet, and dye-sublimation printers to reproduce the colour.

Laser printers

The heart of the laser printer is a small rotating drum - the organic photo-conducting cartridge (OPC) - with a coating that allows it to hold an electrostatic charge. Initially the drum is given a total positive charge. Subsequently, a laser beam scans across the surface of the drum, selectively imparting points of negative charge onto the drum's surface that will ultimately represent the output image. The area of the drum is the same as that of the paper onto which the image will eventually appear, and every point on the drum corresponding to a point on the sheet of paper. In the meantime, the paper is passed through an electrically charged wire, which deposits a negative charge onto it [Pc03].

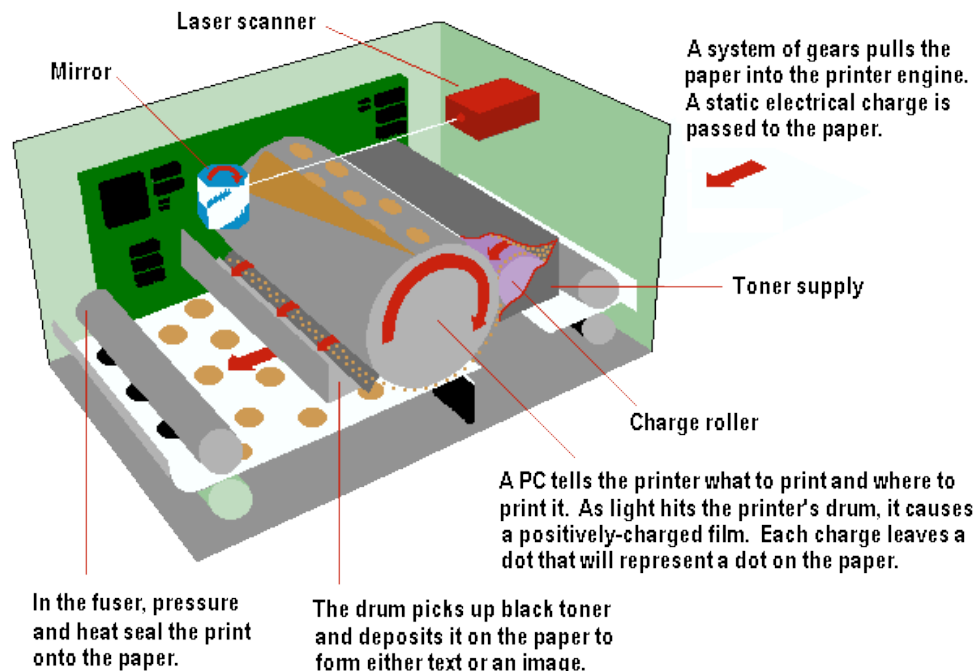


Figure3.8: Laser printer technology, after [Pc03].

In the laser printers, turning the laser on and off as it scans the rotating drum, using a complex arrangement of spinning mirrors and lenses, does the selective charging. The principle is the same as that of a disco mirror ball. The lights bounce off the ball onto the floor, track across the floor and disappear as the ball revolves. In a laser printer, the mirror drum spins incredibly quickly and is synchronised with the laser switching on and off. A typical laser printer will perform millions of switches, on and off, every second [Pc03].

Inside the printer, the drum rotates to build one horizontal line at a time. Clearly, this has to be done very accurately. As the drum rotates to present the next area for laser treatment, the written-on area moves into the laser toner. Toner is very fine black powder, positively charged so that it is attracted to the points of negative charge on the drum surface. Thus, after a full rotation the drum's surface contains the whole of the required black image.

A sheet of paper now comes into contact with the drum, fed in by a set of rubber rollers. This charge on the paper is stronger than the negative charge of the electrostatic image, so the paper magnetically attracts the toner powder. As it completes its rotation it lifts the toner from the drum, thereby transferring the image to the paper. Positively charged areas of the drum do not attract toner and result in white areas on the paper [Pc03].

Toner is specially designed to melt very quickly and a fusing system now applies heat and pressure to the imaged paper in order to add here the toner permanently. The final stage is to clean the drum of any remnants of toner, ready for the cycle to start again.

Laser printers are usually monochrome devices, but can be adapted to colour. It does this by using cyan, magenta, and yellow in combination to produce the different printable colours. Four passes through the electro-photographic process are performed, generally placing toners on the page one at a time or building up the four-colour image on an intermediate transfer surface [Pc03].

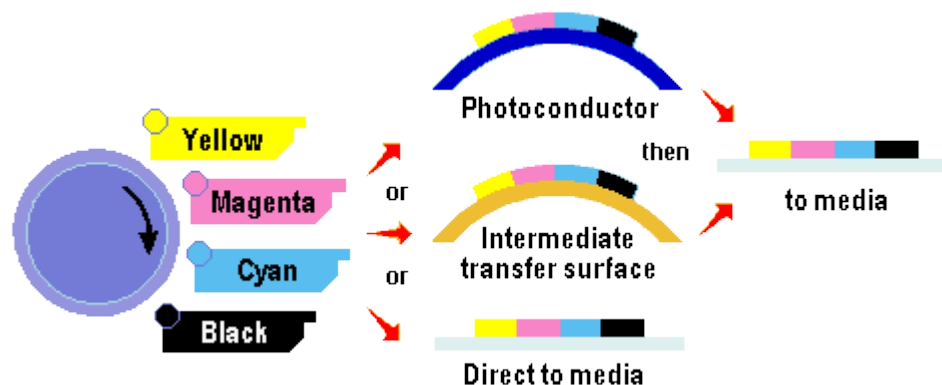


Figure 3.9: Colour laser printer scheme, after [Pc03].

Ink-jet printers

Ink-jet printing, like laser printing, is a non-impact method. Ink is emitted from nozzles as they pass over a variety of possible media, and the operation of an ink-jet printer is easy to visualise: liquid ink in various colours being squirted at the paper to build up an image. A print head scans the page in horizontal strips, using a motor assembly to move it from left to right and back, instead another motor assembly rolls the paper in vertical steps. A strip of the image is printed, and then the paper moves on ready for the next strip. To speed things up, the print head does not print just a single row of pixels in each pass, but a vertical row of pixels at a time [Pc03].

Fundamentally two technologies are used in the ink-jet printers: *continuous* and *drop-on-demand* (see Figure 3.10).

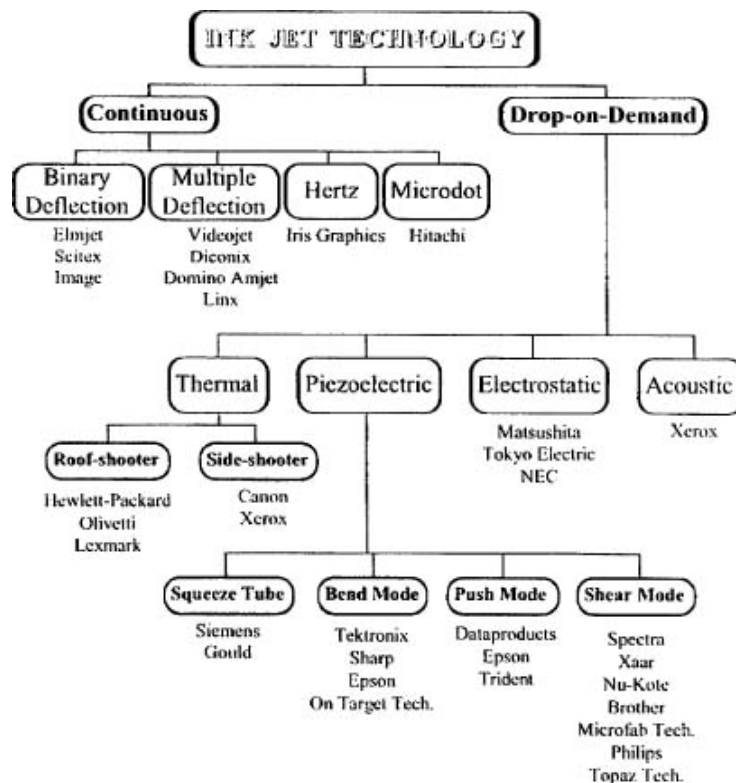


Figure 3.10: Map of the ink-jet technology, after[Le98]

The continuous ink-jet technology basis is to deflect and control a continuous ink-jet droplet stream direction onto the printed media or into a gutter for recirculation by applying an electric field to previously charged ink-jet droplets.

The drop-on-demand (DOD) printer ejects ink droplets only when they are needed to print on the media. This approach eliminates the complexity of drop charging and deflection hardware as

well as the inherent unreliability of the ink recirculation systems required for the continuous ink-jet technology [Le98].

Depending on the drop deflection methodology, the continuous ink-jet can be designed as a *binary* or *multiple deflection system*. In a binary deflection system, the drops are either charged or uncharged. The charged drops are allowed to fly directly onto the media, while the uncharged drops are deflected into a gutter for recirculation (see Figure 3.11). In a multiple deflection system, drops are charged and deflected to the media at different levels (see Figure 3.12). The uncharged drops fly straight to a gutter to be recirculated. This approach allows a single nozzle to print a small image swath [Le98].

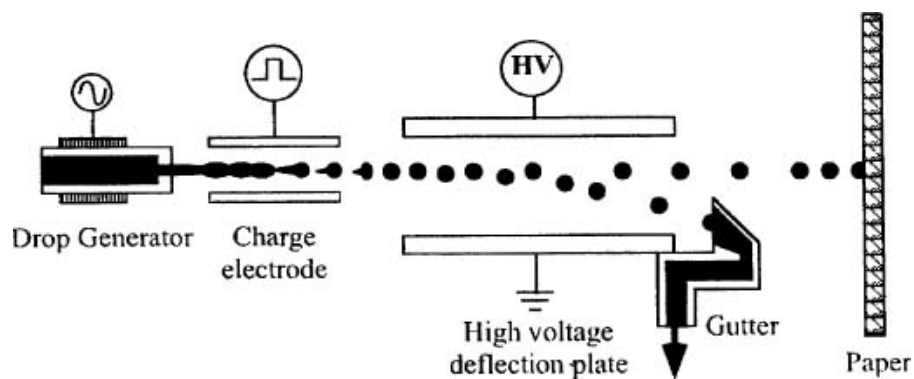


Figure 3.11: Continuous ink-jet binary deflection system, after [Le98].

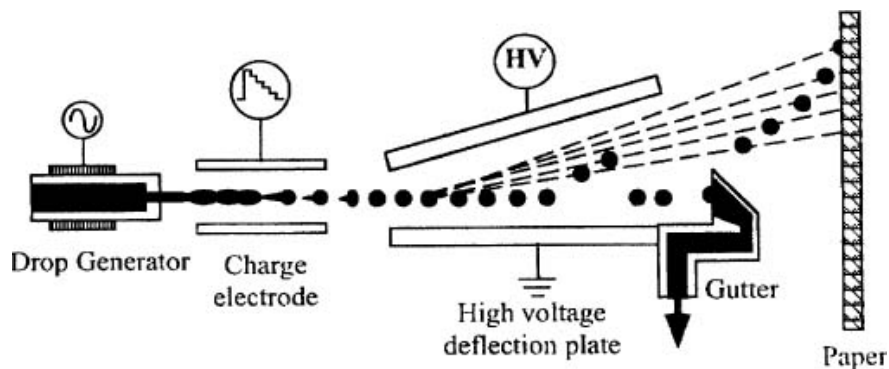


Figure 3.13: Continuous ink-jet multiple deflection system, after [Le98].

The majority of activity in ink-jet printing today is in the drop-on-demand methods. Depending on the mechanism used in the drop formation process, the technology can be categorized into four major methods: *thermal*, *piezoelectric*, *electrostatic* and *acoustic* ink-jet.

Most, if not all, of the drop-on-demand ink-jet printers on the market today are using either the thermal or piezoelectric principle. Both the electrostatic ink-jet and acoustic ink-jet

methods are still in the development stage with many patents pending and few commercial products available.

The thermal ink-jet method was not the first ink-jet method implemented in a product, but it is the most successful method on the market today. Depending on its configuration, a thermal ink-jet can be a roof-shooter (see Figure 3.13) with an orifice located on top of the heater, or a side-shooter (see Figure 3.14) with an orifice on a side located nearby the heater [Le98].

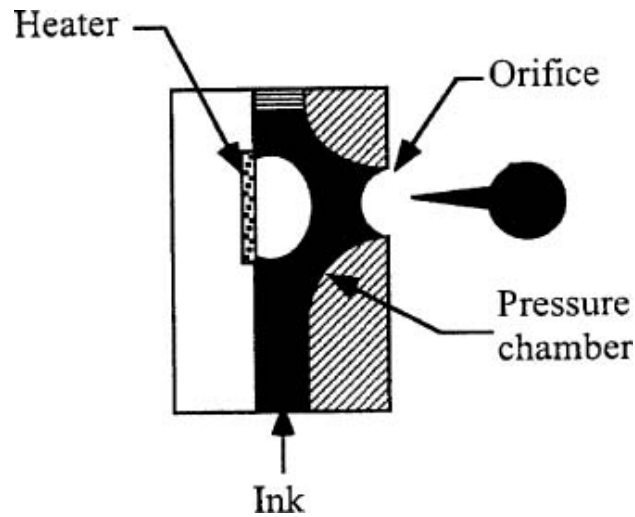


Figure 3.13: Roof-shooter thermal ink-jet, after [Le98].

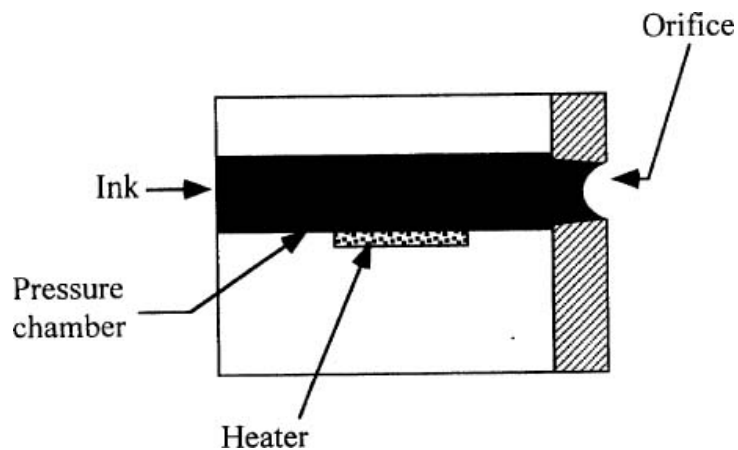


Figure 3.14: Side-shooter thermal ink-jet, after [Le98].

Thermal technology imposes certain limitations on the printing process in that whatever type of ink is used, it must be resistant to heat because the firing process is heat-based. The use of heat in thermal printers creates a need for a cooling process as well, which levies a small time overhead on the printing process [Pc03].

In a basic configuration, a thermal ink-jet consists of an ink chamber having a heater with a nozzle nearby. With a current pulse of less than a few microseconds through the heater, heat is transferred from the surface of the heater to the ink. The ink becomes superheated to the critical temperature for bubble nucleation, for water-based ink, this temperature is around 300°C. When the nucleation occurs, a water vapour bubble instantaneously expands to force the ink out of the nozzle. Once all the heat stored in the ink is used, the bubble begins to collapse on the surface of the heater. Concurrently with the bubble collapse, the ink droplet breaks off and excels toward the paper. The whole process of bubble formation and collapse takes place in less than 10 μ s. The ink then refills back into the chamber and the process is ready to begin again. Depending on the channel geometry and ink's physical properties, the ink refill time can be from 80 to 200 μ s. This process is illustrated in Figure 3.15 [Le98].

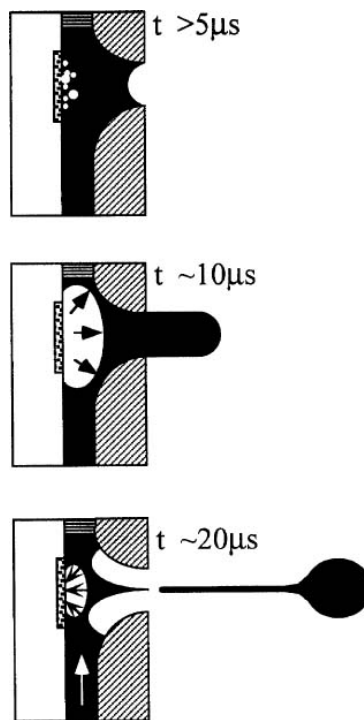


Figure 3.15: Drop formation process in thermal ink-jet printer, after [Le98].

In the piezoelectric ink-jet, depending on the piezo-ceramic deformation mode, the technology can be classified into four main types: squeeze, bend, push, and shear [Le98]. A squeeze-mode ink-jet can be designed with a thin tube of piezo-ceramic surrounding a glass nozzle as in a Gould's impulse ink-jet or with a piezo-ceramic tube cast in plastic that encloses the ink channel as was implemented in a Seimens PT-80 ink-jet printer.

In a typical bend-mode design (see Figure 3.16), the piezo-ceramic plates are bonded to the diaphragm forming an array of bilaminar electromechanical transducers used to eject the ink droplets [Le98].

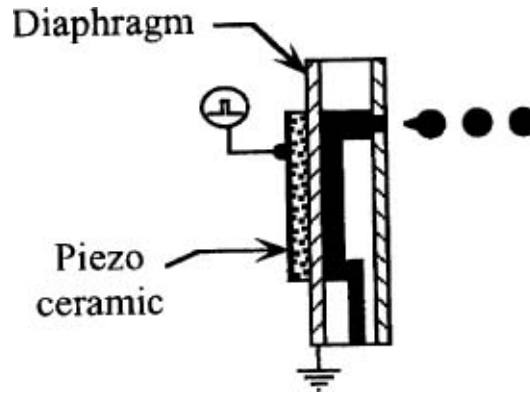


Figure 3.16: A bend-mod piezoelectric ink-jet, after [Le98].

In a push-mode design (see Figure 3.17), as the piezo-ceramic rods expand, they push against ink to eject the droplets. In theory, piezo-drivers can directly contact and push against the ink. However, in practical implementation, a thin diaphragm between piezo-drivers and ink is incorporated to prevent the undesirable interactions between ink and piezo-driver materials [Le98].

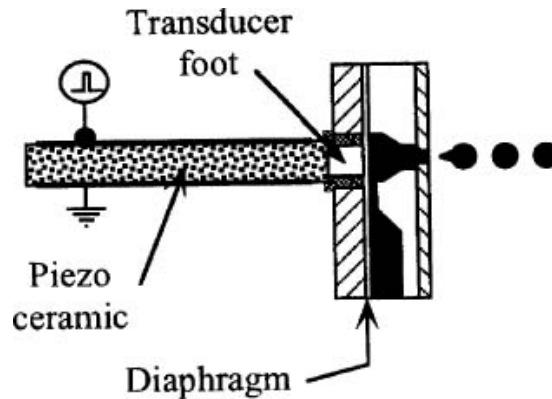


Figure 3.17: A push-mod piezoelectric ink-jet, after [Le98].

In both the bend- and push-mode designs, the electric field generated between the electrodes is in parallel with the polarization of the piezo-material. In a shear-mode print-head, the electric field is designed to be perpendicular to the polarization of the piezo-driver (see Figure 3.18). The shear action deforms the piezo-plates against ink to eject the droplets [Le98].

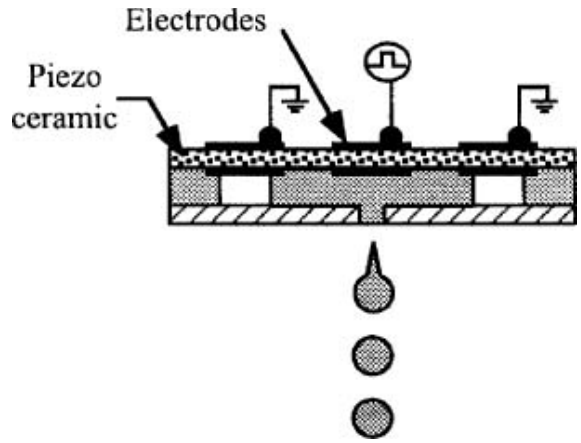


Figure 3.18: A shear-mod piezoelectric ink-jet, after [Le98].

In the piezoelectric drop-on-demand ink-jet method (see Figure 3.19), deformation of the piezo-ceramic material causes the ink volume change in the pressure chamber to generate a pressure wave that propagates toward the nozzle. This acoustic pressure wave overcomes the viscous pressure loss in a small nozzle and the surface tension force from ink meniscus so that an ink drop can begin to form at the nozzle. When the drop is formed, the pressure must be sufficient to expel the droplet toward a recording media. The basic pressure requirement is showed in Figure 3.20.

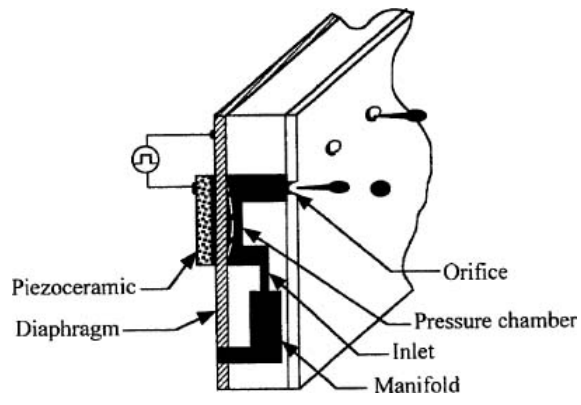


Figure 3.19: Piezoelectric print head, after [Le98].

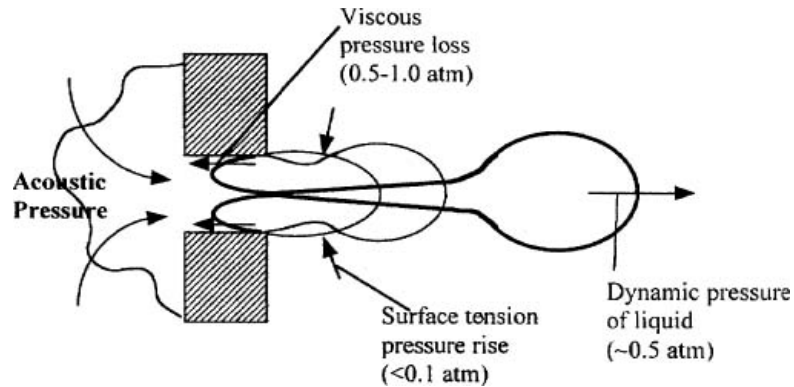


Figure 3.20: Pressure process required for ejecting an ink droplet, after [Le98].

There are several advantages to the piezo method. The process allows more control over the shape and size of ink droplet release. The tiny fluctuations in the crystal allow for smaller droplet sizes and hence higher nozzle density. Also, unlike with thermal technology, the ink does not have to be heated and cooled between each cycle. This saves time, and the ink itself is tailored more for its absorption properties than its ability to withstand high temperatures. This allows more freedom for developing new chemical properties in inks.

Dye-sublimation printers

Dye-sublimation printers are specialized devices widely used in demanding graphic arts and photographic applications. Dye sublimation work (see Figure 3.21) by heating the ink so that it turns from a solid into a gas. The heating element can be set to different temperatures, thus controlling the amount of ink laid down in one spot. In practice, this means that colour is applied as a continuous tone, rather than in dots, as with an ink-jet. One colour is laid over the whole of one sheet at a time, starting with yellow and ending with black. The ink is on large rolls of film which contain sheets of each colour, so for an A4 print it will have an A4-size sheet of yellow, followed by a sheet of cyan, and so on. Dye sublimation requires particularly expensive special paper, as the dyes are designed to diffuse into the paper surface, mixing to create precise colour shades.

There are now some "ink-jet printers" on the market that actually deploy dye-sublimation techniques. The way in which an ink-jet uses the technology differs from a dye-sublimation in that its inks are in cartridge, which can only cover the page one strip at a time. It heats the inks to form a gas, controlled by a heating element, which reaches temperatures of up to 500° C (higher than the average dye sublimation printer).

The benefits, of a dye-sublimation printer, are the superb image quality with very smooth transitions like colour photographs. The obvious drawbacks are the high costs of special paper and transfer rolls. It should only be used if a high proportion of the printed page is covered with colour.

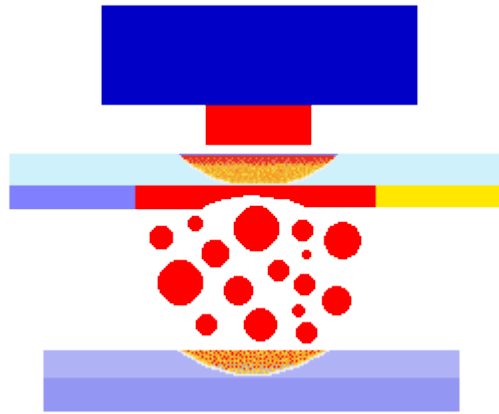


Figure 3.21: Dye-sublimation process, after [Pc03].

In Figure 3.22 is showed a comparison of gamut colours reproduced by different printers: ink-jet, laser, and dye-sublimation. As described for the monitor, the differences are the consequence of different technology used to reproduce the colour, different primaries, and different substrate.

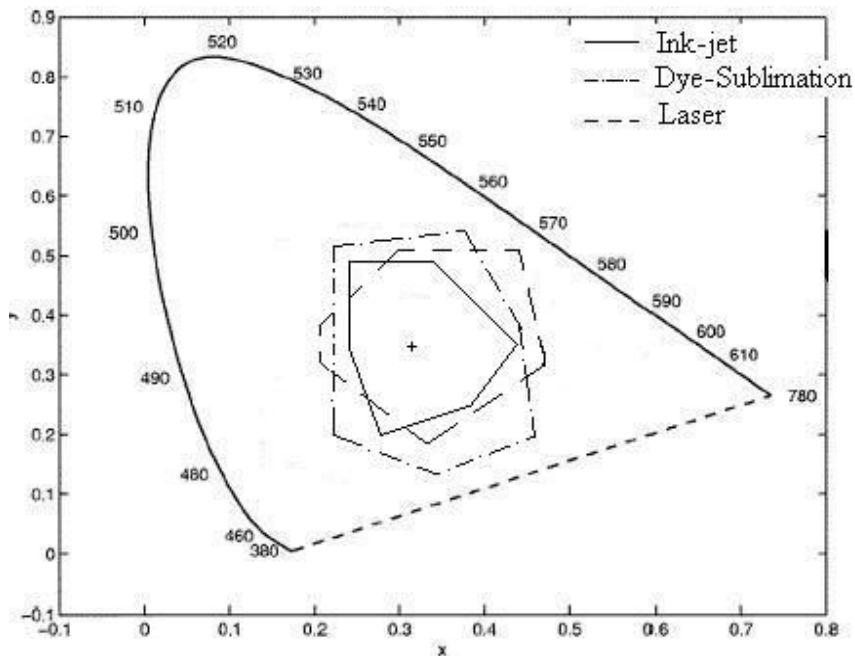


Figure 3.22: Colour gamuts of three different kinds of printer: (continuous line) ink-jet, (dash point line) Dye-sublimation, (dash line) laser.

Print quality

The two main determinants of colour print quality are resolution, measured in dots per inch (dpi), and the number of levels or graduations that can be printed per dot. Generally speaking, the higher the resolution and the more level per dot, the better the overall print quality [Pc03].

In practice, most printers make a trade-off, some opting for higher resolution and others settling for more levels per dot, the best solution depending on the printer's intended use. Graphic arts professionals, for example, are interested in maximising the number of levels per dot to deliver "photographic" image quality, while general business users will require reasonably high resolution so as to achieve good text quality as well as good image quality [Pc03].

The simplest type of colour printer is a binary device in which the cyan, magenta, yellow and black dots are either "on" (printed) or "off" (not printed), with no intermediate levels possible. If ink (or toner) dots can be mixed together to make intermediate colours, then a binary CMYK printer can only print eight "solid" colours (cyan, magenta, yellow, red, green and blue, plus black and white). Clearly this is not a big enough palette to deliver good colour print quality, which is where halftoning comes in [Pc03].

Halftoning algorithms divide a printer's native dot resolution into a grid of halftone cells and then turn on varying numbers of dots within these cells in order to mimic a variable dot size. By carefully combining cells containing different proportions of CMYK dots, a halftoning printer can "fool" the human eye into seeing a palette of millions of colours rather than just a few [Pc03].

Most of the current generation of ink-jet printers require high-quality coated or glossy paper for the production of photo-realistic output, and this can be very expensive. One of the ultimate aims of ink-jet printer manufacturers is to make colour printing media-independent, and the attainment of this goal is generally measured by the output quality achieved on plain copier paper. This has vastly improved over the past few years, but coated or glossy paper is still needed to achieve full-colour photographic quality. Some printer manufacturers, like Epson, even have its own proprietary paper, which is optimised for use with its piezo-electric technology [Pc03].

Recently, "six-colour" ink-jet printers have appeared on the market, specifically targeted at delivering "photographic-quality" output. These devices add two further inks - light cyan and light magenta - to make up for current ink-jet technology's inability to create very small (and

therefore light) dots. Six-colour ink-jets produce more subtle flesh tones and finer colour graduations than standard CMYK devices [Pc03].

3.3 Colorimetric Characterization of Device

In multimedia systems, different color reproduction devices --- while serving the same purpose --- exhibit large discrepancies in their raw output. This is due to the fact that they usually employ different color mixing technologies (additive or subtractive), use different input color spaces and hence have different gamuts, and that their device characteristics can change with time and usage. These facts usually do not permit a faithful matching of colors between devices if no precautions are taken [ArWi01].

Colorimetric characterization is one step in the colorimetric reproduction process that permits faithful image reproduction across different devices. Its goal is to define a mapping function between the device dependent colour spaces in question (such as RGB or CMYK) and device independent colour spaces (such as CIELAB or CIEXYZ), and vice versa.

There are three main approaches to define this mapping function: *physical models*, *empirical models* and *exhaustive measurements* [Fa98]. Physical modeling of imaging devices involves building mathematical models that find a relationship between the calorimetric coordinates of the input (or output) image element and the signals used to drive an output device (or the signals originating from an input device). The advantage of these approaches is that they are robust, typically require few calorimetric measurements in order to characterize the device, and allow for easy re-characterization if some component of the imaging system is modified. The disadvantage is that the models are often quite complex to derive and can be complicated to implement. Physical models are often used for the calorimetric characterization of displays and scanners.

Empirical modeling of imaging devices involves collecting a fairly large set of data and then statistically fitting a relationship between device coordinates and calorimetric coordinates. Empirical models are often higher order multidimensional polynomials, or neural network models. They require fewer measurements than LUT techniques, but they need more than physical models. Empirical models are often used for scanners and printers.

Often the calorimetric characterization of devices requires an exhaustive measurement in order to obtain good performances. The number of measurements depends on the devices repeatability [ArWi01]. Devices with poor repeatability need more measurements those devices with high repeatability property.

Lookup tables can be used to process image data via multidimensional interpolation. This technique has different disadvantages: the large number of measurements that has to be made, difficulties in interpolating the highly nonlinear data and difficult re-characterization if any aspect of the device changes. The advantage of exhaustive measurement techniques is that they require no knowledge of the device physics.

In general a good algorithm for calorimetric characterization must have the following characteristics: small training set, fast response, good accuracy and it must allow for a fast re-characterization.

Chapter 4

Tone Mapping

Introduction

Output display devices, like projectors or screens, commonly used in computer graphics have several technological limitations such as limited colour gamut and contrast, low dynamic range, limited spatial resolution, (usually) limited field of view and non trivial workarounds to achieve stereo capacity.

These limitations, prevent to directly display high dynamic range (HDR) images on output devices. In fact, the luminance range in HDR images exceeds the luminance range of the output devices by several orders of magnitude. The conversion from the HDR to display luminance is known as *Tone Mapping* (TM).

Section 4.1 introduces the concept of *Image Synthesis*, while in Sections 4.2, 4.3 and 4.4 a survey on TM is presented. Finally, in Section 4.5 the concept of TM for printers is discussed.

4.1 Image Synthesis

The creation of images by evaluating a model of light propagation is called *image synthesis* [CoWa93]. The goal of image synthesis is often stated as *photo-realism*. However, although photography produces realistic images, it is a physical process subject to the constraints of camera optics and the chemical nature of film. Should image synthesis really attempt to simulate photography, or should it aim higher? [CoWa93].

A clear understanding of the goal of image synthesis becomes increasingly important as algorithms and computational method grow more sophisticated. In addition to the evaluation of competing approaches, more intelligent algorithms need a basis for deciding how to allocate computational effort and when to end the computation, which requires knowing when the goal has been achieved [CoWa93].

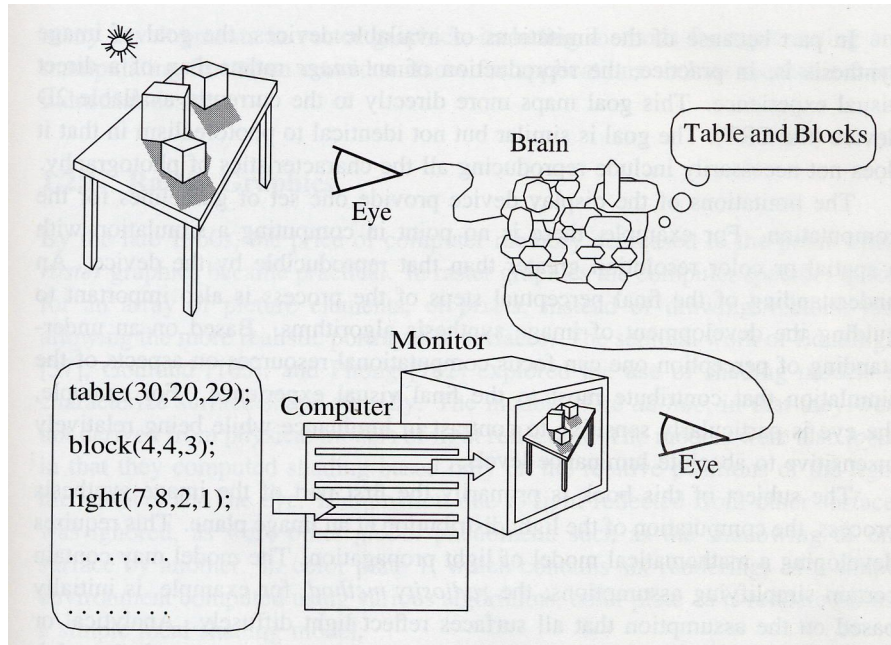


Figure 4.1: Image synthesis process, after [CoWa93].

Perhaps the most far reaching goal for image synthesis is the creation of a visual experience identical to that which would be experienced in viewing the real environment. In Figure 4.1 is shown a simple model of image synthesis process [CoWa93]. The model is divided in two parts: *real world* (top half diagram) and *digital world* (bottom half diagram). In the real world, the light propagates through the scene and eventually enters in the eye with a particular directional and wavelength distribution. The eye and the brain process this information at increasingly higher levels of abstraction, leading ultimately to what is called the visual experience [CoWa93].

In the digital world, a mathematical model is used to simulate the distribution of light energy, and the results are passed to a display device that physically realizes the computed light distribution and sends it to the eye [CoWa93].

There are two problems with this apparently simple approach. First, the computation in step one is arbitrarily expensive. For all practical purpose, there is no end to the details and accuracy with which reality might be simulated [CoWa93]. The second problem is with the display system. In fact it has many limitations: limited colour gamut, dynamic range, spatial

resolution field of view, and stereo-capacity. These limitations reduce the realism of the scene reproduced on the display.

In order to reach realistic images, in the final output of the image synthesis, a *global illumination* model is needed. It takes into account the interreflection of light between surfaces.

The first global illumination model was introduced by Whitted in 1980 [Wh80]. This model consists of a recursive application of ray tracing to evaluate a simple global illumination model accounting for mirror reflection, refraction, and shadows [CoWa93]. Later on the Whitted method has been extended [Co86][Co+84][Am84], and more accurate physically based local reflection models were developed by Cook and Torrance [CoTo82]. In 1984 radiosity methods have been introduced in computer graphics. In image synthesis, radiosity methods are used to solve the interreflection of light between ideal (Lambertian) diffuse surfaces [CoWa93].

Subsequent works have included extension of the radiosity approach to glossy and ideal (mirror) reflection [CoWa93]. Kajiya [Ka86] unified the discussion of global illumination algorithm with the general *rendering equation*. Kajiya applied Monte Carlo integration methods solving the rendering equation and proposed a number of techniques for accelerating the convergence of the solution [CoWa93].

4.2 Tone Mapping

In Section 4.1 the image synthesis goal has been introduced: "produce an image that captures the visual appearance of modelled scenes". The physically based rendering methods try to reach this goal. However simulating the distribution of light energy in the scene does not guarantee that the display images have a realistic visual appearance. There are two reasons for that: first, the HVS operates over a large 10^{12} range of luminance and contrast, while a typical monitor has a maximum contrast of only 10^2 . Second, the visual states of the scene observer and the monitor observer may be very different. We also need to take into account the characteristics of HVS to reproduce the subjective impression [Fe+96]. The TM, is the conversion from the real world to the display information (see Figure 4.2). The TM ideas were originally developed for photography [Wa+97]. In photography (or video), chemistry (or electronics) are used, together with a human actively controlling the scene lighting and the camera that are used to map real world luminance into an acceptable image on a display.

In synthetic image generation, our goal is to avoid active control of lighting and camera settings. We need to have a direct numerical control over display values [Wa+97]. The HVS, as explained in Chapter 1, is able to adapt to the different lighting conditions. This happens when we are going, for example, from a dark room into sunlight and vice-versa. However, the HVS takes time before recovering full sensitivity. After the adaptation time, we are able to see the objects and fine details of the scene. This does not happen in the case of the photography or if we visualize the real scene on the display without having solved the problems exposed above. We have to consider two important criteria's for a reliable tone mapping [Wa+97]: to reproduce the *visibility* and the *subject experience* of the real world observer. The visibility is reproduced when we see the same scene details, which we see in the real scene. In other words, there are not objects obscured in the under-or over-exposed regions in the digital image, and the features are not lost in the middle. Reproducing the subject experiences of the real world observer means reproducing the overall impression of the brightness, contrast, and colour.

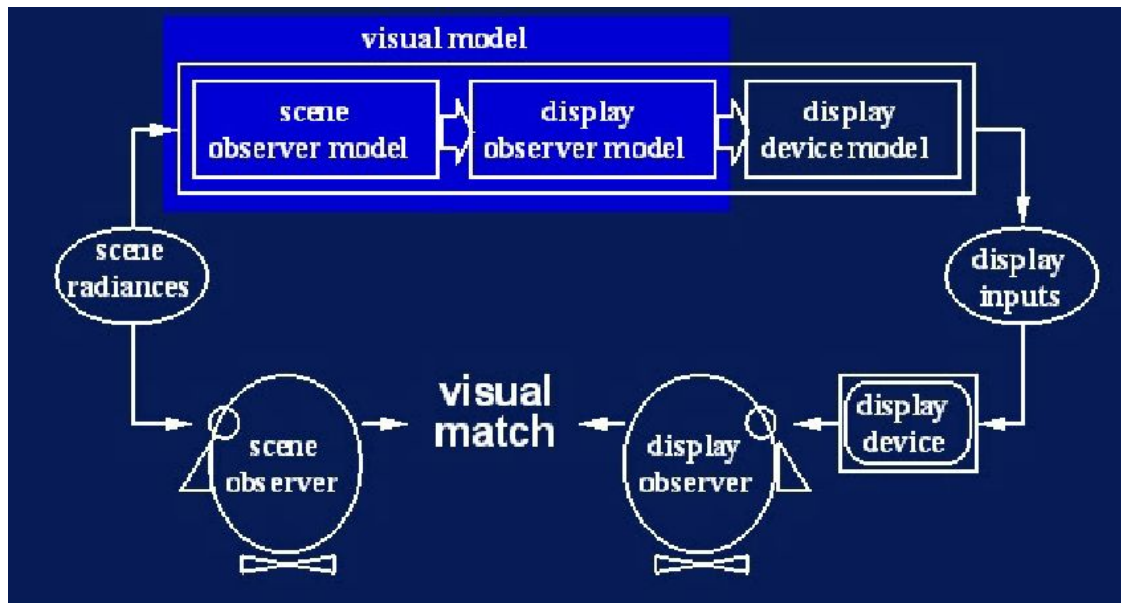


Figure 4.2: Tone reproduction, after [Fe+96].

The *Enhancement* technique is used in computer graphics to improve the displayed scene as well. However, it is not comparable with the TM solution for several reasons [Wa+97]. First, the enhancement works on data already distorted by photography or video recording and collapsed into a limited dynamic range. On the contrary, the TM technique works on the real world luminance data with a potentially high dynamic range. Second, the goal of the enhancement technique is to maximize visibility and contrast; and maintaining the subjective

correspondence with the original view of the scene is irrelevant. On the contrary, the goal of the TM is to simulate the visibility and the contrast, not to maximize it [Wa+97].

Basically, there are two categories of TM operators: global and local. The former are also called tone reproduction curves (TRCs). The latter are called tone reproduction operators (TROs). TRCs operate pixel-wise on the input image which makes the algorithms simple and efficient. In other words, the TRCs apply a single tone-mapping curve (TRC) uniformly to all image data. This globally equal pixel-wise mapping adjusts the intensity of each pixel using a function that is independent of the local spatial context. In contrast to TRCs, TROs use the spatial structure of the image data and attempt to preserve local image contrast. These algorithms allow transforming the same pixel intensity of the input image to different display values, or different pixel intensities to the same display value [DiWa00].

4.2.1 Global operators

Basic Tone Reproduction Function

Schlick [Sc94] defined the TM problem as follow: every intensity value Val of the computed scene has to be quantized (function $Q(Val)$), in order to map one of the N single values in $[0, N-1]$ accepted by a typical visualization device. The mathematical expression in equation 4.1 below, is defined as tone reproduction curve TRC [Sc94]:

$$Q(Val) = \lceil N \cdot F(Val) \rceil, \quad 4.1$$

where F is a function:

$$F : [LoVal, HiVal] \rightarrow [0,1], \quad 4.2$$

where $LoVal$ and $HiVal$ are the lower and the higher pixel values of the scene. Some simple TRC functions have been used in computer graphics community. These functions consider the colour reproduction, and usually ignore brightness reproduction. For example, the most used TRC in computer graphics is the so called *gamma-corrected* linear mapping i.e.,

$$F_{\gamma}(Val) = \left(\frac{Val}{HiVal} \right)^{1/\gamma}, \quad 4.3$$

where γ is the gamma value in the range $[1,3]$. In this function two operators are applied: linear scaling in the range $[0,1]$ (i.e. division by $HiVal$) and a gamma-correction to compress the non-linear response of the visualization device.

However, this model is completely inadequate for high dynamic range scene. For example, the result images in some experiments presented in the work of Schlick [Sc94] are mapped into black or very dark grey.

Another well known *TRC* in computer graphics is the *gamma-corrected clamping* i.e.,

$$F_{p,\gamma}(Val) = (Val / p)^{1/\gamma}, \quad 4.4$$

if $Val < p$, 1 otherwise. Where p is in the range $[LoVal, HiVal]$, while γ is in the range $[1,3]$. The limit of this *TRC* is that a satisfactory value for p is rare to be found [Sc94]. A *TRC* function that has been rarely implemented is the *logarithmic mapping*:

$$F_{p,\gamma}(Val) = \left(\frac{\log(1 + p \cdot Val)}{\log(1 + p \cdot HiVal)} \right)^{1/\gamma}, \quad 4.5$$

where p is in the range $[0, \infty)$, while γ is in the range $[1,3]$. The last *TRC* function we present is the *exponential mapping* that includes both brightness perception and gamma-correction:

$$F_{p,\gamma}(Val) = \left(\frac{Val}{HiVal} \right)^{p/\gamma}, \quad 4.6$$

where p is in the range $[0,1]$, while γ is in the range $[1,3]$.

Tumblin and Rushmeier 1993

Tumblin and Rushmeier [TuRu93] defined a TM operator, built from models of human vision, to convert scene intensities to display intensities.

This operator is based on the super-threshold brightness measurements made by Stevens and Stevens [STST60][STST63] who claimed out that a power-law relation exists between luminance L , and perceived brightness B [Tu+99].

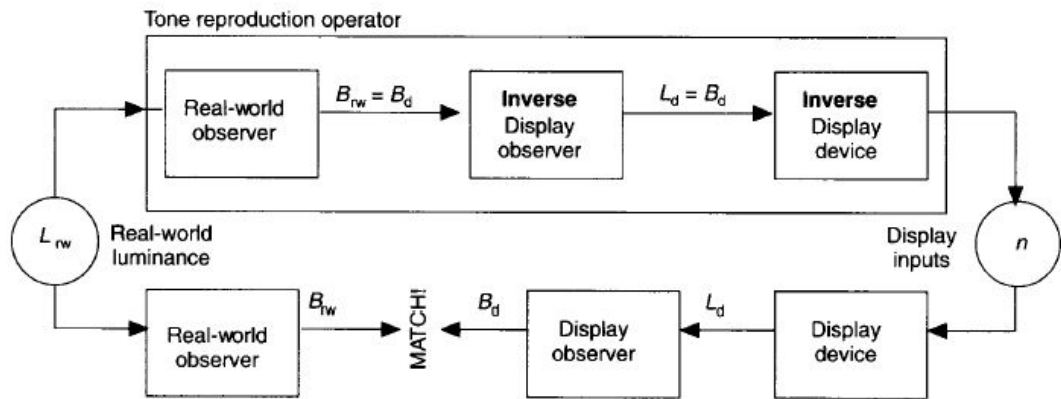


Figure 4.3: Tumblin and Rushmeier 1993 TM operator, after [TuRu93].

This operator is a concatenation of different models: real-world observer, inverse display observer and inverse display device (see Figure 4.3). The author's defined the following real world and display observer equations:

$$B_{rw} = 10^{\beta_{rw}} \cdot L_{rw}^{\alpha_{rw}}, \quad 4.7$$

$$B_d = 10^{\beta_d} \cdot L_d^{\alpha_d} . \quad 4.8$$

Where β and α are functions that describe the changes due to viewed luminance of the sensitivity and the contrast, respectively, and are given by:

$$\alpha(L_{rw}) = 0.41 \cdot \log_{10}(L_{rw}) + 2.92 , \quad 4.9$$

$$\beta(L_{rw}) = -0.41 \cdot \log_{10}(L_{rw})^2 + (-2.584 \cdot \log_{10} L_{rw}) + 2.0208 . \quad 4.10$$

These equations explain the power-law relationship between the value of brightness B and luminance L . This model is also used for photographic tone reproduction. The two observers are connected back-to-back by setting $B_{rw} = B_d$. In this way one can define the display luminance in terms of real-world luminance:

$$L_d = L_{rw}^{\left(\frac{\alpha_{rw}}{\alpha_d}\right)} \cdot 10^{\left[\frac{(\beta_{rw}-\beta_d)}{\alpha_d}\right]} . \quad 4.11$$

The inverse display system model is defined by equation 4.12 below:

$$n = \left[\left(L_d / L_{d \max} \right) - \left(1 / C_{\max} \right) \right]^{1/\gamma} , \quad 4.12$$

where n is the frame buffer value ($0 < R = G = B = n < 1$) used as display device input, $L_{d \max}$ is the maximum screen luminance, C_{\max} is the maximum contrast ratio, while γ is the gamma correction. The final TM operator is obtained by substituting, in the inverse display model of equation 4.12, the display luminance L_d of equation 4.11.

This operator exhibits several serious shortcomings [Tu+99]: first, scenes that approach total darkness are displayed as anomalous middle grey images instead of black. Second, the display contrast for very bright images is unrealistically exaggerated. Also this operator did not address the contrast limitation of displays and was presented in an awkward form that discouraged its use [Tu+99].

Schlick 1994

Schlick proposed a solution for the TM problem, defining the best quantization function, which by passes the limitations of output devices. For instance, this process implies to quantify every floating-point *intensity* value (expressed either as radiance or luminance) computed during the rendering process, in order to map to one of the N single integer values accepted by a typical visualization device [Sc94]. He proposed two quantization techniques: *uniform rational quantization* and *non-uniform rational quantization*. In the uniform rational quantization method, he presented a new quantization scheme called *rational mapping* (see equation 4.13). Schlick also proposed an automatic generation of the parameters p and γ .

$$F_p(Val) = \frac{p \cdot Val}{p \cdot Val - Val + HiVal}, \quad 4.13$$

where p is in the range $[0, \infty)$. This equation is computationally economic compared to the exponential and logarithmic equations 4.6 and 4.5 respectively. In particular, it needs only 1 division, 1 multiplication, 1 subtraction and 1 addition.

The automatic process of parameters generation is based on the assumption, when several viewing parameters are modified, that what really changes on a visualization device is the value M of the darkest grey level that can be clearly distinguished from black. This value is used in the following way: the quantization process should map the smallest non-zero intensity of the picture to the darkest non-black grey of the device $Q(L_oVal) = M$.

With the rational mapping function, the parameter p is equal:

$$p \approx \frac{M}{N} \cdot \frac{HiVal}{LoVal}. \quad 4.14$$

The value of M , can be easily provided by the user without any measuring instrument, as described in Schlick [Sc94].

If the intensity of each pixel is not quantized with the same TM function, the quantization process is called *non-uniform*. This will be explained in Section 4.3.2

The model presented by Schlick [Sc94], works well when applied uniformly to each pixel of a high-contrast scene, and especially in scenes containing strong highlights [TuRu99]. It is quite elegant and practical. Users can find all parameters of the mapping function without photometric measurements of the display device, and can compute the mapping quickly because it does not require transcendental functions [TuRu99].

Also the mapping function preserves contrast for dark image regions and asymptotically compresses image highlights sufficiently to avoid clipping on the display [TuRu99]. However, if it is true that the all parameters of the mapping function can be find without photometric measurements, it is also true that the definition of the parameter M needs an interactive process that must be done off-line, and repeated any time when the display characteristics are changed.

Ward 1994

Ward [Wa94] presented a simpler approach to appearance modelling that also provided a better way to make dark scenes appear dark and brightness scenes appear bright on the display [Tu+99]. He proposed using a light-dependent multiplying factor m to restore the appearance of different lighting conditions. The factor was built using experimental data

(contrast visibility data) obtained by Blackwell [Bl46]. These data showed that the smallest noticeable increase in luminance or “contrast threshold” of a small target on a uniform background grows non linearly as the amount of surrounding light increases [Tu+99]. The scale factor is used to match the threshold of the display to those of the original scene:

$$L_d = m \cdot L_{rw}, \quad 4.15$$

where L_d is the display luminance in cd/m^2 , L_{rw} is the original scene or world luminance in cd/m^2 , and

$$m = \left[\frac{1.219 + L_{da}^{0.4}}{1.219 + L_{rwa}^{0.4}} \right]^{2.5}, \quad 4.16$$

where L_{da} is the display adaptation luminance, a midrange display value, and L_{rwa} is the adaptation luminance for the real world or scene, usually $\log(L_{rwa}) = \text{mean}\{\log(L_{rw})\}$ [Tu+99]. Because Ward’s method scaled image intensities by a constant factor m , it did not change scene contrasts for the display [Tu+99].

Ferwerda et al. 1996

Ferwerda et al. [Fe+96], introduced a model that include the effects of the adaptation in *threshold visibility, colour appearance, visual acuity*, and changes in *visual sensitivity* over the time that are caused by the visual system's adaptation mechanisms.

The model is based on the psychophysics of adaptation measured in experimental studies, and maps image file with photopic luminance *CIE Y*, scotopic luminance *Y'* and *CIE XZ* channels to displayable images in a *RGB* colour space. Their approach is based on Ward’s concept of matching [Wa94]. As in Ward’s method, they converted original scene or image intensities L_{rw} to display intensities L_d , with a multiplicative scale factor m . They determined their m values from a smooth blending of increment threshold data for both rods and cones in the retina [Tu+99]. They first constructed their operator applying Ward’s model without changes to their cone *TVI* data; and extended the Ward’s model to include the rod *TVI* function. Also they took into account the mesopic state applying a weighted function to the results of the photopic and scotopic operators. The final value for the luminance display is

$$L_d: \quad L_d = L_{dp} + K(L_{rwa}) \cdot L_{ds}, \quad 4.17$$

where $K(L_{rwa})$ is a constant that varies from 1 to 0 as the scotopic world adaptation level goes from the bottom to the top of the mesopic range. The values L_{dp} and L_{ds} are the luminance values, compressed in the display luminance range, in the photopic and scotopic range, respectively.

Ferwerda et al. [Fe+96] took into account the visual acuity, removing all spatial frequencies, above the spatial frequencies visible to the world observer, in the image that is presented to the display observer. This was obtained using a Gaussian convolution filter whose power spectrum amplitude at the cut-off frequency is matched to the observer's threshold. Thus, they removed the frequencies in the image which would not be discernable to the world observe, equation 4.18:

$$f^*(\omega_c(L_{rwa})) = \frac{t(L_{rwa})}{L_{rwa}}, \quad 4.18$$

where f^* is the Fourier transform of the convolution filter, and $\omega_c(L_{rwa})$ is the threshold frequency for the world adaptation of the viewer. In this way, a high contrast scene grating at frequency $\omega_c(L_{rwa})$ will be displayed at the threshold of visibility for the display viewer. The light and the dark adaptation are implemented adding to the display luminance a constant value.

Finally, the adaptation state L_{rwa} , for the real world, and the adaptation state L_{da} , for the display system, are taken as half the highest visible luminance and half the maximum value of the display luminance, respectively.

This TM operator can reproduce the visual consequences of visual adaptation. Nevertheless, it does not have a good model for a viewer's state of adaptation. This operator is also not able to capture completely the appearance of the early phases of light and dark adaptation [Fe+96].

Ward et al. 1997

Ward et al. [Wa+97] introduced a TM operator based on iterative histogram adjustment and spatial filtering processes. This operator reduces high scene contrasts to match display abilities, and also ensures that contrasts that exceed human visibility thresholds in the scene remain visible on the display [Tu+99]. They model some foveally dominated local adaptation effects, yet completely avoid halo artefacts or other forms of local gradient reversals [Tu+99].

They also take into account three important limitations of the HVS: glare, colour sensitivity and visual acuity. The way to obtain the glare effect is based on the computation of a low-resolution veil image from foveal sample values. They then interpolated this veil image by adding glare effect to the original image. The equation 4.19 is used to compute the veiling luminance corresponding to a given foveal sample:

$$L_{vi} = 0.087 \cdot \frac{\sum_{j \neq i} \frac{L_j \cdot \cos(\theta_{i,j})}{\theta_{i,j}^2}}{\sum_{j \neq i} \frac{\cos(\theta_{i,j})}{\theta_{i,j}^2}}; \quad 4.19$$

where L_{vi} is the veiling luminance for the pixel i , L_j is the foveal luminance for the pixel j and $\theta_{i,j}$ is the angle between the pixels i and j . The equation 4.19 is a conversion of the integral, representing the glare formula found by Moon and Spencer [MoSp45], to an average over peripheral sample values [Wa+97]. An approximation of the equation 4.19 is presented in order to reduce the computation time. The authors perform the following calculation, in order to derive the final veiled pixel at image position k :

$$L_{pvk} = 0.913 \cdot L_{pk} + L_v(k), \quad 4.20$$

where L_{pk} is the original pixel at image position k and $L_v(k)$ is the interpolated veiling luminance at k . In particular, the $L_v(k)$ function is a bilinear interpolation on the four closest samples in the veil image computed in equation 4.16. The colour sensitivity has been implemented using the equation 4.21:

$$Y' = Y \cdot \left[1.33 \cdot \left(1 + \frac{Y+Z}{X} \right) - 1.68 \right], \quad 4.21$$

where XYZ are the tristimulus values, and Y' is the scotopic luminance. Finally, for the visual acuity a Shaler's data image has been calculated with the equation 1.2 and a variable filter resolution has been implemented with the *mip-map* technique introduced by Williams [Wi83]. At each point in the image, the authors interpolate the local acuity based on the four closest veiled foveal samples and Shaler's data.

Their model can reproduce the visibility, the overall impression of brightness, contrast and colour. In comparison with the other models, it reproduces high level of realism, but as reported in Tumblin et al. [Tu+99] the histogram adjustment is troublesome. In fact the limitations due to the global nature of the technique become obvious when the input scene

exhibits a uniform histogram (see the example of DiCarlo and Wandell [DiWa00]) [DuDo02].

Tumblin and Rushmeier 1998

Tumblin and Rushmeier [TuRu99] presented a TM operator that is an improved version of their model presented in 1993 [TuRu93]. Their new operator is given by:

$$L_d = m(L_{rwa}) \cdot L_{da} \left(\frac{L_{rw}}{L_{rwa}} \right)^{\gamma_{rw}/\gamma_d}, \quad 4.22$$

where L_{da} is the display adaptation luminance, typically between 10-30 cd/m^2 ; L_{rwa} is the scene adaptation luminance, found from scene luminances L_{rw} using $\log(L_{rwa}) = \text{mean}\{\log(L_{rw} + 2.310^{-5} cd/m^2)\}$; γ_d is $\gamma(L_{da})$; and γ_{rw} is $\gamma(L_{wa})$, i.e. Steven's contrast sensitivity for human adapted to the display and the real world scene, respectively. The γ values are computed using equation 4.23:

$$\gamma(L_a) = 2.655 \text{ for } L_a > 100 cd/m^2; 1.855 + 0.4 \cdot \log_{10}(L_a + 2.310^{-5}) \text{ otherwise} \quad 4.23$$

$m(L_{rwa})$ is the adaptation-dependent scaling term to prevent anomalous grey night images:

$$m(L_{rwa}) = \left(C_{\max}^{1/2} \right)^{(\gamma_{wd}-1)}, \quad 4.24$$

where C_{\max} is the maximum available display contrast, and

$$\gamma_{wd} = \left(\frac{\gamma_{rw}}{1.855 + 0.4 \cdot \log(L_{da})} \right). \quad 4.25$$

The term m steadily increases display brightness as the real world scene adaptation luminance L_{rwa} increases towards the upper limits of vision [Tu+99].

Tumblin et al. 1999

In their works Tumblin et al. [Tu+99] presented two methods for displaying high contrast scenes. The first one builds a display scene from several layers of lighting and surface properties. Only the lighting layers are compressed, drastically reducing contrast while preserving most of the scene details. This method is practical only for synthetic scenes where the layers can be retained from the rendering process. The second method interactively adjusts the display scene to preserve local contrasts in a small foveal neighbourhood. Unlike the first method, this technique is usable on any scene. Both methods use a sigmoid function for contrast compression. This function has no effect when applied to small signals but compresses large signals to fit within an asymptotic limit [Tu+99].

Drago et al. 2003

Drago et al. [Dr+03] proposed a high quality TM operator to visualize high contrast scenes on the display. Their method is based on logarithmic compression of luminance values, imitating the human response to light. A bias power function is also introduced to adaptively vary logarithmic bases, resulting in good preservation of details and contrast [Dr+03]. The final luminance display pixel is obtained by equation 4.26:

$$L_d = \frac{L_{d_{\max}}}{\log_{10}(L_{r_{w_{\max}}} + 1)} \frac{\log(L_{r_w} + 1)}{\log \left(2 + \left(\left(\frac{L_{r_w}}{L_{r_{w_{\max}}}} \right)^{\frac{\log(b)}{\log(0.5)}} \right) \cdot 8 \right)}, \quad 4.26$$

where L_{r_w} is the luminance pixel of the scene, $L_{r_{w_{\max}}}$ is the maximum luminance of the scene and both values are normalized by $L_{r_{wa}}$ (luminance world adaptation) and scaled by an exposure factor.

$L_{d_{\max}}$ is the maximum luminance capability of the display medium. To improve contrast in dark areas, the authors proposed some changes to the gamma correction procedure.

To compress the scene luminance values using logarithm functions, which are computed using different bases depending on the scene content, ensure to keep good contrast and visibility in the darkest area and to reinforce the contrast compression in the lightest area. The resulting TM operator is simple and elegant. However, the computation of the logarithmic function and the need of finding the maximum luminance value of the input scene increase the requested computation time.

4.2.2 Local operators

Schlick 1994

The non-uniform quantization process presented by Schlick [Sc94] is applied when the intensity of each pixel is not quantized with the same TM function.

This process takes into account the fact that the subjective perception is not uniform. Indeed the observer does not view a scene as a whole, since his eyes are continuously moving from one point to another, and for each point, on which the eye is focuses, there is an existing surrounding zone that creates some local visual adaptation and this modifies brightness perception [Sc94].

Schlick proposed this empirical scheme to account the non-uniform subjective perception. For each pixel intensity, the value $ZoVal$ is computed according to the average intensity of a given zone surrounding the pixel.

The ratio $ZoVal / MiVal$ characterizes the brightness of a zone, and is used to modify the parameter p in equation 4.14, as follow:

$$p' = p \cdot \left(1 - k + k \cdot \frac{ZoVal}{MiVal} \right), \quad 4.27$$

with k in the range $[0,1]$. This value represents the weight of non-uniformity of the TM operator. The value $MiVal$ represents the geometrical mean:

$$MiVal = \sqrt{LoVal \cdot HiVal}, \quad 4.28$$

which divides the dynamic range in two equal sub-ranges [Sc94]. There is again an open question about the proposed model. For instance, it is difficult to give a meaningful value to k and the author used in his experiments a value $k=0.5$. Moreover, the author proposed three techniques to define a scheme to compute $ZoVal$ for each pixel [Sc94], and he defined an extension to colour scenes. One technique caused halo artefacts, while the results of the tests indicates that the other two techniques were inferior to the uniformly applied mapping function [TuRu99].

Pattanaik et al. 1998

Pattanaik et al. [Pa+98] developed a computational model of adaptation and spatial vision for realistic tone reproduction. This model is based on a multiscale representation of pattern, luminance, and colour processing in the HVS. The model is incorporated into a TM operator that maps the vast range of radiances found in the real and synthetic scenes into the small fixed ranges available on conventional output devices such as CRT's and printers. A complete scheme of the model is shown in Figure 4.4.

This model allows accounting for the changes in threshold visibility, visual acuity, colour discrimination, super threshold brightness, colourfulness and apparent contrast that occur with changes in the level of the illumination in scenes. However, the high computational cost of this method has so far precluded use for the purposes of real-time rendering.

This model seems to be the most faithful to human vision, however, it may still presents halos artefacts [DuDo02].

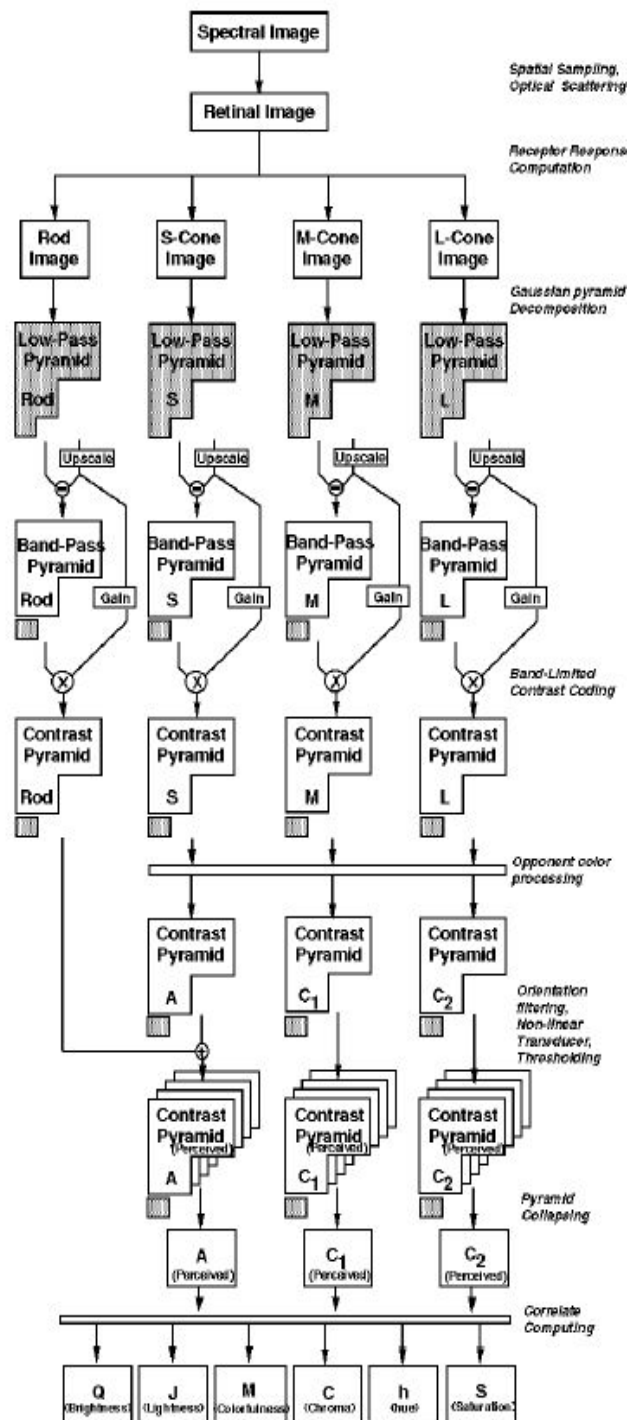


Figure 4.4: Multiscale model, after [Pa+98].

LCIS model

Tumblin and Turk [TuTu99] proposed a TM operator called “*Low Curvature Image Simplifier*” (*LCIS*). This technique is inspired by anisotropic diffusion and mimics the artist’s drawing process in reverse; it selectively removes details from a scene leaving only smoothly shaded regions separated by sharp boundaries. The removed details are recovered

by subtracting the LCIS-smoothed images from the original scene, and then follow the artists scheme for detail-preserving contrast reduction: the contrasts of the simplified image is strongly compressed, then the details with little or no compression are added.

Their method can extract exquisite details from high contrast images. Unfortunately, the solution of their partial differential equation is a slow iterative process [DuDo02]. Moreover, the coefficients of their diffusion equation must be adapted to each image, which makes this model more difficult to use, and the extension to animated sequences unclear [DuDo02].

Ashikhmin 2002

Ashikhmin [As02] defined a TM strategy that consist of three steps. First, an estimation of local adaptation luminance at each pixel in the input scene is performed. Second, a simple function is applied to these values to compress them into required display luminance range. Finally, the details are re-introduced over the image. The author introduced a simplified version of the *TVI* function, as showed in Figure 4.5. The simplified *TVI* form is given by four linear segments, in log-log space, AB, BC, CD and DE. The TM function used by Ashikhmin is reported in equation 4.29 below:

$$TM(L) = L_{d_{\max}} \cdot \frac{C(L_{rw}) - C(L_{rw_{\min}})}{C(L_{rw_{\max}}) - C(L_{rw_{\min}})}, \quad 4.29$$

where $C()$ is the output of the TVI function. A new procedure to estimate the local adaptation is proposed by Ashikhmin. This procedure is based on balancing two opposing requirements, which are faced by HVS: keeping the local contrast signal within reasonable bound while maintaining enough information about image details [As02]. Once the adaptation image and its tone mapped version are produced, the final details can be add over the image using the follow equation:

$$L_d(x, y) = TM(L_{rwa}(x, y)) + \frac{TVI(TM(L_{rwa}(x, y)))}{TVI(L_{rwa}(x, y))} \cdot (L_{rw}(x, y) - L_{rwa}(x, y)) \quad 4.30$$

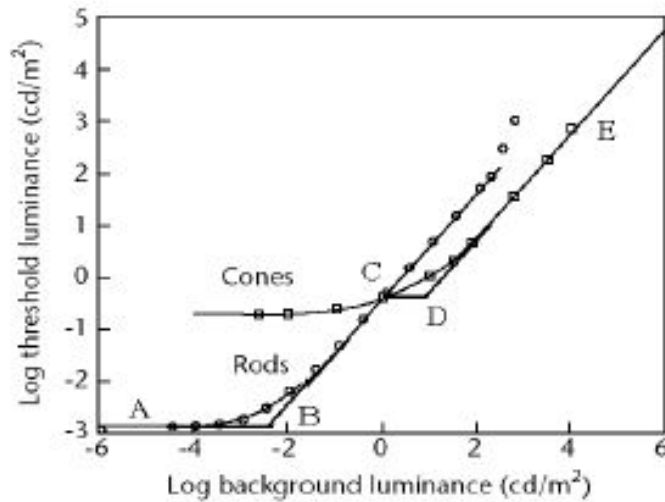


Figure 4.5: TVI Ashikhmin, after [As02].

The main advantage of this model is its modularity. However, some of its drawbacks are: the estimation technique of the local adaptation luminance sometimes creates jumps in adaptation level away from high contrast image features producing artefacts (see [As02]). Moreover, this procedure is not well suited for images where local contrast is uniform everywhere (see [As02]).

Adaptive Gain Control

Pattanaik et al. [PaYe02] introduced a technique to reproduce, on a display, high dynamic range images without introducing any artefacts. The artefacts are usually introduced in the form of halos, around the high contrast edges (i.e. the boundaries that separate the bright areas from dark areas). The authors proposed a detail preserving local gain control approach to reduce the artefacts in the final image. Their technique has the advantage to be simple and not computationally expensive in comparison with LCIS technique [PaYe02].

Photographic Tone Reproduction

Reinhard et al. [Re+02] proposed a TM operator that uses the basic concepts of a technique, called *Zone System*, used by photographers and defined by Ansel Adams (see [Ad80][Ad81][Ad83]), to manage choices in their TM operator. They first apply a scaling that is analogous to setting the exposure in a camera. Then, if needed, they apply automatically *dodging-and-burning* to preserve the image details for very high dynamic range images.

Their operator is able to reduce the halos artefacts and, compared with many existing TM operators, it preserves better fine details.

The main two disadvantages of this tone mapping technique are the large number of needed parameters and the high computational costs. Also the way to set these parameters is unclear.

Fast Bilateral Filtering

Durand and Dorsey [DuDo02] presented a new technique, that reduces the contrast while preserving details. Their technique, is based on a two-scale decomposition of the image into a base layer, encoding large-scale variations, and a detail layer. Only the base layer has its contrast reduced, thereby preserving details. The base layer is obtained using an edge-preserving filter, called *bilateral filter*.

This is a non-linear filter, where the weight of each pixel is computed using a Gaussian in the spatial domain multiplied by a function in the intensity domain that decreases the weight of pixels with large intensity differences.

The resulting TM operator shows interesting results concerning edge preserving filtering and contrast reduction. The authors presented an acceleration technique, that shows promising results but is not supported by enough experimental data.

Gradient Domain model

Fattal et al. [Fa+02] presented a new method which is conceptually simple, computationally efficient, robust and easy to use. They manipulate the gradient field of the luminance image by attenuating the magnitudes of large gradients. A new, low dynamic range image is then obtained by solving a Poisson equation on the modified gradient field. Their approach relies on the assumption introduced by DiCarlo and Wandell [DiWa00] that the HVS is not very sensitive to absolute luminance reaching the retina, but rather responds to the changes of local intensity ratio and reduces the effect of large global differences, which may be associated with illumination differences. Their algorithm is based on a simple observation that any drastic change in the luminance across a high dynamic range image, must give a rise to large magnitude luminance gradients at some scale. Fine details, however, correspond to gradient of much smaller magnitude. Their idea is to identify large gradients at various scales, and attenuate their magnitudes keeping their direction unaltered. A reduced high dynamic range scene is then reconstructed from the attenuated gradient field [Fa+02].

The authors presented an efficient TM technique that shows promising results. Nevertheless, resolving the Poisson equation require a high computation costs.

Other works

A few other computer graphics researchers have modelled the appearance of extremely bright, high-contrast scene features by adding halos, streaks and blooming effects to create

the appearance of intensities well beyond the abilities of the monitor [Tu+99]. Nakamae et al. [Na+90] proposed a model on streaks of light taking into account both refraction and diffraction of light.

Spencer et al. [Sp+95] have presented later on an extensive summary of the optical causes and visual effects of glare and modelled their appearance by using several adjustable low-pass filters on the intensities of the original scene [Tu+99].

4.3 Time-Dependent Tone Mapping

Through the visual adaptation process the HVS adjusts itself to the conditions under which the eyes are exposed to radiant energy. This process takes time, and it is different for the different kinds of adaptation: light, dark, and chromatic adaptation. Indeed, the light adaptation is faster than dark adaptation. Large abrupt changes in scene intensities can cause dramatic compression of visual responses, followed by a gradual recovery of normal vision. Asymmetric mechanisms govern these time-dependent adjustments. Adaptation and its changes over the time have profound effects on the visual appearance of any viewed scene. Continual adjustment helps to keep the visual system acutely sensitive to scene content over a wide range of illumination, but adaptation also tends to hide or obscure any very slow change in scene intensity or spectral content [Pa+00].

4.3.1 Pattanaik et al. 2000

Pattanaik et al. introduced the first time-dependent TM algorithm. It follows the block diagram in Figure 4.6. This algorithm is derived from published quantitative measurements from physiology, psychophysics, colour science, and photography.

It receives as input the viewed scene intensities to retinal-response-like vectors R , and the *appearance model* converts R to appearance vectors Q that express correlates of whiteness/blackness and colourfulness. The upper model pair computes display intensities that match the scene appearance, and the lower pair of inverse model computes display intensities that match the scene appearance (see Figure 4.6). The forward adaptation model is a simplified version of a Hunt's model [Hu95], augmented with exponential filters for time-dependent adaptation mechanisms [Pa+00].

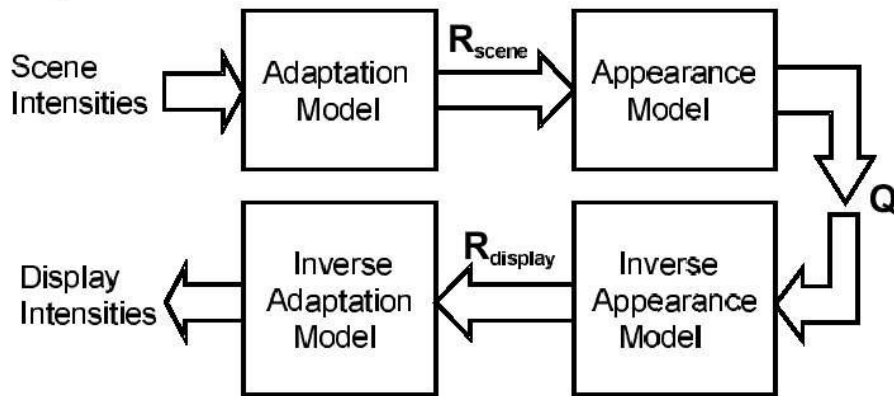


Figure 4.6: Pattanaik et al. operator, after [Pa+00].

The model presented by Pattanaik et al. [Pa+00] can reproduce the time dependency of the HVS, but it presents several drawbacks as will be shown in Chapter 8 (Experimental Results). In particular, for a huge dynamic range scene their model can not reproduce the right scene contrast.

Moreover, the appearance model, used by Pattanaik et al., does not suffices to capture the real appearance of the scene. Indeed, no chromatic adaptation model, that takes into account the different viewing conditions of the two observers (i.e real world, display system), is used.

4.3.2 Durand and Dorsey 2000

The authors [DuDo00] presented an interactive TM process (see Figure 4.7 below).

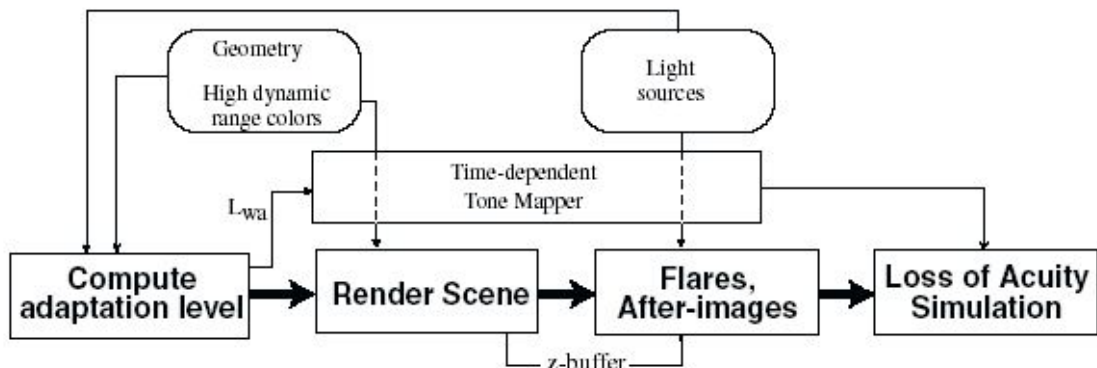


Figure 4.7: Durand Dorsey 2000, after [DuDo00].

This process takes as input a 3D scene and a set of point-light sources. The authors employed a multipass scheme to compute the adaptation level. This is mapped to the displayed colour for each frame. Finally, the authors add glare to improve the appearance of the light sources. Moreover, they used a filtering step to simulate the loss of acuity. They make the assumption that the three RGB components correspond to the three cone types. This assumption is not valid in the application where the colour accuracy is crucial [Hu95], [Fa98]. To solve this

problem, the authors proposed a matrix colour transform at the end of the TM operator. They define a mapping operator modifying the operator of Ferwerda et al. [Fe+96] operators. Hence their operator is based on Ward's model, but introducing a new equation to define the mesopic factor K , used in equation 4.17:

$$K = \frac{\sigma - 0.25 \cdot L_{rwa_{rod}}}{\sigma + L_{rwa_{rod}}}, \quad 4.31$$

where $\sigma = 100cd/m^2$, and $L_{rwa_{rod}}$ is the world adaptation luminance for the rod receptors. They also extend the Ferwerda's et al. model [Fe+96] to time-dependent tone mapping by incorporating a simple model of visual adaptation. They also defined a simple model of light and chromatic adaptation where the time variable is considered. A model of dark adaptation is missing.

Finally, they implemented an interactive version of the glare model proposed by Spencer et al. [Sp+95], as well as a model simulating the loss of visual acuity [DuDo00].

4.4 Real-Time Tone Mapping

In the previous sections several kinds of TM operators have been discussed. These operators can reproduce different aspects as contrast, colour, overall impression of the brightness, time dependency of the HVS, local adaptation, visibility of the objects etc.

The simulation of these complex processes is time consuming, and hence are not feasible in the case of interactive and real time applications.

One possibility of delivering interactivity is implementing a TM operator directly in graphics hardware with support for floating-point colour representation and programmable pixel shaders. Due to the hardware constraints, this approach however imposes severe limitations on the operator: (1) it cannot perform arbitrary data manipulation; (2) its complexity is limited by the maximum number of instructions of a pixel shader [Ar+03].

The direct implementation of the TM operator on graphics hardware was proposed in [Co+01][Dr+03][Go+03]. However, these works do not define a general solution to deliver interactivity of existing TM operators.

Cohen et al. [Co+01] proposed a TM operator suitable for hardware implementation. Due to hardware constraints, this operator is rather simple and does not consider effects as time dependency, chromatic adaptation, and other effects of the HVS.

Drago et al. [Dr+03] presented a hardware implementation of their own global TM operator proposing some ideas to reduce its computation time.

Goodnight et al. [Go+03] proposed an implementation of a suitable modification of the model presented by Reinhard et al. [Re+02] in order to reproduce the time-dependency of the HVS.

Several works tried to reduce the computation time of their own TM operators using some strategies (see [Wa+97][Sc+00][DuDo00][DuDo02]).

In particular, Ward et al. [Wa97] down-sample the image to extract histogram information, which is used to set up the parameters for their TM algorithm.

Scheel et al. [Sc+00] perform ray casting to obtain high dynamic range image samples and use the texture mapping hardware to apply TM on vertices of a model with precomputed radiosity.

Durand and Dorsey [DuDo00] down-sample the image to compute the adaptation luminance and use a lookup table to speed up their interactive time dependent TM algorithm. In their recent work, Durand and Dorsey [DuDo02] also use down-sampling and linear interpolation.

4.5 Tone Mapping for Printers

Several works in computer graphics have addressed to resolve the TM question on display. However, to the best of our knowledges, no works providing a solution on the same question on printer systems have been presented so far. On the contrary, several works in colour science community, connected with the TM question, do exist. In particular, there are three basic areas of particular interest as: *gamut mapping*, *colorimetric characterization of devices* and *colour appearance*.

Gamut mapping is used to map a large gamut of colours, reproducible by a device, on a small gamut of colour reproducible by a different device. As described in Section 3.4, colorimetric characterization is used to map colour of device-dependent colour spaces in a device-independent colour spaces and vice versa. On the contrast, colour appearance is used to predict as the colour is modified if the viewing conditions of the observer are changed. In order to solve the open questions in these areas, different solutions are presented (see [Ar97][ArWi01][Be96][ArWi03][KaAn92][Ka+95][MaAb94][Xi+99][RoBa93]). In colour science the use of HDR scene is concerning the direct acquisition of the real world using special

scanner, digital or traditional photography [LeLe][UrMa97] and try to visualize it on a display or print it on a printer.

These acquisition instruments, are not able to acquire properly the dynamic range of the scenes. In this way the data are compressed in a small range and the information of the original scene can be lost. This is the basic difference between the TM question in computer graphics and what is available in the area of colour science. In computer graphics, there are just a few results on the colorimetric conversion between devices involved in a multimedia system, and only some processes utilized in colour science has been used.

Chapter 5

Novel Time-Dependent Tone Mapping Model

Introduction

The interval of luminance values that can be found in scenes of the real world can be very broad; the HVS has the ability to adapt to these dynamic intervals, but it needs some time to do so.

The cells in our visual system are triggered by a relatively small range of luminance values; large and fast changes in the lighting conditions cause this triggering interval to shrink further, until gradually adapting to the new conditions [Ar+01]. The mechanisms, which control these time-dependent adaptations, are asymmetric; it takes more time to adapt to darkness than to bright surroundings [Pa+00]. These processes that take place entirely inside the retina [Do87] are referred to as *visual adaptation*; it has a substantial impact on the appearance of an image.

Continuous adaptations help to keep the visual system sensible towards the different luminance conditions that can be found in real world scenes [Pa+00]. But not only the perception of the brightness varies with changing viewing conditions, the perception of the colours is affected as well. If we want to capture the visual appearance of a real image completely, it is necessary to consider the colour appearance. The model of chromatic adaptation defined by Von Kries, for example, allows predicting the correct reproduction of colours with respect to varying viewing conditions [Fa98]. Therefore, if we want to visualize a real scene on a monitor we need these models, as the monitor has different viewing conditions than the real world [Ar+01].

In this chapter we present a novel time-dependent algorithm, which is based on an existing model [Pa+00], but includes substantial modifications in order to improve its limitations as unable to capture completely the colour reproduction, and a deficiencies when applied on a very huge dynamic range.

In order to capture it, we introduced a different *luminance mapping* technique and the *Von Kries* chromatic adaptation model. We also, followed other strategies to convert the retinal response values to luminance display values. Our TM operator also takes into account other aspects, like the visibility of objects or the subjective experience when viewing a real scene such as the overall impression of brightness, contrast and colours. Finally we integrated some aspects of the HVS as: glare, visual acuity, and colour sensitivity. Our operator is able to work for low and high dynamic ranges.

5.1 Chromatic Adaptation

The ability of the HVS to change its sensitivity towards determined stimuli in answer to changes of the environmental conditions is referred to as *adaptation*. There are three important types of visual adaptation: Light, Dark and Chromatic adaptation. *Chromatic-adaptation*, as reported in Section 1.3.3, [Fa98] is the human visual system's capability to adjust to widely varying colours of illumination in order to approximately preserve the appearance of object colours.

The adaptation to light or darkness has a substantial impact on the colour appearance, but the chromatic adaptation has even bigger importance. Indeed, it allows perceiving the colour correctly under different viewing conditions. All types of visual adaptation must be included in all models that try to capture the correct colour appearance [Ar+01].

In order to understand better the chromatic adaptation mechanism, we introduce the *corresponding colours* concept. It refers to different stimuli, which are perceived under different viewing conditions but appear as the same colour, have the same colour appearance [Fa98]. For example, one tristimulus value XYZ_1 perceived under determined viewing conditions can appear equal to another stimulus defined by XYZ_2 and a different set of viewing conditions.

The tristimulus values of both colours, along with their corresponding viewing conditions, are referred to as 'corresponding colours'. The tristimulus values of two corresponding colours are rarely numerically identical [Fa98]. A general model of chromatic adaptation

does not include appearance attributes such as lightness, chroma, and hue. But it provides the transformation of tristimulus values from one set of viewing conditions into a different set of viewing conditions [Fa98]. In other words, a model of chromatic adaptation allows predicting the corresponding colours.

A generic model of chromatic adaptation starts from the initial *LMS* values, response of the cones to the initial viewing conditions, and predicts the new signals L_{new} , M_{new} , and S_{new} under the new viewing conditions. In general the colours are expressed using the *CIE* tristimulus values *XYZ*, but fortunately there exists a 3×3 linear transformation between the signals of the *LMS* cones and the *CIE* tristimulus values *XYZ*. A general chromatic adaptation model [Fa98] follows the steps as depicted in Figure 5.1. The first step is the transformation of the *CIE* tristimulus values $X_1Y_1Z_1$, of the first viewing conditions (eg. real world in our case), to cone excitations $L_1M_1S_1$. It is obtained with a simple linear transformation. With the information about the first set of viewing conditions is possible to obtain the cone signals $L_aM_aS_a$ adapted to it. Instead, with the information about the second set of viewing conditions, we determinate the new cone signals $L_2M_2S_2$ adapted to the new visual state (eg. display in our case). The final step is the inverse transformation in the *CIE* tristimulus values $X_2Y_2Z_2$, of the second viewing conditions. It is obtained also with a linear transformation. In this way the final tristimulus values $X_2Y_2Z_2$ are adapted to the new viewing conditions, and only in this way we are able to preserve the appearance of the object colours.

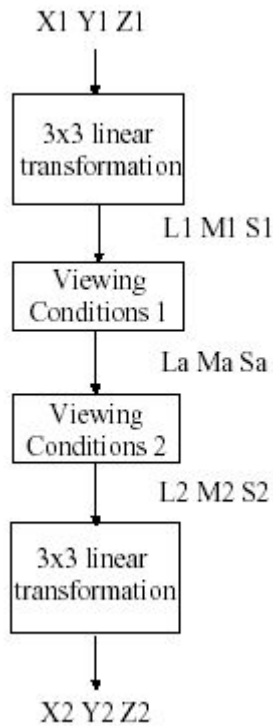


Figure 5.1: General scheme of a chromatic adaptation model, after [Fa98].

5.2 Model Development

The novel time-dependent model presented in this section, is based in principle on the time-dependent model presented by Pattanaik et al. [Pa+00].

In order to facilitate comparisons between the two methods, we first present Pattanaik et al. [Pa+00] model, and then we discuss its limitations and present the solution.

5.2.1 Pattanaik et al. Model 2000

This section provides a complete description of all parts of the time-dependent tone reproduction operator presented by Pattanaik et al. [Pa+00]. The operator follows the scheme in Figure 5.2. It receives an input sequence of scene values expressed in cd/m^2 and gives as output displayable sequence of RGB images.

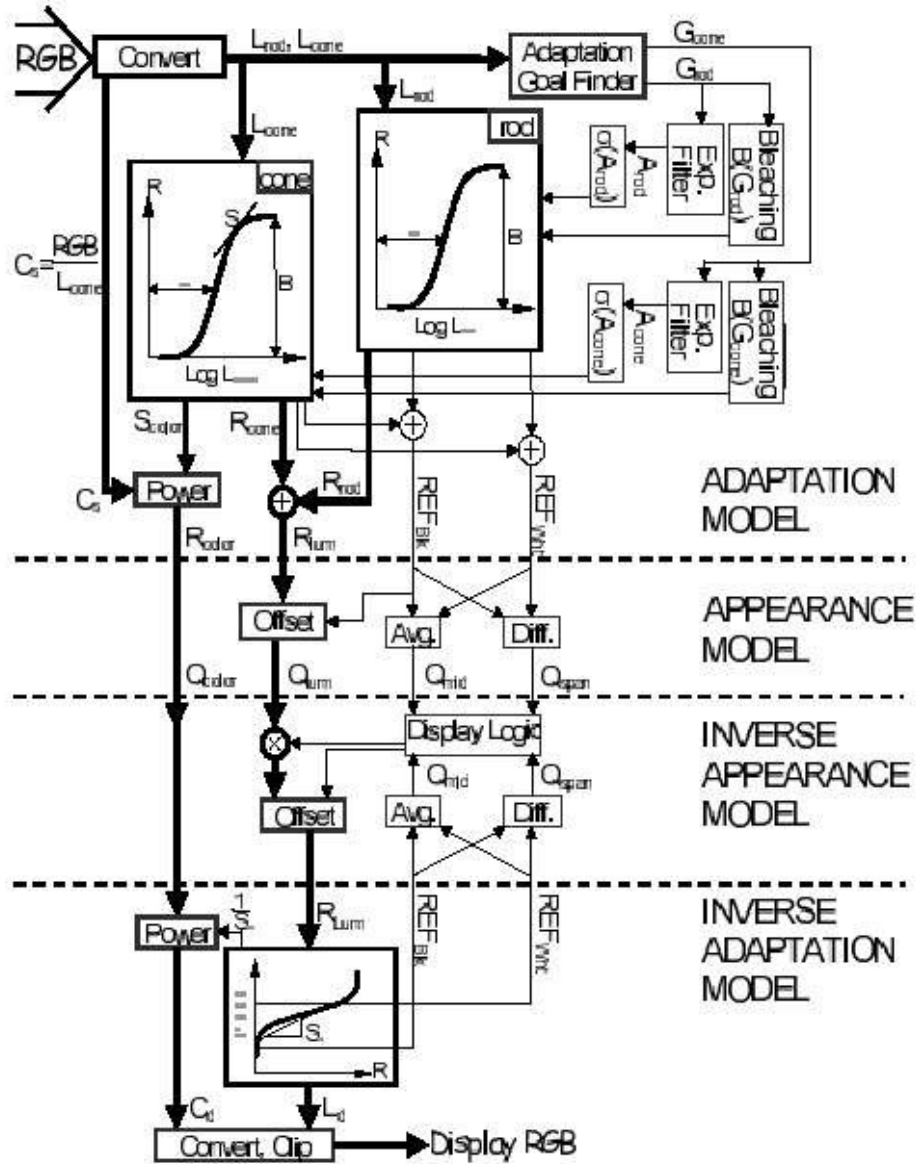


Figure 5.2: Pattanaik et al. time-dependent tone-mapping operator, after [Pa+00].

It is constituted by four phases called: *adaptation model*, *appearance model*, *inverse appearance model* and *inverse adaptation model*.

We describe now these four phases using the same annotation as used in the original paper [Pa+00].

Adaptation model

The adaptation model is a simplification of Hunt's static model of colour vision [Hu95], which adds new time-dependent adaptation components.

As shown in Figure 5.2, a scene in RGB values or radiance is converted into luminance for both rods and cones (CIE standard Y' and Y), labelled L_{rod} and L_{cone} respectively [Pa+00].

The first step is to compute both rod and cone luminance responses using equation 1.1, see Chapter 1, with the maximum response R_{\max} given by a photo pigment bleaching term B :

$$R_{rod} = B_{rod} \cdot \frac{L_{rod}^n}{L_{rod}^n + \sigma_{rod}^n}, \quad 5.1$$

$$R_{cone} = B_{cone} \cdot \frac{L_{cone}^n}{L_{cone}^n + \sigma_{cone}^n}. \quad 5.2$$

The parameter σ , as reported in Chapter 1, is the L value that causes the half-maximum response. Both B and σ are determined by adaptation to overall scene luminance [Pa+00]. The authors approximated the amount of colour compression caused by equations 5.1 and 5.2 at L_{cone} , and apply it to the colour ratio components:

$$R_{color} = \left((red, green, blue) / L_{cone} \right)^{S_{color}}, \quad 5.3$$

where:

$$S_{color} = \frac{nB_{cone}L_{cone}^n\sigma_{cone}^n}{\left(L_{cone}^n + \sigma_{cone}^n\right)^2}. \quad 5.4$$

The parameter n is taken as suggested by Hunt's equal 0.73 [Pa+00]. The authors defined the parameter σ for rods and cones as [Pa+00]:

$$\sigma_{rod} = \frac{2.5874 \cdot A_{rod}}{19000 \cdot j^2 \cdot A_{rod} + 0.2615 \cdot (1 - j^2)^4 \cdot A_{rod}^{1/6}}, \quad 5.5$$

$$\sigma_{cone} = \frac{12.9223 \cdot A_{cone}}{k^2 \cdot A_{cone} + 0.171 \cdot (1 - k^4)^2 \cdot A_{cone}^{1/3}}, \quad 5.6$$

$$\text{where: } j = \frac{1}{5 \cdot 10^5 \cdot A_{rod} + 1}, k = \frac{1}{5 \cdot A_{cone} + 1}. \quad 5.7$$

The authors, unlike the Hunt's model, used the time varying adaptation amounts (A_{rod}, A_{cone}), and the slower asymmetric effects from photo pigment bleaching, regeneration and saturation effects (B_{rod}, B_{cone}) dynamically. The authors used two forms of exponential smoothing filters applied to the adaptation goals signals G_{rod} and G_{cone} computed for every frame [Pa+00]. To compute the goal signals different solutions have been proposed [Pa+00], and the best choice may depend on the application.

The fast, neurally-driven adaptation values A are computed from the goal values G using simple fixed exponential filters F :

$$F = 1 - \exp(-T / t_0), \quad 5.8$$

where T is the time necessary for the adaptation for the single frame. Instead the constant t_0 is equal to $t_{0,rod} = 150ms$ for the rods photo receptors, and to $t_{0,cone} = 80ms$ for the cone photo receptors [Pa+00]. They do not distinguish between multiplicative and subtractive adaptation because the former is usually completed within one or two frame times [Pa+00]. Finally the value of A is computed following equation 5.9:

$$A_i = A_{i-1} + (J_A - K_A), \quad 5.9$$

where the index i describe the frame number, and for the first frame the value of A is equal to the goal value G . The function J_A and K_A are fixed scale factors [Pa+c00]:

$$J_A = F \cdot G_i, \quad 5.10$$

$$K_A = F \cdot A_{i-1}. \quad 5.11$$

The time-dependent bleaching factor B is computed following the filter proposed in [Pa+00], that is an extension of the Hunt' static expression in order to include pigment kinetics.

$$B_i = B_{i-1} - (J_B - K_B), \quad 5.12$$

where B in the first frame is equal to:

$$B_1 = \frac{const}{const + G_1}. \quad 5.13$$

The *const* value is the same constant used in the Hunt's static expression [Hu95]. The function J_B and K_B are computed following [Pa+c00]:

$$J_B = \frac{T \cdot B_{i-1} \cdot G_i}{const \cdot t_0}, \quad 5.14$$

$$K_B = \frac{T \cdot (1 - B_{i-1})}{t_0}, \quad 5.15$$

where the constant t_0 is equal to $t_{0,rod} = 400s$ for the rods photo receptors, and to $t_{0,cone} = 110s$ for the cones photo receptors.

Visual Appearance Model

Pattanaik et al. [Pa+00] define a simple visual adaptation model which assumes that a human can assign equivalent appearance to dim displays and very bright or very dark scenes by a simple linear mapping of visual responses. The model determines "reference white" and

“reference black” responses from among the current visual response against the reference standards.

As shown in Figure 4.6, their appearance model computes luminance appearance values Q_{Lum} by subtracting reference black response REF_{black} from R_{Lum} , where $R_{Lum} = R_{cone} + R_{rod}$.

They followed the Hunt’s suggestion determining the reference white as five times the current adaptation level and the reference black as 1/32 the intensity of the reference white.

The final responses to reference white and black are [Pa+00]:

$$REF_{white} = R_{rod} \Big|_{L_{rod}=5A_{rod}} + R_{cone} \Big|_{L_{cone}=5A_{cone}} \quad 5.16$$

$$REF_{black} = R_{rod} \Big|_{L_{rod}=5/32A_{rod}} + R_{cone} \Big|_{L_{cone}=5/32A_{cone}} \quad 5.17$$

Instead the midrange of visual responses are defined as $Q_{span} = (REF_{white} - REF_{black})$ and $Q_{mid} = 0.5(REF_{white} + REF_{black})$. Finally Q_{Color} values are set by R_{Color} values of the scene [Pa+00].

Tone Reproduction Operator

The final Tone Reproduction operator is derived assembling to the adaptation and appearance model the inverse appearance and adaptation model as shown in Figure 4.6 [Pa+00]. In this way the scene appearance values are converted backwards into display intensity.

Inverse Appearance Model

This model translates the visual appearance into the display. Pattanaik et al. [Pa+00] did the following assumption: “the display minimum and the maximum values will evoke REF_{black} and REF_{white} responses in the viewer, and their inverse appearance model attempts to map scene appearance values Q to a display observer’s response values with a little distortion“.

The inverse appearance model is based on the following four rules:

1. IF the display can directly reproduce the scene visual responses, do so. Exactly cancel the offset to R_{Lum} that was applied by the forward appearance model. ELSE
2. IF scene $Q_{span} > displayQ_{span}$, compress and offset scene Q_{Lum} to match scene REF_{white} and REF_{black} to display REF_{white} and REF_{black} . ELSE

3. IF $sceneQ_{mid} > displayQ_{mid}$, offset scene Q_{Lum} downwards only enough to ensure $sceneREF_{white} \leq displayREF_{white}$. ELSE
4. Offset scene Q_{Lum} upwards only enough to ensure $displayREF_{black} \leq sceneREF_{black}$.

Inverse Adaptation Model

The aim of this model is to find RGB display values for a given set of visual response values R_{Lum} and R_{Color} [Pa+00]. Pattanaik et al., for simplicity, made the following assumptions: first, assumed that the display device gamma is 1.0, forcing proportionality between RGB and display intensity values. Second, assumed that the display observer have fixed steady-state adaptation amounts. The display luminance value is derived inverting equation 1.1. The colour component is not derived by the inversion of equation 5.3, for simplicity, but instead they compute a constant display S_d value from a forward-difference of the slope of equation 1.1 measured between the display REF_{white} and REF_{black} [Pa+00]. They raise the colour appearance value Q_{Color} to the power of $1/S_d$ to convert it to display colour ratio C_d .

5.2.2 Novel Time-Dependent Tone Mapping Model

In this section we present the modifications introduced in the original time-dependent tone-mapping operator of Pattanaik et al. [Pa+00]. Their model has been designed to simulate the time adaptation process of the HVS and achieved good results. However, following aspects have not been taken into consideration: compression of the HDR that does not allow reproducing the visibility and the contrast correctly, chromatic adaptation, limitations of the HVS (glare, visual acuity, and colour sensitivity). These limitations are also shown in Section 8.1, where the experimental results are presented.

The technique used to compress the HDR of the input scene is too weak for large HDR. As shown in Figure 5.3 we have an output scene, used in several tone-mapping works, obtained by using the original model of Pattanaik et al. [Pa+00]. In this case the visibility of the objects and the contrast are not reproduced properly.



Figure 5.3: Scene with luminance range $[0.37; 18828]$ cd/m^2 [S. Crone, L. Fawler and J. Kerrgan 1997]; scene obtained by using the model of Pattanaik et al. [Pa+00].

A way to overcome these limitations is to modify the original TM operator. We propose a new luminance mapping technique, and introduce a chromatic adaptation model. We also integrated in our TM operator the HVS limitations. In particular, we modified the following steps: *Adaptation*, *Visual Appearance*, *Inverse Appearance* and *Inverse Adaptation* models.

Adaptation model

Concerning the colour response value R_{Color} used by Pattanaik et al. [Pa+00], to discard the complex colour calculation performed by Hunt's model, we used a different strategy also

used in several TM operators. Our strategy consists in multiplying by the ratio $\frac{L_d}{L_{rw}}$ the colour components of the corresponding input pixel. As discussed in Ashikhmin's work [As02], this is the simplest way to deal with colour that provides acceptable results. See, e.g. Figure 5.4, where a comparison of some images obtained using both operators is presented.



Figure 5.4: Colour comparison between images obtained with the two operators: (left) Pattanaik et al. operator [Pa+00]; (right) our operator. [S. Crone, L. Fawler and J. Kerrgan 1997].

From the images in Figure 5.4, one can understand that the colour reproduction with our operator is comparable with the colour reproduction of the Pattanaik et al. operator [Pa+00]. Our method is also faster than the method used by Pattanaik et al., since it requires just one division (compare it with equations 5.3 and 5.4).

We introduced another modification in the dynamic adaptation model for the parameters A_{rod} , A_{cone} , and B_{rod} , B_{cone} . This influences also the compression of the high dynamic range of the original scene. For the parameter A we used the exponential filter expressed in equation 5.8 without updating with equation 5.9 used by Pattanaik et al. [Pa+00]. In our case the parameter T (i.e. time in milliseconds) must be specified for each frame and the parameters A and B will be updated consequently. To allow the parameter B depending to the time T , equations 5.14 and 5.15 have to be modified as follows:

$$J_B = \frac{T \cdot G_i}{const \cdot t_0}, \quad 5.18$$

$$K_B = -\frac{T}{t_0}. \quad 5.19$$

Visual Appearance Model

We used different parameters for the visual appearance model. Indeed, we did not follow the strategy suggested by Hunt as used in the model of Pattanaik et al.

For the reference white REF_{white} , and for the reference black REF_{black} we simply used the minimum and maximum luminance scene values as used in several TM models. This simplification does not introduce visible artefacts as shown by experimental results (see Section 8.1).

For the display system we used the maximum and minimum luminance, reproducible by the monitor, as white and black reference values (REF_{white_d} and REF_{black_d}) both lie in the range $[0.0, 1.0]$.

Inverse Appearance Model

To compute the luminance mapping of the retinal luminance response of the real scene in the range of the display, we have to consider two different possibilities:

- 1 The value of the reference luminance of the scene is contained in the range of the display luminance. In this case, no operation is needed and the value R_{Lum} is not modified.

- 2 The value of the reference luminance of the scene lies outside the range of the display luminance. In this case the new reference luminance value R_{Lum} can be determined by mapping the retinal response of the scene R_{Lum} into the luminance range of display $[0.0, 1.0]$.

$$R_{Lum} = \frac{R_{Lum} - REF_{black}}{REF_{white} - REF_{black}}, \quad 5.20$$

and we used the rules specified in the Pattanaik et al. model [Pa+00].

If the resulting value is larger than 1.0 , it has to be clipped to 1.0 . This strategy is in practice used in many applications in order to normalize a huge range in a smaller one. Equation 5.20 has also the property to compress HDR scene to the display range while trying to convey the overall impression of brightness [As02].

In Figure 5.4 one can observe that, in the last phase of adaptation, our compression method achieves better results than that of Pattanaik et al. model [Pa++00]. Indeed the image on the right (obtained with our method), no areas over-exposed are present. This is in contrast with the image obtained with the model of Pattanaik et al. (see the image on the left).

Inverse Adaptation Model

Once we have obtained the new retinal answer R_{Lum} , compressed in the low dynamic range of the display, the next step is to compute the luminance value of the display.

This can be obtained inverting the equation $R_{Lum} = R_{cone} + R_{rod}$. This essentially amounts in solving a second-degree equation. By contrast, at this stage, the model of Pattanaik et al. [Pa+00] use an approximation. Before calculating the luminance value, one needs to determine the coefficients B_{cone_d} and B_{rod_d} using equations 5.21 and 5.22, for the display.

The coefficients A_{cone_d} and A_{rod_d} for the display are equal to the work display luminance values in the range $[0.0, 1.0]$.

$$B_{cone_d} = \frac{2 \cdot 10^6}{2 \cdot 10^6 + A_{cone_d}}, \quad 5.21$$

$$B_{rod_d} = \frac{0.04}{0.04 + A_{rod_d}}. \quad 5.22$$

The coefficients σ_{cone_d} and σ_{rod_d} , for the display, are determined using equations 5.5 and 5.6, with the parameters A_{cone_d} and A_{rod_d} . The form of the second-degree equation is

$$L_{cone_d}^{2n} a + L_{cone_d} \cdot b + c = 0, \quad 5.23$$

where a , b and c come from the following equations:

$$a = B_{cone_d} + B_{rod_d} - R_{Lum}, \quad 5.24$$

$$b = B_{cone_d} \cdot \sigma_{rod_d}^n \cdot r^n + B_{rod_d} \cdot \sigma_{cone_d}^n + R_{Lum} \cdot \sigma_{cone_d}^n + \sigma_{rod_d}^n \cdot ratio^n, \quad 5.25$$

$$c = -R_{Lum} \cdot \sigma_{cone_d}^n \cdot \sigma_{rod_d}^n \cdot ratio^n, \quad 5.26$$

we also consider the L_{rod_d} luminance value

$$L_{rod_d} = \frac{L_{cone_d}}{ratio}. \quad 5.27$$

Where $ratio = \frac{L_{cone}}{L_{rod}}$.

Since we are also interested in achieving a correct colour reproduction on the display, we have to consider the differences in viewing conditions between the real world and the display.

Von Kries Model

Different chromatic adaptation models (e.g. Von Kries, Naytani, Guth's, and Fairchild's) do exist.

We used the Von Kries model to predict the colours data under the viewing conditions of the display. We applied it to the tristimulus values XYZ_d obtained as output of our time dependent operator. Our choice is due to the simplicity of this model and the fact that every other chromatic adaptation model also follows the same hypothesis.

Every chromatic adaptation model permits to compute the corresponding colours. Moreover using these models one can predict the colour matches across changes in viewing conditions.

Any physiologically plausible model of chromatic adaptation must act on the cone responses [Fa98]. Thus, in applications that use the CIE colorimetric values, $CIEXYZ$ tristimulus values are transformed into cone responses LMS with a linear transformation [Fa98].

$$L = 0.400 \cdot X + 0.708 \cdot Y - 0.081 \cdot Z$$

$$M = -0.226 \cdot X + 1.165 \cdot Y + 0.046 \cdot Z \quad 5.29$$

$$S = 0.000 \cdot X + 0.000 \cdot Y + 0.918 \cdot Z$$

The modern interpretation of the Von Kries hypothesis in terms of a chromatic adaptation model is expressed by the following equation:

$$XYZ_2 = Matrix^{-1} \cdot LMS_{\max 2} \cdot \frac{1}{LMS_{\max 1}} \cdot Matrix \cdot XYZ_1, \quad 5.30$$

where *Matrix* is the matrix that describes the linear relation between CIE tristimulus values *XYZ* and the cone responses *LMS* (see equation 5.27). The values LMS_{\max} represent the cone responses of white point; it is a diagonal matrix with diagonal values LMS_{\max} . The value $1/LMS_{\max}$ is a diagonal matrix too, with diagonal values $1/LMS_{\max}$. *XYZ* represents the pixel tristimulus values.

The indices *1* and *2* represent the different viewers in the real world and in the display respectively. In order to have success with this operation one has to normalize the tristimulus values in the range $[0.0, 1.0]$. The final outputs of our model, after data denormalization, are the new tristimulus values XYZ_{new_d} of the image.

The last step consist in transforming the new tristimulus values XYZ_{new_d} to *RGB* display values, which is obtained with a linear transformation.

5.3 Human Visual System Limitations

To reproduce contrast, visibility of the objects, time dependency, and visual adaptation phenomena has profound effects on the visual appearance of any viewed scene, but this is not enough to offer a truly realistic image reproduction. To this purpose we also need to reproduce the well known limitations of the HVS, such as *visual acuity*, *glare* and *colour sensitivity*.

Visual acuity is the capacity of the visual system to resolve spatial details in light environments, and loose its ability to resolve fine details in dark environments. Glare effect is caused by bright sources in the periphery of the visual field through scattering of light in the lens of the eye, which in turn obscures foveal vision. A realistic treatment of colour sensitivity has to account for the loss of colour vision in dark environments [Wa+97].

5.3.1 Integration of the Human Visual System Limitations in the Time-Dependent Model

The basic idea we implemented is shown in Figure 5.5. We start from an image, which contains HDR values for each pixel, and for each pixel in the image we apply the glare model on the cone luminance value. On the new cone luminance value the visual acuity information

is applied to obtain the updated luminance value. The colour sensitivity model uses this value to calculate a luminance value for the rod receptors. These new values are given to the time dependent algorithm. Is this algorithm that generates luminance values compressed in the low dynamic range of the display.

In the next subsections we explain how we implemented the glare effect, visual acuity and colour sensitivity.

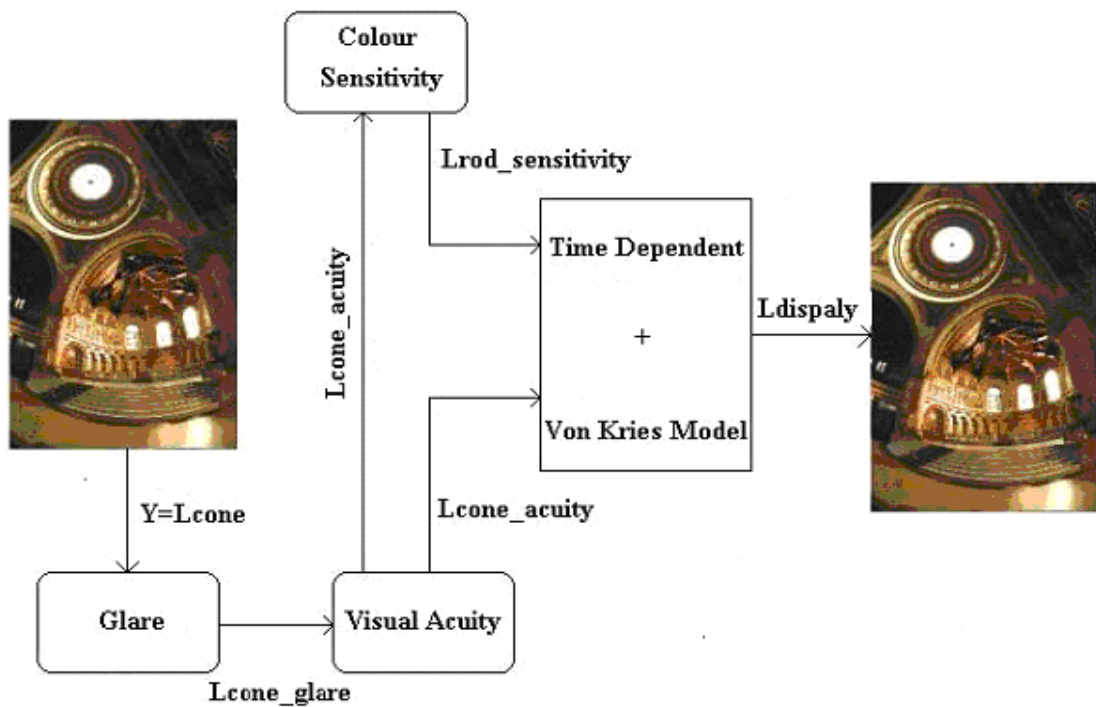


Figure 5.5: Integration of the human visual limitations in the Time-Dependent TM operator, [ArWi02].

5.3.2 Glare

There are several models, which take the glare effect into account. The first one is based on the work of Holladay [Ho26] and Moon and Spencer [MoSp45], and was proposed by Ward et al.[Wa+97].

Spencer et al. [Sp+95] proposed a second model, that is very precise and predicts the all aspects of glare, like *flare* and *bloom*, but it is computationally expensive. This is achieved through a quantitative model, which directly models the bloom, ciliary corona and the

lenticular halo. Finally, generates a filter which is applied to the image in question. Durand and Dorsey [DuDo00] presented a modification of this model.

Also Nakamae et al. [Na+90], in their driving simulator, presented two versions of glare: *blooming* and *streaking*. Blooming represents the hazy fog that is visible around the luminaries; instead streaking represents the case where star-like smears emanate from the luminaires [Ch+93]. In Chiu et al. [Ch+93] has been presented the implementation of blooming.

The glare effect implemented, in this thesis work, is based on the work of Ward et al. [Wa+97] where we compute the veil on the all original scene, and not only to a low resolution of it, using a mask of size 3x3 pixels. In this way we do not need a bilinear interpolation step, as specified in Ward et al. work [Wa+97].

5.3.3 Visual Acuity

The model of visual acuity implemented, in this thesis work, is based on the model explained by Ferwerda et al. [Fe+96]. We use the same Gaussian convolution filter with its Fourier transform given by

$$f^*(R_c(L_{rwa})) = \frac{t(L_{rwa})}{L_{rwa}}, \quad 5.31$$

where L_{rwa} is the local adaptation luminance of the real scene, $R_c(L_{rwa})$ is the threshold frequency, and $t(L_{rwa})$ is the threshold luminance for the world viewer. $R_c(L_{rwa})$ is calculated with Shaler's formula, equation 1.2. However, $t(L_{rwa})$ is calculated by using Ward's model [Wa94]. For every pixel we calculate the visual acuity $R(L_{rw})$ with the Shaler formula, with the pixel luminance of the real scene as input, and we apply the equation 5.31 to the pixel that has visual acuity over the threshold frequency. With this operation we remove all spatial frequencies above those visible to the world observer in the image, which is presented to the display observer.

There is a second model which implements the visual acuity, and which appears in [Wa+97]. This model implements a variable resolution filter using an image pyramid and interpolation, similar to the *mip-map* technique introduced by Williams [Wi83]. But the computational cost of this model is very high.

5.3.4 Colour Sensitivity

To simulate the loss of colour information in dark environments, we use the model presented by Ferwerda et al. [Fe+96].

In the scotopic level there is no colour vision and poor acuity, in this range only the rod receptors are active. The mesopic level is characterized by the fact that both the rod and cone receptors are active, and finally in the photopic level one has both good colour vision and acuity. The latter is due to the cone receptors, which are responsible for colour vision, being finally fully active.

With this model we modify only the luminance value for the rod receptors simulating this defect, and we are able to simulate the passage between the scotopic and the photopic through the mesopic level. The new value for the luminance in the scotopic level is obtained by

$$Y' = L_{rod} = Y \cdot \left[1.33 \cdot \left(1 + \frac{Y+Z}{X} \right) - 1.68 \right]. \quad 5.32$$

Where X , Y and Z are the *CIE* tristimulus values. The value of the luminance Y is the luminance value for the cone photo receptors.

Chapter 6

Real Time Tone Mapping Model

Introduction

Many TM operators have been proposed in the last decade. An important subclass of these are global TM operators. They consider each pixel in an image separately, and are therefore well suited for acceleration. Although these operators do not take into account the neighbourhood of a pixel (like local TM operators), they can still simulate many important effects related to the HVS.

Some global operators evaluate just a simple function, but many of them perform complex computations for each image pixel [TuRu93][Wa+97][Pa+00][Dr+03][Fe+96][Ma+97]. While convincing results can be obtained, the high execution times of the complex operators prevent their usage in interactive applications.

One possibility of delivering interactivity is implementing a TM operator directly in graphics hardware with support for floating point colour representation and programmable pixel shaders. Due to the hardware constraints, this approach however imposes severe limitations on the operator: (1) it cannot perform arbitrary data manipulation; (2) its complexity is limited by the maximum number of instructions of a pixel shader.

We propose a novel acceleration framework that delivers interactivity to complex global TM operators. Interactivity brings the user a qualitatively different understanding of the behaviour of different operators as (he) she can study their response by interactive manipulation of the image or the parameters of the operator. The framework can be implemented either as a pure software acceleration technique, or it can perform some algorithmically simple but

computationally costly operations in the graphics hardware. In the latter case, the proposed framework also provides the benefit of an efficient integration of the method into the flow of the graphics hardware-rendering pipeline and permits to use complex global TM operators in real-time rendering applications.

This Chapter is organized as follows: Section 6.1 provides an overview of the proposed method. Section 6.2 discusses the different steps of the proposed framework. Section 6.3 discusses an implementation of the method on the graphics hardware.

6.1 Overview

The proposed framework consists of four steps: *sampling*, *tone-mapping*, *fitting* and *reconstruction*. As an input we take a HDR image. The *sampling* algorithm produces a set of samples that form a compact representation of the luminance distribution in the image. The samples are passed to the TM operator that assigns each sample a luminance in the colour space of the display device (*tone-mapping* step). The *fitting* algorithm that finds interpolation coefficients for the point-sampled *tone-mapping curve* (*tone reproduction curve* TRC) processes the result. Finally the *reconstruction* algorithm applies the interpolated TRC on all pixels of the input image [Ar+03]. The framework is depicted in Figure 6.1:

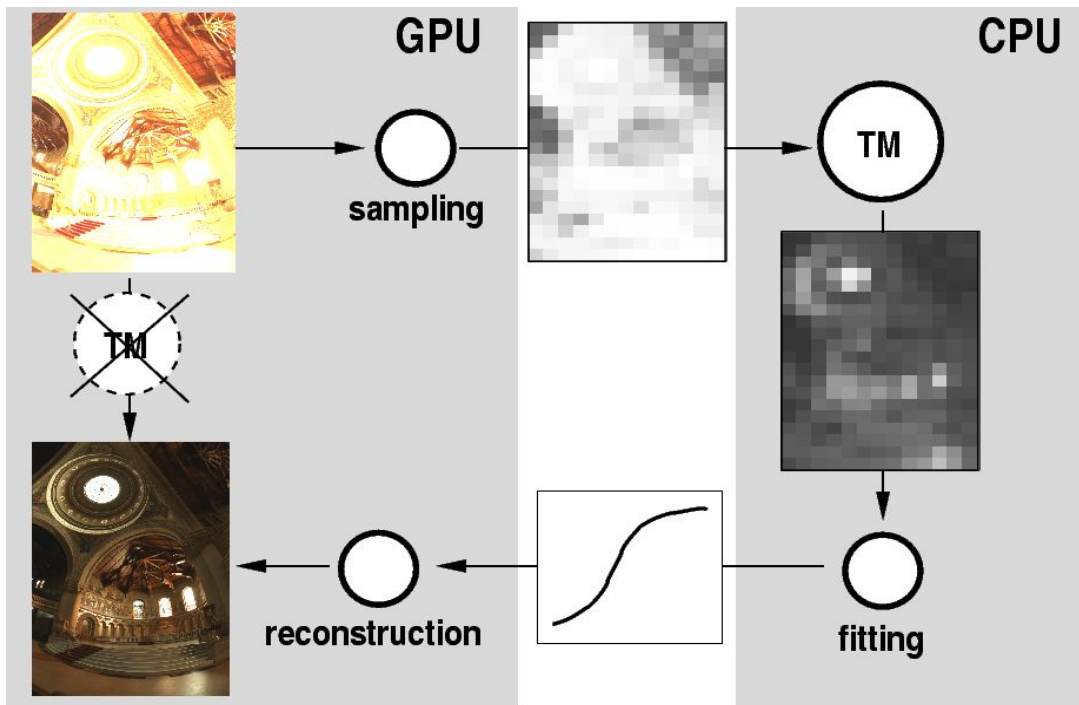


Figure 6.1: Overview of the proposed framework. The sampling and the reconstruction are suitable for an implementation on the GPU, the TM and the fitting are performed on the CPU [Ar+03].

Figure 6.1 shows a possible subdivision of the computation between the graphics hardware and the general-purpose processor. Sampling and reconstruction are relatively simple algorithms, but they are applied on all pixels of the input image. Tone mapping and fitting are more complicated algorithms, but they are applied on a relatively small number of samples. Thus using the suggested subdivision of the computation we exploit the raw computational power of graphics hardware while keeping the framework open for arbitrarily complex global TM operator.

Transferring only the sampled image and the coefficients established by the fitting algorithm minimizes the overhead of transferring data between the CPU and GPU. The main data flow takes place on the GPU [Ar+03].

6.2 Framework

In this section we discuss the four main steps of our framework with more details.

6.2.1 Sampling

The goal of the sampling algorithm is to compute a compact representation of the luminance distribution in the input image. The set of samples should provide an accurate representation of the histogram of the input image, while keeping the number of samples small. To avoid maintaining a HDR histogram, we sample directly in the image domain. We have used two techniques that can be seen as representatives of two extremes random sampling and down sampling, and one technique that combines the advantages of both filtered random sampling.

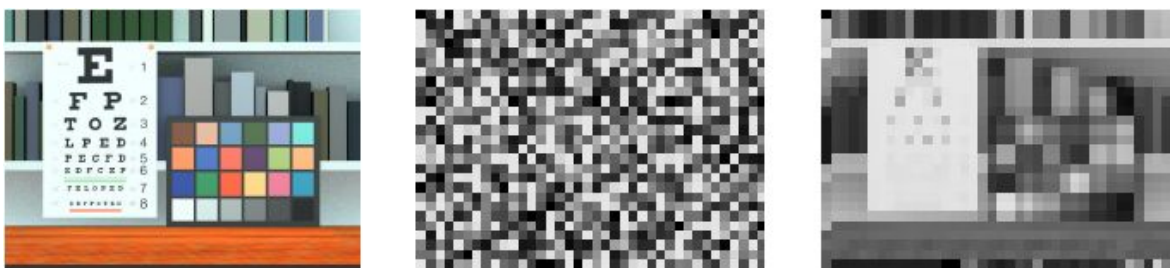


Figure 6.2: The input image and the luminance samples. (left) The input image (1000x700 pixels). (centre) Random sampling (1024 samples). (right) Down sampling (1024 samples), [Ar+03].

Random Sampling

Random sampling takes a specified number of samples from the input image. The samples are taken at image coordinates given by the Halton sequence.

The Halton sequence can be used like a random number generator to produce points in the interval $[0,1]$. The coordinates pixel (x,y) are obtained just multiply the Halton values, in 2D space, for the dimensions of the input image, as showed in equation 6.1:

$$\begin{aligned} x &= H_{1dimension} \cdot X_{size}, \\ y &= H_{2dimension} \cdot Y_{size} \end{aligned} \quad 6.1$$

where H represent the Halton sequence for two-dimensional space, and X_{size} is the height of the input image, instead Y_{size} is the width of the input image.

The Halton sequence is even less random than a random number generator. In contrast, the Halton sequence does a better job of filling in the region, avoiding the large gaps that can occur with a random number generator [Ha03].

Down Sampling

Down sampling subdivides the image into n regions and computes an average luminance for each region, which corresponds to the application of a box filter on the samples. The number n describes the resolution of the final image. For example, if the dimension of the final image must be 1000 pixels we subdivide the input image in 1000 equal regions and in each region is computed an average luminance value that represents a pixel of the final image.

A similar technique was used by Scheel et al. [Sc+00] to compute a global adaptation luminance. See Figure 6.2 for an illustration of the resulting images using these two sampling techniques.

Filtered Random Sampling

Filtered random sampling combines random sampling with down sampling in order to improve the ability of the algorithm to capture both high frequency as well as low-frequency information. It consists to apply, on every pixel obtained by random sampling technique (see section *Random sampling*), a box filter, of defined size, that computes an average luminance value of the luminance pixel contained inside in the box filter. An example of the final image obtained with this sampling technique is shown in Figure 6.3.



Figure 6.3: The input image and the luminance samples. (left) The input image (1000x700 pixels). (right) Filtered Random sampling (1024samples).

We performed several experiments in order to define the most suitable size of the box filter usable by our framework. These experiments are discussed in Chapter 8.

6.2.2 Tone Mapping

Once a small set of samples is obtained with the sampling phase, applied on the input image, it is sent to the TM operator in order to be mapped on the low dynamic range of the display device. The TM operator is only applied on this restricted set of samples, of the input image, in order to pass this information to the fitting algorithm (section 6.2.3) that captures the behaviour of the tone reproduction curve of the TM operator. We assume that the operator works only with the luminance component Y (or real world luminance L_{rw}) of *CIE XYZ* colour space. The TM operator maps each sample in the low dynamic range luminance in the restricted device range. In other words, the mapping represents a 1D point-sampled TRC as defined in equation 6.2.

$$TRC(L_{rw_i}) = L_{d_i} . \quad 6.2$$

Where L_{rw_i} represent the real world luminance value of the i -th pixel of the input image contained in the small set of samples obtained with the sampling phase. Instead, L_{d_i} represents the corresponding luminance value mapped in the restricted luminance range of the display device.

In the case a colour transformation is required; it can be applied on the whole image during the reconstruction phase, Section 6.2.4.

6.2.3 Fitting

The fitting algorithm aims is to capture the behaviour of the TM curve, or TRC using the information received by the TM phase. The fitting phase has the following features:

- Accurately capture the point-sampled TRC.

- Keep computational costs low.

To reproduce accurately the behaviour of the TM operator, minimize the error introduced by the reduce information obtained from the sampling phase, we need to respect the first features.

However, we need to be as fast as possible in reducing the final computation time of the framework. To preserve these two characteristics we tried two different kinds of interpolation models: piecewise linear interpolation and high order interpolation using natural cubic spline. Note that before applying the interpolation algorithm, we first sort the samples according to the real world luminance L_{rw} .

In the case of piecewise linear interpolation, we need just to compute the interpolation coefficients k and d between the boundaries real world luminance samples pixel L_{rw_i} and $L_{rw_{i+1}}$ in the generic i -th interval. Linear interpolation in that interval gives the following interpolation equation:

$$L_d = k \cdot L_{rw} + d, \quad 6.3$$

and the coefficients k and d are:

$$k = \frac{L_{d_{i+1}} - L_{d_i}}{L_{rw_{i+1}} - L_{rw_i}}; d = L_{d_i} - k \cdot L_{rw_i}, \quad 6.4$$

where L_{d_i} and $L_{d_{i+1}}$ are the boundaries luminance values mapped in the low dynamic range of the display for the i -th interval.

In the case of natural cubic spline more coefficients, have to be determined. Indeed, if the natural cubic spline is as equation 6.5, we have to compute four coefficients [Pr+92]:

$$L_d = a \cdot L_{d_i} + b \cdot L_{d_{i+1}} + h \cdot L'_{d_i} + l \cdot L''_{d_{i+1}}, \quad 6.5$$

where L'_{d_i} and $L''_{d_{i+1}}$ are the first and the second derivatives of the boundaries luminance values L_{d_i} and $L_{d_{i+1}}$ for the i -th interval, respectively. These coefficients are computed as follows

(see [Pr+92]):

$$a = \frac{L_{rw_{i+1}} - L_{rw_i}}{L_{rw_{i+1}} - L_{rw_i}}; b = 1 - a. \quad 6.6$$

$$h = \frac{1}{6} \cdot (a^3 - a) \cdot (L_{rw_{i+1}} - L_{rw_i})^2; l = \frac{1}{6} \cdot (b^3 - b) \cdot (L_{rw_{i+1}} - L_{rw_i})^2 \quad 6.7$$

Natural cubic spline slightly increases the accuracy but it does not compensate the higher computational costs, that are approximately 2 times the computational costs to perform the piece-wise linear interpolation.

6.2.4 Reconstruction

The reconstruction of the TM curve proceeds as follows: given the HDR luminance L_{rw} , using a binary search, we locate an interval i of the interpolated curve that covers L_{rw} . For the linear interpolation the device luminance L_d is extrapolated as:

$$L_d = k_i \cdot L_{rw} + d_i; \quad L_{rw_i} \leq L_{rw} \leq L_{rw_{i+1}}. \quad 6.8$$

For the natural cubic spline, L_d is expressed as:

$$L_d = a_i \cdot L_{d_i} + b_i \cdot L_{d_{i+1}} + h_i \cdot L_{d_i}' + l_i \cdot L_{d_{i+1}}''. \quad 6.9$$

Finally, the colour of each pixel of the output image is computed by multiplying the colour

components of the corresponding input pixel by the ratio $\frac{L_d}{L_w}$. As discussed in Ashikhmin's (see work [As02]), this is the simplest way to deal with colour which provides acceptable results.

Look-up-table

To accelerate the reconstruction algorithm, we can re-sample the interpolation coefficients and store them in a lookup table. Given a world luminance, the lookup table can be directly accessed to obtain the corresponding device luminance value. To improve the accuracy of the reconstruction we can use a logarithmic scale to capture the low dynamic part of the TM curve more accurately. Indeed, the logarithmic scale has the property to change the distribution of the samples of the TM curve. This means that for the low dynamic range of the TM curve the interval between two samples is decreased. This gives more information. By contrast for the high dynamic range of the TM curve, the interval between two samples, is increased.

A similar technique was used by Durand and Dorsey [DuDo00] and Scheel et al.[Sc+00]. Indeed, they computed the down-samples pixel using directly the TM operator whereas our method re-samples the interpolated TM curve. The advantage of our method is that we can treat the TM operator as a black box without altering its TM curve.

This is more general, since we do not have to know which parameters the TM operator uses.

6.3 Hardware

We integrated the proposed framework to the rendering pipeline of graphics hardware with the support for floating point per-pixel operations. In particular we used the NVIDIA GeForceFX card, which supports 32 bits per colour component and pixel shaders (also called fragment programs) of up to 1024 instructions [Nv03].

Following Figure 6.1, we ported two steps of the framework to the graphics hardware: the sampling and the reconstruction. The corresponding algorithms are implemented as fragment programs written in NVIDIA's Cg language [Nv03]. In this section we discuss specific issues associated with the hardware implementation.

6.3.1 Sampling

The sampling algorithms described in this section, are the hardware implementation of the algorithms described in Section 6.2.1. In this section we will give just a description of their hardware implementation on the GPU.

Random Sampling

Random sampling is implemented by dependent texture lookups. We generate a small texture containing values of the Halton sequence. The number of the desired samples gives the size of the texture. This texture is rendered into the frame buffer as follows: for each pixel we use the corresponding texel as indices to the input image (treated as a secondary texture) and the pixel is set to the colour of the addressed pixel.

Down Sampling

Down sampling is implemented by subsequent rendering of textured quads of smaller resolution. Initially we use the input image as a texture and render it on a quad with $1/k$ of the resolution of the input image. The fragment program computes an average of k^2 texels for each pixel covered by the quad. If the input image consists of p pixels after n steps we obtain

$\frac{p}{k^{2n}}$ samples assuming the dimensions of the image are a power of k .

Filtered Random Sampling

Filtered random sampling is a simple extension of random sampling where a predefined image region around the input image pixel is averaged in the fragment program to obtain the sample value.

6.3.2 Tone mapping and Fitting

To maintain the flexibility of the original TM operator and to overcome the hardware constraints of the graphics cards, the Tone mapping and the Fitting steps are implemented on the CPU, as shown in the Framework structure of Figure 6.1.

After the sampling step, about 1000 samples are transferred from the GPU to the CPU using a frame buffer read-back. Apparently, this operation seems to be the bottleneck of the Framework. However we transfer only a reduced number of samples (1000) in respect to the original dimension of the input image. Moreover this operation is not time consuming, as shown in Section 8.2.

As described in Section 6.2.2, the Tone mapping step consists to apply one of the pre-existing global TM operators. The TM operator is applied on the samples and the fitting is used to find the coefficients of the interpolated TRC as described in Section 6.2.3. For the motivations already explained in Section 6.2.3, we used the piecewise linear interpolation instead of cubic spline.

As described in Section 6.2.3 the fitting algorithm is used to interpolate the point-sampled TRC in order to tone map all pixels of the input image. The coefficients obtained from the fitting algorithm are sent to the graphics card as a texture. Each texel of the texture represents coefficients of the interpolation for one interval between the samples. The coefficients are encoded as colour components of the texel.

6.3.3 Reconstruction

Given the real world luminance of the input image L_{rw} , the reconstruction phase can be divided in two tasks:

- 1) Find the i -th interval that covers the luminance value L_{rw} . In this interval the correspondent coefficients, determinate with the fitting algorithm, are stored.
- 2) Applying the interpolation algorithm to the input pixel using these coefficients.

To perform the first task we tried to use the binary search as used in the pure CPU implementation. For the second task, as described in the previous section, we used the piecewise linear interpolation instead of cubic spline.

Preliminary tests showed the high computation costs of the binary search, even if it is implemented on the GPU. In order to avoid it a look-up table can be created. This works as explained in Section 6.2.4. In general, from these preliminary tests, we can see that the binary

search implemented on the GPU is three times slower than the lookup table implemented also on the GPU. More information about the performances, both in terms of time and quality, are presented in Section 8.2.

A description of the implementation on the GPU of these two techniques: binary search and look-up table is given in the next two subsections.

Binary Search

To perform a binary search we extend the texture of coefficients with a row containing the world luminance values of the samples. The binary search algorithm locates the appropriate coefficients of the interpolated TRC using this row.

Lookup Table

The lookup table for reconstruction is constructed by *rendering* the coefficients established by the fitting algorithm into the frame buffer. The coefficients of an interval are rendered as a narrow horizontal textured quad. The vertical limits of the quad are determined by the world luminance of the sample points that bound the associated interval. The texture of the quad consists of the interval coefficients. Each column of the frame buffer represents then a fixed size interval of world luminance. Given a luminance value, the reconstruction algorithm can then directly access the lookup table without the necessity of performing a binary search.

6.3.4 Discussion

As described before the framework consists of four basic steps, and a smart working subdivision between CPU and GPU is presented. This subdivision has been designed to provide a general solution; for instance it allows to any pre-existing global TM operator to be used as a black box without modifying its original implementation. The generality, of our approach, also concerns the fact that the rendering pipeline does not need any changes and the framework can be easily integrated.

The two steps that are computationally critical are the sampling and the reconstruction steps, since they have to be applied on the whole input image. For instance, the idea to implement these two steps on the GPU was directed to reduce the computation time and to maintain the generality of the framework at the same time. The latter is due to the fact that the TM operator is implemented on the CPU and it can be substituted by any another existing operator, without changing the framework and the rendering pipeline. The sampling step should be critical if an unstable sampling technique is adopted. Thus, the information needed to capture the behaviour of the TM operator is not properly extracted.

The framework can be affected by a flicker noise present in the image sequence (flickering problem). This happens for two reasons: first, the sampling on the input image can not capture the exact image parameters like maximum and minimum luminance used by the TM operator. Second, also a direct implementation of a TM operator on the GPU, is affected by the flickering. This is also due to the fact that an interactive application can often suffer from large temporal discontinuities in dynamic range.

Smoothing those discontinuities can be done in two ways: reusing the pixel information, captured by sampling, in the last ten frames, or incorporating a model of time-dependent adaptation.

The bottleneck of this framework could appear the transfer of the data between CPU and GPU and vice versa. In practice this does not constitute any limitation since the transferred data are limited to a small number of samples obtained by sampling. In the performed experiments (see Chapter 8) usually the number of samples does not exceed 1000. The tone-mapping step is also not time critical, since it is performed on the samples obtained with the sampling step. For instance this step is left on the CPU.

Chapter 7

Tone Mapping for Printers

Introduction

The printer device is often used for printing HDR scenes, and like the display, presents some problems when a HDR scene is printed out. In fact the printer, as the display, reproduces the scene in a low dynamic range, also lower than the dynamic range of the display. The consequence of it is that the characteristics of the original scene such as contrast, brightness, details etc. are lost, and the final output scene is not comparable with the original HDR scene. The problem in the case of the printer is more complex than for the display device. Indeed a complex characterization model must be included that performs the colour reproduction on the device. In this chapter a simple framework, that defines the steps that are necessary to print a HDR scene, are presented (Section 7.1). In Section 7.2, a novel colorimetric characterization method is presented. Finally in Section 7.3 a discussion on the presented method is presented.

7.1 Framework

Uroz and Marimon [UrMa97] presented a flowchart (see Figure 7.1) for the reproduction of paintings on a large format ink-jet printer. The objective is to obtain an equivalent match between the original and the copy. Several steps constitute it: *image capture*, *scanning*, *image correction and printing*. This framework can be extended to be used in the context of HDR images.

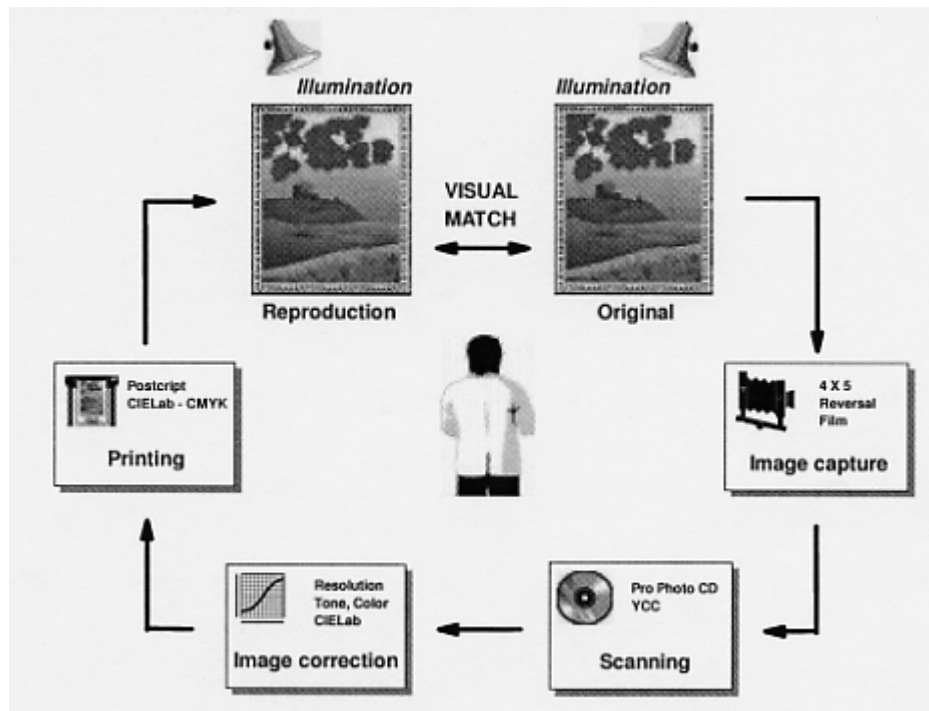


Figure 7.1: Framework for the reproduction of paintings using large format inkjet printer. The objective is to obtain an equivalent match between the original and the copy, after [UrMa97].

Indeed one of the several methods proposed in Computer Graphics, that capture HDR images [DeMa97] can be used, instead the scanning step. Once the image has been captured, it can be direct visualized on a display system or printed on a printer. For this purpose, new methods have to be proposed in order to work with this high dynamic range and reproduce on a small dynamic range the characteristics of the original scene. In order to capture this goal, it is not only necessary to compress the HDR scene in a small dynamic range (printer), but also to have a colorimetric characterization model able to achieve the colour reproduction on the printer device.

Several solutions, proposing different techniques, have been presented [Ar97][ArWi01][Be96][ArWi03][KaAn92][Ka+95][MaAb94][Xi+99][RoBa93].

Two basic steps which are part of the framework in Figure 7.1 are presented in this section: *tone reproduction* and *colorimetric characterization of a colour printer*. The tone reproduction problem, or tone mapping has already been introduced in Chapter 4.

Many TM operators have been presented in the last ten years [TuRu93][Wa+97][Pa+00][Dr+03][Fe+96][Ma+97][Pa+00][Pa+98][Sc94][As02][Re+02][Fa+02][DuDo00][DuDo02][TuTu99][Tu+99][Wa94]. They address the problem basically on a display device; but on the other hand the mapping problem has been formulated in a general way, in the sense that the low dynamic range used in order to map the original HDR scene has been defined without

restrictions. It means that they are also suitable to be used on a printer device, but in this case the colorimetric characterization problem is more complex than for the display device. The basic reason of it is that there is no a priori knowledge about the shape of the colorimetric characterization function of the printer. In the next section a novel model, that addresses how to resolve this problem, is presented.

7.2 Colorimetric Characterization of a Colour Printer

In multimedia systems, different color reproduction devices while serving the same purpose exhibit large discrepancies in their raw output. This is due to the fact that they usually employ different color mixing technologies (additive or subtractive), use different input color spaces and hence have different gamuts, and that their device characteristics can change with time and usage. These facts usually do not permit a faithful matching of colors between devices if no precautions are taken.

Colorimetric characterization is one step in the colorimetric reproduction process that permits faithful image reproduction across different display devices. Its goal is to define a mapping function between the device dependent color spaces in question (such as RGB or CMYK) and device independent color spaces (such as CIELAB or CIEXYZ), and vice versa. There are three main approaches to defining this mapping function: *physical models*, *empirical models* and *exhaustive measurements* [Fa98]. Physical modeling of imaging devices involves building mathematical models that find a relationship between the colorimetric coordinates of the input (or output) image element and the signals used to drive an output device (or the signals originating from an input device). The advantage of these approaches is that they are robust, typically require few colorimetric measurements in order to characterize the device, and allow for easy recharacterization if some component of the imaging system is modified. The disadvantage is that the models are often quite complex to derive and can be complicated to implement. Physical models are often used for the colorimetric characterization of displays and scanners.

Empirical modeling of imaging devices involves collecting a fairly large set of data and then statistically fitting a relationship between device coordinates and colorimetric coordinates. Empirical models are often higher order multidimensional polynomials, or neural network models. They require fewer measurements than exhaustive techniques, but they need more than physical models. Empirical models are often used for scanners and printers.

Often the colorimetric characterization of printers requires an exhaustive measurement in order to obtain good performances. Typically 9x9x9 samples of the device drive signals are sampled and colorimetrically measured. Many more measurements have to be used for devices with poor repeatability.

Lookup tables can be used to process image data via multidimensional interpolation. This technique has different disadvantages: the large number of measurements that has to be made, difficulties in interpolating the highly nonlinear data and difficult recharacterization if any aspect of the device changes. The advantage of exhaustive measurement techniques is that they require no knowledge of the device physics.

There is a large number of publications on the colorimetric characterization of printers that propose different models for solving this problem: Kang and Anderson [KaAn92] propose the application of neural networks and polynomial regression. Albanese et al. [Al+95] and Tominaga [To96] have used feed forward neural networks trained by back propagation and obtained promising results. However, their approach also has some disadvantages: the need for a big training set (several hundred to several thousand samples), high computational cost, and a comparatively large maximum color error for high quality color reproductions. One of these problems has been solved by Artusi et al. [Ar+98]: in their work they reduced the size of the training set to 216 measured samples, while retaining a maximum error that is comparable to --- in some cases even better than --- previous approaches. There are no references to be found in the literature about the use of radial basis function networks for the colorimetric characterization of printers, but there is a wealth of other publications about them and their applications (such as Orr [Or96], Bishop [Bi96], Carozza and Rampone [CaRa99] and Lee [Le99]).

In general a good algorithm for colorimetric characterization must have the following characteristics: small training set, fast response, good accuracy, and it must allow for a fast recharacterization. This thesis work present a modification of an existing learning algorithm [CaRa99] to train radial basis function networks to solve the problem discussed so far, namely the colorimetric characterization of printers. This learning algorithm has fast training and test phases, go novel colorimetrod accuracy, and it also requires a comparatively small training set.

The work we present is novel in seven ways: to begin with, this is the first work that uses radial basis function networks to resolve the colorimetric characterization of printers. Second, we used a new learning model to train such networks; our approach is based on a

modification of the proposal by Carozza and Rampone [CaRa99]. Third, we use only 125 measured samples for the training of the network. Fourth, the computational costs for this training are very low when compared to previous techniques and allow to use this model in consumer products. Fifth, it is a general model which one can also use to define other transformations between color spaces. Sixth, it is possible to have a fast recharacterization of the device because the computational cost of the training phase is low. Finally, it improves on the performance of multiple polynomial regression and tetrahedral interpolation [Ar Wi03]. In Subsection 7.2.1 a background for radial basis function networks (RBFN) is presented. In Subsection 7.2.2 our novel colorimetric characterization model for colour printers is presented.

7.2.1 Background

The colorimetric characterization problem can be seen as a regression problem, where it is necessary to resolve a linear problem of the form $Ax=b$. In other words we have available a set of input values A , a set of output values b and we need to find the correlation between the input and output x that defines the colorimetric characterization function.

There are two main types of regression problems in statistics [Or96]: parametric and non-parametric. In parametric regression the form of the functional relationship between the dependent and independent variables is known, but may contain parameters whose values are unknown, and it is possible to successfully estimate the desired result from the training set.

In the case of non-parametric regression there is no, or very little, a priori knowledge about the form of the true function which is being estimated. The colorimetric characterization problem, presented in this thesis work, is a non parametric regression problem, because one does not know the mapping function properties the algorithm will arrive at in advance. There are different approaches to resolve non parametric regression problems; when one uses equation systems in this context they may contain many free parameters that have no physical meaning in the problem domain (interpolation models, multiple polynomials regression), or one can instead use neural networks.

Neural Networks

The base of a neural network is a formal neuron. It is defined as a series of inputs, a series of outputs and by a function that maps specific inputs to series of outputs [Bi91]. Neural networks consist of collections of connected formal neurons. Each formal neuron computes a simple non-linear function F on the weighted sum of its input. The function F is referred to as

activation function and its output is defined as the activation of the formal neuron. Long-term knowledge is stored in the network in the form of interconnection weights that link such formal neurons.

There are different neural network structures [Ar97]: total connected networks, partial connected networks, multilayer networks (feed forward, feedback), and auto-associative networks.

In a neural network, the learning phase is a process where a set of weights is defined that produces a desired response as a reaction to certain input patterns [Bi91]. There are two main techniques for the learning phase: supervised learning and non supervised learning. In supervised learning the function is learned from samples, which a teacher supplies. This set of samples, referred to as the training set, contains elements which consist of paired values of the independent (input) variable and the dependent (output) variable [Or96]. In the case of non-supervised learning, it reaches an internal model that captures the regularity in the inputs without taking other information into account [Ar97].

Basis Functions

A linear model for a function $f(x)$ can be expressed in the following form [Or96]:

$$f(x) = \sum_{j=1}^m w_j h_j(x). \quad 7.1$$

The model f is expressed as a linear combination of a set of m fixed functions h_j , often referred to as basis functions. The flexibility of f , its ability to fit many different functions, derives only from the freedom to choose different values for the weights w_j . The basis functions and any parameters, which they might contain, are fixed. If the basis function parameters can also change during the learning process, the model is considered non-linear. Any set of functions can be used as a basis set, although it is desirable that they are differentiable. There are many different classes of functions that one can use as basis functions, for example:

- *Fourier series*

$$h_j(x) = \sin\left(\frac{2\pi j(x - \theta_j)}{m}\right). \quad 7.2$$

- *Logistic functions*

$$h(x) = \frac{1}{1 + \exp(b'x - b_0)}. \quad 7.3$$

- Polynomial functions

Radial functions are a special class of basis function. Their characteristic feature is that their response decreases (or increases) monotonically with the distance from a central point. The center c , the distance scale, and the precise shape of the radial function r are parameters of the model, and are fixed if it is a linear model. Two possible examples of radial functions are:

- *Gaussian*

$$h(x) = \exp\left(\frac{-(x-c)^2}{r^2}\right). \quad 7.4$$

- *Multiquadratic*

$$h(x) = \frac{\sqrt{r^2 + (x-c)^2}}{r}. \quad 7.5$$

Radial Basis Function Networks

Radial Basis Function Networks (RBFN) are derived from the exact interpolation problem [Bi96] by the introduction of several changes. The exact interpolation problem attempts to map every input point exactly onto a corresponding target point. The Radial Basis Function (RBF) approach introduces a set of basis functions equal to the number of input points. In contrast, the following modifications are necessary for the introduction of RBFN:

- The number of basis functions does not have to be the same as the number of input points, and is typically smaller.
- The bias parameters are included in the sum term of the linear model from equation 7.1.

In the case of a non-linear model there are two more modifications if the basis function can move, change size, or if there is more than one hidden layer:

- There is no constraint that the centers of basis functions have to be input points; instead, determining these centers is part of the training process.
- Instead of a unique parameter r , every basis function has a parameter r_j , the value of which is obtained during the training process.

An example of a traditional RBFN with one hidden layer is shown in Figure 7.2. Each of n components of the input vector x feeds forward to m basis functions whose outputs are linearly combined with weights w_j into the network output. This example could be a linear model of RBFN if the parameters of the basis function H_j , in the hidden layer, do not change during the learning process. Instead if they change during the learning process the RBFN is

non-linear. Also, if there is more than one hidden layer of basis functions H_j in the structure of the RBFN, the network is a non-linear model.

There are two stages for the training phase: determining the basis function parameters, and the finding of appropriate weights.

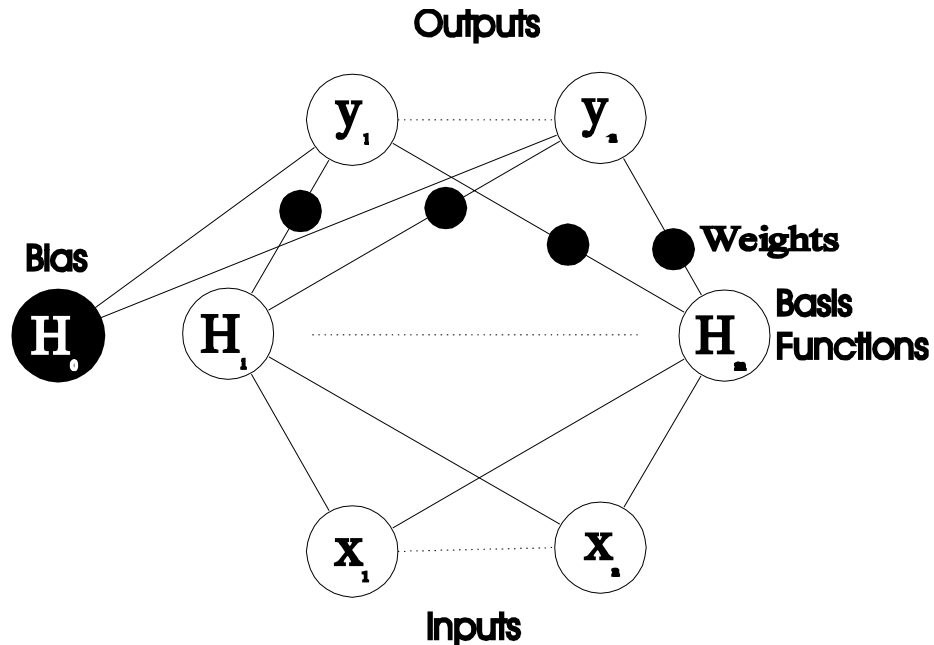


Figure 7.2: Radial Basis Function Network RBFN, after [ArWi01].

Linear Network models

In the case of a linear model, the parameters of the basis functions are fixed, and the goal is to minimize the sum of the squared errors in order to obtain the optimal weights vector [Or96]:

$$S = \sum_{i=1}^p (y_i - f(x_i))^2, \quad 7.6$$

where p is the number of pattern, and (x_i, y_i) are the input and output vector targets of the respective training set. The optimal weights, in matrix notation, are:

$$W = A^{-1} H^T Y, \quad 7.7$$

where H is referred to as design matrix and is the output of the RBF, A^{-1} is the covariance matrix of the weights W , and the matrix Y is the output target.

In many cases this amounts to an over fitting problem and the main effect of this is that the neural network loses its generalization capacity. In order to counter the effects of over-fitting it is possible to utilize results from *regularization theory*. Regularization theory suggests

attaching a term called *regularization parameter* k in equation 7.6, in order to obtain a weight vector, which is more robust against noise in the training set.

In regularization theory, there are two main techniques: global ridge regression, where one uses unique regularization parameters k for all basis functions, and local ridge regression, where there is a regularization parameter k_j for every basis function j . For the case of global ridge regression one has to modify equation 7.6 as follows:

$$C = \sum_{i=1}^p (y_i - f(x_i))^2 + k \sum_{j=1}^m w_j^2, \quad 7.8$$

where m is the number of the basis function. In the case of local ridge regression equation 7.6 has to be modified to:

$$C = \sum_{i=1}^p (y_i - f(x_i))^2 + \sum_{j=1}^m k_j w_j^2. \quad 7.9$$

Forward Selection

One way to give linear models the flexibility of non-linear models is to go through a process of selecting a subset of basis functions from a larger set of candidates [Or96]. In linear regression theory [Ra88] subset selection is well known and one popular version is forward selection in which the model starts empty ($m=0$) and the basis functions are selected one at a time and added to the network. The basis function to add is the one which most reduces the sum squared errors in equation 7.6; this is repeated until no further improvements are made. There are different criterions to decide when to stop the forward selection process: generalised cross-validation (GCV) [Go+79], unbiased estimate of variance (UEV) [EfTi93], final predictor error (FPE) [Ma73] and the Bayesian information criterion (BIC) [Sc78]. An efficient method of performing forward selection is the orthogonal least squares method as discussed in Orr [Or95]; it is based on the orthogonalisation of the columns of the design matrix. This involves a particular form of the covariance matrix, which consists of a triangular and a diagonal matrix; this fact can be used to greatly accelerate the computation.

Non-Linear Network models

In the non linear model the basis function parameters are not fixed, and it is possible to estimate them during the learning process. This gives more flexibility to the network model. In literature there is a large number of publications that propose different models to estimate these basis function parameters. In this section we present two existing models, also

implemented in this thesis work [CaRa99] [Le+99]. In particular we present in detail the model of Carozza [CaRa99], in order to understand the differences with the model presented in this paper.

In Carozza [CaRa99] a new algorithm for function approximation from noisy data was presented. The authors proposed an incremental supervised learning algorithm for RBFN. It added a new node at every step of the learning process, and the basis function center c and the output connection weights are settled in accordance with an extended chained version of the Nadaraja--Watson estimator. The output network for the neuron m , in accordance with the Nadaraja--Watson estimator, is

$$net_m(x) = \frac{\sum_{j=1}^m w_j \exp\left(-\frac{\|x - x_{i_j}\|^2}{2r_j^2}\right)}{\exp\left(-\frac{\|x - x_{i_j}\|^2}{2r_j^2}\right)}, \quad 7.10$$

where the index i indicates the number of patterns in the Training set ($i=1 \dots p$). The goal is to minimize the empirical risk

$$E_m = 1/p \sum_{i=1}^p (y_i - net_m(x_i))^2. \quad 7.11$$

The chained version of the estimator 7.10 obtained for an incremental approach is

$$net_m(x) = \frac{net_{m-1}(x)den_{m-1}(x) + w_m \exp\left(-\frac{\|x - x_{i_m}\|^2}{2r_m^2}\right)}{den_m(x)}, \quad 7.12$$

where

$$den_m(x) = \sum_{j=1}^m \exp\left(-\frac{\|x - x_{i_j}\|^2}{2r_j^2}\right). \quad 7.13$$

In order to reduce the empirical risk given by 7.11, the output weights for the neuron j are chosen as

$$w_j = y_{i_j} - net_{j-1}(x_{i_j}), \quad 7.14$$

where the output y_j is such that

$$(y_{i_j} - net_{j-1}(x_{i_j}))^2 = \max_{i=1,\dots,p} (y_i - net_{j-1}(x_i))^2. \quad 7.15$$

Moreover, since this quantity is affected by normalization factor den_j , the authors multiply w_j for this factor, and arrive at the final chained version of 7.10

$$net_m(x) = \frac{net_{m-1}(x)den_{m-1}(x)}{den_m(x)} + w_m \exp\left(-\frac{\|x - x_{i_m}\|^2}{2r_m^2}\right), \quad 7.16$$

The variance r of the basis functions is determined by an empirical risk driven rule based on a genetic like optimization technique. Applying the mutation generates a population of individuals

$$r_{j,new}^2 = r_j^2(1 + alpha), \quad 7.17$$

where $alpha$ is a random number in a fixed range $[-a, a]$. This mutation step is iterated a fixed number of times if the empirical risk 7.11 associated with $r_{j,new}^2$ is less than the same quantity associated with r_j^2 .

On a different note, Lee et al. [Le+99] introduces the concept of *robust* RBFs and makes suggestions on how to choose a function candidate, which fulfills this role.

7.2.2 Proposed Model

Modified Estimator for RBFN Weights

The proposed model is a modification of an existing one [CaRa99]. In particular we have modified the estimation of the weights by introducing a pseudo inverse matrix [Bi96] instead of using the extended chained version of the Nadaraja--Watson estimator for updating the weights. The pseudo inverse method works by resolving the following general system of linear equations:

$$HW = Y, \quad 7.18$$

where H is the matrix of the basis functions h design matrix of dimension (*number of input vectors* \times *number RBF*), Y is the matrix of output vectors of dimension (*number of output vectors* \times 3), and W is the weights matrix of dimension (*number of RBF* \times 3). The number 3 indicates the three dimensional space of the input and output vectors. In this equation there

appears a Moore--Penrose pseudo inverse [Ma00] in the form of the matrix B , which has the same dimensions as H^t , and that has to satisfy the following four conditions:

$$HBH = H$$

$$BHB = B$$

HB is Hermitian

BH is Hermitian

7.19

The solution of the linear system 7.18, through the pseudo inverse matrix, is the matrix W 7.20:

$$W = H^t (HH^t)^{-1} Y . \quad 7.20$$

There are different methods to resolve the linear system 7.18: Gauss-Jordan elimination, Gaussian elimination with Back-substitution, LU decomposition, and Singular value decomposition (SVD) etc. We used the SVD method because in comparison with the other methods it gives satisfactory results and is the fastest algorithm [Pe+92]. Let take a general matrix X . This method is based on the following theorem of linear algebra: any $M \times N$ matrix X whose number of rows M is greater than or equal its number of columns N , can be written as the product of an $M \times N$ column--orthogonal matrix U , an $N \times N$ diagonal matrix Q with positive or zero elements (the singular values), and the transpose of an $N \times N$ orthogonal matrix V .

In other words the matrix X can be written as:

$$X = UQV^t . \quad 7.21$$

We have three different cases, about the form of the matrix X :

- 1) In the case the matrix X is square, this means that $N=M$, then U , V , and W are all square matrices of the same size. Their inverses are also trivial to compute, in fact U and V are orthogonal, and so their inverses are equal to their transposes. Instead Q is diagonal, so its inverse is the diagonal matrix whose elements are the reciprocals of the element q_j . In this way the inverse of the matrix X is:

$$X^{-1} = V \left[\text{diag} \left(\frac{1}{q_j} \right) \right] U^t , \quad 7.22$$

in our case $X = HH^t$ and the value of the unknowns matrix W is:

$$W = H^t V \left[\text{diag} \left(\frac{1}{q_j} \right) \right] U^t Y . \quad 7.23$$

If the value of the term q_j is equal zero, we need to replace the term $1/q_j$ with zero, do not have a division by zero.

- 2) If there are fewer linear equations M than unknowns N , then you are not expecting a unique solution. Usually there will be an $N-M$ dimensional family of solutions. In this case is possible to augment your left-hand side matrix with rows of zeros underneath its M nonzero rows, until it is filled up to be square, of size $N \times N$. In this way the matrix becomes to be square and we can apply the SVD in the way explained for the square matrix, equation 7.23.
- 3) If there are more equations than unknowns, we are in the case of over-determinate set of linear equations, and the equation for the square case, equation 7.23, can be apply without modification.

The output of the network trained with our model is not calculated as in the model proposed by Carozza [CaRa99], but rather by a sum of products of basis function output weights. The RBFN model used in our experiments adopts only one hidden layer, and three dimensional Gaussian basis functions for the nodes in the hidden layer with the same variance r . In our model there are some parameters: the average c and the variances r for the basis function, the weights W , *times* as the number of the training epochs and E as initial error (empirical risk). Other initializations depend on the application at hand. After preliminary experiments, we chose the following values to be suitable for our particular case: $c = \text{random}(0,1)$, $r = 0.5$, $W = \text{random}(0,1)$, *times* = 10. The initial error E has to be large in order for error reduction to work (100 times more than the acceptable error), and the node numbers of the hidden layer N begins from 1.

Our model

In this subsection we present our algorithm in pseudo code format, followed by a brief explanation.

Parameter initialization

while (termination criterion is not met)

if ($N > 1$)

$r_N = 0.5$;

Compute the average error E of the output model with respect to the output target

/ Update of basis function parameter c */*

$c =$ the input vector with index of the patterns with maximum error

Compute the weights W with the pseudo inverse

end if

/ Update of basis function parameter r */*

for ($j = N$ to 1)

while ($l \leq \text{times}$)

$\alpha = \text{random}(-0.5, 0.5)$

$r_{\text{new}} = r_j(1 + \alpha) + \text{epsilon}$

Compute average error E_{new} of the output model with respect to the output target

if ($E_{\text{new}} < E$)

$r_j = r_{\text{new}}$

$E = E_{\text{new}}$

end if

$l = l + 1$

end while

end for

*Compute the weights W with the pseudo inverse and use the basis function
parameter update*

$N = N + 1$

end while.

The parameter initialization defines the initial values for the parameters of the radial basis functions: average c and variance r , and other network parameters such as: the weights W , the

number of epochs *times*, the initial value of the hidden number *N* and the error *E*. In order to obtain it we follow the initialization suggested above.

This model has the capacity to generate the structure of the RBFN. In fact, it adds a new node *j* during each iteration and initializes its parameters, the three dimensional vector c_j , r_j and the three dimensional vector w_j . This operation stops when the termination criterion is satisfied.

In the case of the first node, $N = I$, we only need to perform the initialization of the parameters r_j and w_j because the average c_j is chosen randomly. In the other cases the average c_j is the input vector with index of the output model with the maximum error with respect to the output target.

The updating of the weights parameters is performed with the pseudo inverse technique 7.23; the variance parameter r_j of the radial basis functions is updated following the technique proposed by Carrozza [CaRa99].

This technique consists of using the following equation for each node *j*:

$$r_{new} = r_j(1 + alpha) + epsilon, \tag{7.24}$$

where the terms *alpha* and *epsilon*, after preliminary experiments, were set to the following initial values that are suitable for our application: for alpha a random value in the range from -0.5 to 0.5, and epsilon was set to 0.01.

We update the value r_j for the node *j* with the new value obtained with equation 7.24 only if the new error value E_{new} , computed with the new parameters r , c and W , is less than the old error value. We repeat the updating operation for the parameter r_j up to the maximum epochs condition is respected $l \leq times$. In case the error condition is not matched till the maximum value of epochs *times*, the new values for the parameters r_j remain equal at the old ones.

This operation is repeated for all nodes *j* of the hidden layer. When the updating has been performed we compute the new value for the weights W with the pseudo inverse technique 7.23, with the new parameters of the basis functions.

7.3 Discussion

The neural network approach, when applied to the colorimetric characterization problem, presents some limitations that have so far limited the more widespread use of such methods [ArWi03]: high computational costs, large number of training samples needed and the definition of a suitable termination condition for the learning process in the case of a direct colorimetric characterization function (from an independent device colour space as CIELAB to a dependent device colour space as CMY) is requested.

High computational costs

High computational costs are related to the neural network structure, and the algorithm used in order to train the system. Also a high number of training samples influence the computation time. To simplify the neural network structure and the learning algorithm, reduce the requested computation time for the training phase. The model presented defines an automatic way to find the best structure for the neural network without using more neurons (in the hidden layer) than necessary.

Number of Training Samples

The number of training samples is requested to be as small as possible, to reduce the time necessary to measure the colour samples using a spectrophotometer. Also a small number of training samples reduces the computation time requested for the learning phase. In Chapter 8, where experiment results are presented, is shown that the number of the training samples has been reduced to 125 samples, in comparison to the traditional techniques that normally use 729 samples, without reducing the quality performances.

Suitable Termination Condition

To define a suitable termination condition for the learning process in the case of a direct colorimetric characterization function is still an open question. Usually a termination condition is defined as an average colour error, of the training samples, between the original colour and the predicted one. It does not present problems in the case of an inverse colorimetric characterization function, from a dependent device colour space as CMY to an independent device colour space as CIELAB, because the average colour error is computed in a pseudo uniform colour space as the CIELAB.

Instead, if we use the same strategy in the case of a direct colorimetric characterization function, the average colour error is computed in a dependent device colour space as CMY that is a not uniform colour space. It means that the error information that we get does not

respect the human perception, and it does not help define the right neural network structure obtained by the learning process. As discussed in [ArWi03], a possible solution is to adopt the computation of the average colour error in the CIELAB colour space also for the direct mapping function. In order to do it we need to use the inverse mapping function for each termination check, which in turn also implies that a neural network for this inverse case ought to be used. This solution is not feasible because the computational cost of the inverse network-learning phase is significant, and too much accuracy is lost through the repeated conversions. The solution that we adopted has been to generate a precise LUT with 729 uniform samples in CMY space with our type of RBF network, which was trained with 125 samples. It can be used with any of the most used colorimetric characterization techniques to perform the direct mapping function (i.e. interpolation, multiply polynomials regression etc.). In other words, the basic idea was to generate a big and precise LUT, only from 125 samples measured, with the RBF network proposed and to be used with a more stable colorimetric characterization model.

Chapter 8

Experimental Results

Introduction

The experimental results have been divided into three parts and described in different section. In Section 8.1 the experiments on the novel time dependent tone mapping operator are presented. In Section 8.2 we described the experiments on the framework proposed to accelerate any pre-existing complex global tone mapping operator. Finally, in Section 8.3 we present the experiments on the novel colorimetric characterization model usable inside the tone mapping framework for printer.

8.1 Time-Dependent Tone Mapping Operator

In this section we present the obtained results of the novel time-dependent tone mapping operator. In particular, an exhaustive comparison with the original time-dependent tone mapping operator presented by Pattanaik et al. [Pa+00] was performed. The experiment are divided into four subsections: *luminance compression*, *colour reproduction*, *visual effects* and *chromatic adaptation*.

Luminance compression

In general a tone mapping operator has to deal with the compression of huge HDR of the input scene. To do this, one has to consider several aspects such as contrast reproduction, the visibility of object etc. When a TM operator solve some of these aspects, it usually negligees the others. This fact prevents the use of these operators in all applications. It is the case of the Pattanaik et al. model [Pa+00], that on the one hand reproduce carefully the dynamic visual

adaptation of the HVS, on the other hand presents deficiencies in the compression of huge luminance dynamic range. Figure 8.1 (left) shows an output scene obtained with the Pattanaik et al. model [Pa+00]. The same scene obtained using our operator is shown in Figure 8.1 (right). Both images are obtained in the last step of adaptation.



Figure 8.1: Hotel room [Crone, Fawler and Kerrigan 97] (3000 x 1950); image obtained with the Pattanaik et al. model [Pa+00] (left), image obtained with our operator (right). The original image has luminance range equal to $[0.37; 18828] \text{ cd}/m^2$.

One can note that the contrast is better reproduced in the scene obtained with our operator than for the same scene obtained with the original Pattanaik model. In Figure 8.2 we show further images, with different contrast ratio, obtained with our model. In all of these cases the algorithm used for luminance compression can reproduce the contrast and the visibility of objects, without showing any deficiencies for very huge HDR luminance.



Figure 8.2: Flat [Ward, Shirley, Debevec, Malik 1997], with luminance range equal to $[1.78; 3248] \text{ cd}/m^2$ (left); Office_light [Ward 1997], with luminance range equal to $[2.26; 8379] \text{ cd}/m^2$ (right). Images obtained with our operator.

Colour reproduction

As is well known, the colour is an important aspect of the real world and must be considered to reproduce a scene realistically. In Figure 8.1 and 8.2 it is possible to observe how our operator can reproduce the colour in the final scene.

The solution proposed by Pattanaik et al. [Pa+00] requires some computational steps, as discussed in Section 5.2.1, that can increase the final computation time. Our simple solution (see section 5.2.2 Adaptation Model), that achieves good results compared with the results of the Pattanaik model, requires just simple operations reducing the final computation time.

Visual effects

As described in Chapter 1, the HVS presents some limitations under particular lighting conditions. This can reduce or increase the ability of the HVS to see fine details and colours of the objects in the scene. In our model we integrated three basic visual effects: *glare*, *visual acuity* and *colour sensitivity*. The glare introduces an impression of great brightness and interferences in the visibility of the object near the light source. We tested the glare as shown by the example in Figure 8.3.



Figure 8.3: Hotel room [Crone, Fowler and Kerrigan 97] (3000 x 1950); image obtained with our operator but without added the glare (left), image obtained with our operator with added the glare (right). The original image has luminance range equal to $[0.37; 18828] \text{ cd}/m^2$.

Remarkably enough the glare is well reproduced (see also Figure 8.4). In particular, the details in the window and in the close bulb lamp are lost causes of the glare (right image). By contrast, in the image on the left reproduced without applying the glare, the details of the window and the bulb lamp are much better visible.

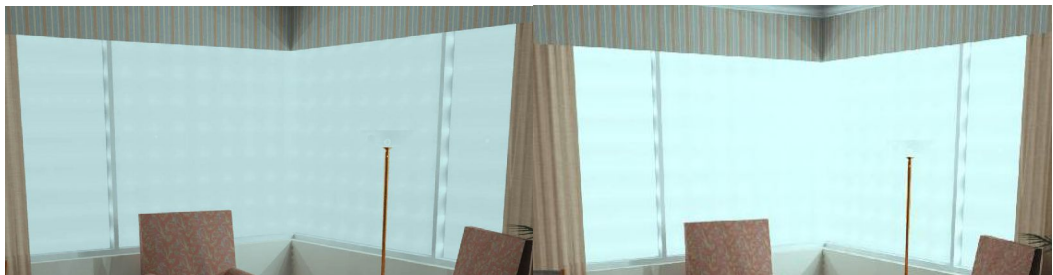


Figure 8.4: Particular of the window in the Hotel room image: (left) without glare, (right) with glare. The details in the window (image on the right) are lost causes of the glare. By contrast the details in the window, are present in the case the glare is not applied (image on the left).

The acuity is a measure of the HVS ability to resolve spatial details [Fe+96]. Figure 8.5 shows an example in which the visual acuity is changing according with the light conditions. On the left a scene without adding the visual acuity is reported. By contrast, on the right a scene with added the visual acuity is shown. The details in the window and in the lamps (bulbs) are better visible in the scene on the right than in that in the left (see also the particulars in Figure 8.6).



Figure 8.5: Hotel room [Crone, Fawler and Kerrigan 97] (3000 x 1950); image obtained with our operator but without added the visual acuity (left), image obtained with our operator with added the visual acuity (right). The original image has luminance range equal to $[0.37; 18828]$ cd/m^2 .



Figure 8.6: Particular of the window in the Hotel room image: (left) without visual acuity, (right) with visual acuity. The details in the window (image on the right) are more visible and in particular the bulb of the lamp is well defined.

Chromatic Adaptation

An important aspect of the HVS is the *chromatic-adaptation*. It represents the capability to adjust to widely varying colours of illumination in order to approximately preserve the appearance of object colours. In other words, if the viewer conditions of the two observers (real observer and display observer) are different we need to consider the new viewing conditions of the display observer in order to reproduce correctly the real scene on the display.

The results of the chromatic adaptation are shown in Figure 8.1, where the images obtained with the two operators are presented. The image obtained with the original Pattanaik model (left), is not adapted to the chromaticity of the display white point. In this case, the image appears too reddish. By contrast, the image obtained with our operator (right) is chromatic adapted to the display white point. This results in a better image.

8.2 Real Time Tone Mapping Operator

The experiments, of the framework proposed in Chapter 6, are organized in the following manner: first, we tested the software implementation of the all framework in order to find the suitable setting for the final configuration of its hardware implementation. Second, we performed the experiments of the best configuration of the framework proposing the subdivision of the implementation between CPU and GPU.

Software Implementation

We conducted a series of tests to capture the behaviour of different parts of the framework in dependence on the type of the sampling algorithm, the number of samples, and the type of interpolation. For the first series of tests, we used the tone-mapping operator of Pattanaik et al. [Pa+00]. To evaluate the time performance, we measured the time of the sampling, the tone mapping, the fitting, the reconstruction and the total time. The results are summarized in Table 8.1.

<i>Sampling</i>	# <i>Samples</i>	<i>Time</i>					<i>Speed-up</i> [-]
		<i>Sampling</i> [ms]	<i>TM</i> [ms]	<i>Fitting</i> [ms]	<i>Reconst.</i> [ms]	<i>Total</i> [ms]	
<i>No acceleration</i>	-	-	-	-	-	54852	1.0
<i>Halton</i>	300	0.642	3.232	0.012	3118	3122	17.57
	1000	2.287	7.212	0.029	3381	3391	16.18
	2000	5.780	10.968	0.055	3423	3440	15.94
	3000	7.823	13.071	0.074	3469	3490	15.72
<i>down-sampling</i>	300	2024	3.604	0.014	3131	5159	10.63
	1000	2024	13.384	0.132	3522	5557	9.870
	2000	2051	20.075	0.162	3729	5800	9.457
	3000	2032	30.080	0.300	3879	5941	9.233
<i>flt. Rand.</i> <i>3 x 3</i>	300	1.098	3.498	0.013	3222	3226	17.00
	1000	3.833	11.260	0.047	3610	3625	15.13
	2000	7.906	19.992	0.162	3813	3841	14.28
	3000	12.127	29.502	0.330	3963	4005	13.70

Table 8.1: Evaluation of the framework in software using Pattanaik's operator on the Hotel room [Ar+03].

Our framework provides a speed-up of Pattanaik's operator in the range between 9.23 and 17.6. For example using 1,000 samples, our method accelerates the operator 16.18 times on an image of 3000x1950 pixels. We can see, in Figure 8.7, that down-sampling provides more accurate results than random sampling at the cost of a slight increase in computational time as shown in table 8.1. When the number of samples increases, the accuracy of random sampling becomes comparable with down-sampling. In order to reduce the increased of

computational costs of down-sampling, we tried the filtered random sampling approach. Preliminary results have shown good results using a filter window size of 3×3 and 2000 samples. In Figure 8.7 we can see that the results for filtered random sampling are comparable to down-sampling, but the computational costs are comparable with the computational costs of random sampling (Table 8.1). For all measurements, the time of the reconstruction is the major component of the total computational time. This justifies the subdivision of the computation between the GPU and CPU, which suggests implementing the reconstruction in the GPU.

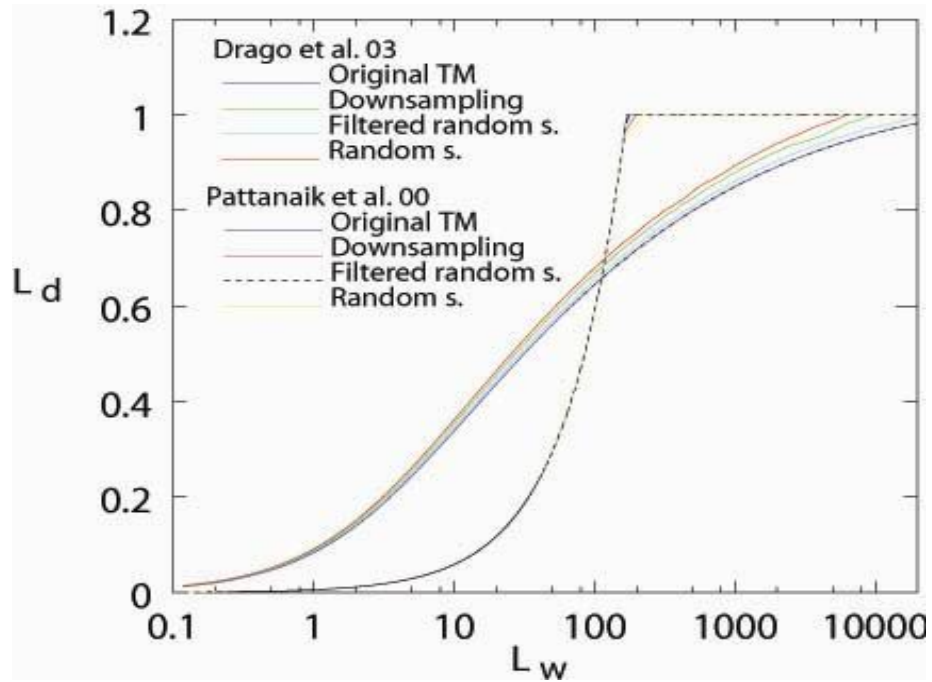


Figure 8.7: Fitting accuracy of the framework using random sampling, down-sampling with 1000 samples, and filtered random sampling with 2000 samples. The tone mapping operators used are Drago et al. [Dr+03] and Pattanaik et al. [Pa+00], both on the Memorial church image [Ar+03].

In Figure 8.8 two examples of images, obtained with two different operators Ashikhmin [As02] and Drago et al. [Dr+03] (left), are shown and the same images obtained applying the framework on these two operators using random sampling (centre) and down-sampling (right) with 1000 samples, respectively. It is possible to observe how the down-sampling (right image) is able to preserve the quality of the original image (left image), compared to the random sampling.



Figure 8.8: Hotel room [Crone, Fawler and Kerrigan 97] (3000 x 1950); The top images are obtained using the original Ashikhmin tone mapping operator (left), the Framework using random sampling (centre), and the Framework using down-sampling (right). The bottom images are obtained original Drago et al. tone mapping operator (left), the Framework using random sampling (centre), and the Framework using down-sampling (right).

We also evaluated our method using other TM operators, namely the operator of Tumblin et al. [TuRu93] and the operator of Drago et al. [Drago03].

For each input image we measured the results obtained by direct application of tone mapping and the results obtained by applying our technique using random sampling with 1000 samples and linear interpolation. Table 8.2 summarizes the results. The data in table 8.2 are related to the Framework completely implemented in software.

<i>image</i>	<i>Operator</i>	<i>Time</i>		<i>Speed-up</i> [-]
		<i>original</i> [ms]	<i>acceleration</i> [ms]	
<i>Hotel</i> (3000 x 1950)	<i>Pattanaik et al.</i> [Pa+00]	54852	3391	16.18
	<i>Tumblin</i> [TuRu93]	5657	3344	1.70
	<i>Drago et al.</i> [Dr+03]	9359	3336	2.81
<i>Airport</i> (1024 x 705)	<i>Pattanaik et al.</i> [Pa+00]	6815	426	16.00
	<i>Tumblin</i> [TuRu93]	687	404	1.70
	<i>Drago et al.</i> [Dr+03]	1154	400	2.98

Table 8.2: Evaluation of the Framework, in software, applying three tone mapping operators using random sampling (1000 samples) and linear interpolation on three different images [Ar+03].

We can observe that all three complex global operators are speed up, and that the time performance of the framework is practically independent from the tone mapping operator used. The accelerated time in Table 8.2 is practically the same for all tested operators.

Figure 8.9 shows other results of our framework applied on different tone mapping operators. In the first row the images are obtained with the original tone mapping operators, instead in the second row the images are obtained applying our framework on these operators using 1000 samples and down-sampling.

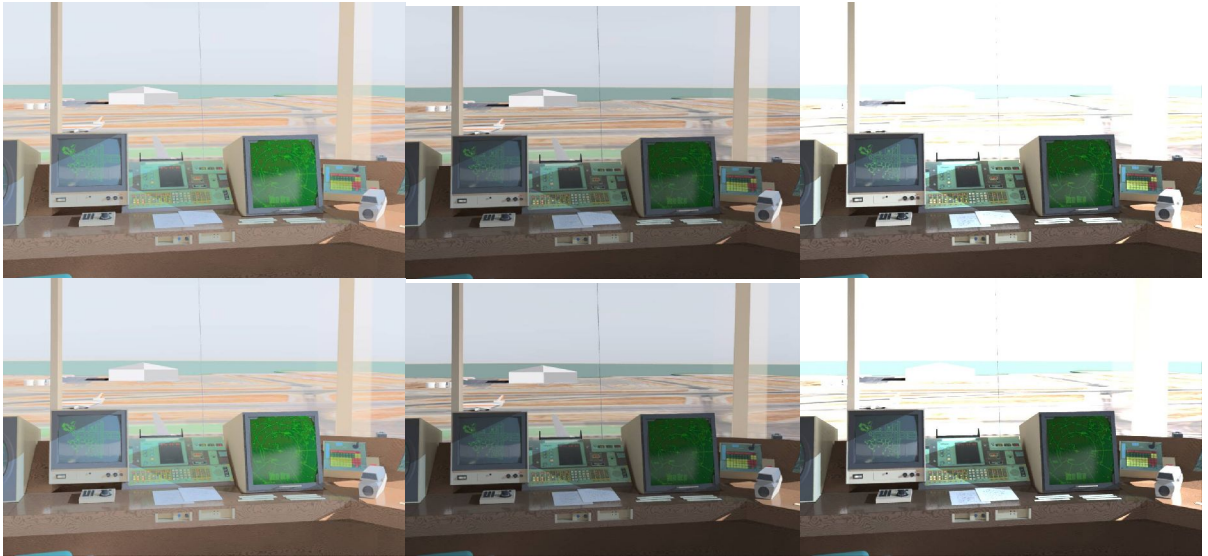


Figure 8.9: The Airport [C. Ehrlich, G. Ward, 97] (1024 x 705); The top images are obtained using the original tone mapping operators, the bottom images are obtained by applying our framework. (Left) Ashikhmin [As00]; (center) Drago et al. [Dr+03]; (right) Ward [Wa94].

Hardware Implementation

For an evaluation of the hardware implementation of the method we used a simple OpenGL interactive application, which uses high dynamic range environment maps. The tests were conducted for the tone mapping curve produced by the following tone-mapping operators: Ashikhmin [As02], Drago et al. [Dr+03], Exponential, and Tumblin [TuRu93]. In the case of Ashikhmin's operator, we apply it directly to the pixel luminance without computing the luminance adaptation.

We measured the time performance of the method in dependence on the type of the sampling technique. We also measured the time of the actual tone mapping (executed on the CPU), and the total frame time. We also measured the time performances of a direct GPU implementation of the tone mapping operator used in these experiments. Note that we did not encounter any significant delays due to latency between CPU and GPU because only a very small image is read back from the graphics card into main memory. The results are summarized in Table 8.3.

<i>sampling</i>	<i>samples</i>	<i>Time</i>		<i>FPS</i> [-]
		<i>TM</i> <i>CPU</i> [ms]	<i>Total</i> [ms]	
<i>random</i>	1024	1.23	18.71	56
<i>downs.</i>	1024	1	21.42	48
<i>filt. rand. 3x3</i>	1024	1.01	18.87	56
<i>direct GPU</i>	-	-	19.76	51

Table 8.3: Evaluation of the real-time application using Ashikhmin's tone mapping operator accelerated by our framework in hardware, and its direct implementation on the GPU. The image resolution is 512 x 512 [Ar+03].

The results show that the hardware implementation provides real-time performance to Ashikhmin's tone mapping operator. It is interesting to note that our framework can even compete in terms of speed against a direct hardware implementation of the quite simple Ashikhmin operator, albeit at reduced quality due to the sampling. The filtered random sampling technique does not cause noticeable performance degradation as compared to random sampling, but it provides better image quality. We also tried to use the binary search, in the reconstruction phase, but the results in terms of computation time were three times slower than the performances obtained with the LUT solution.

Our original experiments with the hardware implementation revealed significant temporal aliasing in the form of flickering. This occurred especially when random sampling was used, because the input parameters of the used tone mapping algorithms, i.e., minimum, maximum or average, were not well captured by the small sample set used in the implementation. In order to reduce these artefacts, we reuse each sample for a certain number of frames as discussed by Scheel et al. [Sc+00]. Note that the measurements in Table 8.3 already reflect this improvement. Each sample was reused for 8 frames, which reduced the flickering by a significant amount. An interesting observation was that even the direct implementation of the Ashikhmin operator exhibited noticeable flickering in dynamic scenes. It happens because an interactive application can often suffer from large temporal discontinuities in the dynamic range.

Such discontinuities can be smoothed also by incorporating a model of time-dependent adaptation.

8.2.1 Summary

We have presented a framework that delivers interactive performance to complex global TM operators. When integrated into the rendering pipeline, the proposed framework allows

studying the response of complex global tone mapping operators interactively. It also enables to select interactively the most appropriate tone mapping operator for a given application. The structure of the framework is designed for easy integration into the rendering pipeline of recent graphics hardware. Unlike previous TM techniques using graphics hardware, our method does not require modification of the rendering pipeline as it works as a post-process on a HDR image. We proposed an efficient subdivision of the workload between the CPU and the GPU. The TM operator is applied on the CPU, which maintains the generality of the method. The GPU resolves the computationally simple but costly stages of the algorithm. An important feature of the framework is that the GPU implementation is simple and it does not need to be modified for an application with a different tone mapping operator. The hardware implementation of our method shows a speed-up of one order of magnitude compared to the pure software solution. This proves the potential of the proposed technique for interactive rendering applications. The modularity of the proposed framework enables to concentrate on improving each of its parts independently.

8.3 Colorimetric Characterization Model for Ink-jet Printers

The first step in our research was to find a good learning algorithm to train the RBFN function with small training sets. In order to do this, we evaluated different learning's algorithm in the following order: first linear models with forward selection and local ridge regression (*GCV* [Go+79], *UVE* [EfTi93], *FPE* [Ma73], *BIC* [Sc78]); we used the Matlab implementation of these algorithms by Orr [O96]. Then we considered non-linear models as proposed by Lee [Le99] and Carozza [CaRa99]; we implemented these ourselves in C. Our proposed own non-linear model was again implemented in Matlab. All these algorithms were then compared to multiple polynomial regression and tetrahedral interpolation.

When we wanted to find out whether a particular learning algorithm is adequate to solve the posed problem, we first tested it only for the conversion $CMY \rightarrow CIELAB$ on an Epson Stylus Pro5000, and only if this preliminary test turned out favourably, we conducted further experiments on other printers and with the conversion $CIELAB \rightarrow CMY$. How to reproduce the training and test sets is presented in Subsection 8.3.1. In Subsections 8.3.2 are presented the all experiments results. Finally a summary of the results is presented in Subsection 8.3.3.

8.3.1 Training and Test sets

The colorimetric patterns of the training and test sets, which we used in these experiments, are formed by pairs of three dimensional vectors. One of these vectors specifies the *CMY* coordinates, and the other specifies the *CIELAB* coordinates of a printed colour. The colour sets for the training and test phases are obtained by printing a number of hues, specified in *CMY* space, in squares of approximately 1 cm^2 at the highest resolution the printer has to offer. These colour swatches are then measured with a SPECTROLINO spectrophotometer produced by GretagMacbeth.

For the training phase of our experiments we used four different sets of this kind, labelled *Training1* through *Training4*. The sets *Training1* and *Training2* were made up of 729 and 125 colours, respectively, which were obtained by uniform sampling in *CMY* space. The sets *Training3* and *Training4* consisted of 392 and 252 colours, which were obtained as suggested by Moroni [Mo96]. The Test set contains 777 colours obtained by random sampling of *CMY* space. The error of the models was calculated in *CIELAB* space according to the formula

$$\Delta E = \sqrt{(L - L')^2 + (a - a')^2 + (b - b')^2} \quad 8.1$$

where (L, a, b) is the output of the models and (L', a', b') is the target output. The experiments were conducted using several different ink--jet printers, namely an *Epson Stylus Pro5000* (with photo quality paper), an *Olivetti Artjet 20* (with coated paper) and a *HP2000C* (with cut sheet paper). The code for the algorithms tested in these experiments was written in C and Matlab.

8.3.2 Experimental Results

In this subsection we present the all experimental results that compare different techniques used to perform the colorimetric characterization of an ink-jet printer.

These results are summarized from table 8.4 to table 8.15, where in the columns for the training and test sets we show the average error on the left and the maximum error on the right of the respective cells. The maximum error has particular importance because it represents the main parameter to evaluate if a colorimetric characterization model is usable or not in real application. For instance a model that has a better maximum error is preferable, if the divergences in the average error are not higher than 3 cdu (cdu is the unit of colour difference) in CIElab coordinates.

We began the experiments by initially testing existing learning algorithms, specifically those proposed by Golub [Go+79], Efron [EfTi93], Mallows [Ma73] and Schwarz [Sc78] on the colour printer *Epson Stylus Pro 5000*; we used two training sets (labelled *Training1* and *Training2* in tables) and tested the networks with a set labelled *Test*. The results are shown in Table 8.1 for the function $CMY \rightarrow CIELAB$.

RBFN	Traning1		Test		Training2		Test	
	Avg.	Max	Avg.	Max	Avg.	Max	Avg.	Max
<i>GCV</i>	0.287	1.182	0.823	5.551	0.525	1.487	3.025	10.867
<i>UEV</i>	0.114	0.486	0.823	5.249	0.016	0.067	3.006	10.635
<i>FPE</i>	0.157	0.706	0.808	5.188	0.016	0.067	3.006	10.635
<i>BIC</i>	0.185	0.754	0.827	5.570	0.016	0.067	3.006	10.635

Table 8.4: Error comparison of $CMY \rightarrow CIELAB$ conversion using the initial RBFN, with the selection criterions (*GCV*, *UEV*, *FPE*, *BIC*), without ridge regression for the *Training1*, *Traning2* and *Test* datasets on the *Epson Stylus Pro5000*.

Our results demonstrate that this model is already able to improve the performance of multiple polynomial regressions for polynomials up to 60 terms and tetrahedral interpolation, shown in table 8.5, in the case of the set *Training1*. However, this does not extend to the set *Training2*, where the unmodified RBFN approach fares no better than the conventional techniques. This is probably due to the fact that in this case the network encounters overfitting problems as mentioned in Chapter 7. It is characterized by the fact, as showed in Table 8.4 in the column of Avg. and Max error, that with some learning methods (*UEV*, *FPE*, *BIC*) we obtain the same values for the error using the *Training2*. In order to resolve this problem we have tried to use regularization theory (*local* and *global ridge regression*), but this failed to improve the results. These results are shown in table 8.6, and also in this case we encountered the same problem but for both training set (*Training1* and *Training2*).

<i>Method</i>	<i>Traning1</i>		<i>Test</i>		<i>Training2</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>60 terms</i>	2.390	10.970	3.170	10.970	2.490	7.670	3.220	9.790
<i>69 terms</i>	2.310	10.920	3.100	10.910	2.450	7.630	3.170	10.110
<i>87 terms</i>	1.810	8.040	2.510	8.190	1.720	5.450	2.650	8.270
<i>105 terms</i>	1.630	8.790	2.300	8.660	1.470	4.370	2.570	7.670
<i>Tetrahedral</i>	0.0	0.0	0.810	5.01	0.0	0.0	2.110	7.490

Table 8.5: Error comparison of $CMY \rightarrow CIELAB$ conversion using regression, with different polynomials and tetrahedral interpolation for the *Training1*, *Traning2* and *Test* datasets on the *Epson Stylus Pro5000*.

<i>RBFN</i>	<i>Traning1</i>		<i>Test</i>		<i>Training2</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>GCV</i>	0.477	2.171	0.834	4.818	0.525	1.487	3.024	10.873
<i>UEV</i>	0.469	2.118	0.827	4.841	0.035	0.100	3.003	10.637
<i>FPE</i>	0.469	2.118	0.827	4.841	0.035	0.100	3.003	10.637
<i>BIC</i>	0.522	2.295	0.867	4.653	0.035	0.100	3.003	10.637

Table 8.6: Error comparison of $CMY \rightarrow CIELAB$ conversion using the initial RBFN, with the selection criterions (*GCV*, *UEV*, *FPE*, *BIC*), with ridge regression for the *Training1*, *Traning2* and *Test* datasets on the *Epson Stylus Pro5000*.

Another approach was to generate more training sets with smaller numbers of samples compared to the original large set *Training1*. We produced two such sets in the way suggested by Moroni [Mo96], labelled *Training3* and *Training4*, with 392 and 252 samples respectively. Results from test runs with these sets are reported in table 8.7, 8.8 and 8.9 show that there are indeed improvements with respect to multiple polynomial regression with the set *Training3*, and equal performance with *Training4*.

<i>RBFN</i>	<i>Traning3</i>		<i>Test</i>		<i>Training4</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>GCV</i>	0.223	1.030	1.296	4.127	0.200	0.787	3.429	11.637
<i>UEV</i>	0.099	0.378	1.577	5.231	0.119	0.600	3.723	16.176
<i>FPE</i>	0.109	0.432	1.528	5.108	0.192	0.735	3.446	11.746
<i>BIC</i>	0.138	0.578	1.496	4.531	0.192	0.735	3.446	11.746

Table 8.7: Error comparison of $CMY \rightarrow CIELAB$ conversion using the initial RBFN, with the selection criterions (*GCV*, *UEV*, *FPE*, *BIC*), without ridge regression for the *Training3*, *Traning4* and *Test* datasets on the *Epson Stylus Pro5000*.

<i>RBFN</i>	<i>Traning3</i>		<i>Test</i>		<i>Training4</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>GCV</i>	0.390	1.849	1.286	4.347	0.355	1.027	2.811	9.003
<i>UEV</i>	0.366	1.738	1.271	4.383	0.310	0.939	2.995	9.046
<i>FPE</i>	0.390	1.849	1.286	4.347	0.313	0.938	2.984	8.900
<i>BIC</i>	0.390	1.849	1.286	4.347	0.330	0.966	2.902	9.000

Table 8.8: Error comparison of $CMY \rightarrow CIELAB$ conversion using the initial RBFN, with the selection criterions (*GCV*, *UEV*, *FPE*, *BIC*), with ridge regression for the *Training3*, *Traning4* and *Test* datasets on the *Epson Stylus Pro5000*.

<i>Method</i>	<i>Traning3</i>		<i>Test</i>		<i>Training4</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>60 terms</i>	2.386	8.114	3.315	10.674	2.470	9.217	3.603	10.934
<i>69 terms</i>	2.328	8.432	3.271	10.863	2.372	9.320	3.731	9.920
<i>87 terms</i>	1.771	6.066	2.669	8.644	1.771	5.708	3.271	8.096
<i>105 terms</i>	1.556	5.810	2.543	7.501	1.461	5.642	3.497	10,043

Table 8.9: Error comparison of $CMY \rightarrow CIELAB$ conversion using regression, with different polynomials for the *Training3*, *Traning4* and *Test* datasets on the *Epson Stylus Pro5000*.

However, compared to the still large size of the new reduced training sets the improvement is rather small and also in this case we encountered the over-fitting problem encountered with the others methods (Table 8.7 and Table 8.8). The methodology for generation of the

condensed training set proposed by Moroni [Mo96] apparently does not allow for the desired increase in efficiency.

In this phase of the experiments we also tried other innovative learning algorithms found in literature (such as Lee [Le99] and Carozza [CaRa99]), but the results we obtained were of poor quality. We then altered our strategy and decided to modify an existing learning algorithm. Our choice here has been to modify the learning algorithm proposed by Carozza [CaRa99], mainly because it does not get significantly more complex when it is modified, and also because it does not have inherent convergence problems like the algorithm of Lee [L99]. The results we obtained are shown in tables 8.10 and 8.11; the tests were done on an Epson Stylus Pro5000 for the function $CMY \rightarrow CIELAB$.

	<i>Traning2</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>Proposed RBFN</i>	0.797	1.893	1.831	6.763

Table 8.10: Initial error measurements of $CMY \rightarrow CIELAB$ conversion using the proposed *RBFN* for the Training2 and Test datasets on the Epson Stylus Pro5000.

The results show how it is possible to obtain performance that is better than that of multiple polynomial regression and tetrahedral interpolation with only 125 samples. This is shown in the last two columns beginning at the right of Table 8.5 for both sets (Training and Test) and both error metrics (average and maximum). In order to make sure that these results are consistently reproducible over time we repeated the experiment in May, June, and July, reprinting the training and the test samples in each case; the progression of the error over time is shown in Table 8.11.

<i>Month</i>	<i>Traning2</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>May</i>	0.927	1.980	2.024	6.833
<i>June</i>	0.744	1.537	2.043	5.780
<i>July</i>	0.784	1.737	2.210	5.699

Table 8.11: Subsequent error measurements of $CMY \rightarrow CIELAB$ conversion using the proposed *RBFN* for the Training2] and Test] datasets on the Epson Stylus Pro5000 measured at one month intervals.

The results are in line with the first experiment reported in Table 8.10. In order to validate our model we performed similar tests on two more ink-jet colour printers from other

manufacturers, namely a HP2000C and an Olivetti Artjet20. The results are shown in Table 8.12 for the Epson Stylus Pro5000, in Table 8.13 for the HP2000C and in Table 8.14 for the Olivetti Artjet20. The data is also compared against results from multiple polynomial regression; all tests were done for the function $CMY \rightarrow CIELAB$.

<i>Method</i>	<i>Traning2</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>Proposed RBFN</i>	0.797	1.893	1.831	6.763
<i>Regression 60 terms</i>	2.495	7.674	3.228	9.798
<i>Regression 69 terms</i>	2.458	7.639	3.176	10.111
<i>Regression 87 terms</i>	1.721	5.453	2.654	8.277
<i>Regression 105 terms</i>	1.475	4.337	2.571	7.676

Table 8.12: Error comparison of $CMY \rightarrow CIELAB$ conversion using the proposed RBFN and regression with different polynomials for the Training2 and Test datasets on the Epson Stylus Pro5000.

<i>Method</i>	<i>Traning2</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>Proposed RBFN</i>	2.876	5.774	3.691	9.800
<i>Regression 60 terms</i>	6.610	19.141	5.396	18.794
<i>Regression 69 terms</i>	6.516	19.344	5.728	18.995
<i>Regression 87 terms</i>	5.182	11.187	5.502	13.077
<i>Regression 105 terms</i>	4.607	10.714	4.744	13.072

Table 8.13: Error comparison of $CMY \rightarrow CIELAB$ conversion using the proposed RBFN and regression with different polynomials for the Training2 and Test datasets on the Hewlett Packard HP 2000 C.

<i>Method</i>	<i>Traning2</i>		<i>Test</i>	
	<i>Avg.</i>	<i>Max</i>	<i>Avg.</i>	<i>Max</i>
<i>Proposed RBFN</i>	1.752	4.084	2.537	6.660
<i>Regression 60 terms</i>	1.901	10.268	2.236	9.739
<i>Regression 69 terms</i>	1.851	10.400	2.266	9.864
<i>Regression 87 terms</i>	1.514	9.100	1.994	8.664
<i>Regression 105 terms</i>	1.305	9.100	1.936	8.664

Table 8.14: Error comparison of $CMY \rightarrow CIELAB$ conversion using the proposed RBFN and regression with different polynomials for the Training2 and Test datasets on the Olivetti Artjet 20.

The results show that there is an improvement over multiple polynomial regressions for every ink-jet colour printer. The results shown in tables 8.12 to 8.14 demonstrate that our model has general validity.

The final problem that we discuss in this thesis work is the definition of the function $CIELAB \rightarrow CMY$ using our model; now the main problem is that of the definition of a suitable termination condition for the learning process. If we use the same condition that we used for the definition of the function $CMY \rightarrow CIELAB$ and compute the error in CIELAB space, is necessary to use the inverse transformation for each termination check, which in turn also implies that a neural network for this inverse case ought to be used. This solution is not feasible because the computational cost of the inverse network-training phase is significant, and too much accuracy is lost through the repeated conversions. Neither is it possible to compute the error in CMY space, because this approach is inherently incapable of knowing when to stop the learning process.

The solution that we have adopted has been to generate a LUT with 729 uniform samples in CMY space with our type of RBF network, which was trained with 125 samples. We then used this LUT with multiple polynomial regressions on the set Test. We compared this result to the result obtained with a LUT of 729 uniform samples, which were printed and measured from CMY space, and which were also used with multiple polynomials regression on the same set of samples. The results are reported in Table 8.15; this test was performed on an ink--jet printer of the type Epson Stylus Pro5000.

<i>Regression</i>	<i>Avg.</i>	<i>Max</i>
<i>LUT (729) printed</i>	2.244	9.894
<i>LUT (729) created by proposed RBFN</i>	2.290	8.890

Table 8.15: Error of *CIELAB* \rightarrow *CMY* conversion Test datasets on the Epson Stylus Pro5000.

These results show that it is possible to generate a LUT from only 125 initial printed and measured samples with our method, compared to 729 samples used by multiple polynomial regression or interpolation models. Our model permits a fast re-characterization of ink--jet colour printers because it needs only 125 printed and measured samples, and in addition its training phase is very fast. On a Pentium II Celeron system with 128 Mbyte of RAM the training time with the initial 125 samples is approximately 10 minutes, and the time to generate the 729 entry LUT is one additional minute.

8.3.3 Summary

We have presented a new learning algorithm, which is a modification of a known technique that trains the RBFN model for the colorimetric characterization of colour printers. Our algorithm needs a training set of only 125 samples in order to train the RBFN. With this model is even possible to generate a LUT of 729 samples, beginning with only 125 printed and measured samples, and to use this LUT with other standard algorithms of colorimetric characterization.

The computational cost is very low in the training and testing phases, and is even better than the performance of other standard colorimetric characterization models (e.g. multidimensional polynomials regression and tetrahedral interpolation). It can be expressed as a difference between the maximum error obtained with the different approaches, and in our case the improvement is between 1.1 and 3.2 cdu in CIELab coordinates.

In our opinion, the results suggest that may be possible to use this algorithm in consumer products, because we have been able to resolve the two problems that have so far limited the more widespread use of such methods: high computational cost and the large number of training samples needed. The small size of the training set also permits a fast re-characterization of devices.

We believe that there are several possible ways to evolve these models for colorimetric characterization problems: investigation of different mathematical models for the estimation of the basis function parameters, research on different mathematical models for the estimation of the weights, introduction of one or two more hidden layers in the structure of the RBFN, and eventually experiments that involve combinations of these new techniques.

Chapter 9

Conclusions and Future Work

In this thesis we presented new ideas on how to accelerate processes in the context of the TM problem. We also introduced a modification of an existing time-dependent TM methodology. More precisely:

- In Chapter 5 we presented a modification of an existing time-dependent TM model and we integrated some effects of the HVS in it. In order to take into account the different viewing conditions of the two observers (real world and display), a chromatic adaptation model was also integrated.

- In Chapter 6 we defined a new framework, applicable to monitor systems, which delivers interactive performances to complex global tone mapping operators. When integrated into the rendering pipeline, the proposed framework allows one studying the response of complex global tone mapping operators interactively. It also enables to select interactively the most appropriate tone mapping operator for a given application. This addresses to two basic questions:

- *Generality*: Our framework can be applied to any pre-existing global TM operator without modifying it. It can be also easily integrated in the rendering pipeline without any modification.

- *Computational Complexity*: Our framework reduces the complexity of the original TM operator used, since apply it on a subset of the original image. In this way, the computational complexity is independent from the specific existing TM operator.

- In Chapter 7 we introduced a methodology to accelerate a framework applicable to a printer. More precisely we presented a new learning algorithm, which is a modification of a known technique that trains the RBFN model for the colorimetric characterization of colour printers. Our modification solves two basic problems that reduce the use of neural networks in real application:

- *Computational costs*: A small training set of just 125 samples is used to train the RBFN. This reduces computation costs.
- *Quality performances*: Despite the reduced number of samples used in the training phase, the performances are comparable and even better than the performances of standard methods used for this purpose.

The solutions proposed in this thesis are far from being final. The integration of a time dependent adaptation model in a local operator, or the extension of the existing time dependent TM operators to work locally, are two possible directions for future work. A possible way to extend the model of Pattanaik et al. [Pa+00] in order to preserve local image contrast, is to modify the exponential filter used in the Dynamic Adaptation model (section 5.2.1 Adaptation model) to be updated for each frame locally. This also allows simulating the local adaptation mechanisms that are characteristic of the HVS during the visual adaptation process.

The framework proposed in Chapter 6 is applicable only to a global TM operators, but fails when it is applied to local TM operators. Indeed, a local TM operator uses the spatial structure of the image data and attempts to preserve local image contrast. This permits to transform the same pixel intensity of the input image to different display values, or different pixel intensities to the same display value [DiWa00].

To capture the behaviour of a local TM operator one needs to modify the framework described in this thesis. In particular the sampling and the reconstruction steps need several changes.

A local TM operator works applying a filter on a subset of the original input image. Thus, the sampling step needs to be modified to capture the real information needed for the local TM operator. One way to modify it is to sample the input image and transfer to the TM operator also the neighbourhood pixels, included in the size of the filter kernel, of the sampled pixel. In this way, we can transfer the information that the TM operator needs, but restricted only on a subset of the input image. Hence the aim of the original framework (i.e. to reduce the total amount of operations that the operator has to perform on the whole image) is retained. Also

the reconstruction step will be affected by the modification of the sampling step. Indeed this step uses both data: the sampled pixels and the response of the local TM operator applied on the sampled pixels. Finally, it has to reproduce the original behaviour of the TM operator. Our opinion is that an algorithm suitable for this purpose should be a learning algorithm like a neural network. For instance, a neural network has the ability to learn from a set of samples and to generalize the behaviour learned.

In this way, the neural network can learn the behaviour of the TM operator on the fly and can generalize it on the whole input image.

We believe that several ways to evolve the colorimetric characterization model proposed in Chapter 7, are possible. For example, investigate different mathematical models for the estimation of the basis function parameters, and of the weights, introduce one or two more hidden layers in the structure of the RBFN, and finally perform experiments involving combinations of these new techniques. Different learning algorithms suitable for neural networks like Mixture Models [Bi96], can also be analysed. Mixture Models are a type of density models which comprise a number of component functions, usually Gaussian. These component functions are combined to provide a multimodal density. Mixture models are a semi-parametric alternative to non-parametric histograms (which can also be used as densities) [Bi96] and provide greater flexibility and precision in modelling the underlying statistics of the sample data. They can smooth over gaps resulting from sparse sample data and provide tighter constraints in assigning object membership to colour-space regions. Such precision is necessary to obtain the best possible results from colour-based pixel classification for qualitative segmentation requirements. Gaussian mixture models can also be viewed as a form of generalised radial basis function network in which each Gaussian component is a basis function or 'hidden' unit. The component priors can be viewed as weights in an output layer.

Bibliography

[Ad82]

E.H. Adelson.

Saturation and adaptation in the rod system. *Vision Research* , 22, 1299-1312, 1982.

[Ag79]

G. A. Agoston.

Color Theory and its Application in Art Design. Spring Verlag, 1979.

[Am84]

J. Amanatides.

Ray tracing with cones.

Computer Graphics (SIGGRAPH'84 Proceedings) **18**:3 (July 1984), pp. 129-135.

[Ar97]

A. Artusi.

Applicazione di algoritmi di apprendimento alla caratterizzazione colorimetrica di stampanti a colori. Tesi di laurea in Informatica, Univ. Degli studi di Milano, A.A 1996-97.

[Ar+01]

A. Artusi, C. Faisstauner and A. Wilkie.

A New Time-Dependent Tone Mapping Model. Technical Report TR-186-2-01-21, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria, November 2001.

<ftp://ftp.cg.tuwien.ac.at/pub/TR/02/TR-186-2-01-21Paper.pdf>

[ArWi02]

A. Artusi and A. Wilkie.

A New Realistic Tone Mapping Model. Technical Report TR-186-2-02-03, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria, November 2001.

<ftp://ftp.cg.tuwien.ac.at/pub/TR/02/TR-186-2-02-3Paper.pdf>

[Ar+03]

A. Artusi, J. Bittner, M. Wimmer and A. Wilkie.

Delivering Interactivity to Complex Tone Mapping operators. *Proceedings Eurographics Symposium on rendering EGSR03*, pp. 38-44, June 2003.

[ArWi01]

A. Artusi and A. Wilkie.

Colour Printer Characterization Using Radial Basis Function Networks. *Proc. SPIE Colour Imaging Conference: Device-Independent Colour, Colour Hardcopy, and Graphics Arts VI*, January 2001(San Jose California USA).

[ArWi03]

A. Artusi and A. Wilkie.

Novel Colour Printer Characterization Model.

Journal of Electronic Imaging, Volume 12, Issue 3, pp. 448-458 , July 2003.

[Be96]

R. S. Berns.

Methods for characterizing CRT displays. Displays Volume 16 Number 4 (1996).

[Bi96]

C. M. Bishop.

Neural Networks for Pattern Recognition. Calendor Press Oxford, 1996.

[Bi+91]

J. M. Bishop, M.J. Bushnell, and S. Westland.

Application of Neural Networks to the Computer Recipe Prediction. Color research and application, Volume 16, Number 1, 3-9, February 1991.

[Bl46]

H. R. Blackwell.

Contrast Thresholds of Human Eye.

Journal Opt. Soc. Am. 36, 11 (Nov.), pp. 624-643. 1946.

[Br+02]

D. H. Brainard, D. G. Peli and T. Robson.

Display Characterization. In Encyclopedia of Imaging Science and Technology. J. Hornak (ed.), Wiley: 172-188.

[CaRa99]

M. Carozza, and S. Rampone.

Function approximation from noisy data by an incremental RBF network. Pattern Recognition, volume 32, n. 12, 2081-2083, 1999.

[Co+01]

J. Cohen, C. Tchou, T. Hawkins, and P. Devebec.

Real-Time High-Dynamic Range Texture Mapping. graphics3.isi.edu/~jcohen/egwr-01-cohen.pdf, EGWR 2001.

[CoWa93]

M. F. Cohen and J. R. Wallace.

Radiosity and Realistic Image Synthesis. Accademic Press Professional, 1993.

[Co86]

R. L. Cook.

Stochastic sampling in computer graphics.

ACM transaction on Graphics **5**:1 (Jan. 1986), pp. 51-72.

- [Co+84]
R. L. Cook, T. Porter, and L. Carpenter.
Distributed ray tracing.
Computer Graphics (SIGGRAPH'84 Proceedings) **18**:3 (July 1984), pp. 137-145.
- [CoTo82]
R. L. Cook, and K. E. Torrance.
A reflection model for computer graphics.
ACM Transaction on Graphics **1**:1 (1982), pp. 7-24.
- [DeMa97]
P. E. Debevec and J. Malik.
Recovering High Dynamic Range Radiance Maps From Photographs.
Proceedings of SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series,
pp. 369-378 (August 1997, Los Angeles, California). Addison Wesley. Edited by Turner
Whitted. ISBN 0-89791-896-7.
- [DiWa00]
J. C. DiCarlo and B. A. Wandell.
Rendering high dynamic range images. SPIE conferences, 2000
- [Do87]
J. E. Dowling.
The Retina: An Approachable Part of the Brain. Cambridge:Belknap, 1987.
- [DuDo00]
F. Durand, and J. Dorsey.
Interactive Tone Mapping. Proceedings Eurographics Workshop on Rendering EGWR02,
2000.
- [DuDo02]
F. Durand, and J. Dorsey.
Fast Bilateral Filtering for the Display of High-Dynamic-Range Images. ACM Computer
Graphics (Proc. of SIGGRAPH)'2002, pp. 257—266, 2002.
- [Du87]
H. J. Durrett.
Color and the Computer.
Accademic Press Inc. Harcourt Brace Javanovich, Publishers, 1987.
- [Dr+03]
F. Drago, K. Myszkowski, T. Annen, and N. Chiba.
Adaptive Logarithmic Mapping For Displaying High Contrast Scenes. Eurographics EG2003.
- [EfTi93]
B. Efron, R. J. Tibshirani.
An Introduction to the Bootstrap. Chapman and Hall, 1993.

[FaWy98]

M. D. Fairchild, and D. Wyble.

Colorimetric Characterization of the Apple Studio Display (Flat Panel LCD).

Munsell Color Science Lab., Rochester Institute of Technology, Rochester NY, Technical report, July 1998.

[Fa98]

M. D. Fairchild.

Colour Appearance Models. Addison Wesley, 1998.

[Fa+02]

R. Fatah, D. Lischinski, and M. Werman.

Gradient Domain High Dynamic Range Compression. ACM Computer Graphics (Proc. of SIGGRAPH)'2002, pp. 249—256, 2002.

[Fe+96]

J. A. Ferwerda, S. N. Pattanaik, P. Shirley, and D. P. Greenberg.

A Model of Visual Adaptation for Realistic Image Synthesis. ACM Computer Graphics (Proc. of SIGGRAPH)'96, pp. 249--258, 1996.

[FoRo98]

A. Ford, and A. Roberts.

Colour Space Conversion.

www.poynton.com/PDFs/coloureq.pdf.

[Go+79]

G.H. Golub, M. Heat, and G. Wahba.

Generalized cross-validation as a method for choosing a good ridge parameters. Technometrics, 21(2), 215-223, 1979.

[Ha03]

http://www.math.iastate.edu/reu/2001/voronoi/halton_sequence.html

[He34]

S. Hecht.

Vision II: the nature of the photoreceptor process. In C. Murchison (Ed.). A handbook of general experimental psychology. Worcester, Massachusetts: Clark university Press 1934.

[He92]

R. P. Hemenger.

Source of intraocular light scatter from inversion of an empirical glare function. Applied Optics. Vol. 31 (19):3687--3693, 1992.

[Hu95]

R. W. G. Hunt.

The Reproduction of Colour. Chapter, Fountain Press, England, 1995.

[Hu84]

R. W. G. Hunt.

Colorimetry. IBA Technical review, n. 22, pp. 16-27, November 1984.

[JuWy75]

D. B. Judd, and G. Wyszecki

Colour in Business, Science and Industry. Jhon Wiley and Sons, 3rd Edition, 1975.

[Ka86]

J. T. Kajiya.

The rendering equation.

Computer Graphics (SIGGRAPH'86 proceedings) **20**:4 (Aug. 1986), pp. 143-150.

[KaAn92]

H. R. Kang, and P. G. Anderson

Neural Network Applications to the Colour Scanner and Printer calibrations.

Journal of Electronic Imaging, Vol. 1(2), pp. 125-135. April 1992.

[Ka+95]

J. M. Kasson, S. I. Nin, W. Plouffe, and J. L. Hafner.

Performing Color Space Conversions with Three-Dimensional Linear Interpolation.

Journal of Electronic Imaging, Vol. 4(3), pp. 226-250, (July 1995).

[Le98]

H. P. Le.

Progress and Trends in Ink-jet Printing Technology.

Journal of Imaging Science and Technology, Vol.(42), number 1, January/February 1998.

[Le+99]

Lee et al.

Robust radial basis function neural networks. IEEE trans, on systems, man and cybernetics, vol. 29, 674-685, 1999.

[LeLe]

R. Lenz, and U. Lenz.

The Marc project: colorimetric acquisition and printing of paintings with 20000x20000 pixels with an electronic camera.

SPIE vol. 2950.

[LuRi87]

M. R. Luo, and B. Rigg.

BFD(l:c) colour-difference formula, Part I-Development of the formula. J. Soc. Dyers. Col., 103(1987), pp. 86-94.

BFD(l:c) colour-difference formula, Part I-Performance of the formula. J. Soc. Dyers. Col., 103(1987), pp. 126-132.

[LuGu99]

M. R. Luo, and S. S. Guan.

Investigation of parametric effects using small colour differences.
Color Res. Appl., Vol. 24, pp. 331-343, 1999.

[Ma73]

C. Mallows .

Some comments on Cp. Technometrics, 15, 661-675, 1973.

[MaAb94]

G. Marcu, and S. Abe.

CRT and Ink Jet Printers Models for Device Independent Color Reproduction in Image Transmission. IS&T and SID's 2nd Color Imaging Conference: Color Science, System and Applications (1994), pp. 143-148.

[Ma+97]

K. Matkovic, L. Neumann, and W. Purgathofer.

A Survey of Tone Mapping Techniques.

13th Spring Conference on Computer Graphics, pp. 163-170. Comenius University, Bratislava, June 1997.

[Ma00]

Mathworks.

Matlab Function reference, <http://www.mathworks.com>, 2000.

[McSm95]

R. McDonald, and K. J. Smith.

CIE94 - a new colour-difference formula. Journal of Society of Dyers and Colour, Vol. 111, pp. 376-379, 1995.

[MoSp45]

P. Moon and D. Spencer.

The Visual Effect of Non-Uniform Surrounds. J. Optical Soc. Am., Vol. 35 (3):233-248, 1945.

[Mo96]

N. Moroni

Barcelona, HP Res. Labs, Personal Communication 1996.

[Or95]

M. J. L. Orr

Regularization in the selection of radial basis function centers. Neural Computation, 7(3), pp. 606-623, 1995.

[Nv03]

NVIDIA 2003. Graphics Hardware Specifications. <http://www.NVIDIA.com>

[Or96]

M. J. L. Orr.

Introduction to Radial Basis Function Networks. Center of Cognitive Science, University of Edinburgh, 1996.

[Pa+98]

S. N. Pattanaik, J. A. Ferwerda, M. Fairchild, and D. P. Greenberg.

A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display. ACM Computer Graphics (Proc. of SIGGRAPH)'98, 287--298, 1998.

[Pa+00]

S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg.

Time-Dependent Visual Adaptation For Fast Realistic Image Display. ACM Computer Graphics (Proc. of SIGGRAPH)'00, 47--53, 2000.

[Pa+c00]

S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg.

Time-Dependent Visual Adaptation For Fast Realistic Image Display. ACM Computer Graphics (Proc. of SIGGRAPH)'00, 47--53, 2000. Matlab code.

[Pc03]

<http://www.pctechguide.com/13inkjets.htm>

<http://www.pctechguide.com/12lasers.htm>

[Pe+92]

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery.

Numerical Recipes in C The Art of Scientific Computing. Second Edition, Cambridge University press, 1992.

[Po93]

C. A. Poyton.

“Gamma” and its Disguises: The Nonlinear Mappings of Intensity in Perception, CRTs, Film and Video. SMPTE Journal, pp. 1099-1108. December 1993.

[Pr+92]

W.H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery.

Numerical Recipes in C. Second Edition, Cambridge University Press, 1992.

[Ra88]

J.O. Rawlings.

Applied Regression Analysis. Wadsworth & Brooks/Cole, Pacific Grove, CA, 1988.

[RoBa93]

R. Rolleston, and R. Balasubramanian

Accuracy of various Types of Neugebauer Model.

IS&T and SID's Color Imaging Conference: Transforms & Transportability of Color, pp. 32-82.1993.

- [Sa02]
G. Sharma.
LCDs Versus CRTs Color Calibration and Gamut Considerations.
Proceedings of the IEEE, Vol. 90, No. 4, April 2002.
- [Sc+00]
A. Scheel, M. Stamminger, and H. P. Seidel.
Tone Reproduction for Interactive Walkthroughs. Proceedings Eurographics '2000, Vol. 19 (3) 2000.
- [Sc94]
Christophe Schlick.
Quantization Techniques for Visualization of High Dynamic Range Pictures. Proceedings Fifth Eurographics Workshop on Rendering 1994, 7--18, 1994.
- [Sc78]
G. Schwarz.
Estimating the dimension of a model. Annals of Statistics, 6, 461-464, 1978.
- [Sh37]
S. Shaler.
The Relation Between Visual Acuity and Illumination. Journal General Psychology, Vol. (21):165--188, 1937.
- [Si53]
G. Simpson.
Oculars Halos and Coronas. British Journal of Pothalmology 37, 450--486, 1953.
- [Sp90]
L. Spillman, and J.S. Werner.
Visual perception: the neurophysiological foundations. San Diego Accademic press 1990.
- [Sp+95]
G. Spencer, P. Shirley, K. Zimmerman, and D. P. Greenberg.
Physically-Based Glare Effects for Computer Generated Images. ACM Computer Graphics (Proc. of SIGGRAPH)'95, 325--334, 1995.
- [St98]
G. Starkweather.
Color Space Interchange Using sRGB.
<http://white.stanford.edu/~brian/psy221/reader/Starkweather.sRGBWhitePaper.pdf>.
- [StSt60]
S. S. Stevens, and J. C. Stevens.
Brightness Function: Parametric Effects of Adaptation and Contrast.
Journal Opt. Soc. Am. 50, 11 (Nov.) (Program of the 1960 Annual Meeting), 1139.

[StSt63]

J. C. Stevens, and S.S Stevens.
Brightness Function: Effects of Adaptation.
Journal Opt. Soc. Am. 53, 3 (March), pp. 375-385.

[St+96]

M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta.
A Standard Default Color Space for the Internet – sRGB. Version 1.10, November 5, 1996.
<http://www.w3.org/Graphics/Color/sRGB.html>

[Tr91]

D. Travis.
Effective Color Displays. Academic Press, 1991.

[TuRu93]

J. Tumblin, and H. Rushmeier.
Tone Reproduction for Realistic images.
IEEE Computer graphics and Applications, Vol. 13 (6):42--48, 1993.

[Tu+99]

J. Tumblin, J. K. Hodgins, and B. K. Guenter.
Two Methods for Display of High Contrast Images.
ACM Transaction on Graphics, Vol. 18, No. 1, January 1999, pp. 56-94, 1999.

[TuTu99]

J. Tumblin, and L. Turk.
LCIS: A Boundary Hierarchy For Detail-Preserving Contrast Reduction.
Proceedings ACM SIGGRAPH Conference 1999 Los Angeles, pp.83—90, 1999.

[UrMa97]

J. Uroz , and R. Marimon
Printing Reproduction of Fine-Art Paintings with the HP DesignJet 2500 CP large Format Printer.
Technical report HP Labs Barcelona, September 1997.

[Wa94]

G. Ward.
A Contrast-based Scale Factor for Luminance Display.
In P.S. Heckbert (Ed.), Graphic gems IV, Boston: Academic Press Professional, 1994.

[Wa+97]

G. Ward-Larson, H. Rushmeier, and C. Piatko.
A Visibility Matching Tone Reproduction Operator for High Dynamic range Scene. IEEE Transaction on Visualization and Computer Graphics, Vol. 3 (4):291--306, 1997.

[Wh80]

T. Whitted.
An improved illumination model for shaded display.
Communication of the ACM 23:6 (1980), pp. 343-349.

[WySt82]

G. Wyszecki, and W. S. Stiles.

Color Science: Concepts and Methods, Quantitative Data and Formulae.

Jhon Wiley and Sons, 2nd Edition, 1982.

[Xi+99]

M. Xia, E. Saber, G. Sharma, and A. M. Tekalp.

End-to-End Color Printer Calibration by Total least Squares Regression.

IEEE Transactions on Image Processing, Vol. 8, No. 5, pp. 700- 716, May 1999.