Linköping Studies in Science and Technology
Dissertation No. 810

Efficient Image Retrieval with Statistical Color Descriptors

Linh Viet Tran

INSTITUTE OF TECHNOLOGY
LINKÖPING UNIVERSITY

Department of Science and Technology
Linköping University, Norrköping, Sweden
May 2003

Efficient Image Retrieval with Statistical Color Descriptors

Department of Science and Technology
Campus Norrköping, Linköping University
SE 601-74 Norrköping
Sweden

ISBN 91-7373-620-1 ISSN 0345-7524
Printed in Sweden by UniTryck, May 2003

Abstract

Color has been widely used in content-based image retrieval (CBIR) applications. In such applications the color properties of an image are usually characterized by the probability distribution of the colors in the image. A distance measure is then used to measure the (dis-)similarity between images based on the descriptions of their color distributions in order to quickly find relevant images. The development and investigation of statistical methods for robust representations of such distributions, the construction of distance measures between them and their applications in efficient retrieval, browsing, and structuring of very large image databases are the main contributions of the thesis. In particular we have addressed the following problems in CBIR.

Firstly, different non-parametric density estimators are used to describe color information for CBIR applications. Kernel-based methods using non-orthogonal bases together with a Gram-Schmidt procedure and the application of the Fourier transform are introduced and compared to previously used histogram-based methods. Our experiments show that efficient use of kernel density estimators improves the retrieval performance of CBIR. The practical problem of how to choose an optimal smoothing parameter for such density estimators as well as the selection of the histogram bin-width for CBIR applications are also discussed.

Distance measures between color distributions are then described in a differential geometry-based framework. This allows the incorporation of geometrical features of the underlying color space into the distance measure between the probability distributions. The general framework is illustrated with two examples: Normal distributions and linear representations of distributions. The linear representation of color distributions is then used to derive new compact descriptors for color-based image retrieval. These descriptors are based on the combination of two ideas: Incorporating information from the structure of the color space with information from images and application of projection methods in the space of color distribution and the space of differences between neighboring color distributions. In our experiments we used several image databases containing more than 1,300,000 images. The experiments show that the method developed in this thesis is very fast and that the retrieval performance achieved compares favorably with existing methods. A CBIR system has been developed and is currently available at `http://www.media.itn.liu.se/cse`.

We also describe color invariant descriptors that can be used to retrieve images of objects independent of geometrical factors and the illumination conditions under which these images were taken. Both statistics- and physics-based methods are proposed and examined. We investigated the interaction between light and material using different physical models and applied the theory of transformation groups to derive geometry color invariants. Using the proposed framework, we are able to construct all independent invariants for a given physical model. The dichromatic reflection model and the Kubelka-Munk model are used as examples for the framework.

The proposed color invariant descriptors are then applied to both CBIR, color image segmentation, and color correction applications. In the last chapter of the thesis we describe an industrial application where different color correction methods are used to optimize the layout of a newspaper page.

Acknowledgements

Since I started my Ph.D. studies, many people have helped and supported me along the way. Without them, the thesis would obviously not have looked the way it does now.

The first person I would like to thank is my supervisor, Associate Professor Reiner Lenz. I have been working with him since November 1998 when I started my M.Sc. thesis. During these years he has helped and assisted me in many ways. His enthusiastic engagement in my research and his never-ending stream of ideas have been absolutely essential for the results presented here. I am very grateful that he has spent so much time with me discussing different problems ranging from philosophical issues down to minute technical details.

I would also like to express my gratitude to Professor Björn Kruse, my co-supervisor, for his support and assistance throughout my research. Colleagues in the media group at the Department of Science and Technology, Campus Norrköping provided an enjoyable working environments. My warm thanks are due to present and past members of the group for being good friends as well as helpful colleagues. In particular, special thanks to Arash Fayyazi, Daniel Nyström, Li Yang, Linda Johansson, Sasan Gooran, and Thanh Hai Bui for their friendship, support and many interesting discussions in both scientific matters and life, Helena Rosander Johansson and Sophie Lindesvik for administrative help, Peter Eriksson and Sven Franzén for technical support.

I wish to extend my sincere thanks to those who have read all or parts of previous versions of this thesis and made useful comments. Many thanks to Ivan Rankin for proof-reading and being helpful with comments and answering questions regarding my abuse of the English language. I am grateful to Professor Peter Meer for helpful discussions and for kindly providing the mean shift algorithm developed at the Robust Image Understanding Laboratory, Rutgers University. I thank Dr. Theo Gever, Dr. Harro Stokman, and their colleagues at the Intelligent Sensory Information Systems group, University of Amsterdam for useful discussions and being host of my visit in the group. My interest in color invariants started at the time I stayed in their group.

This work has been carried out within the VISIT (Visual Information Technology) program under the ”Content-based search in image and video databases” project, which is fully financial supported by the Swedish Foundation for the Strategic Research (SSF). Thanks must go to the SSF Foundation and the VISIT program for the funding.

I am thankful to all my friends, both here and there, for their encouragement and help, specially to Thanh's family: Thanh, Hà, Tý and Ti.

A final and heartfelt thanks go to my parents and my brother for everything they have given me.

In this printed version of the thesis, several figures are either not in color, or not printed well enough. The interested reader is recommended to view the **electronic version** of the thesis at `http://www.itn.liu.se/~lintr?thesis` and `http://www.itn.liu.se/publications/thesis`. **Demos and illustrations** of the projects described in the thesis as well as the **list of publications** which have been published can be also found at `http://www.itn.liu.se/~lintr?demo` and `http://www.itn.liu.se/~lintr?publication`.

Hồ Hoàn Kiếm, Lake of the Restored Sword, Hà nội, Việt Nam.

The cover page illustrates a three-dimensional RGB color histogram and a snapshot of the color-based search engine developed in the thesis. The background picture is a side view of Hoàn Kiếm lake in the center of my hometown. The name Hoàn Kiếm (Lake of the Restored Sword) originates from a legend. When King Lê Lợi was taking a dragon-shaped boat on the lake after the victory over foreign invaders, the Golden Tortoise Genie came out of the water to reclaim the sacred sword that had been given to him by the Dragon King to save the homeland. Since then, the lake has been called the Restored Sword Lake, or the Sword Lake for short. The Sword Lake is not only a beauty-spot, but also a historical site representing the spiritual heart of the capital. The tiny Tortoise Pagoda situating in the middle of the lake is often used as the emblem of Hà nội. This is also the kilometer zero marker from which all the roads in Việt Nam start.

Contents

Chapter 1

INTRODUCTION

1.1 Motivation

Recent years have seen a rapid increase in the size of digital image collections together with the fast growth of the Internet. Digital images have found their way into many application areas, including Geographical Information System, Office Automation, Medical Imaging, Computer Aided Design, Computer Aided Manufacturing, Robotics. There are currently billions of web pages available on the Internet using hundreds of millions (both still and moving) images (Notes, 2002). However, we cannot access or make use of the information in these huge image collections unless they are organized so as to allow efficient browsing, searching, and retrieval over all textual and image data.

The straightforward solution to managing image databases is to use existing keyword or text-based techniques. Keywords are still a quite common technique to provide information about the content of a given database, but to describe the images to a satisfactory degree of concreteness and detail, very large and sophisticated keyword systems are needed. Another serious drawback of this approach is the need for well-trained personnel not only to annotate keywords to each image (which may take up to several minutes for one single image, and several years for a large image database) but also to retrieve images by selecting good keywords. These manual annotations are highly time-consuming, costly, and dependent on the subjectivity of human perception. That is, for the same image content, different (even well-trained) people may perceive the visual content of the images differently. The perceptional subjectivity and the annotation impreciseness may cause unrecoverable mismatches in later retrieval processes. Furthermore, a keyword-based system is very hard to change afterwards. Therefore, new approaches are needed to overcome these limitations.

Content-based image retrieval represents a promising and cutting-edge technology to address these needs. The fundamental idea of this approach is to generate automatically image descriptions directly from the image content by analyzing the content of the images. Such techniques are being developed by many research groups and commercial companies around the world. Financially supported by the Swedish Foundation for Strategic Research (SSF), the VISIT[1] (VISual Information Technology) program has been running one such project in which we were involved.

Given a query image, a content-based image retrieval system retrieves images from the image database which are similar to the query image. In a typical situation, all the images in the database are processed to extract the selected features that represent the contents of the images. This is usually done automatically once when the images are entered into the database. This process assigns to each image a set of identifying descriptors which will be used by the system later in the matching phase to retrieve relevant images. The descriptors are stored in the database, ideally in a data structure that allows efficient retrieval in the later phase.

Next a query is posted in the matching phase. Using the same procedures that were applied to the image database the features for the query image are extracted. Image retrieval is then performed by a matching engine, which compares the features or the descriptors of the query image with those of the images in the database. The matching mechanism implements the retrieval model adopted according to the selected metric, or similarity measure. The images in the database are then ranked according to their similarity with the query and the highest ranking images are retrieved. Efficiently describing the visual information of images and measuring the similarity between images described by such pre-computed features are the two important steps in content-based image retrieval.

Recent efforts in the field have focused on several visual descriptors to describe images, such as color, texture, shape, and spatial information of which color is the most widely used feature for indexing and retrieving images since it is usually fast, relative robust to background, small distortions, and changes of image size and orientation.

[1] Detailed information about our project, the VISIT (VISual Information Technology) program, our sponsors, and partners can be found at the VISIT homepage, http://visit.cb.uu.se.

Everyone knows what color is, but the accurate description and specification of color is quite another story. Color has always been a topic of great interest in various branches of science. Despite this, many fundamental problems involving color, especially in human color perception where brain activities play an important role, are still not fully understood. Low-level properties of human color perception are, however, successfully modelled within the colorimetric framework. In this framework, we see that statistical methods are powerful tools for describing and analyzing such huge datasets of images. In the thesis we describe in particular our research in the application of color-based features for content-based image retrieval[2]. Other visual features such as texture and shape, as well as other topics like multi-dimensional indexing techniques, system design, query analysis, user interface, are beyond the scope of the thesis.

1.2 Contributions of the Thesis

The proposed techniques in this thesis can be classified as statistics-based methods to improve retrieval performance for color-based image retrieval (CBIR) applications. Specifically, the following four problems are discussed in the thesis:

Estimating color distributions: In CBIR applications the color properties of an image are characterized by the probability distribution of the colors in the image. These probability distributions are very often approximated by histograms (Rui et al., 1999; Schettini et al., 2001). Well-known problems with histogram-based methods are: the sensitivity of the histogram to the placement of the bin edges, the discontinuity of the histogram as a step function and its deficiency of using data in estimating the underlying distributions compared to other estimators (Silverman, 1986; Scott, 1992; Wand and Jones, 1995). These problems can be avoided by using other methods such as kernel density estimators. However, our experiments have shown that straightforward application of kernel density estimators in CBIR provides unsatisfactory retrieval performance. Using good density estimators does not guarantee good retrieval performance (Tran and Lenz, 2003a). This explains why there are few papers using kernel density estimators in CBIR[3]. To improve the retrieval performance of CBIR applications, we propose two different kernel-based methods. These new

[2] The CBIR abbreviation is widely used for both ”content-based image retrieval” and ”color-based image retrieval” terms. To distinguish between them, we will state the meaning before use the CBIR abbreviation.

[3] We found only one paper (Gevers, 2001) using kernel-based methods for reducing noise in CBIR. However, the experiments described in the paper used a very small database of 500 images of several objects taken under different combinations of changing light sources and camera view points.

methods are based on the use of non-orthogonal bases together with a Gram-Schmidt procedure and a method applying the Fourier transform. Our experiments show that the proposed methods performed better than traditional histogram-based methods. Fig. 1.1 illustrates one of our results.

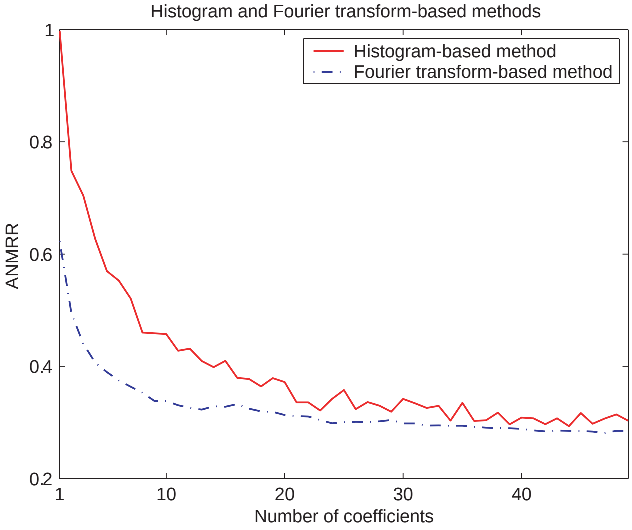


Figure 1.1: Retrieval performance of the histogram and Fourier transform-based method using triangular kernel. The detailed description of ANMRR will be described in chapter 3. Briefly, the lower values of ANMRR indicate better retrieval performance, 0 means that all the ground truth images have been retrieved and 1 that none of the ground truth images has been retrieved.

Like other density estimators, the histograms and kernel density estimators are both sensitive to the choice of the smoothing parameter (Silverman, 1986; Scott, 1992; Wand and Jones, 1995). This parameter in turn influences the retrieval performance of CBIR applications. Such influences are investigated in (Tran and Lenz, 2003c) for both histogram-based and kernel-based methods. Particularly for histogram-based methods, we show that the previously applied strategy (Brunelli and Mich, 2001) of applying statistical methods to find the theoretically optimal number of bins (Sturges, 1926; Scott, 1979; Rudemo, 1982; Scott, 1985; Devroye and Gyorfi, 1985; Scott, 1992; Kanazawa, 1993; Wand, 1996; Birge and Rozenholc, 2002) in image retrieval applications requires further research.

Distance measures between color distributions: We investigated a new differential geometry-based framework to compute the similarity between color distributions (Tran and Lenz, 2001c; Tran and Lenz, 2003b). This framework allows us to take the properties of the color space into account. The framework is theoretically of interest since many other similarity measures are special cases of it. Some examples to illustrate the general framework are also presented.

Compressing feature space: An efficient implementation of a content-based image retrieval system requires a drastic data reduction to represent the content of images since current modern multi-dimensional indexing techniques only work efficiently when the dimension of the feature space is less than 20 (Weber et al., 1998; Rui et al., 1999; Ng and Tam, 1999; Schettini et al., 2001).

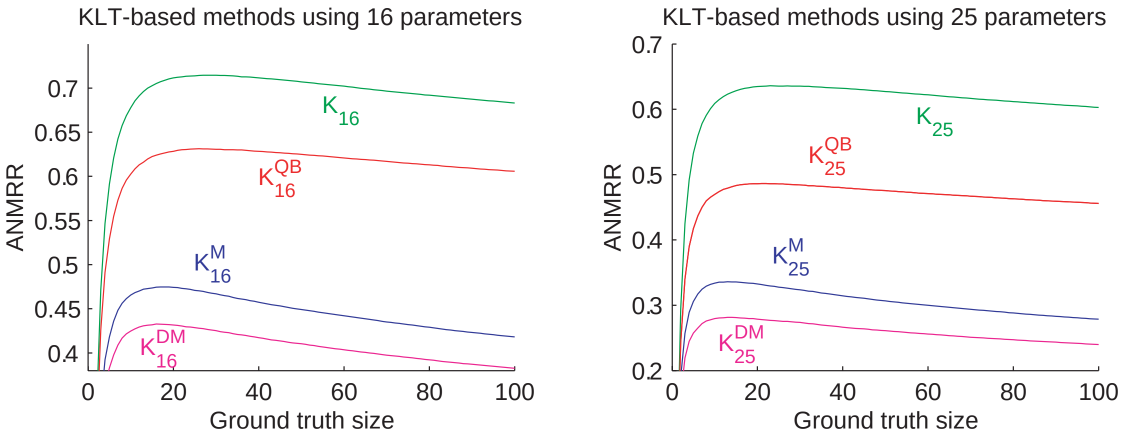


Figure 1.2: ANMRR of 5,000 queries from the Matton database of 126,604 images using different KLT-based histogram compression methods compared to the full histogram-based method. 5,000 query images were selected randomly outside the training set.

It is well-known that the optimal way to reduce the dimension of feature vectors is the Karhunen-Loève Transform (KLT). It is optimal in the sense of minimizing the mean squared error of the L_2-distance between the original and the approximated vectors. However, a straightforward application of the KLT to color feature vectors gives poor results since KLT treats the color feature vector as an ordinary vector and ignores the properties of the underlying color distribution. Also the properties of image retrieval applications where we are only interested in similar images were not considered previously. Therefore we introduced several KLT-based representation methods for color distributions (Tran and Lenz, 2001b; Tran and Lenz, 2002b) which are based on two ideas: application of KLT on a metric which utilizes color properties, and KLT on the space of local histogram differences in which only similar images are considered in the compression process. The experiments on different image databases ranging from one thousand to more than one million images show that the method developed using both the ideas described above is very fast and that the retrieval performance achieved compares favorably with existing methods. Fig. 1.2 shows an example of the superior performance of our proposed method K^{DM} over other methods.

Color invariants: The color image (either captured by a camera or scanned by a scanner) depends at least on the following factors: the physical properties of the scene, the illumination, and the characteristics of the camera. This leads to a problem for many applications where the main interest is in the content of the scene. Consider, for example, a computer vision application which identifies objects by color. If the colors of the objects in a database are specified for tungsten illumination (reddish), then object recognition can fail when the system is used under the very blue illumination of blue sky. This happens because the change in the illumination alters object colors far beyond the tolerance required for reasonable object recognition. Thus the illumination must be controlled, determined, or otherwise taken into account. Shadows, highlights, and other effects of geometry changes are also sources of problems in many applications. A typical unwanted problem in segmentation is that objects with complicated geometry are usually split into many small objects because of shadowing and highlight effects.

Color features which are invariant under such conditions are often used in many applications. Both physics-based (Tran and Lenz, 2003d; Lenz et al., 2003b) and statistics-based (Lenz et al., 1999; Lenz and Tran, 1999) methods are investigated in the thesis. The proposed physics-based methods use the dichromatic reflection model and the Kubelka-Munk model. They are derived mainly for invariants against geometry changes using the theory of transformation groups. Using the proposed framework, all independent invariants of a given physical model can be constructed by using standard symbolic mathematical software packages. The invariant features, however, are quite noisy because of the quantization error and few unrealistic assumptions of the underlying physical processes. A robust region-merging algorithm is proposed to reduce the effect of noise in color image segmentation application. Fig. 1.3 shows an example of the segmented results by the proposed robust region-merging method.

The proposed statistical method is based on the normalization of moments of image. Many statistics-based color constancy methods assume that the effect of an illumination change can be described by a matrix multiplication with a diagonal matrix. Here we investigate the general case of using a full 3×3 matrix. This normalization procedure is a generalization of the channel-independent color constancy methods since general matrix transformations are considered.

All these methods are then used in the following applications:

Color-based image retrieval: In order to evaluate the retrieval speed and performance of different representation methods and similarity measures in image retrieval, we implemented a color-based image retrieval system for both web-based and stand-alone applications. The web-based version is available at `http://www.media.itn.liu.se/cse`. An example of the search results from the demo using a database of 126,604 images is illustrated in Fig. 1.4.

The size of the image database is very important when comparing different methods in image retrieval. One algorithm might work well for a small set of images, but totally fail when applied to a large database. For a realistic comparison, a database of a few hundred images does not seem good enough because the retrieval results are probably similar for different methods. The properties of the images in the database also affect the performance of different algorithms.

In our experiments, we have used different image databases of different size and contents. The following four different image databases of totally more than 1,300,000 images are used:

Corel database: consists of 1,000 color images (randomly chosen) from the Corel Gallery.

MPEG-7 database: consists of 5,466 color images and 50 standard queries (Zier and Ohm, 1999). The database is designed to be used in the MPEG-7 color core experiments.

Matton database: consists of 126,604 color images. These images are low-resolution images of the commercial image database[4] maintained by Matton AB in Stockholm, Sweden.

TV database: consists of 1,058,000 color images, which are grabbed from over 2 weeks video sequences of MTV-Europe and BBC-World TV channels (one frame is captured every second). Fig. 1.5 shows an example of the search results on the TV database.

BabyImage project: The investigated color constancy and color normalization methods are applied in an industrial color correction project. This project was done in cooperation with the "Östgöta Correnspondenten" daily newspaper published in Linköping in which we show that a simple application of conventional, global color constancy and color normalization algorithms produces poor results. Segmenting the images into

[4] Text-based search of the image database is available at `http://www.matton.se`, and color-based search is at `http://www.media.itn.liu.se/cse`. The color-based search on the TV database using more than one million images is also available here.

relevant regions and applying local correction algorithms lead to much better results. In Fig. 1.6 some of these results of the new method are illustrated. The figure shows two color images taken under two different conditions, corresponding to the left and the middle image. The middle one was then corrected resulting in the right image so that it should look similar to the left one.

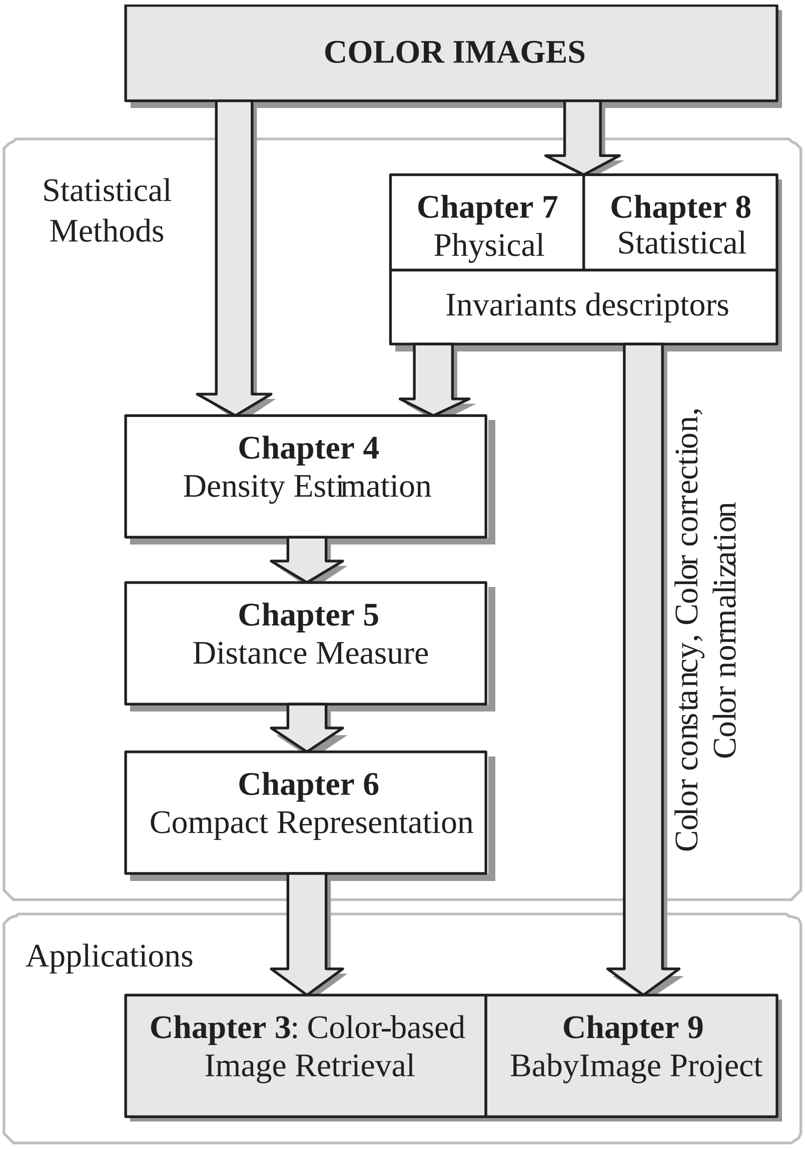

COLOR IMAGES
Statistical Methods
Chapter 7 Physical
Chapter 8 Statistical
Invariants descriptors
Chapter 4 Density Estimation
Chapter 5 Distance Measure
Chapter 6 Compact Representation
Color constancy, Color correction, Color normalization
Applications
Chapter 3: Color-based Image Retrieval
Chapter 9 BabyImage Project

Figure 1.7: Thesis outline.

1.3 Thesis Outline

The thesis consists of 10 chapters. The background information and literature review are briefly covered in the next two chapters. Basic facts about color are summarized in chapter 2. The chapter provides a background on how color images are formed, how colors are described in digital images, and which factors influence the color properties of images. Chapter 3 reviews some background material on content-based image retrieval. It describes features useful for content-based image retrieval and investigates similarity measures between color images based on such pre-computed features.

The contributions of the thesis are presented from chapter 4 to chapter 9. Briefly, chapter 4 presents our investigations in estimating color distributions for image databases. The topic of how to measure the distances between such distributions is discussed in chapter 5, in which we develop a new similarity measure of color distributions based on differential geometry. Chapter 6 presents several new KLT-based compression methods for representing color features in CBIR. Chapter 7 deals with physics-based color invariants using different physical models while the moment-based color image normalization, is presented in chapter 8. Chapter 9 describes the BabyImage project, which is an application of the color correction, color constancy and color normalization methods discussed in chapter 8.

Finally, conclusions and future work is presented in chapter 10. A summary of the thesis layout is illustrated in Fig. 1.7

Chapter 2

FUNDAMENTALS ON COLOR

Perhaps everyone knows what color is, but the accurate description and specification of color is quite another story. Color as a science is still fairly young however it involves many different branches of science such as material science, physics, chemistry, biological science, physiology, psychology. This chapter presents a very brief description on fundamentals of color which are of interest for the rest of the thesis.

2.1 Physical Basis of Color

Objects are visible only because light from them enters our eyes. Without light nothing can be seen. However, ”The rays, to speak properly, are not colored; in them there is nothing else than a certain power and disposition to stir up a sensation of this or that color” – as Sir Isaac Newton said. Thus it is important to understand that color is something we humans impose on the world. The world is not colored, we just see it that way. This entails that the task of defining the word color provides interesting challenges and difficulties. Even the most dedicated color scientists who set out to write the *International Lighting Vocabulary* could not write down a very satisfactory definition[1].

[1] Details of the definition from the *International Lighting Vocabulary* and discussion about it can be found in (Fairchild, 1997)

A simple and reasonable working definition of color which is used in the thesis is the following: Color is our human response to different wavelengths of light[2].

In everyday language we speak of "seeing"[3] objects, but of course it is not the objects themselves that we see. What we see is light that has been reflected from, transmitted through, or emitted by objects. For instance, though it is something of a simplification, one can say that an object that appears blue reflects or transmits predominately blue light. The object may be absorbing other wavelengths of light, or the available light may be primarily in the wavelengths we recognize as blue, but the final result is that the object appears blue. Color, therefore, can be seen as the result of the interaction of three elements: an illuminant (a light source), an object, and an observer (the person who experiences the color). The following sections discuss the above mentioned three elements as well as other factors that influence the process of forming color.

2.2 Light Sources

As we have mentioned earlier, without a light source, there is nothing to see. So what is light? There are several ways to think of light. The classical description says light is an electromagnetic wave. This means that it is a varying electric and magnetic field, which spreads out or propagates from one place to another. This wave has amplitude, which tells us the brightness of the light, wavelength, which tells us about the color of the light, and an angle at which it is vibrating, called polarization. The modern quantum mechanical description, however, says that light can also be considered to be particles called photons. These carry energy and momentum but have no mass. Both descriptions are correct and light has both wave-like and particle-like properties.

Light covers a broad range of phenomena with sound and radio waves at one end and gamma rays at the other. Visible light which is our main interest, is somewhere towards the middle of this spectrum tucked in between infrared waves and ultra violet waves ranging from about 380nm to 780nm, see Fig. 2.1.

Light can be produced by a variety of methods. The most widely occurring light sources are incandescence, which is the method of emitting light by heating

[2] One should note the corollary of this definition that we *"can not"* really measure color itself. When we talk about *"measuring color"* what we are really measuring is not any inherent quality or even our response to various wavelengths of light (someday we may be able to measure the electro-chemical signals in the brain and directly connect them with color term, but that days seems far off), but rather the stimulus that creates it

[3] Perception is not only a passive measurement of the incoming signals. New results from brain research show that perception is a process in which the brain actively analyzes information. It is probably more accurate to say that we see with our brain than to say we see with our eyes. A recent overview over some relevant facts is (Zeki, 1999).

an object. It has been known that solids and liquids emit light when their temperatures are above about 1000K. The amount of power radiated depends on the temperature of the object. Correlated color temperature, CCT[4], of the object can be used to describe the spectral properties of the emitted light. For example, direct sunlight has a CCT of about 5500K. Typical indoor daylight has a CCT of about 6500K. Some examples of daylights are illustrated in Fig. 7.13.

Tungsten lamps are other examples of incandescent light sources, but their CCTs are much lower than that of daylight. Typical tungsten filament lamps have a CCT of about 2600-3000K. Light can also be produced by letting electric current pass through gases, or certain semiconductors, phosphors.

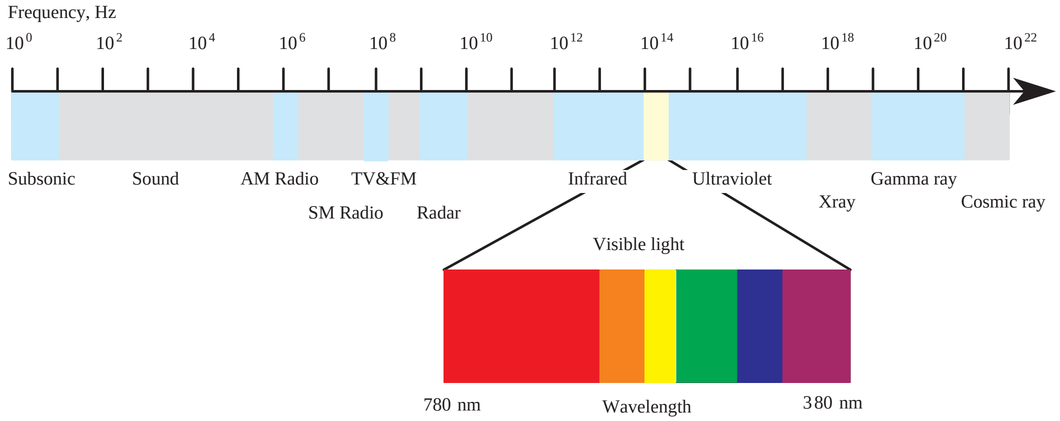


Figure 2.1: Classification of the electromagnetic spectrum with frequency and wavelength scales.

Fig. 2.2 shows the spectral power distributions of three light sources: a Sylvania Cool White Fluorescent tube light[5], which is a typical white neon light, the CIE (Commission International de l'Éclairage or International Commission on Illumination) illuminant D65, which is a mathematical representation of a phase of daylight having a CCT of 6504 K, and the CIE illuminant A, which is a mathematical representation of tungsten halogen (incandescent) having a CCT of 2856K. Clearly the CIE illuminant A has more radiant power in the red region compare to the CIE illuminant D65, thus its color should be warmer than the color of the CIE illuminant D65.

[4] Correlated color temperature is the temperature of the Planckian radiator whose perceived color most closely resembles that of a given stimulus seen at the same brightness and under specified viewing conditions

[5] The spectral data of the Sylvania Cool White Fluorescent tube light source was measured at the Computer Science Laboratory, Simon Fraser University, Vancouver, Canada, http://www.cs.sfu.ca/research/groups/Vision/, see (Funt et al., 1998) for a detailed description.

2.3 Objects

When the illumination light reaches an object (or surface), many complicated processes will occur. These processes can basically be divided into two different classes. The first class is related to the discontinuities of optical properties at the interface such as reflection, surface emission, etc. and the second class is volume-related and depends on the optical properties of the material of the object. A brief summary of the most important processes are given below.

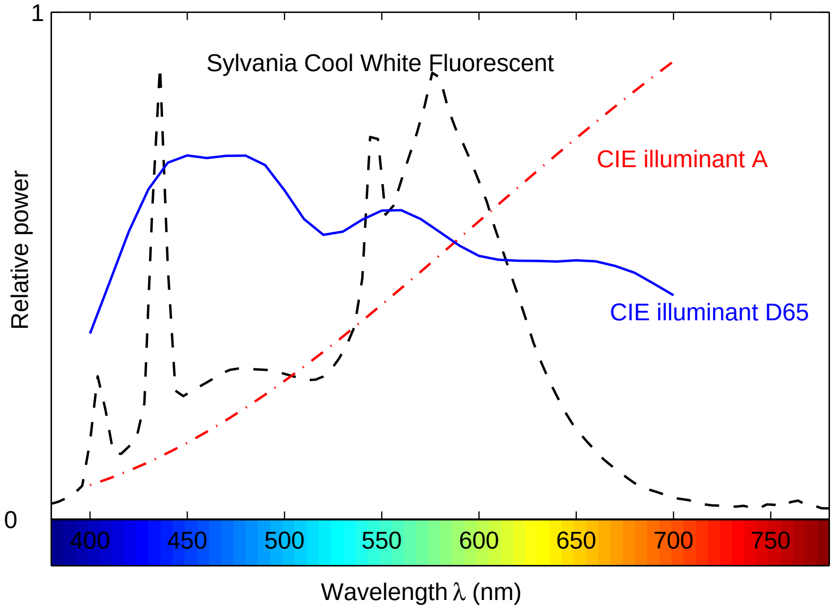


Figure 2.2: The relative spectral power distributions for a Sylvania Cool White Fluorescent (dash line), the CIE illuminant D65 (solid line) and the CIE illuminant A (dash-dot line) light sources. The curves describe the relative power of each source's electromagnetic radiation as a function of wavelength.

Reflection : When light hits the surface of an object, it must pass through the interface between the two media, the surrounding medium and the objects. Since the refractive indices of the two media are generally different, part of the incident light is reflected at the interface. It behaves like a mirror, meaning that the angle of reflectance is equal to the angle of incidence. The reflected ray and the normal of the surface lie in one plane. The ratio of the reflected radiant flux to the incident at the surface is called the reflectivity and it depends on the angle of incidence, the refractive indices of the two media meeting at the interface and the polarization state of the radiation.

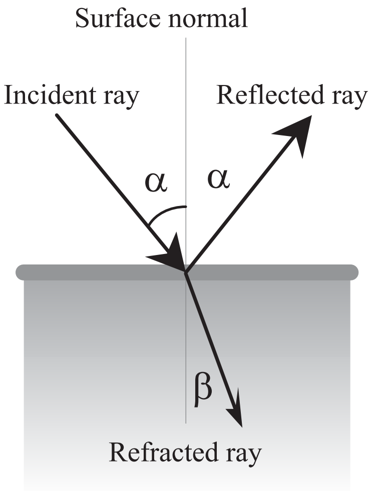


Figure 2.3: When a ray of light hits the interface between two optical media with different index of refraction, part of the incident light is reflected back. The other part transfers into the medium, and its direction is changed at the interface.

Refraction, Absorption, Scattering : For many materials such as dielectric materials, not all incident light is reflected at the interface, part of it penetrates into the object. See Fig. 2.3. When travelling inside the medium, the light hits pigments, fibers or other particles from time to time. It is either absorbed and converted into different energy forms, or scattered in different directions. The light keeps hitting particles and is increasingly scattered until some of it arrives back at the surface. Some fraction of the light then exits from the material while the rest is reflected back, see Fig. 7.6.

Thermal Emission : Emission of electromagnetic radiation occurs at any temperature. The cause of the spontaneous emission of electromagnetic radiation is thermal molecular motion, which increases with temperature. During emission of radiation, thermal energy is converted to electromagnetic radiation and the object cools down.

Depending on the chemical and physical properties of the object and other factors, the amount of light that is reflected back (which, in this case, might consist of reflected light directly from the interface, reflected light inside the object, or light emitted from the objects) will vary at different wavelengths. This variation is described in terms of the spectral reflectance (or the spectral transmittance) characteristics of the object. The color of the object can be defined on the basic of such spectral properties.

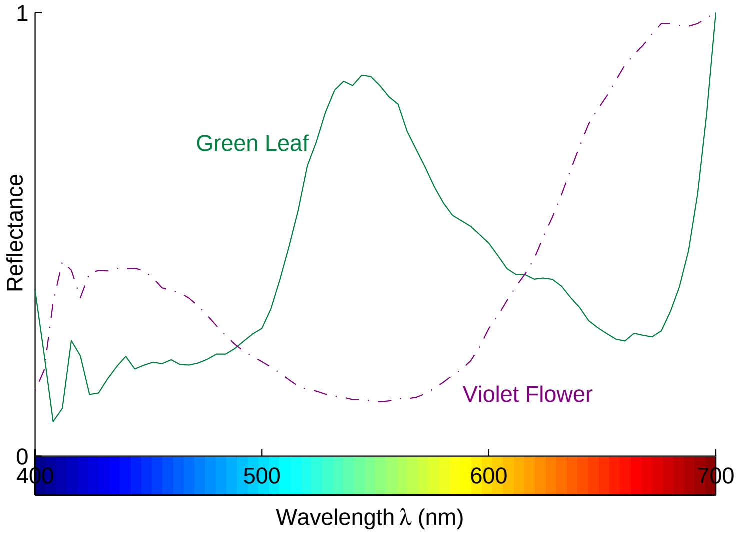


Figure 2.4: Spectral reflectance of a green leaf and a violet flower.

As examples, the spectral reflectance of a green leaf and a violet flower[6] are shown in Fig. 2.4. The green leaf reflects the light mainly in the green region while the violet flower reflects the light in the red and blue regions.

The light that enters a sensor is called a color stimulus. For example, when the violet flower characterized by the spectral reflectance in Fig. 2.4 is illuminated with the Sylvania Cool White Fluorescent or the CIE standard C light sources as shown in Fig. 2.2, the color stimuli will have the spectral power distributions shown in Fig. 2.5 and Fig. 2.6. The spectral power distribution of this stimulus is the product of the spectral power distribution of the light source and the object. It is calculated by multiplying the power of the light source and the reflectance of the object at each wavelength.

2.4 Human Color Vision

To be able to describe color, we need to know how people respond to light. Our eyes contain two types of sensors, rods and cones, that are sensitive to light. The rods are essentially monochromatic, with a peak sensitivity at around 510nm. They contribute to peripheral vision and allow us to see in relatively dark conditions. But they do not contribute to color vision. You have probably noticed that on a dark night, even though you can see shapes and movement, you see very little color.

[6] The spectral data of two objects, a green leaf and a violet flower, was measured at the Department of Physics, University of Kuopio, Finland, see (Parkkinen et al., 1988) for a detailed description.

The sensation of color comes from the second set of photo-receptors in our eyes, the cones. Our eyes contain three different types of cones, which are most properly referred to as the L, M, and S cones, denoting cones sensitive to light of long wavelength (having maximal sensitivity at 575nm), medium wavelength (535nm), and short wavelength (445nm), respectively.

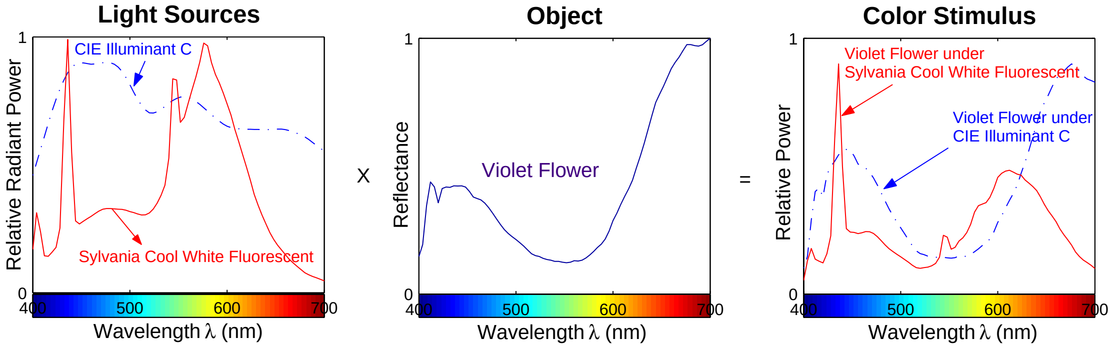


Figure 2.5: Spectral power distributions of a violet flower, illuminated with two difference light sources: the Sylvania Cool White Fluorescent and the CIE standard C light source.

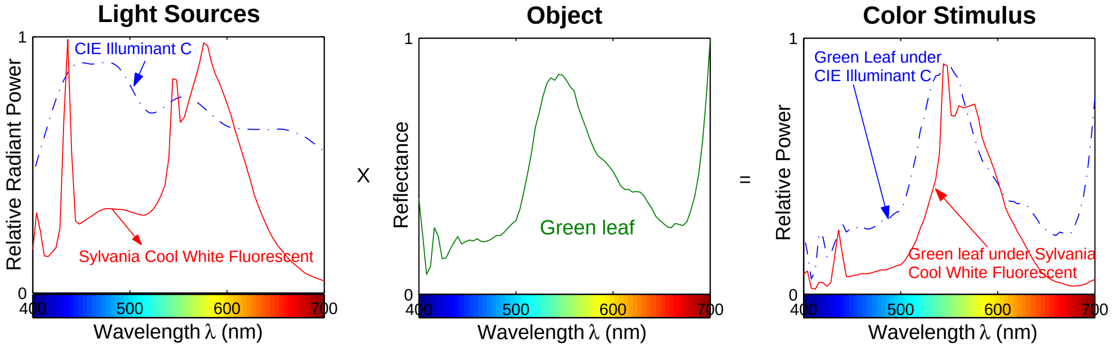


Figure 2.6: Spectral power distributions of a green leaf, illuminated with two difference light sources: the Sylvania Cool White Fluorescent and the CIE standard C light source.

The cones respond to light in a complex manner in which our brain is actively involved. This process does not only simply receive the signal from each cone, but also compares each signal to that of its neighbors, and assigns feedback weighting to the raw signals. One reason why such weighting is necessary is that we have many more L and M cones than S cones. The relative population of the L, M, and S cones is approximately 40:20:1. Many other complicated processes have happened before the concept of color is formed in our brain.

Many of such processes are still not fully understood (Fairchild, 1997). More facts and information about the human vision system can be found in many books, for example (Wyszecki and Stiles, 1982; Zeki, 1999)

2.5 Color Image Formation

The process of how digital color images, which are taken by a digital camera, or scanned by a scanner, are formed, however, much more easier to understand. The color stimulus reaches the sensors of the camera and is recorded here. The spectral characteristics of the sensors inside the camera, or the sensitivity functions of the sensors are the most important properties of the camera.

Mathematically, we could formulate (in a simplified way) the process of how a color image is formed inside a camera as follows: We denote the light energy reaching a surface by $E(\lambda)$ where λ is the wavelength. For a given scene and viewing geometry, the fraction of total light reflected back or transmitted through the object is denoted by $R(\lambda)$. A vision system then samples image locations with one or more sensor types. In our case, the locations are simply image pixels, and the sensor types are the red, green, and blue camera channels. The response of the i^{th} sensor, $\rho_i(x, y)$, is often modelled by (given the sensor response functions $f_i(\lambda)$)

$$\rho_i(x, y) = k \int_\lambda f_i(\lambda) R(x, y, \lambda) E(\lambda) d\lambda \tag{2.1}$$

where k is a normalization factor.

Here we assumed that the optoelectronic transfer function of the whole acquisition system is linear. This assumption based on the fact that the CCD sensor is inherently a linear device. However, for real acquisition systems this assumption may not hold, due for example to electronic amplification nonlinearities or stray light in the camera. Appropriate nonlinear corrections may be necessary (Maître et al., 1996).

This model can also be assumed for the human visual system, see for example (Wyszecki and Stiles, 1982), and forms the basis for the CIE colorimetry standard. Fig. 2.7 shows a (simplified) example of how the sensor responses are computed using Eq. 2.1.

Eq. 2.1 describes a very simple model of how the recorded image depends on the physical properties of the scene, the illumination incident on the scene, and the characteristics of the camera. This dependency leads to a problem for many applications where the main interest is in the physical content of the scene. Consider, for example, the color-based image retrieval application to search similar objects by color. If the images in a database are taken under tungsten illumination (reddish), then the search could fail when the system is used under the very blue illumination of sunlight. Such a change in the illumination affects colors of images far beyond the tolerance required for retrieval methods based on raw color comparison. Thus the illumination must be controlled, determined, or at least taken into account in this case. This topic will be discussed in more detail in chapters 7 and 8.

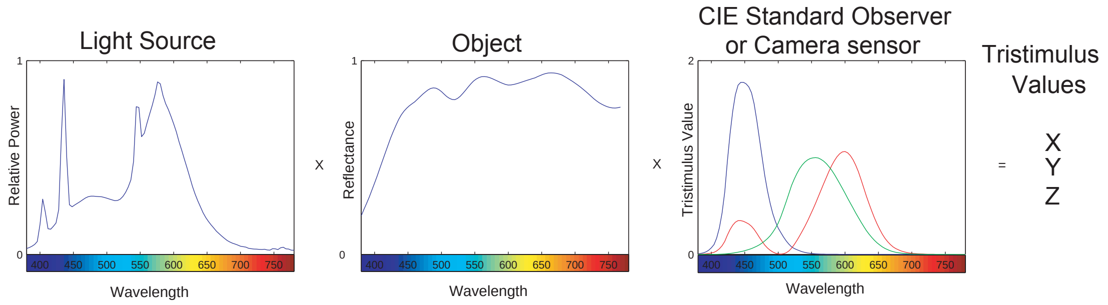


Figure 2.7: How colors are recorded.

2.6 Color Spaces

The space of color spectra reaching a point on the retina is high-dimensional. However the color vision system of most human beings consists of three independent color receptors. The visual system thus maps a high-dimensional input, the spectral distribution of light, onto a low-dimensional (three dimensions) output where each point in the visual scene is assigned one color. Obviously, information is being lost in the process, but it seems reasonable that the visual system is attempting to preserve as much of the information (in some sense) as possible. A discussion of this topic (the connection between the statistics of natural scenes and the properties of human perception) is beyond the framework of this thesis. Interested readers can find a good introduction to the current discussion in (Willshaw, 2001). Here we just discuss properties of some projection methods, or color spaces, which are used in the thesis.

RGB Color Space: The most popular color space is RGB which stands for Red-Green-Blue. This is a device-dependent color space[7] and normally

[7] Recently, Hewlett-Packard and Microsoft proposed the addition of support for a standard color space, sRGB which stand for standard RGB (Stokes et al., 2000). The goal of sRGB is to develop a simple solution that solves most of the color communication problems for office, home and web users, by which sRGB is a device-independent color space. More information can be found on (Süsstrunk et al., 1999), `http://www.srgb.com` or `http://www.w3.org/Graphics/Color/sRGB.html`

used in Cathode Ray Tube (CRT) monitors, television, scanners, and digital cameras. For a monitor the phosphor luminescence consists of additive primaries and we can simply parameterize all colors via the coefficients (α, β, γ), such that $C = \alpha R + \beta G + \gamma B$. The coefficients range from zero (no luminescence) to one (full phosphor output). In this parametrization the color coordinates fill a cubical volume with vertices black, the three primaries (red, green, blue), the three secondary mixes (cyan, magenta, yellow), and white as in Fig. 2.8.

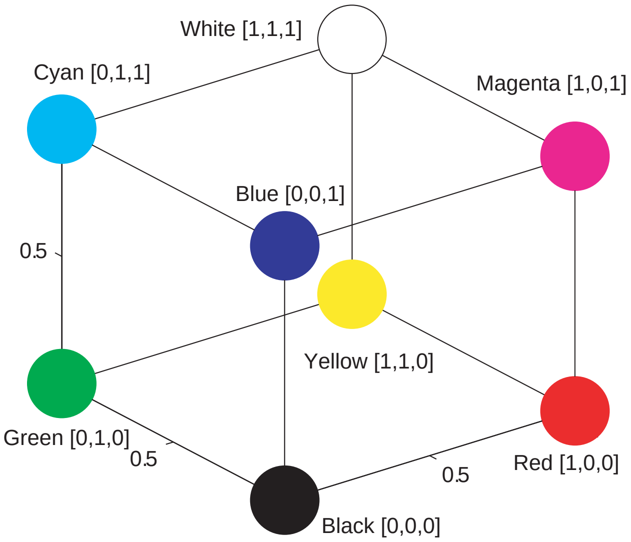


Figure 2.8: RGB Color spaces.

There are many different variations of RGB spaces; some of them were developed for specific imaging workflow and applications, others are standard color spaces promoted by standard bodies and/or the imaging industry. However they share the following important points:

- They are perceptually non-linear. Equal distances in the space do not in general correspond to perceptually equal sensations. A step between two points in one region of the space may produce no perceivable difference while the same increment in another region may result in a noticeable color change.
- Because of the non-linear relationship between RGB values and the intensity produced, low RGB values produce small changes. As many as 20 steps may be necessary to produce a JND (Just Noticeable Difference) at low intensities whereas a single step at high intensities may produce a perceivable difference.

- This is not a good color description system. Without considerable experience, users find it difficult to give RGB values of colors. What is the RGB value of ”medium brown”. Once a color has been chosen, it may not be obvious how to make subtle changes to the nature of color. For example, changing the ”vividness” of a chosen color will require unequal changes in the RGB components.

HSV and HSL Color Spaces: The representation of the colors in the RGB space is adapted for monitors and cameras but difficult to understand intuitively. For color representation in user interfaces, the HSV and HSL color spaces are usually preferred. Both models are based on the color circle mapped on the RGB cube: the edge progression that visits the vertices Red, Yellow, Green, Cyan, Blue, Magenta in this cyclical order. When the RGB cube is seen along the gray direction, this edge progression appears like a regular hexagon which has the structure of the classical color circle. The difference between the two models is the definition of the white points as illustrated in Fig. 2.9

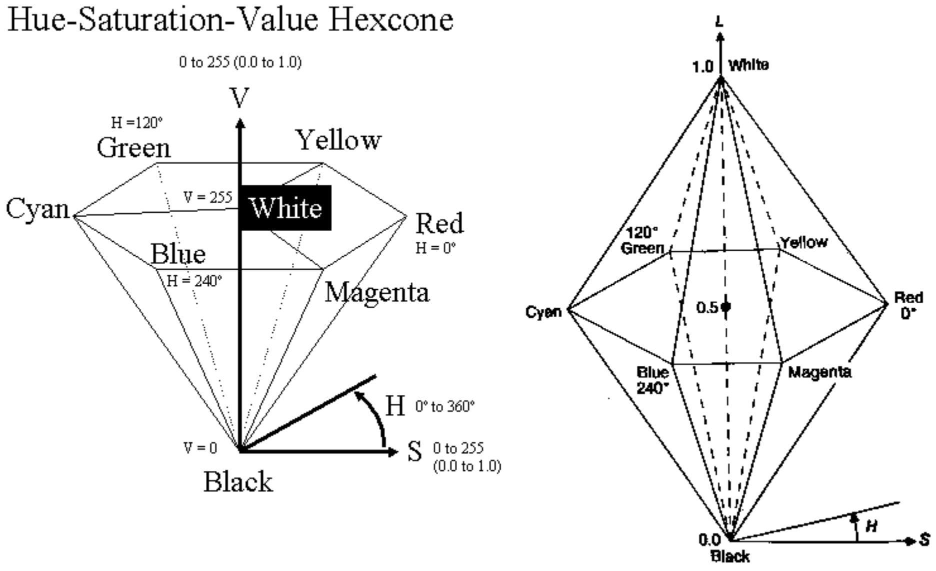


Figure 2.9: A cross-section view of the HSV(left) and HLS(right) color spaces.

Still both models are perceptually non-linear. Another subtle problem implicit in these models is that the attributes are not really themselves perceptually independent. It is possible to detect an apparent change in Hue, for example, when it is the parameter Value that is actually being changed.

Finally, perhaps the most serious departure from perceptual reality resides in the geometry of the models. The color spaces label those colors reproducible on a computer graphics monitor and this implies that all colors on planes of constant V are of equal brightness. This is not the case. For example, maximum intensity blue has a lower perceived brightness than maximum intensity yellow.

CIE Color Spaces: We have seen in the previous section that we need the spectral space to describe the physical properties of color. This implies that we need a way of reducing or converting spectral space calculations. We also saw that in many cases we are more concerned with the difference between a pair of colors. Color difference evaluation is essential for industrial color quality control. Throughout the years, a number of attempts have been made at developing color difference equations and uniform color spaces.

In 1931, the CIE adopted one set of color matching functions to define a Standard Colorimetric Observer (see Fig. 2.10) whose color matching characteristics are representative of the human population having normal vision.

The CIE Standard describes a color by a numeric triple (X,Y,Z). The X, Y, and Z values are defined as:

$$\begin{aligned} X &= k \int_{\lambda} E(\lambda) S(\lambda) \bar{x}(\lambda) d\lambda \\ Y &= k \int_{\lambda} E(\lambda) S(\lambda) \bar{y}(\lambda) d\lambda \\ Z &= k \int_{\lambda} E(\lambda) S(\lambda) \bar{z}(\lambda) d\lambda \\ k &= \frac{100}{\int_{\lambda} E(\lambda) \bar{y}(\lambda) d\lambda} \end{aligned} \tag{2.2}$$

where X, Y, and Z are the CIE tristimulus values, $E(\lambda)$ is the spectral power distribution of the light source, $S(\lambda)$ is the spectral reflectance of a reflective object (or spectral transmittance of a transmissive object). $\bar{x}(\lambda), \bar{y}(\lambda)$, and $\bar{z}(\lambda)$ are the color matching functions of the CIE Standard Colorimetric Observer, and k is a normalizing factor. By convention, k is usually determined such that $Y = 100$ when the object is a perfect white. A perfect white is an ideal, non-fluorescent, isotropic diffuser with a reflectance (or transmittance) equal to unity throughout the visible spectrum.

The CIE has also recommended two other color spaces designed to achieve more uniform and accurate models: CIE LAB for surfaces and and CIE LUV for lighting, television, video display applications respectively. The perceptual linearity is particular considered in these color spaces.

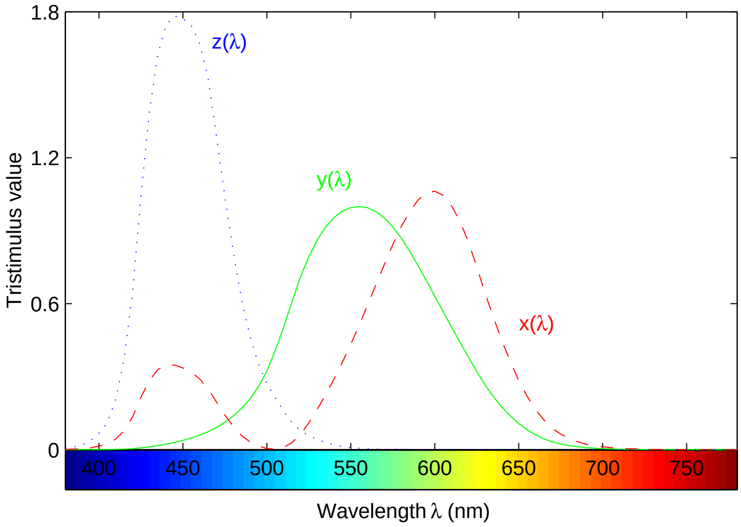


Figure 2.10: CIE Standard Colorimetric Observer, 2^o.

In the CIE LAB color space, three components are used: L* is the luminance axis, a* and b* are respectively red/green and yellow/blue axes, see Fig. 2.11. Although CIE LAB provides a more uniform color space than previous models, it is still not perfect, see for example (Luo, 1999). CIE LAB values are calculated from CIE XYZ by

$$L* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16, & \text{if } \left(\frac{Y}{Y_n}\right) > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right), & \text{if } \left(\frac{Y}{Y_n}\right) \leq 0.008856 \end{cases} \tag{2.3}$$

$$a* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \tag{2.4}$$

$$b* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \tag{2.5}$$

where

$$f(x) = \begin{cases} x^{1/3}, & \text{if } x > 0.008856 \\ 7.787x + 16/116, & \text{if } x \leq 0.008856 \end{cases} \tag{2.6}$$

The constants X_n, Y_n, and Z_n are the XYZ values for the chosen reference white point. When working with color monitors good choices could be something close to D65's XYZ coordinates.

As CIE LAB, CIE LUV is another color space introduced by CIE in 1976. This color space has 3 components which are L*, u* and v*. The

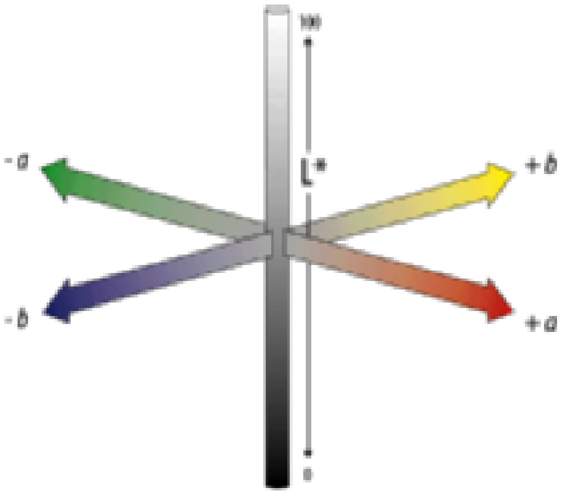


Figure 2.11: CIE LAB color space.

L* component defines the luminancy, and u*, v* define chromaticities. CIE LUV is very often used in calculations involving small color values or color differences, especially with additive colors. The CIE LUV color space is very popular in the television and video display industries. CIE LUV can be computed from CIE XYZ by

$$L* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16, & \text{if } \left(\frac{Y}{Y_n}\right) > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right), & \text{if } \left(\frac{Y}{Y_n}\right) \leq 0.008856 \end{cases} \tag{2.7}$$

$$u* = 13L * (u' - u'_n) \tag{2.8}$$

$$v* = 13L * (v' - v'_n) \tag{2.9}$$

$$u' = \frac{4X}{X + 15Y + 3Z} \tag{2.10}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \tag{2.11}$$

$$u'_n = \frac{4X_n}{X_n + 15Y_n + 3Z_n} \tag{2.12}$$

$$v'_n = \frac{9Y_n}{X_n + 15Y_n + 3Z_n} \tag{2.13}$$

where the tristimulus values X_n, Y_n, and Z_n are those of the white object color stimulus. The interested reader is referred to (Wyszecki and Stiles, 1982) for more detailed information.

Opponent Color Space: There is evidence that human color vision uses an opponent-color model by which certain hues were never perceive to occur together. For example, a color perception is never described as redish-greens or bluish-yellows, while combinations of red and yellow, red and blue, green and yellow, and green and blue are readily perceived. Based on this observation, the opponent color space is proposed to encode the color into opponent signal as follows:

$$\begin{aligned} rg &= R - G \\ by &= 2B - R - G \\ wb &= R + G + B \end{aligned} \tag{2.14}$$

where R, G, and B represent red, green, and blue channels, respectively, in RGB color space (Lennie and D'Zmura, 1988).

Chapter 3

CONTENT-BASED IMAGE RETRIEVAL

3.1 Visual Information Retrieval

The term "Information retrieval" was coined in 1952 and gained popularity in the research community from 1961 (Jones and Willett, 1977). The concept of an information retrieval system is to some extent self-explanatory from the terminological point of view. One may simply describe such a system as one that stores and retrieves information. As a system it is therefore composed of a set of interacting components, each of which is designed to serve a specific function for a specific purpose, and all these components are interrelated to achieve a goal, which is to retrieve information in a narrower sense.

In the past, information retrieval has meant textual information retrieval, but the above definition still holds when applied to Visual Information Retrieval (VIR). However, there is a distinction between the type of information and the nature of the retrieval of text and visual objects. Textual information is linear while images are bi-dimensional, and videos are three dimensional (one dimension is time). More precisely, text is provided with an inherent starting and ending point, and with a natural sequence of parsing. Such a natural parsing strategy is not available for images and videos.

There are generally two approaches to solutions for the VIR problem based on the form of the visual information: attribute-based and feature-based methods. Attribute-based methods rely on traditional textual information retrieval and Rational Database Management System (RDBMS) methods as well as on human intervention to extract metadata about a visual object and couple it together with the visual object as a textual annotation. Unfortunately, manual assignment of textual attributes is both time-consuming and costly. Moreover the manual annotations are very much dependent on the subjectivity of human perception. The perception subjectivity and annotation impreciseness may cause unrecoverable mismatches in later retrieval processes.

Problems with text-based access to images and videos have prompted increasing interest in the development of feature-based solutions. That is, instead of being manually annotated by text-based keywords, images would be extracted using some visual features such as color, texture, and shape, and be indexed based on these visual features. This approach relies heavily on results from computer vision. In this thesis our discussion will focus on some specific features, particularly color-based features for general image searching applications or content-based image retrieval applications. However, there is no single best feature that gives accurate results in any general setting. Usually a customed combination of features is needed to provide adequate retrieval results for each content-based image retrieval application.

3.2 Functions of a Typical CBIR System

A typical Content-based Image Retrieval (CBIR) system deals not only with various sources of information in different formats (for example, text, image, video) but also user's requirements. Basically it analyzes both the contents of the source of information as well as the user queries, and then matches these to retrieve those items that are relevant. The major functions of such a system are the following:

1. Analyze the contents of the source information, and represent the contents of the analyzed sources in a way that will be suitable for matching user queries (space of source information is transformed into feature space for the sake of fast matching in a later step). This step is normally very time consuming since it has to process sequentially all the source information (images) in the database. However, it has to be done only once and can be done off-line.

2. Analyze user queries and represent them in a form that will be suitable

for matching with the source database. Part of this step is similar to the previous step, but applied only to the query image.

3. Define a strategy to match the search queries with the information in the stored database. Retrieve the information that is relevant in an efficient way. This step is done online and is required to be very fast. Modern indexing techniques can be used to reorganize the feature space to speed up the matching processing.
4. Make necessary adjustments in the system (usually by tuning parameters in the matching engine) based on feedback from the users and/or the retrieved images.

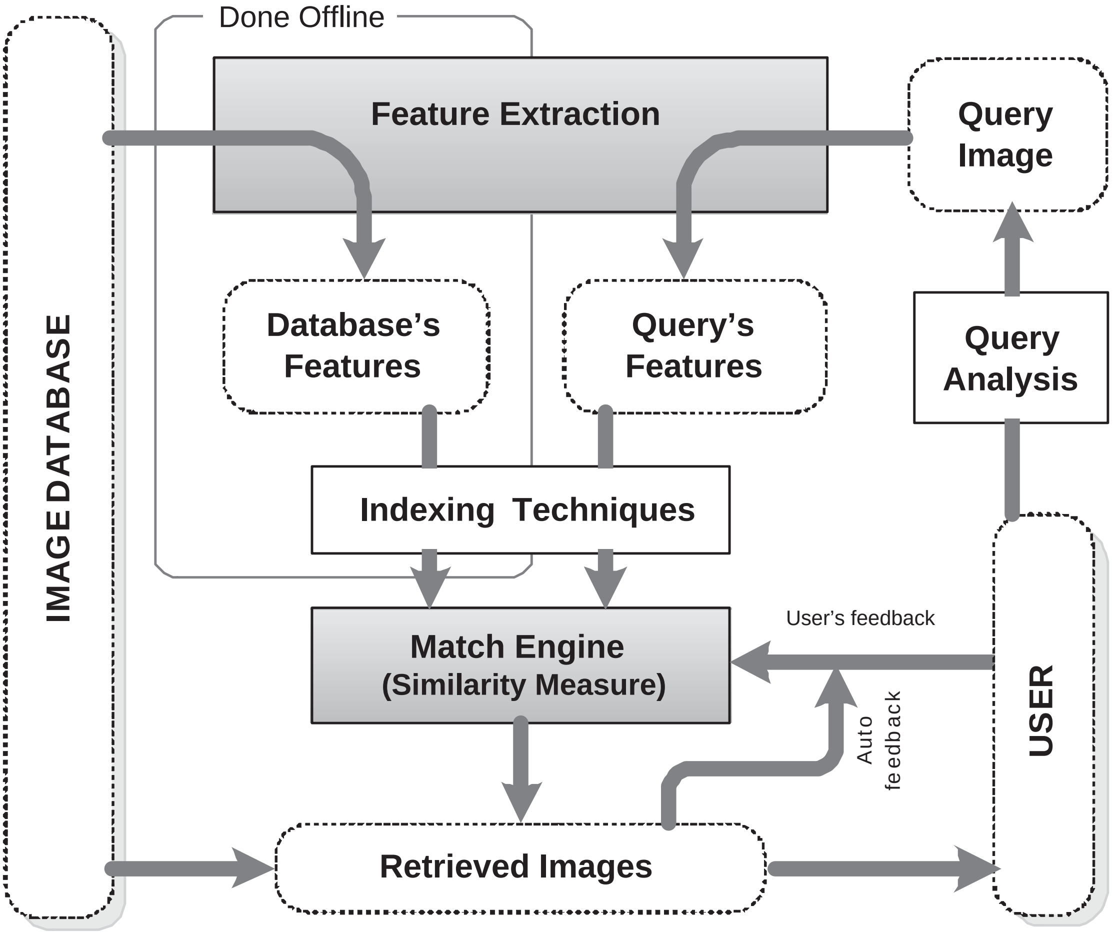


Figure 3.1: Broad outline of a Content-based Image Retrieval System.

It is evident from the above discussion that on the one side of a Content-based Image Retrieval system, there are sources of visual information in different formats and on the other there are the user queries. These two sides are linked through a series of tasks as illustrated in Fig. 3.1. Some of these tasks (such as user query analysis, multi-dimensional indexing) are briefly discussed here while the two most important tasks: "*Analyze the contents of the source information*" (Feature extractions) and "*Define a strategy to match the search queries with the information in the stored database*" (similarity measures), will be described in more detail later in dedicated sections in which color is emphasized.

User Query

There are many ways one can post a visual query. A good query method is the one which is natural to the user as well as capturing enough information from the user to extract meaningful results. The following query methods are commonly used in content-based image retrieval research:

Query by Example (QBE): In this type of query, the user of the system specifies a target query image upon which the image database is to be searched and compared against. The target query image can be a normal image, a low resolution scan of an image, or a user drawn sketch using graphical interface paint tools. A prime advantage of this type of system is that it is a natural way for expert and general users to search an image database.

Query by Feature (QBF): In the QBF type system, users specify queries by explicitly specifying the features they are interested in searching for. For example, a user may query an image database by issuing a command to "retrieve all images whose left quadrant contains 25% yellow pixels". This query is specified by the use of specialized graphical interface tools. Specialized users of an image retrieval system may find this query type natural, but general users may not. QBIC (Flickner et al., 1995) is an example of an existing content-based image retrieval system that uses this type of query method.

Attribute-based queries: Attribute-based queries use the textual annotations, pre-extracted by human effort, as a primary retrieval key. This type of representation entails a high degree of abstraction which is hard to achieve by fully automated methods because an image contains a large amount of information which is difficult to summarize using a few keywords. While this method is generally faster and easier to implement, there is an inherently high degree of subjectivity and ambiguity present as we have mentioned previously.

Which query method is most natural? To the general user, probably attribute-based queries are, with QBE systems a close second. A typical user would probably like to query content-based image retrieval systems by asking natural questions such as "Give me all my pictures from two years ago." or "Find all images on the Internet with a computer keyboard." Mapping this natural language query to a query on image database is extremely difficult to do using automated methods. The ability of computers to perform automatic object recognition on general images is still an open research problem. Most research and commercial efforts are therefore focused on building systems that perform well with QBE methods.

Multi-dimensional Indexing

To make content-based image retrieval truly scalable to large image databases, efficient multidimensional indexing techniques need to be explored. There are three major research communities contributing in this area: computational geometry, database management, and pattern recognition. The existing popular multidimensional indexing techniques include the bucketing algorithm, *k-d* tree, priority *k-d* tree, quad-tree, *K-D-B* tree, *hB* tree, *R*-tree and its variants R^+ tree and R^* tree.

The history of multidimensional indexing techniques can be traced back to the mid 1970s, when cell methods, quad-tree, and *k-d* tree were first introduced. However, their performances were far from satisfactory. Pushed by the urgent demand of spatial indexing from GIS and CAD systems, Guttman proposed the *R*-tree indexing structure (Guttman, 1984). Based on his work, many other variants of *R*-tree were developed (Sellis et al., 1987; Greene, 1989). In 1990, Beckmann and Kriegel proposed the best dynamic *R* tree variant, R^* tree in (Beckmann et al., 1990). However, even the R^* tree is not scalable to dimensions higher than 20 (Faloutsos et al., 1993; Weber et al., 1998; Rui et al., 1999; Ng and Tam, 1999).

3.3 Feature Extraction

Feature (content) extraction is the basis of content-based image retrieval. In a broad sense, features may include both text-based features (key words, annotations) and visual features (color, texture, shape, faces). Within the visual feature scope, the features can be further classified as low-level features and high-level features. The former include color, texture, and shape features while the latter is application-dependent and may include, for example, human faces and fingerprints. Because of perception subjectivity, there does not exist a single best presentation for a given feature. As we will soon see, for any given feature there exist multiple representations which characterize the feature from different perspectives.

3.3.1 Color

Color is the first and most straightforward visual feature for indexing and retrieval of images (Swain and Ballard, 1991; Rui et al., 1999; Schettini et al., 2001). It is also the most commonly used feature in the field.

A typical color image taken from a digital camera, or downloaded from the Internet normally has three color channels (Gray images have only one channel, while multi-spectral images could have more than three channels). The values of this three-dimensional data from the color image, however, do not give us an exact colorimetric description of the color in the image, but the position of these pixels in the color space. Pixels having values of (1,1,1) will appear differently in color in different color spaces. Thus a full description of a typical color image should consist of the two-dimensional spatial information telling where the color pixel is in the spatial domain, the color space we are refereing to, and the three-dimensional color data telling where the color pixel is in this color space.

Here the color space is assumed to be fixed, the spatial information in the image is ignored, and the color information in a typical image can be considered as a simple three-dimensional signal.

One- or two-dimensional color signals are also widely used in CBIR especially in applications where robustness against image capturing conditions is important. Chromaticity information in the form of the xy- or ab-coordinates of the CIE XYZ and CIE LAB systems can be used in intensity independent applications. Hue information was used in applications where only the differences between materials of objects in the scene are important. It has been shown (Gevers and Smeulders, 1999; Geusebroek et al., 2001) that the hue is invariant under highlights, shadowing, and geometry changes of viewing and illumination angles.

If we consider color information of an image as a simple one-, two-, or three-dimensional signal, analyzing the signal by using multivariate probability density estimation is the most straightforward way to describe the color information of the image. The histogram is the simplest tool. Other ways of describing color information in CBIR include the use of dominant colors, or color signatures, and color moments.

Color histogram

Statistically, a color histogram is a way to approximate the joint probability of the values of the three color channels. The most common form of the histogram is obtained by splitting the range of the data into equally sized bins. Then for each bin, the number of points from the data set (here the colors of the pixels in an image) that fall into each bin are counted and normalized to total points, which gives us the probability of a pixel falling into that bin.

Details of color histograms will be discussed in Chapter 4 when different ways of describing the underlying color distributions are presented. For the sake of simplicity, given a color image $I(x, y)$ of size $X \times Y$, which consists of three channels $I = (I_R, I_G, I_B)$, the color histogram used here is

$$h_c(m) = \frac{1}{XY} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \begin{cases} 1 & \text{if I(x,y) in bin m,} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where a color bin is defined as a region of colors.

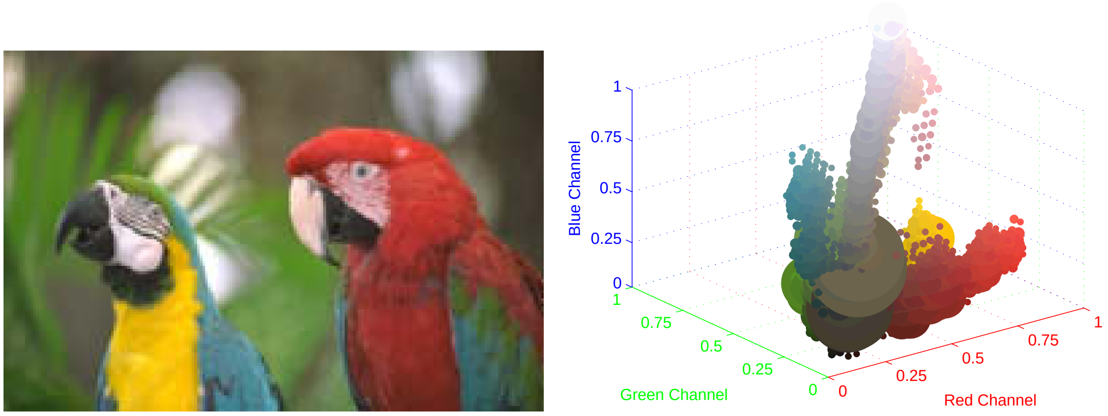


Figure 3.2: A color image and its over-smoothed three-dimensional RGB color histogram.

The regions in the color space can be defined in a non-parameterized way by non-parametric clustering algorithms, or simply given by fixed borders in some color space. For example in RGB color space, if we divide each channel R,G, and B into 8 equally intervals with a length of 32: $0 \cdot 31, 32 \cdot 63, \cdots, 224 \cdot 255$, we will have an 8 by 8 by 8 color histogram of $8 \times 8 \times 8 = 512$ color bins. An example of how a color histogram looks is shown in Fig. 3.2, in which the three-dimensional histogram was made in RGB color space. The left side of Fig. 3.3 shows another example of a one-dimensional hue histogram[1] of the same image as in Fig. 3.2 in which we divided the hue information into 32 equal bins. The right side of Fig. 3.3 is the estimated hue distribution given by a kernel-based

[1] One important property of hue is its circular nature as an angle in most color coordinate systems. This is important for the selection of the processing method. Ignoring this constraint leads to misleading results as demonstrated in Fig. 3.3. This figure shows an example of the estimated hue density distribution. The histogram method on the left results in an estimation of the hue distribution which is wrong in the red area since it does not take into account the circular nature property of the hue. This problem can be solved by using a kernel density estimator with an extended support. The estimated density using a kernel density estimator is depicted on the right of Fig. 3.3.

method. Details of the kernel-based method in describing color distributions will be discussed in chapter 4.

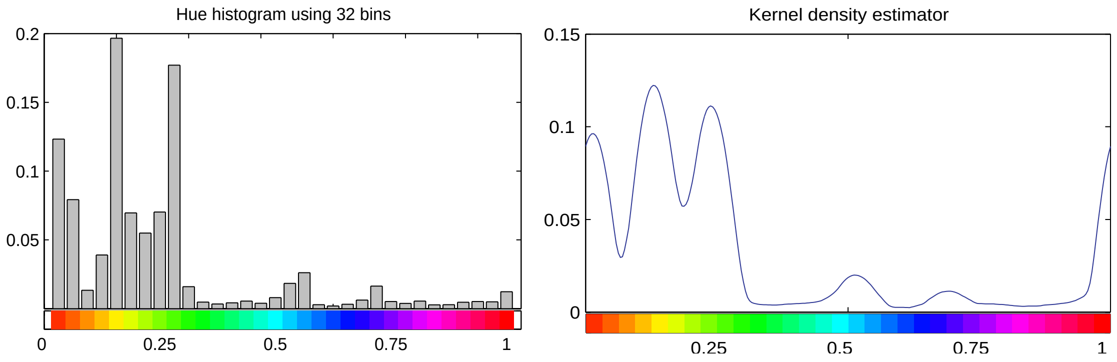


Figure 3.3: The hue density distribution of the parrots image in Fig. 3.2 estimated by histogram and kernel-based methods. The histogram fails to describe the circular nature of the hue in the red region.

There are two important parameters that need to be specified when constructing a histogram in this way: the bin width and the bin locations. It is not very difficult to see that the choice of the bin width has an enormous effect on the appearance of the resulting histogram. Choosing a very small bin width results in a jagged histogram, with a separate block for each distinct observation. A very large bin width results in a histogram with a single block. Intermediate bin widths lead to a variety of histogram shapes between these two extremes. The positions of the bins are also of importance to the shape of the histogram. Small shifts of the bins can lead to a major change in the shape of the histogram.

Considering that most color histograms are very sparse, see Fig. 3.2, and thus sensitive to noise, Stricker and Orengo (Stricker and Orengo, 1996) proposed using the cumulated color histogram. Their results demonstrated the advantages of the proposed approach over the conventional color histogram approach. However the approach has the disadvantage in the case of more than one dimensional histograms, that there is no clear way to order bins.

The color histogram is the most popular representation of color distributions since it is insensitive to small object distortions and is easy to compute. For example, Fig. 3.4 shows images of the same ball but taken under five different viewing positions[2] and their corresponding color histograms, which are very similar.

[2] The images of the ball were taken at the Computer Science Laboratory, Simon Fraser University, Vancouver, Canada, `http://www.cs.sfu.ca/research/groups/Vision/`.

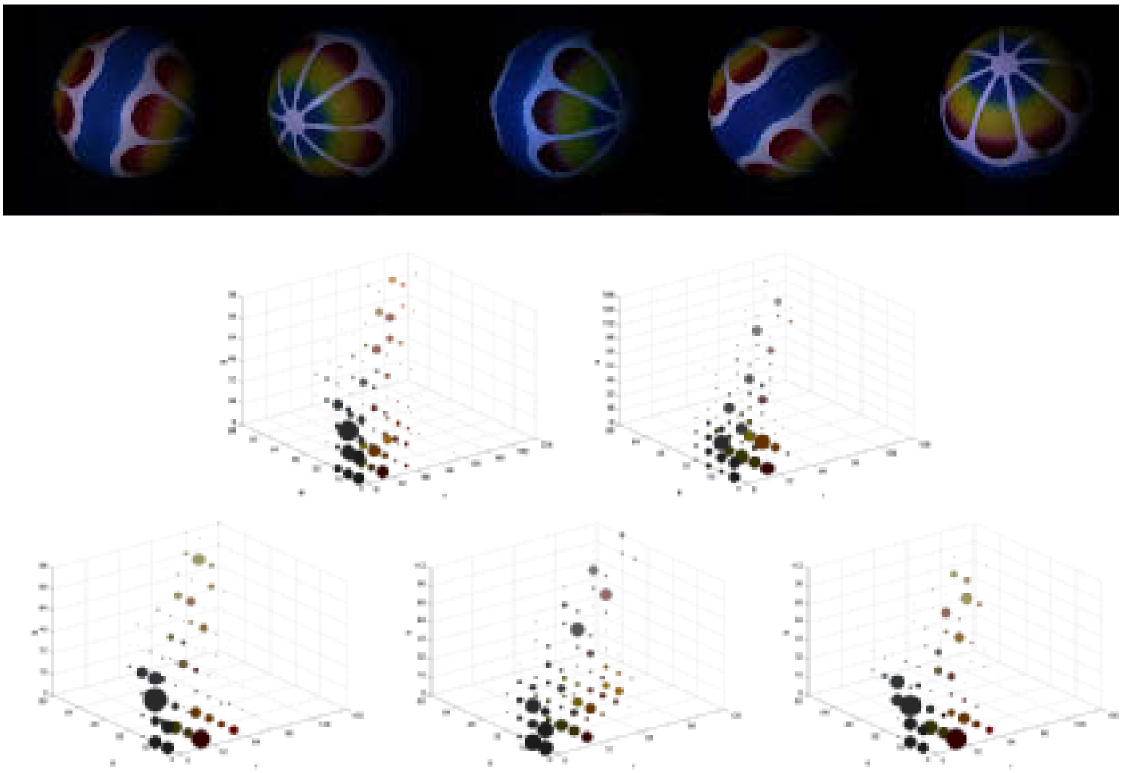


Figure 3.4: Color images of the same object taken under different views and their color distributions.

Dominant Colors

Based on the observation that the color histograms are very sparse and normally a small number of colors are enough to characterize the color information in a color image, dominant colors are used to characterize the color content of an image. A color clustering is performed in order to obtain its representative dominant colors and its corresponding percentage. Each representative color and its corresponding percentage form a pair of attributes that describe the color characteristics in an image region.

The dominant color histogram feature descriptor F is defined to be a set of such attribute pairs:

$$F = \{\{c_i, p_i\}, i = 1..N\} \tag{3.2}$$

where N is the total number of color clusters in the image, c_i is a 3-D color vector, p_i is its percentage, and $\sum_i p_i = 1$. Note that N can vary from image to image.

Color Moments

Color moments are the statistical moments of the probability distributions of colors. In (Stricker and Orengo, 1996) color moments are used; only the first three moments of the histograms of each color channel are computed and used as an index, and the image is represented only by the average and covariance matrix of its color distribution. Detailed descriptions of color moments can be found in section 8.2.

Color Correlogram

Huang and colleagues (Huang et al., 1997) use color correlograms, which considers the spatial correlation of colors. A color correlogram of an image is a table indexed by color pairs, where the k^{th} entry for (i,j) specifies the probability of finding a pixel of color j at a distance k from a pixel i in the image. Due to the high complexity of this method, the autocorrelogram is used instead which captures spatial correlation between identical colors only.

3.3.2 Texture

Texture is widely used and intuitively obvious but has no precise definition due to its wide variability. One existing definition states that ”an image region has a constant texture if a set of its local properties in that region is constant, slowly changing, or approximately periodic”.

There are many ways to describe texture: Statistical methods often use spatial frequency, co-occurrence matrices, edge frequency, primitive length etc. From these many simple features such as energy, entropy, homogeneity, coarseness, contrast, correlation, cluster tendency, anisotropy, phase, roughness, directionality, flames, stripes, repetitiveness, granularity are derived. These texture description methods compute different texture properties and are suitable if texture primitive sizes are comparable with the pixel sizes.

Syntactic and hybrid (combinations of statistical and syntactic) methods such as shape chain grammars, or graph grammars are more suitable for textures where primitives can easily be determined and their properties described. There are many review papers in this area. We refer interested readers to (Weszka et al., 1976; Ohanian and Dubes, 1992; Ma and Manjunath, 1995; Randen and Husoy, 1999) for more detailed information.

3.3.3 Shape

Defining the shape of an object is often very difficult. Shape is usually represented verbally or in figures, and people use terms such as elongated, rounded.

Computer-based processing of shape requires describing even very complicated shapes precisely and while many practical shape description methods exists, there is no generally accepted methodology of shape description.

Two main types of shape features are commonly used: boundary-based and region-based features. The former uses only the outer boundary of the shape while the latter uses the entire shape region. Examples of the first type include chain codes, Fourier descriptors, simple geometric border representations (curvature, bending energy, boundary length, signature), and examples of the second include area, Euler number, eccentricity, elongatedness, and compactness. Some review papers in shape representation are (Li and Ma, 1995; Mehtre et al., 1997)

3.3.4 High-level Features

The vast majority of current content-based image retrieval research is focused on low-level retrieval methods. However, some researchers have attempted to bridge the gap between low-level and high-level retrieval. They tend to concentrate on one of two problems. The first is scene recognition. It can often be important to identify the overall type of scene depicted by an image, both because this in an important filter which can be used when searching, and because this can help in determining whether a specific object is present. One system of this type is IRIS (Hermes, 1995), which uses color, texture, region and spatial information to derive the most likely interpretation of the scene, generating text descriptors which can be input to any text-based retrieval system. Other researchers have identified simpler techniques for scene analysis, using low-frequency image components to train a neural network (Oliva, 1997), or color neighborhood information extracted from low-resolution images to construct user-defined templates (Ratan and Grimson, 1997)

The second focus of research activity is object recognition, an area of interest to the computer vision community for many years. Techniques are now being developed for recognizing and classifying objects with database retrieval in mind. The best-known work in this field is probably that of (Forsyth, 1997), who has attracted publicity by developing a technique for recognizing naked human beings in images, though his approach has been applied to a much wider range of objects, including horses and trees. All these techniques are based on the idea of developing a model of each class of objects to be recognized, identifying image regions which might contain examples of the objects, and building up evidence to confirm or rule out the object's presence.

3.4 Similarity Measures

Once features of images in the database are extracted and the user's query is formed, the search results are obtained by measuring the similarity between the pre-extracted features of the image database and the analyzed user's query.

The similarity measure should ideally have some or all of the following basic properties:

Perceptual Similarity: The feature distance between two images is large only if the images are not "similar", and small if the images are "similar". Images are very often described in feature space and the similarity between images is usually measured by a distance measure in the feature space. Taking into account the properties of this space for human perception and the underlying properties of the feature vectors representing the images is very important in improving the perceptual similarity property of the proposed similarity measure.

Efficiency: The measure should be computed rapidly in order to have fast response in the search phase. Typical CBIR applications require a very fast response, not longer than a few seconds. During that short period of time, the search engine normally has to compute thousands of distances depending on the size of the image database. The complexity of the distance measure is therefore important.

Scalability: The performance of the system should not deteriorate too much for large databases since a system may search in databases containing millions of images. A naive implementation of CBIR computes all the distances between the query image and the images in the databases. These distances are then sorted to find out the most similar images to the query image. The complexity of the search engine is therefore proportional to the size of the image database (or $O(N)$ if we say N is the number of images). Multi-dimensional indexing techniques (as mentioned in section 3.2) could be used to reduce the complexity to $O(log(N))$. However it has been reported that the performance of current indexing techniques is reduced back to a sequential scanning (Weber et al., 1998; Rui et al., 1999) when the number of dimensions that need to be indexed is greater than 20. So one has to consider this factor when dealing with very large image databases.

Metric: The problem of whether the similarity distance should be a metric or not is not decided yet since human vision is very complex and the mechanisms of the human visual system are not fully understood. We prefer the similarity distance to be a metric since we consider the following properties as very natural requirements.

- Constancy of self-similarity: The distances between an image to itself should be equal to a constant independent to the image (preferable to be zero).

  $$d(A, A) = d(B, B);$$

- Minimality: An image should be more similar to itself than to other images.

  $$d(A, A) < d(A, B);$$

- Symmetry: It is unreasonable if we say image A is similar to image B but image B is not similar to image A.

  $$d(A, B) = d(B, A);$$

- Transitivity: It is also unreasonable if image A is very similar to image B, and B in turn very similar to C, but C is very dissimilar to A.

  However this transitivity property may not hold for a series of images. Even if image I_i is similar to image I_{i+1} for all $i = 1..N$ this does not mean that image I_1 similar to image I_N. In a video sequence, for example, each frame is similar to its neighbor frames but the first and the last frame of the sequence can be very different.

Robustness: The system should be robust to changes in the imaging conditions of the database images. For example if images in the database are taken under tungsten illumination (reddish), the retrieval system should be able to find these objects even if the query object was taken under daylight illumination (blueish).

Many (dis)similarity measures have been proposed, but none of them has all the above properties. We list here some of the most commonly used.

- Histogram intersection (Swain and Ballard, 1991):

  This is one of the first distance measures in color-based image retrieval. The distance defined is based on the size of the common part of two color histograms. Given two color histograms h_1 and h_2 as in Eq. 3.1, the distance between them can be defined as

  $$dist_{HI} = 1 - \sum_{i=1}^{N} min(h_{1i}, h_{2i}) \tag{3.3}$$

  This distance measure is fast since it is based on a very simple formula. However it is not a metric and no color information is used when deriving the distance. This may lead to undesirable results.

- $L_1 distance$ (Stricker and Orengo, 1996), the Minkowski-form distance L_p: The Minkowski-form distance L_p between two histograms is defined as

$$dist_{Mp} = \left(\sum_i \mid h_{1i} - h_{2i} \mid^p \right)^{1/p} \tag{3.4}$$

- Quadratic form (Hafner et al., 1995): the distance between two N-dimensional color histograms h_1 and h_2 is defined as

$$dist_{QF} = (h_1 - h_2)^{'} A(h_1 - h_2) \tag{3.5}$$

  where $A = [a_{ij}]$ is a matrix and the weights a_{ij} denote the similarity between bins i and j. A popular choice of a_{ij} is given by

$$a_{ij} = 1 - (d_{ij}/d_{max})^k \tag{3.6}$$

  where d_{ij} is the distance between color i and color j (normally d_{ij} is the Euclidean distance between the two colors in some uniform color spaces like La^*b^* or Lu^*v^*) and $d_{max} = max_{ij}(d_{ij})$. k is a constant controlling the weight between neighboring colors.

  Alternatively, another common choice for a_{ij} is (Hafner et al., 1995)

$$a_{ij} = exp(-k(d_{ij}/d_{max})^2) \tag{3.7}$$

- The Earth Mover Distance (EMD) (Rubner et al., 1998) is based on the minimal cost to transform one distribution to the other. If the cost of moving a single feature unit in the feature space is the ground distance, then the distance between two distributions is given by the minimal sum of the costs incurred to move all the individual features. The EMD can be defined as the solution of a transport problem which can be solved by linear optimization:

$$dist_{EMD} = \frac{\sum_{ij} g_{ij} d_{ij}}{\sum_{ij} g_{ij}} \tag{3.8}$$

  where d_{ij} denotes the dissimilarity between bins i and j, and $g_{ij} \geq 0$ is the optimal flow between the two distributions such that the total cost

$$dist_{EMD} = \sum_{ij} g_{ij} d_{ij} \tag{3.9}$$

  is minimized, subject to the following constraints:

$$\begin{aligned} &\sum_i g_{ij} \leq h_{1i} \\ &\sum_j g_{ij} \leq h_{2i} \\ &\sum_{ij} g_{ij} = min(h_{1i}, h_{2i}) \end{aligned} \tag{3.10}$$

for all i and j. The denominator in Eq. 3.8 is a normalization factor that permits matching parts of distributions with different total mass. If the ground distance is a metric and the two distributions have the same amount of total mass, the EMD defines a metric. As a key advantage of the EMD each image may be represented by different bins that adapt to their specific distribution. When marginal histograms are used, the dissimilarity values obtained for the individual dimensions must be combined into a joint overall dissimilarity value.

Other distance measures which are also of interest are

- The Kolmogorov-Smirnov distance was originally proposed in (German, 1990). It is defined as the maximal discrepancy between the cumulative distributions
$$dist_{Mp} = \max_i \mid h^c_{1i} - h^c_{2i} \mid \tag{3.11}$$
where h^c is the cumulative histogram of histogram h

- A Statistics of the Cramer/Von Mises type based on cumulative distributions is defined
$$dist_C = \sum_i (h^c_{1i} - h^c_{2i})^2 \tag{3.12}$$

- The χ^2 statistic is given by
$$dist_\chi = \sum_i \frac{\left(h_{1i} - \hat{h}_i\right)^2}{\hat{h}_i} \tag{3.13}$$
where
$$\hat{h}_i = \frac{h_{1i} + h_{2i}}{2}$$
denotes the joint estimate.

- The Kullback-Leibler divergence is defined by
$$dist_{KL} = \sum_i h_{1i} log \frac{h_{1i}}{h_{2i}} \tag{3.14}$$

- The Jeffrey-divergence is defined by
$$dist_{JD} = \sum_i \left(h_{1i} log \frac{h_{1i}}{\hat{h}_i} + h_{2i} log \frac{h_{2i}}{\hat{h}_i} \right) \tag{3.15}$$

- The Weighted-Mean-Variance was proposed in (Manjunath and Ma, 1996). This distance is defined by

$$dist_{WMV} = \frac{\mu_1 - \mu_2}{\sigma(\mu)} + \frac{\sigma_1 - \sigma_2}{\sigma(\sigma)} \tag{3.16}$$

  where μ_1, μ_2 are the empirical means and σ_1, σ_2 are the standard deviations of the two histogram h_1, h_2. $\sigma(.)$ denotes an estimate of the standard deviation of the respective entity.

- Bhattacharyya-distance (Fukunaga, 1990) is defined

$$\begin{aligned} &d_B^2(N(\mu_1, \Sigma_1), N(\mu_2, \Sigma_2)) = \\ &\frac{1}{8}(\mu_1 - \mu_2)'\Sigma^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\ln\frac{\det\Sigma}{\sqrt{\det\Sigma_1 \det\Sigma_2}} \end{aligned} \tag{3.17}$$

  where $\Sigma = 0.5 \times (\Sigma_1 + \Sigma_2)$

- Mahalanobis distance (Fukunaga, 1990) is given by

$$d_M^2(N(\mu_1, \Sigma), N(\mu_2, \Sigma)) = (\mu_1 - \mu_2)'\Sigma^{-1}(\mu_1 - \mu_2) \tag{3.18}$$

For more detailed descriptions, we refer to the cited papers. (Puzixha et al., 1999) provides a comprehensive comparison over many different distance measures.

3.5 Evaluating Retrieval Performance for CBIR

Once a content-based image retrieval application had been developed, the next crucial problem is how to evaluate its performance, both retrieval performance and complexity (or the time for searching and for creating the pre-computed feature database). For evaluating the retrieval performance, many papers in the field were often ignored or restricted simply to printing out the results of one or more example queries which are easily tailored to give a positive impression. Some other papers either used performance measures borrowing from information retrieval (TREC, 2002), or developed new measures for content-based image retrieval (Gunther and Beretta, 2001; Manjunath et al., 2001; Benchathlon, 2003)[3].

In this section, basic problems in evaluating performance of content-based image retrieval systems are addressed briefly. Then a more detailed description

[3] The Benchathlon network is a non-profit organization that aims at gathering CBIR people under a single umbrella to create a favorable context for developing a new CBIR benchmarking framework. More information can be found on their website at `http://www.benchathlon.net/`

of the MPEG-7 Color/Texture Core Experiment Procedures is given. These are used widely in evaluating the retrieval performance of our experiments described in the thesis.

Basic Problems in CBIR performance Evaluation

In order to evaluate a CBIR application, an image database and a set of queries with ground truth are needed. The queries are put to the CBIR application to obtain the retrieval results. A performance method is then needed to compare these retrieved results with the ground truth images.

A common way of constructing an image database for CBIR evaluation is to use Corel photo CDs, each of which usually contains 100 broadly similar images. Most research groups use only a subset of the collection, and this can result in a collection of several highly dissimilar groups of images, with relatively high within-group similarity. This can lead to great apparent improvement in retrieval performance: e.g. it is not too hard to distinguish sunsets from underwater images of fish. Another commonly used database is the VisTex database at MIT, Media Lab, which contains more than 400 primarily texture images. Some other candidates includes the standard collection of 5466 color images from MPEG-7 (Zier and Ohm, 1999), the image database from University of Washington at `http://www.cs.washington.edu/research/imagedatabase/groundtruth/` and the Benchathlon collection at `http://www.benchathlon.net/img/done/`.

One of the problems in creating such an image collection is that the size of the database should be large enough, and the images should have enough diversity in different domains. For text-based retrieval, it is quite normal to have millions of documents (TREC, 2002) whereas in CBIR most systems work with only few thousand images, some even with fewer. Ways to get a huge collection of images include collecting them from the Internet and sampling image frames from TV channels.

Once images are collected, the next task in evaluating performance of CBIR application is to define a set of queries and their ground truth based on the input image database. It can be done by:

- Using these collections with a pre-defined subset: A very common technique is to use sets of images with different topics such as the Corel collections. Relevant judgements are given by the collection itself since it contains distinct groups of annotated images. Grouping is not always based on the visual similarity but often on the objects contained.

- Simulating a user: The ground truth images are simulated from the query image using some model. A very common way to generate ground truth from a query image is by adding noise, down-sampling, or up-sampling the query image.

- User judgements: The collection of real user judgements is time-consuming and only the user who knows what he or she expects as a retrieval result of a given query image. Experiments show that user judgements for the same image often differ (Squire and Pun, 1997).

When the image database is collected and the queries and their ground truth are selected, the query images are presented one by one to the search engine of the CBIR application, the retrieved results are then compared to the ground truth of the corresponding query image. Several different methods can be applied here to compare the two sets: the ground truth and the actual retrieved images.

- The straightforward way is by asking users to judge the success of a query by looking at the two sets.
- A single value is computed from the two sets telling us how well the query was retrieved by the system. Examples are: rank of the best match, average rank of relevant images, precision, recall, target testing, error rate, retrieval efficiency, correct and incorrect detection.
- A graph can be used to illustrate the relation of two of the above values, for example the precision vs. recall graph, the precision vs. number of retrieved images, recall vs. number of retrieved images graph or retrieval accuracy vs. noise graph.

The MPEG-7 Color/Texture Core Experiment Procedures

Under the framework of the MPEG-7 standardization process, procedures for color and texture core experiments are defined so that different competing technologies can be compared. It consists of the description of the input image database, the standard queries and their corresponding ground truth images, and the benchmark metric.

- CCD, The Common Color Dataset, consists of 5466 color images, 56KB for each image on average, and the average size is 260x355 pixels.
- CCQ, The Common Color Queries, consists of 50 queries and their ground truth images all selected from the image database. The length of the ground truth ranging from 3 to 32 images with average length is 8 images.
- ANMRR, The Average Normalized Modified Retrieval Rank, is defined as follows:

Consider a query q with its ground truth of $G(q)$ images. Invoking the query q against the image database causes a set of images to be retrieved. In the best case, all images in the ground truth set $G(q)$ of the query q would be returned as an exact match of the ground truth vector sequence and would thus correspond to a perfect retrieval score.

However, most image retrieval algorithms are less than perfect, so images that are member of $G(q)$ may be returned either out of order, or in correct sequence but interspersed with incorrect images, or as an incomplete subset when not all the images of $G(q)$ are found, or even none of the ground truth images is found in the worst case.

A ranking procedure is used to take into account all such possibilities. A scoring window $W(q) > G(q)$ is associated with the query q such that the retrieved images contained in $W(q)$ are ranked according to an index $r = 1, 2, \cdots, W$ as depicted in Fig. 3.5.

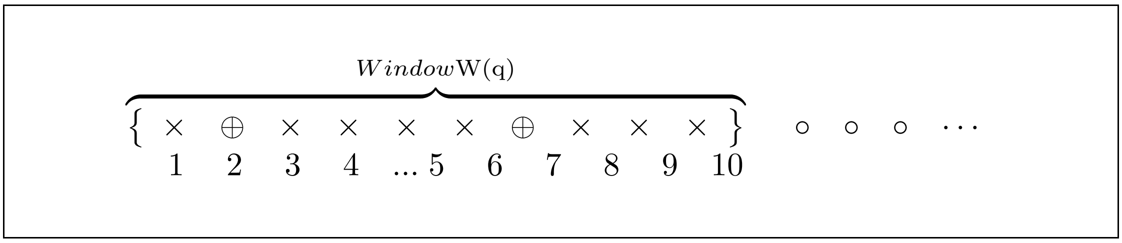


Figure 3.5: Retrieved images with scoring window $W(q)$ and two correct images of rank 2 and 7. ANMRR = 0.663.

A step function θ is defined as

$$\theta(w-g) = \begin{cases} 1 & \text{iff} : w \approx g \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

which is zero unless there is a match (denoted by $\approx$) between the retrieved image of index w and any ground truth image g.

The number of correct images returned in the window $W(q)$ is given by

$$R_{correct}(q) = \sum_{w=1}^{W(q)} \theta(w-g) \tag{3.20}$$

and the number of missed images is

$$R_{missed}(q) = G(q) - R_{correct}(q) \tag{3.21}$$

Now the average retrieval rank $AVR(q)$ can be defined as

$$AVR(q) = \frac{1}{G(q)} \left\{ \sum_{w=1}^{W(q)} w \cdot \theta(w-g) \right\} + \frac{R_{missed}(q) \cdot Pen(q)}{G(q)} \tag{3.22}$$

The first term $\sum_{w=1}^{W(q)} w \cdot \theta(w-g)$ is the sum of the ranks of the correct images and $Pen(q)$ is a penalty for the missed images. Since the missed images lie outside the scoring window $W(q)$, the value of the penalty must exceed the rank of the last entry in $W(q)$: $Pen(q) > W(q)$.

It is important to note that the value of the retrieval rank is affected only by the position of the correct images in the scoring window, not their order with respect to the sequence specified by the ground truth vector. If A and B are correct images in the ground truth set, then the retrieved sets $\{A, B\}$ and $\{B, A\}$ have equal retrieval rank.

In the case of perfect score, all images in the ground truth set are found with rank from 1 to $G(q)$ and the number of missed images $R_{missed} = 0$. The best average retrieval rank is given by

$$AVR_b(q) = \frac{1 + G(q)}{2} \tag{3.23}$$

In the worst case, no ground truth images are found in the window $W(q)$, so the number of incorrect images $R_{missed}(q) = G(q)$, and the worst average retrieval rank is given by

$$AVR_w(q) = Pen(q) \tag{3.24}$$

These extremes define an interval $[AVR_b(q), AVR_w(q)]$, within which any average retrieval rank $AVR(q)$ must lie. For the purpose of comparisons, it is preferable to normalize this interval onto the unit interval $[0 \cdots 1]$ via the normalized modified retrieval rank (NMRR) given by:

$$NMRR(q) = \frac{AVR(q) - AVR_b(q)}{AVR_w(q) - AVR_b(q)} = \frac{AVR(q) - 0.5 \cdot (1 + G(q))}{Pen(q) - 0.5 \cdot (1 + G(q))} \tag{3.25}$$

It is then straightforward to define the average normalized modified retrieval rank (ANMRR) as average NMRR over all NQ queries.

$$ANMRR(q) = \frac{1}{NQ} \sum_{q=1}^{NQ} NMRR(q) \tag{3.26}$$

Specifically in this thesis, we used the window size equal to two times the ground truth size $W(q) = 2 \cdot G(q)$ and the penalty function $Pen(q) = 1.25 \cdot$

$W(q) = 2.5 \cdot G(q)$. Under such conditions, the ANMRR can be reduced to the following form:

$$ANMRR(q) = 1 - \frac{1}{NQ} \sum_{q=1}^{NQ} \frac{\sum_{w=1}^{2G(q)} \{2.5G(q) - w\}\theta(w - g)}{G(q)\{2G(q) - 0.5\}} \tag{3.27}$$

Some examples may help to give a feeling for this measure: The ANMRR of the retrieval result in Fig. 3.5 is 0.663. Suppose that we have a query with 30 ground truth images; if only one ground truth image is missed in the retrieval result, the ANMRR is 0.055 if the incorrect image is found as 1st rank, and ANMRR is 0.011 if it is found in the last rank. If we missed the first five images, we get ANMRR=0.262, and if the last 5 images were wrong then ANMRR=0.072.

3.6 CBIR Systems

In recent years, content-based image retrieval has become a highly active research area. Many image retrieval systems, both commercial and research systems, have been built. In the following discussion, we briefly describe some of the well-known CBIR systems that have been developed.

IBM's QBIC

QBIC, standing for Query By Image Content, is the first commercial content-based image retrieval system. Its system framework and techniques had profound effects on later image retrieval systems. QBIC supports mainly queries based on example images, user-constructed sketches and drawings, and selected color and texture patterns.

In the process of image indexing, QBIC has used fully automatic unsupervised segmentation methods along with a foreground/background model to identify objects in a restricted class of images. Robust algorithms are required in this domain because of the textured and variegated backgrounds. QBIC also has semiautomatic tools for identifying objects. One is an enhanced flood-fill technique. Flood-fill methods start from a single object pixel and repeatedly add adjacent pixels whose values are within some given threshold of the original pixel. Another outlining tool to help users track object edges is based on the ”snakes” concept developed in computer vision research. This tool takes a user-drawn curve and automatically aligns it with nearby image edges. It finds the curve that maximizes the image gradient magnitude along the curve.

After object identification, QBIC will compute the features of each object and image. They are as following.

- Color:

  The color feature used in QBIC are the average (R,G,B), (Y,I,Q), (L,a,b), and MTM (Mathematical Transform to Munsell) coordinates, and a k-element color histogram (Faloutsos et al., 1993).

- Texture:

  QBIC's texture feature is an improved version of the Tamura texture representation (Tamura et al., 1978); i.e. combinations of coarseness, contrast, and directionality (Equitz and Niblack, 1994). For color images, these measures are computed on the luminance band, which is computed from the three color bands. The coarseness feature describes the scale of the texture and is efficiently calculated using moving windows of different sizes. The contrast feature describes the vividness of the pattern, and is a function of the variance of the gray-level histogram. The directionality feature describes whether or not the image has a favored direction, or whether it is isotropic, and is a measure of the "peakedness" of the distribution of gradient directions in the image.

- Shape:

  Shape features in QBIC are based on a combination of area, circularity, eccentricity, and major axis orientation, plus a set of algebraic moment invariants (Scassellati et al., 1994; Faloutsos et al., 1993). All shapes are assumed to be non-occluded planar shapes allowing each shape to be represented as a binary image.

- Sketch:

  QBIC allows images to be retrieved based on a rough user sketch. The feature needed to support this retrieval consists of a reduced resolution edge map of each image. To compute edge maps, QBIC converts each color image to a single band luminance, computes the binary edge image and reduces the edge image to size 64 × 64.

Once the features are described, the similarity measures are used to get similar images. In the search step, QBIC distinguishes between "scenes" (or images) and "objects". A scene is a full color image or single frame of video and an object is a part of a scene. QBIC computes the following features:

- Objects: average color, color histogram, texture, shape, location.

- Images: average color, color histogram, texture, positional edges (sketch), positional color (draw)

QBIC is one of the few systems which takes into account the high dimensional feature indexing. In its indexing subsystem, KLT is first used to perform dimension reduction and then R^*-tree is used as the multidimensional indexing structure (Lee et al., 1994; Faloutsos et al., 1994). In its new system, text-based keyword search can be combined with content-based similarity search. The online QBIC demo is at `http://wwwqbic.almaden.ibm.com`.

Virage

Virage is a content-based image search engine developed at Virage Inc. Similar to QBIC, Virage (Bach et al., 1996) supports visual queries based on color, composition (color layout), texture, and structure (object boundary information). But Virage goes one step further than QBIC. It also supports arbitrary combinations of the above four atomic queries. The users can adjust the weights associated with the atomic features according to their own emphasis. Jeffrey et al. further proposed an open framework for image management. They classified the visual features (primitive) as general (such as color, shape, or texture) and domain specific (face recognition, cancer cell detection, etc.). Various useful primitives can be added to the open structure, depending on the domain requirements. To go beyond the query by example mode, Gupta and Jain proposed a nine-component query language framework in (Gupta and Jain, 1997). The system is available as an add-on to existing database management systems such as Oracle or Informix.

RetrievalWare

RetrievalWare is a content-based image retrieval engine developed by Excalibur Technologies Corp. From one of its early publications, we can see that its emphasis was the application of neural nets to image retrieval (Dow, 1993). Its more recent search engine uses color, shape, texture, brightness, color layout, and aspect ratio of the image, as query features. It also supports the combinations of these features and allows the users to adjust the weights associated with each feature. Its demo page is at `http://vrw.excalib.com/cgi-bin/sdk/cst/cst2.bat`.

VisualSeek and WebSeek

VisualSEEk (Smith and Chang, 1996) is a visual feature search engine and WebSEEk (Smith and Chang, 1997) is a World Wide Web oriented text/image search engine, both of which have been developed at Columbia University. Main research features are spatial relationship query of image regions and visual feature extraction from compressed domain. The visual features used in their systems are color sets and wavelet transform-based texture features. To speed up the retrieval process, they also developed binary tree-based indexing algorithms. VisualSEEk supports queries based on both visual features and their spatial relationships. This enables a user to submit a sunset query as red-orange color region on top and blue or green region at the bottom as its ”sketch”. WebSEEk is a web-oriented search engine. It consists of three main modules, i.e. image/video collecting module, subject classification and indexing module, and search, browse, and retrieval module. It supports queries based on both keywords and visual content. The on-line demos are at `http://www.ee.columbia.edu/sfchang/demos.html`.

Photobook

Photobook (Pentland et al., 1996) is a set of interactive tools for browsing and searching images developed at the MIT Media Lab. Photobook consists of three subbooks from which shape, texture, and face features are extracted, respectively. Users can then query on the basic of the corresponding features in each of the three subbooks. In its more recent version of Photobook, FourEyes, Picard et al. proposed including the human users in the image annotation and retrieval loop. The motivation for this was based on the observation that there was no single feature which can best model images from each and every domain. Furthermore, human perception is subjective. They proposed a ”society of models” approach to incorporate the human factor. Experimental results show that this approach is effective in interactive image annotation.

Netra

Netra is a prototype image retrieval system developed in the UCSB Alexandria Digital Library (ADL) project (Ma and Manjunath, 1997). Netra uses color, texture, shape, and spatial location information in the segmented image regions to search and retrieve similar regions from the database. Main research features of the Netra system are its Gabor filter-based texture analysis, neural net-based image thesaurus construction and edge flow-based region segmentation. The on-line demo is at `http://maya.ece.ucsb.edu/Netra/netra.html`.

Chapter 4

ESTIMATING COLOR DISTRIBUTIONS FOR IMAGE RETRIEVAL

In content-based image retrieval applications, the color properties of an image are very often characterized by the probability distribution of the colors in the image. These probability distributions are usually estimated by histograms although the histograms have many drawbacks compared to other estimators such as kernel density methods.

In this chapter we investigate whether using kernel density estimators instead of histograms could give better descriptors of color images. Experiments using these descriptors to estimate the parameters of the underlying color distribution and in color-based image retrieval (CBIR) applications were carried out in which the MPEG-7 database of 5466 color images with 50 standard queries are used as the benchmark. Noisy images are also generated and put into the CBIR application to test the robustness of the descriptors against noise. The results of our experiments show that good density estimators are not necessarily good descriptors for CBIR applications. We found that the histograms perform better than simple kernel-based method when used as descriptors for CBIR applications. Two modifications to improve the simple kernel-based method are proposed. Both of them show a better retrieval performance in our experiments.

In the second part of the chapter, optimal values of important parameters in the construction of these descriptors, particularly the smoothing parameters or the bandwidth of the estimators, are discussed. Our experiments show that using over-smoothed bandwidth gives better retrieval performance.

4.1 Introduction

Color is widely used for content-based image retrieval. In these applications the color properties of an image are characterized by the probability distribution of the colors in the image. These probability distributions are very often approximated by histograms (Rui et al., 1999; Schettini et al., 2000). Well-known problems of histogram-based methods are: the sensitivity of the histogram to the placement of the bin edges, the discontinuity of the histogram as a step function and its deficiency in using data in estimating the underlying distributions compared to other estimators (Silverman, 1986; Scott, 1992; Wand and Jones, 1995).

These problems can be avoided by using other methods such as kernel density estimators. To the best of our knowledge there are, however, only a few papers (Gevers, 2001) that use kernel density estimators in image retrieval. So the question is thus why methods like kernel density estimators are not more widely used in estimating the color distributions in image retrieval applications; even though they have theoretical advantages in estimating the underlying color distributions? Is it because kernel density estimators are time-consuming or are kernel based methods unsatisfactory for image retrieval?

In this chapter we first compare the performance of histograms and different kernel density estimator methods in describing the underlying color distribution of images for image retrieval applications. Our experiments show that simple kernel-based methods using a set of estimated values at histogram bin centers give bad retrieval performance. We therefore propose two different kernel-based methods to improve the retrieval performance. These new methods are based on the use of non-orthogonal bases together with a Gram-Schmidt procedure and a method applying the Fourier transform.

Like other density estimators, the histograms and kernel density estimators are both sensitive to the choice of the smoothing parameter (Silverman, 1986; Scott, 1992; Wand and Jones, 1995). This parameter in turn influences the retrieval performance of CBIR applications. Our experiments show that the proposed methods do not only lead to an improved retrieval performance but that they are also less sensitive to the selection of the smoothing parameter. In particular the retrieval performance of the Fourier-based method for hue distribution is almost independent of the value of the smoothing parameter if it lies in a reasonable range. For histogram-based methods, we investigate the selection of the optimal number of histogram bins for CBIR. This parameter was previously often chosen heuristically without explanation (Rui et al., 1999; Schettini et al., 2000). We will also show that the previously applied strategy (Brunelli and Mich, 2001) of applying statistical methods to find the theoretically optimal number of bins (Sturges, 1926; Scott, 1979; Rudemo, 1982; Scott, 1985;

Devroye and Gyorfi, 1985; Scott, 1992; Kanazawa, 1993; Wand, 1996; Birge and Rozenholc, 2002) to image retrieval applications requires further research.

The chapter is organized as follows: in the next section, histogram and kernel-based methods are briefly described. Their performance in CBIR applications are compared in section 4.3. Section 4.4 presents our proposed kernel-based methods to improve the retrieval performance. The discussion of the optimal bin-width of the histogram is continued in Section 4.5 with emphasis on color-based image retrieval applications.

4.2 Non-parametric Density Estimators

Methods to estimate probability distributions can be divided into two classes, parametric or non-parametric methods. Parametric density estimation requires both proper specification of the form of the underlying sampling density $f_\theta(x)$ and the estimation of the parameter vector θ. Usage of parametric methods has to take into account two types of bias: the estimation of θ and incorrect specification of the model f_θ. Non-parametric methods make no assumptions about the form of the probability density functions from which the samples are drawn. Non-parametric methods require therefore more data than parametric methods because of the lack of a ”parametric backbone”. A typical color image contains 100,000 color pixels, and the structure of its underlying distribution can (and will) vary from image to image. Therefore non-parametric methods are more attractive in estimating color distributions of images.

4.2.1 Histogram

The oldest and most widely used non-parametric density estimator is the histogram. Suppose that $\{X_1, ..., X_N\}$ is a set of continuous real-valued random variables having common density f in an interval (a, b). Let $\mathbb{I} = \{I_m\}_M$ be a partition of (a, b) into M disjoint, equally sized intervals, often called bins, such that $a = t_0 < t_1 < \ldots < t_M = b, t_{i+1} = t_i + (b - a)/M$. Let h denote the length of the intervals, also called the smoothing parameter or the bin-width, and $H_m = \#\{n : X_n \in I_m, 1 \leq n \leq N\}$ be the number of observations in bin I_m. The histogram estimator of f, with bin-width h and based on the regular partition $\mathbb{I} = \{I_m\}_M$ at a point $x \in I_m$ is given by:

$$\hat{f}_H(x, h) = \frac{1}{Nh} \cdot H_m \tag{4.1}$$

One of the main disadvantages of histograms is that they are step functions. The discontinuities of the estimate originate usually not in the underlying density but are often only artifacts of the selected bin locations. To overcome this limitation the frequency polygon was proposed in (Scott, 1992). It is the continuous version of the histogram which is formed by interpolating the midpoints of a histogram. Still, both the histogram and the frequency polygon share their dependency on the choice of the positioning of the bin edges, especially for small sample sizes. For multivariate data, the final shape of the density estimate is also affected by the orientation of the bins. For a fixed bin-width, there is an unlimited number of possible choices of placements of the bin edges. Further information about the effect of the placement of bin edges can be found in (Simonoff and Udina, 1997). In (Scott, 1985) Scott proposed averaging over shifted meshes to eliminate the bin edge effect. This can be shown to approximate a kernel density estimator which is described in the next section.

4.2.2 Kernel Density Estimators

We define a kernel as a non-negative real function K with $\int K(x)dx = 1$. Unless specified otherwise, integrals are taken over the entire real axis. The kernel estimator $\hat{f}_K$ at point x is defined by

$$\begin{aligned} \hat{f}_K(x,h) &= \frac{1}{Nh}\sum_{n=1}^{N} K\{(x - X_n)/h\} \\ &= \frac{1}{N}\sum_{n=1}^{N} K_h(x - X_n) \end{aligned} \tag{4.2}$$

As before, h denotes the window width, also called the smoothing parameter or the bandwidth, and N denotes the number of sample data and the scaled kernel is $K_h(u) = h^{-1}K(u/h)$.

The kernel K is very often taken to be a symmetric, unimodal density such as the normal density. There are many different kernel functions but in most applications their performance is comparable. The choice between kernels is therefore often based on other grounds such as computational efficiency (Wand and Jones, 1995). Multivariate densities can also be estimated by using high dimensional kernels.

The analysis of the performance of estimators requires the specification of appropriate error criteria for measuring the error when estimating the density at a single point as well as the error when estimating the density over the whole real line. The mean squared error (MSE) and its expected value, mean integrated squared error (MISE) are widely used for this purpose. Scott shows the deficiency of the histogram method over the kernel density estimator (Scott, 1979). The mean integrated squared error (MISE) of the histogram is asymptotically inferior to the kernel density estimator since its convergence rate is $O(n^{-2/3})$ compared to the kernel estimator's $O(n^{-4/5})$ rate.

Naturally the superior performance of kernel density estimators in estimating the underlying probability distributions over the histogram-based method suggests that the application of kernel based methods in CBIR instead of histograms might improve the retrieval performance. In the next section we will examine whether better estimators always give better retrieval performance or not.

4.3 Density Estimators for CBIR

Our aim is to describe the color information of images by a set of numbers and use these numbers for indexing the image database. For histograms, we can use the histogram values as the descriptors of the images. For kernel density estimators there are many more options to choose such a set of numbers. A straightforward way is to sample them at points on a grid such as the centers of the corresponding histogram bins. These descriptors, derived from histograms and kernel-based methods, are compared in the following experiments.

In the experiments we first computed the hue values from RGB images using the conversion in (Plataniotis and Venetsanopoulos, 2000, p.30). The following sets of 16 numbers are computed to represent the hue distribution in an image:

- The histogram-based method uses 16 bins of one-dimensional hue histograms with bin-width = 1/16. The bin centers are located at $X_{16} = \{1/32 : 1/16 : 31/32\}$ (in Matlab-notation).
- The kernel-based method uses the normal density as the kernel to estimate the values of the hue distributions at the 16 positions X_{16}. The bandwidths are chosen by using either a constant bandwidth for all color images in the database, or using different bandwidths for different images. In this case the bandwidth is optimized for each image and a normalization process is needed to compensate the differences between bandwidths. Here we normalized the coefficients by a factor so that their sum equals 1.

There are many methods of automatically selecting an "Optimal" value of the bandwidth h for kernel density estimators but none of them is the overall "best" method. Wand and Jones (Wand and Jones, 1995) suggest that the Solve-The-Equation (STE) method offers good overall performance. We chose STE to find the optimal bandwidth in this set of experiments (it should, however, be mentioned that the STE method is worst when the underlying distribution has large peaks, which is not the case for hue and color distributions of images).

For each of the kernel-based methods mentioned above, an optimal bandwidth value is chosen together with an over-smoothed, (10% of the optimal value), and an under-smoothed, (10 times the optimal value) value. In total seven methods are compared. The histogram-based estimation is denoted by H and the kernel-based methods by K_x. In detail the experiments are denoted as follows:

H Histogram method using bin centers at $X_{16} = \{1/32 : 1/16 : 31/32\}$

K_I Kernel-based method using Eq. 4.2 to estimate hue density at 16 positions X_{16}. Optimal bandwidths are computed for each image using the STE algorithm.

K_{IU} The same as K_I except using an undersmoothed bandwidth, which is 10% of the optimal bandwidth for the image.

K_{IO} Bandwidth is oversmoothed, ie. 10 times the optimal bandwidth for the image.

K_D Bandwidth is the mean value of the optimal bandwidths for all images in the database.

K_{DU} Bandwidth is undersmoothed, ie. 10% of the value used in K_D

K_{DO} Bandwidth is oversmoothed, ie. 10 times the value used in K_D

These descriptors are then used to describe color images in an image retrieval application. The MPEG-7 database with 5466 color images and 50 standard queries is used to compare the retrieval performance of different methods. The average results are shown in Table 4.1.

Method	ANMRR
H	0.38
K_{IU}	0.57
K_I	0.47
K_{IO}	0.43
K_{DU}	0.54
K_D	0.45
K_{DO}	0.38

Table 4.1: Compare histogram and standard kernel-based method in CBIR. ANMRR of 50 standard queries.

In all our CBIR experiments, the Euclidian distance of their descriptors is used to compute the distance between images. The retrieval performance is measured using the Average Normalized Modified Retrieval Rank (ANMRR). The detailed description of ANMRR has been presented in section 3.5. Just mentioned briefly that the lower values of ANMRR indicate better retrieval performance.

We also did the same experiments for the two-dimensional chromaticity descriptors xy (from the CIEXYZ system). In this case we used $8 \times 8 = 64$ numbers as descriptors. For three-dimensional RGB color distributions we computed an $8 \times 8 \times 8 = 512$ dimensional description of all images in the MPEG-7 database. The results are collected in Table 4.2.

Method	(x,y)	RGB
H	0.38	0.23
K_{IU}	0.69	0.77
K_I	0.62	0.70
K_{IO}	0.64	0.54
K_{DU}	0.69	0.79
K_D	0.56	0.71
K_{DO}	0.41	0.45

Table 4.2: Retrieval performance of different methods in CBIR using estimated chromaticity density (xy) and RGB density as the color descriptors of images.

In the next experiment, we selected a set of 20 images, 10 of them from standard queries, and the other 10 were standard image processing images such as Lenna, Peppers, Mandrill, Parrots, etc. From each of these 20 images a new set of 20 images was generated by adding noise and sub-sampling the images. This resulted in a set of 420 images. The parameters that control the generated images are:

- the percentage of sampled pixels
- the percentage of pixels with added noise and
- the range of the noise magnitudes

The noise is uniformly distributed. Each set of 20 generated images is intended to have similar color distributions as the original image. We then take these 20 images as the ground truth when retrieving the original image. The average results of 20 different queries are collected in Table 4.3.

Our experiments show that histogram-based methods outperform simple kernel-based methods in color-based image retrieval applications. This may be one of the reasons why we found only one paper (Gevers, 2001) using kernel-based methods for image retrieval (in this paper kernel-based methods are shown to be robust against noise in image retrieval application using a small dataset of 500 images). Another reason is that kernel-based methods are very time-consuming. Using the KDE toolbox (Baxter et al., 2000) each kernel-based method takes about two days of computation with a standard PC to estimate the color distributions at 512 points of all images in the MPEG-7 database.

Method	Hue desc.	(a,b) desc.	RGB desc.
H	0.36	0.20	0.16
K_{IU}	0.64	0.68	0.66
K_I	0.44	0.53	0.68
K_{IO}	0.52	0.52	0.32
K_{DU}	0.53	0.63	0.67
K_D	0.42	0.53	0.65
K_{DO}	0.31	0.28	0.25

Table 4.3: Compare histogram and standard kernel-based method in CBIR. ANMRR of 20 queries based on 420 noise-generated images.

4.4 Series Expansions and Kernel-based Descriptors in CBIR

Computational complexity and low retrieval performance are the two main reasons that suggest that histograms are better for CBIR than kernel-based descriptors. The features are, however, only computed once when the images are entered into the database and they can therefore be computed off-line. The limited retrieval performance is more critical. In this section we present two applications of kernel density estimators in CBIR and show that they improve the retrieval performance in CBIR and make them superior to the histogram method.

4.4.1 Basis expansions

Instead of simply using the estimated values of the underlying distribution at only few specific values, one could expand the full distribution using M coefficients $\{\alpha_m\}_M$ in a series expansion (in some predefined system given by basis functions $\{b_m(x)\}_M$)

$$f(x) \approx \hat{f}_K(x) = \sum_{m=1}^{M} \alpha_m b_m(x) \tag{4.3}$$

If the basis functions $\{b_m(x)\}_M$ is orthogonal, the coefficients $\{\alpha_m\}_M$ can be computed simply as

$$\alpha_m = \left\langle \hat{f}_K, b_m \right\rangle \tag{4.4}$$

If the basis functions $\{b_m(x)\}_M$ is not orthogonal, the Gram-Schmidt algorithm can be used to compute the coefficients $\{\alpha_m\}_M$ as follows:

$$
\begin{aligned}
\alpha_1 &= \left\langle \hat{f}_K, b_1 \right\rangle \\
\alpha_2 &= \left\langle \hat{f}_K - \alpha_1 \cdot b_1, b_2 \right\rangle \\
&= \left\langle \hat{f}_K, b_2 \right\rangle \ - \ \alpha_1 \cdot \langle b_1, b_2 \rangle \\
&\ldots\ldots \\
\alpha_m &= \left\langle \hat{f}_K, b_m \right\rangle \ - \ \sum_{i=1}^{m-1} \alpha_i \langle b_i, b_m \rangle
\end{aligned} \tag{4.5}
$$

Here $\langle f, g\rangle$ denotes the scalar product of the functions $f(x)$ and $g(x)$. In the following it is mainly defined as the integral $\langle f, g\rangle = \int f(x)g(x)\ dx$ but other definitions are possible and useful. Since both the functions $\{b_m(x)\}_M$ and the kernel K are known the coefficients $\{\alpha_m\}_M$ can be analytically computed using Eq. 4.5. For the case where the kernel is a Gaussian and the basis functions are shifted Gaussians centered equally at $\{Y_m\}_M$ using the same standard deviation s:

$$
b_m(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x - Y_m)^2}{2s^2}\right\} \tag{4.6}
$$

they are computed with the help of the following derivations:

$$
\hat{f}_K(x) = \frac{1}{N}\sum_{n=1}^{N} K_h(x - X_n) = \sum_{m=1}^{M} \alpha_m b_m(x) \tag{4.7}
$$

Here the first equation is the definition of the density estimate and the second equation describes the fact that the estimate is expanded in the shifted Gaussians. We thus have:

$$
\begin{aligned}
\left\langle \hat{f}_K, b_m \right\rangle &= \int \hat{f}_K(x) b_m(x) dx \\
&= \frac{1}{(2\pi N h)} \sum_{n=1}^{N} \int \exp\left\{-\frac{(x - X_n)^2}{2h^2} - \frac{(x - Y_m)^2}{2s^2}\right\} dx \\
&= \frac{s}{N\sqrt{2\pi(h^2 + s^2)}} \sum_{n=1}^{N} \exp\left\{-\frac{(X_n - Y_m)^2}{2(h^2 + s^2)}\right\}
\end{aligned} \tag{4.8}
$$

and

$$
\langle b_k, b_l \rangle = \int b_k(x) b_l(x) dx = \frac{s}{2\sqrt{\pi}} \cdot \exp\left\{-\frac{(Y_k - Y_l)^2}{4s^2}\right\} \tag{4.9}
$$

The Gram-Schmidt procedure in Eq. 4.6 can be easily extended to higher dimensions. The following is the solution for the d-dimensional case:

$$\hat{f}_K(x_1,\ldots,x_d) = \frac{\sum_{n=1}^{N} K(x_1 - X_{1i},\ldots,x_d - X_{di})}{N\prod_{j=1}^{d} h_j} \tag{4.10}$$

$$\begin{aligned} &b_m(x_1,\ldots,x_d) = \\ &(2\pi)^{-d/2}\exp\big\{-\frac{(x_1 - Y_{m1})^2 + \ldots + (x_d - Y_{md})^2}{2s^2}\big\} \end{aligned} \tag{4.11}$$

$$\begin{aligned} &\Big\langle \hat{f}_K(x_1,\ldots,x_d), b_m(x_1,\ldots,x_d)\Big\rangle = \\ &\frac{1}{N}\left(\prod_{j=1}^{d}\frac{s}{\sqrt{2\pi(h_j^2+s^2)}}\right)\cdot\sum_{n=1}^{N}\exp\left\{-\sum_{j=1}^{d}\frac{(X_{ij}-Y_{mj})^2}{2(h_j^2+s^2)}\right\} \end{aligned} \tag{4.12}$$

$$\langle b_k, b_l\rangle = \left(\frac{s}{2\sqrt{\pi}}\right)^d \exp\left\{-\sum_{j=1}^{d}\frac{(Y_{kj}-Y_{lj})^2}{4s^2}\right\} \tag{4.13}$$

We tested these algorithms in an experiment using the hue distributions from the MPEG-7 database. Here we have to specify two parameters: the smoothing parameter h and the width s of the Gaussian. Table 4.4 presents some of the results with different values of h and s.

s	h/h_{opt}	ANMRR	s	h/h_{opt}	ANMRR
0.01	0.1	0.410	0.05	0.1	0.388
0.01	0.3	0.409	0.05	0.3	0.388
0.01	1	0.406	0.05	1	0.389
0.01	3	0.397	0.05	3	0.391
0.01	10	0.370	0.05	10	0.403
0.025	0.1	0.373	0.1	0.1	0.480
0.025	0.3	0.373	0.1	0.3	0.480
0.025	1	0.373	0.1	1	0.480
0.025	3	0.371	0.1	3	0.481
0.025	10	0.374	0.1	10	0.491

Table 4.4: Gram-Schmidt method for hue distributions of MPEG-7 database.

Our experiments show that with good choices of the smoothing parameter h and the width of the Gaussian s, the Gram-Schmidt-based method gives a better retrieval performance than the histogram and simple kernel-based methods. For example if h is chosen as 10 times the optimal value given by the STE algorithm and $s = 0.01$, then the ANMRR is 0.37 which is smaller than values given by both histogram and simple kernel-based method given in Table 4.1

The experiments also show that the Gram-Schmidt method is less sensitive to the choice of the smoothing parameter h compared to the simple kernel-based methods. However, it is still sensitive to the choice of the basis which is the width s of the Gaussians in this example.

4.4.2 Fourier transform-based method

Using the Fourier transform is another way to describe the estimated hue distributions. It is well-known that the Fourier transform is the optimal transform for many problems that are (like the hue distributions) defined on a circle. In our application it is especially interesting that the Fourier transform of shift-invariant processes is closely related to the Karhunen-Loéve transform of these processes. Computing the Fourier coefficients of the hue-distributions and keeping only the most important coefficients is thus a promising approach to obtaining a compressed description of hue distributions. This approach will be developed in the following.

Given the estimated hue distribution

$$f_K(x,h) = (Nh)^{-1} \sum_{n=1}^{N} K\left\{(x - X_n)/h\right\}$$

as in Eq. 4.2, its Fourier transform $\mathcal{F}_K(y,h)$ is computed as the follows:

$$\begin{aligned}
\mathcal{F}_K(y,h) &= \int \hat{f}_K(x,h) \cdot \exp(-ixy)dx \\
&= \frac{1}{Nh} \int \sum_{n=1}^{N} K\{(x - X_n)/h\} \cdot \exp(-iyx)dx \\
&= \frac{1}{N} \sum_{n=1}^{N} \int K(t) \cdot \exp\{-iy(ht + x_n)\}dt \qquad (4.14) \\
&= \frac{1}{N} \left\{ \sum_{n=1}^{N} \exp(-iyx_n) \right\} \int K(t) \cdot \exp(-iyht)dt \\
&= \frac{1}{N} \left\{ \sum_{n=1}^{N} \exp(-iyx_n) \right\} \mathcal{K}(yh)
\end{aligned}$$

where $\mathcal{K}$ is the Fourier transform of the kernel K. It should be noted here that the factor $\sum_{n=1}^{N} \exp(-iyx_n)$ of the Fourier transform $\mathcal{F}_K(y,h)$ in Eq. 4.14 is independent of the kernel and the smoothing parameter h. It can thus be computed from the data once and then new estimates with different kernels and smoothing parameters can be computed without accessing the data again.

The distance between two images I_1, I_2 is defined as the distance between the two corresponding hue distributions $f_1(x,h)$ and $f_2(x,h)$. Using Parseval's formula it is given by

$$\begin{aligned} d(I_1,I_2) &= d(f_1(x,h), f_2(x,h)) = \langle f_1(x,h), f_2(x,h) \rangle \\ &= \frac{1}{2\pi} < \mathcal{F}_1(y,h), \mathcal{F}_2(y,h) > \end{aligned} \tag{4.15}$$

In our case the Fourier transform is actually a Fourier series since the functions are all defined on the circle. We can thus describe the two Fourier transforms by selecting the coefficients of the most important frequencies

$$\left\{\eta_{(1,m)}, \eta_{(2,m)}\right\} \qquad \text{with } m = 0, \ldots M$$

and approximate the distance between the two images by the inner product of two low dimensional vectors:

$$d(I_1,I_2) \approx \frac{1}{2\pi} \sum_m \eta_{(1,m)} \cdot \overline{\eta_{(2,m)}} \tag{4.16}$$

Method	M_L	M_D
Biweight kernel, $h = 0.2$	0.4786	0.4954
Biweight kernel, $h = 0.05$	0.4749	0.4946
Biweight kernel, $h = 0.008$	0.4748	0.4945
Biweight kernel, $h = 0.0056$	0.4748	0.4945
Biweight kernel, $h = 0.001$	0.4748	0.4945
Biweight kernel, $h = 0.0002$	0.4748	0.4945
Triangular kernel, $h = 0.001$	0.4748	0.4945
Normal kernel, $h = 0.001$	0.4748	0.4945
Epenechnikov kernel, $h = 0.001$	0.4748	0.4945

Table 4.5: The retrieval performance improvement of the M_L method over M_D method of selecting the coefficients of the most three important frequencies for CBIR.

The straightforward way (we call this method M_D) of selecting the coefficients of the most important frequencies is to take the lowest frequencies, which gives us the best solution to reconstruct the underlying density. However, it has been shown in (Tran and Lenz, 2001b) that for image retrieval applications where only similar images are of interest, the retrieval performance can be improved by choosing the frequencies which give the best solution for reconstructing the differences between similar densities. We call this method M_L. In detail the coefficients of the most important frequencies in M_L method are obtained as follows:

- 100 images are randomly chosen from the image database, called set S. Take each image in set S as the query image and find the 50 most similar images from the database.
- Estimate the differences between the query image and the 50 most similar images, and their Fourier coefficients. Totally $100 \times 50 = 5000$ entries are computed.
- The coefficients of the most important frequencies are selected as the frequencies which give the biggest mean of the magnitude for the whole set of the above 5000 entries.

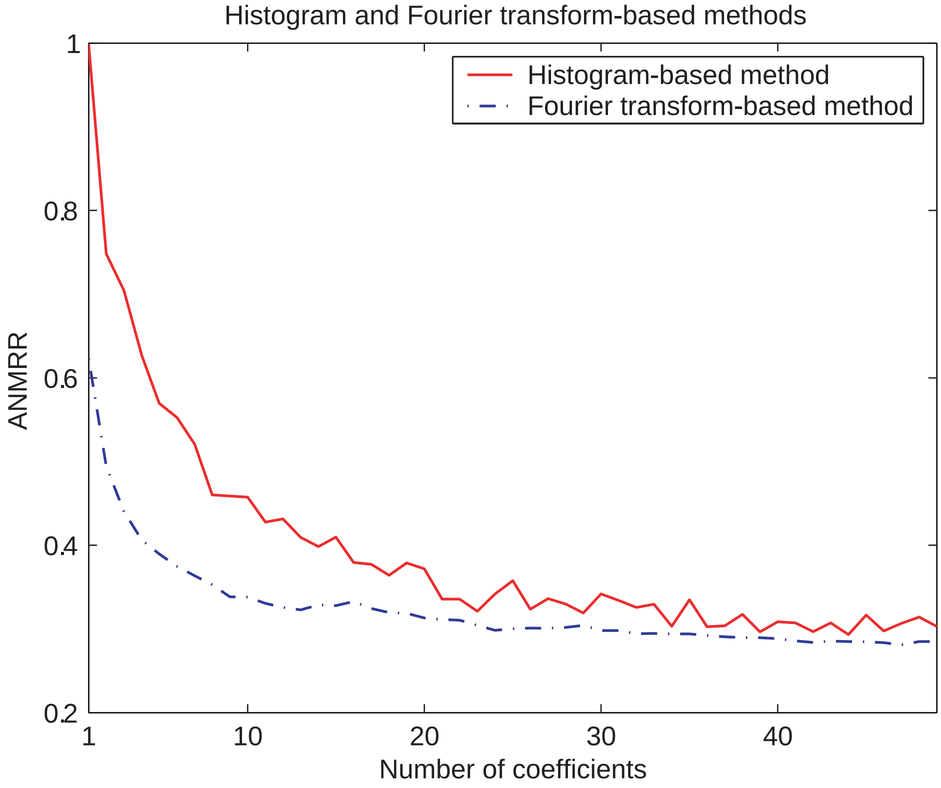


Figure 4.1: Retrieval performance of histogram and Fourier transform-based method using triangular kernel, the smoothing parameter $h = 0.0056$.

Our experiments show that only small improvement is achieved by using the M_L method. The most clear case is when 3 coefficients are used. Some of the comparisons are presented in Table 4.5. This method, too, can be generalized to higher dimensional spaces of chromaticity (2-D) and color distributions (3-D). In the following we select a few results obtained in our experiments.

We evaluated the performance of the method with the MPEG-7 database. Fig. 4.1 shows an example of the retrieval performance of the Fourier transform method using a triangular kernel with smoothing parameter $h = 0.0056$. It shows that the Fourier Transform method has a better performance than the histogram method, especially for a small number of parameters. The next figure, Fig. 4.2, illustrates the dependency of the retrieval performance of the Fourier transform-based method on the smoothing parameter h. The different curves correspond to different numbers of Fourier coefficients used.

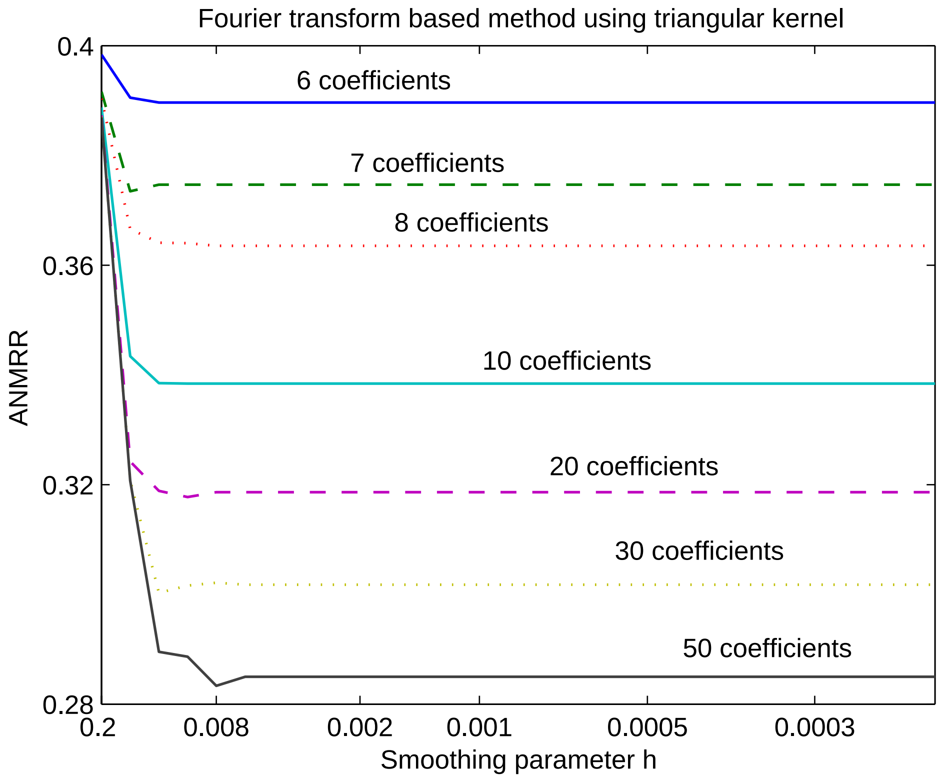


Figure 4.2: Retrieval performance of Fourier transform-based method using triangular kernel with different smoothing parameters.

From the results of our experiments we draw the following conclusions:

- Using the same number of coefficients, the Fourier transform-based method gives a better retrieval performance than the histogram and the Gram-

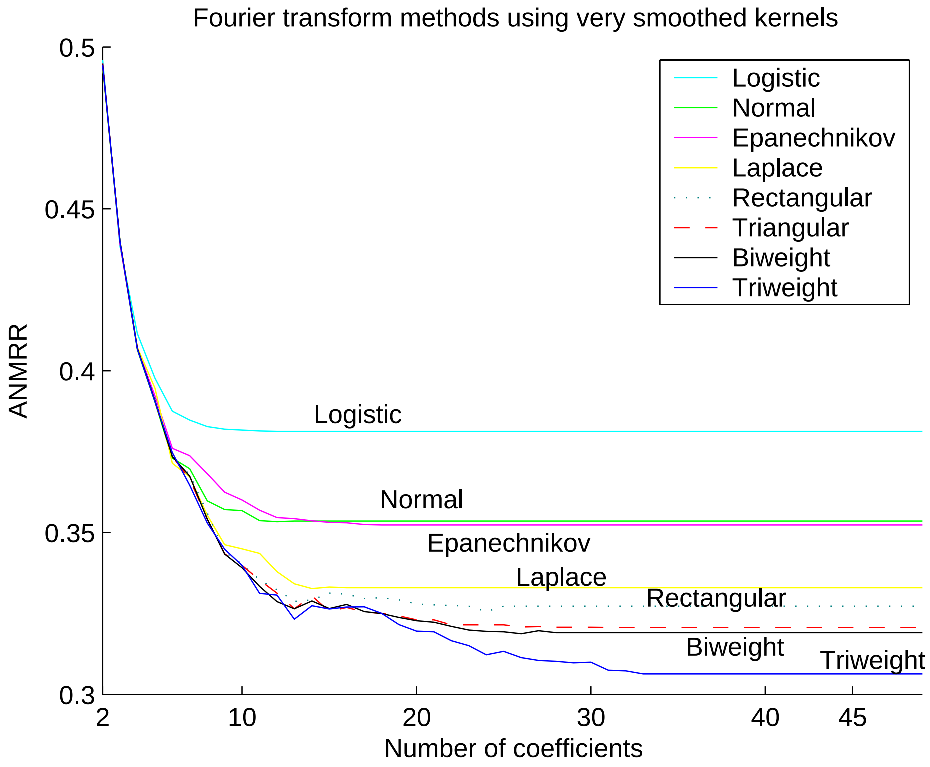


Figure 4.3: Retrieval performance of Fourier transform-based method using different kernels with smoothing parameter $h = 0.05$.

Schmidt method. For example using 10 Fourier coefficients gives a retrieval performance as good as using 23 coefficients from a histogram. This is illustrated in Fig. 4.1.

- Using a larger smoothing parameter h gives a better retrieval performance. However the performance does not change for h below 0.005. We tested 30 different smoothing parameters ranging from 0.0001 to 0.2. Fig. 4.2 shows how the retrieval performance depends on both the number of coefficients and the smoothing parameters.

- Using different kernels gives comparable retrieval performance when the kernel is not over-smoothed. When $h < 0.01$ all kernels had identical retrieval properties. Seven different kernels (Epenechnikov, Biweight, Triweight, Normal, Triangular, Laplace, Logistic, detailed definition of kernels can be found in (Wand and Jones, 1995)) were tested. Fig. 4.3 illustrates the retrieval properties of different kernels when an over-smoothed kernel with $h = 0.05$ is used. For values of h below 0.01 there is no difference between the different kernels.

4.5 Optimal Histogram Bin-width

For very large image databases it is computationally very expensive to use the kernel density estimators to estimate the color distributions for all images in the database. Thus the histogram method, which is much faster and gives comparable retrieval performance, is still an attractive alternative in many cases. We saw earlier that the most important, freely selectable parameter is the size of the bin width. It is often selected by rules of thumb or using statistical methods. In this section we investigate the retrieval performance of several rules to select the bin width. We will show that these existing statistical methods are not very useful in image database retrieval applications since their goal is the faithful description of statistical distributions whereas the goal of the database search is a fast comparison of different distributions.

Finding an optimal number of bins of histograms is an active research problem in statistics. There are many papers in the field describing how to find this optimal number of bins in order to estimate the underlying distribution of given generic data (Sturges, 1926; Akaike, 1974; Scott, 1979; Rudemo, 1982; Scott, 1985; Devroye and Gyorfi, 1985; Scott, 1992; Kanazawa, 1993; Wand, 1996; Birge and Rozenholc, 2002). It is optimal in the sense of minimizing some statistics-based error criteria (such as the MSE or MISE). In most CBIR papers, however, this parameter is selected without further comment. One paper that investigates this problem is (Brunelli and Mich, 2001) in which two algorithms (Sturges, 1926; Scott, 1979) have been applied to find the optimal number of bins of histograms. The reason why they are appropriate for CBIR applications is also discussed in this paper.

The first, and oldest, method they used is a rule of thumbs suggested by Sturge (Sturges, 1926). It is given by:

$$h = \frac{\Delta}{1 + \log_2(n)} \tag{4.17}$$

where Δ is the range of the data, n is the number of data entries and it gives $1 + \log_2(n)$ bins. Such methods are still in use in many commercial software packages for estimating distributions although they do not have any type of optimality property (Wand, 1996). The optimal number of bins (Sturges, 1926) given by Sturges depends mainly on the number of data entries, which is the size of the image in this case. For small sized images of around 200x160 pixels, it always gives around 16 bins independently of the properties of the underlying color distribution.

The second method they used was introduced by Scott (Scott, 1979) with an optimal bin-width:

$$h_{Scott} = 3.49\hat{\sigma}n^{-1/3} \tag{4.18}$$

where $\hat{\sigma}$ is an estimate of the standard deviation. This method is similar to some other methods like (Devroye and Gyorfi, 1985; Kanazawa, 1993; Freedman and Diaconis, 1981). They are all based on the evaluation of the optimal asymptotic value of the bin-width. In such methods, unfortunately, the optimal bin-width is asymptotically of the form $\hat{C}n^{-1/3}$ where $\hat{C}$ is a function of the unknown density to be estimated and its derivative. Since an estimation of $\hat{C}$ involves complicated computations, most authors suggest a rule of thumbs to evaluate it, typically pretending that the true density is normal. Some optimal bin width of the estimators in these classes are given below:

$$h_{Devroye} = 2.72\hat{\sigma}n^{-1/3} \tag{4.19}$$

$$h_{Kanazawa} = 2.29\hat{\sigma}n^{-1/3} \tag{4.20}$$

where $\hat{\sigma}$ is an estimate of the standard deviation. A more robust version is given by Freedman and Diaconis (Freedman and Diaconis, 1981) using the Inter-Quartile Range (IQR) value:

$$h_{Freedman} = 2\text{IQR}n^{-1/3} \tag{4.21}$$

Method	hue	x	y	R	G	B
Scott as in Eq. 4.18	68	170	231	65	67	67
Freedman in Eq. 4.21	87	293	559	78	82	81
Devroye in Eq. 4.19	87	118	297	83	86	86
Kanazawa in Eq. 4.20	103	259	353	098	102	102
Scott in Eq. 4.23	68	67	89	14	15	15
Akaike in Eq. 4.22	749	-	-	-	-	-
Birge [Birge 2002]	681	-	-	-	-	-

Table 4.6: Theoretically optimal number of bins.

There are other classes of methods for estimating the optimal number of bins. Methods based on cross-validation have the advantage of avoiding the estimation of an asymptotic function and directly provide a bin-width from the data (Rudemo, 1982). Different penalties like the ones in (Akaike, 1974; Birge and Rozenholc, 2002) can also be used to improve the results, see (Birge and Rozenholc, 2002) for a comparison of different methods in estimating probability distributions. The optimal number of bins estimated by Akaike's method is given by

$$nb_{Akaike} = \sup\left\{\sum_{k}^{nb} N_k \log(\frac{nb \cdot N_k}{n}) + 1 - nb\right\} \tag{4.22}$$

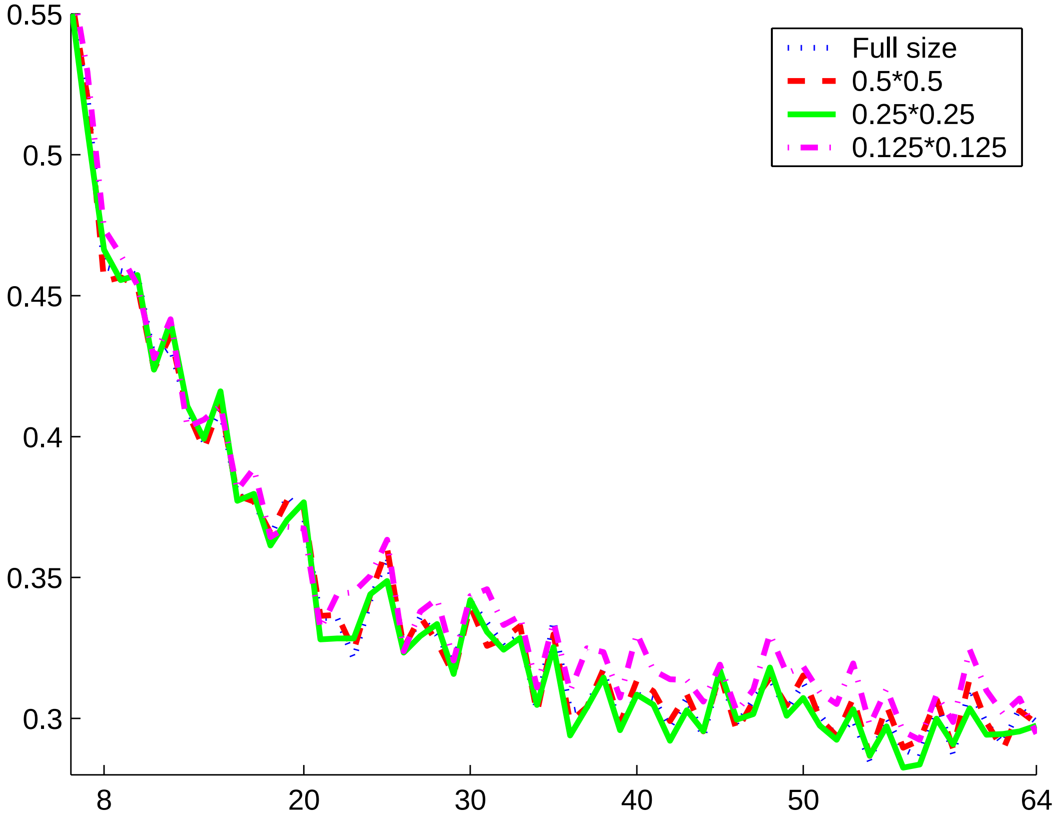


Figure 4.4: Average value of ANMRR of 50 standard queries on the MPEG-7 database. Images are described by one-dimensional hue histograms using different numbers of bins ranging from 8 to 64 and different down-sampling methods to test the effect of image size on retrieval performance. For each image, 4 hue histograms are computed from: 1-the original image, 2-the down-sample image with sampling factor $k = 1/2 = 0.5$ in both vertical and horizontal directions, 3-the down-sample image with $k = 1/4 = 0.25$, and 4-the down-sample image with $k = 1/8 = 0.125$.

There are few investigations of multivariate histograms. Scott (Scott, 1992) has proposed an algorithm for estimating the optimal bin-width of multivariate histograms as follow:

$$h_{ScottM} = 3.49\hat{\sigma}n^{-\frac{1}{2+d}} \tag{4.23}$$

where d is the number of dimensions of the underlying multivariate histograms.

Using the above procedures, we computed the theoretical optimal bin-width for the estimation of the hue, (x, y), and (R, G, B) distributions of the images in the MPEG-7 database. The results are collected in Table 4.6.

In order to evaluate the optimal number of bins given by statistics-based method for CBIR, We did some experiments using different bin-width for color image retrieval. These results will then be compared to the results obtained from the statistical methods shown in Table 4.6. We used the MPEG-7 database with 50 standard queries. Images are described by hue, (x,y), and RGB color histograms using different bin-widths. The results are collected in Fig. 4.5, 4.6, and 4.7. They showed that the empirical methods give much smaller values than the values given by statistical methods (Akaike, 1974; Scott, 1979; Rudemo, 1982; Scott, 1985; Devroye and Gyorfi, 1985; Scott, 1992; Kanazawa, 1993; Wand, 1996; Birge and Rozenholc, 2002).

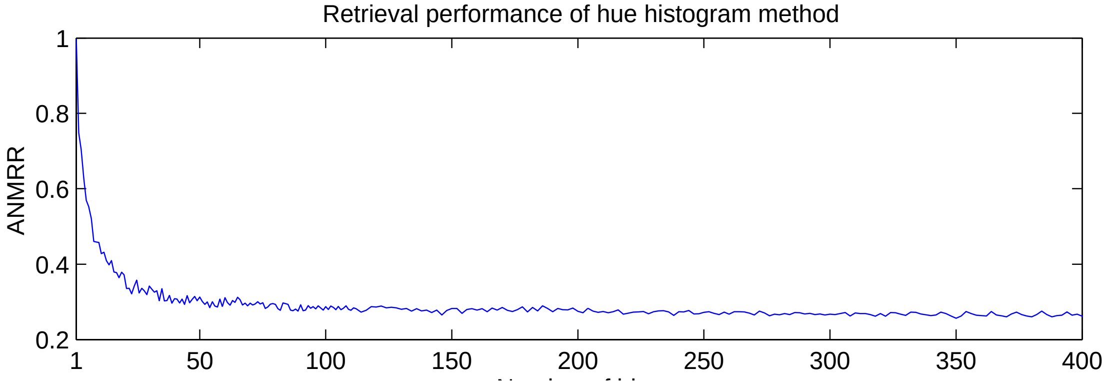


Figure 4.5: Average of ANMRR of 50 standard queries on the MPEG-7 database. Images are described by one-dimensional hue histograms using different numbers of bins ranging from 1 to 400. A closer look at values between 1 and 50 is shown in Fig. 4.1. Values between 20 and 30 seem to be the best number of bins of one-dimensional hue histograms since the retrieval performance does not increase significantly when the number of bins gets over 20.

The statistical methods all recommend that the number of bins increases with the sample size. This is reasonable from a statistical estimation point of view but it is a drawback for CBIR applications since those applications require descriptions with as few parameters (bins) as possible for efficient search. The next experiment also shows that the empirical retrieval performance is almost independent of the image size suggesting a different strategy to select the bin number. In Fig. 4.4 we measure the retrieval performance for 50 standard queries on the MPEG-7 database using different image sizes: original size, 1/4, 1/16, and 1/64 image size. It shows that the performance is almost independent of the size of images. The results in (Brunelli and Mich, 2001) (based on Eq. 4.18) are valid only for small images (which is the case for their video and image databases).

The reason why all the statistical methods (Sturges, 1926; Akaike, 1974; Scott, 1979; Rudemo, 1982; Scott, 1985; Devroye and Gyorfi, 1985; Scott, 1992; Kanazawa, 1993; Wand, 1996; Birge and Rozenholc, 2002) fail when

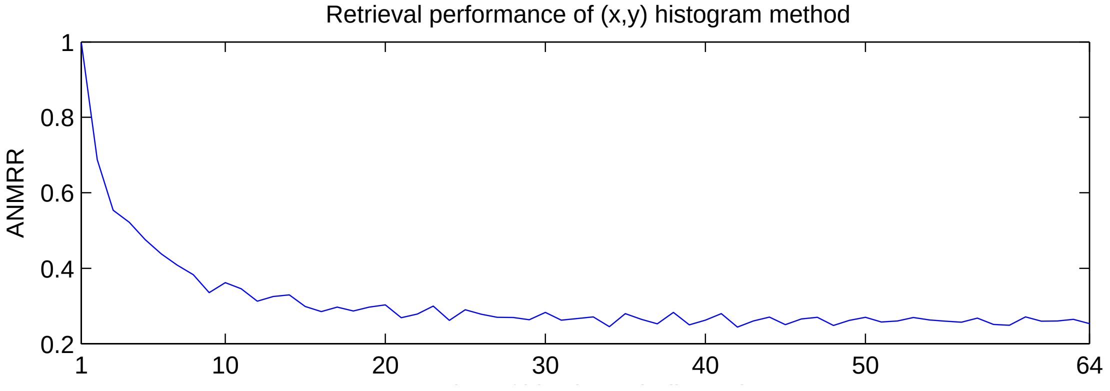


Figure 4.6: Average of ANMRR of 50 standard queries on the MPEG-7 database. Images are described by two dimensional (x,y) chromaticity histograms using different numbers of bins ranging from 1 to 64 in each dimension x and y making the number of bins in two-dimensional space range from 1 to $64^2 = 4096$. Using 8 to 10 intervals in each direction x and y seems to be the best value for the number of bins in each dimension in this case since the retrieval performance does not increase significantly when the number of bins exceeds 10.

applied to CBIR applications is that they all define their own cost functions which is integrated over the whole support (the mean integrated squared error, MISE, is very often used) in order to optimize the bin-width h. CBIR applications, however, use only a few estimated values from the data set as a compact description of the image, not all the data. Another important issue is that CBIR applications require fast response, a compact descriptor using only few parameters and giving a reasonable retrieval performance in many cases is more useful than a very complicated descriptor with just a slightly better retrieval performance. This is seen in Fig. 4.5, Fig.4.6, and Fig.4.7 which present results from our experiments using hue, (x,y), and RGB histograms. They all show that the improvement in retrieval performance is very small when the number of bins increase more than some threshold. Particularly for 3-D RGB histogram, the retrieval performance decreased when too many bins were used. So there is definitely a clear difference between the optimal number of bins given by the best value based on statistical criteria and the optimal bins for color-based image retrieval. Also we see that over-smoothed bin-width works better for image retrieval. This explains why a good estimator does not always give good descriptors for image retrieval as our experiments have confirmed in the previous sections.

A very simple way to take into account the influence of the deficiency of using too many bins in CBIR is to define a penalty as the number of bins increases. For example, a modified version of Akaike's method (Akaike, 1974)

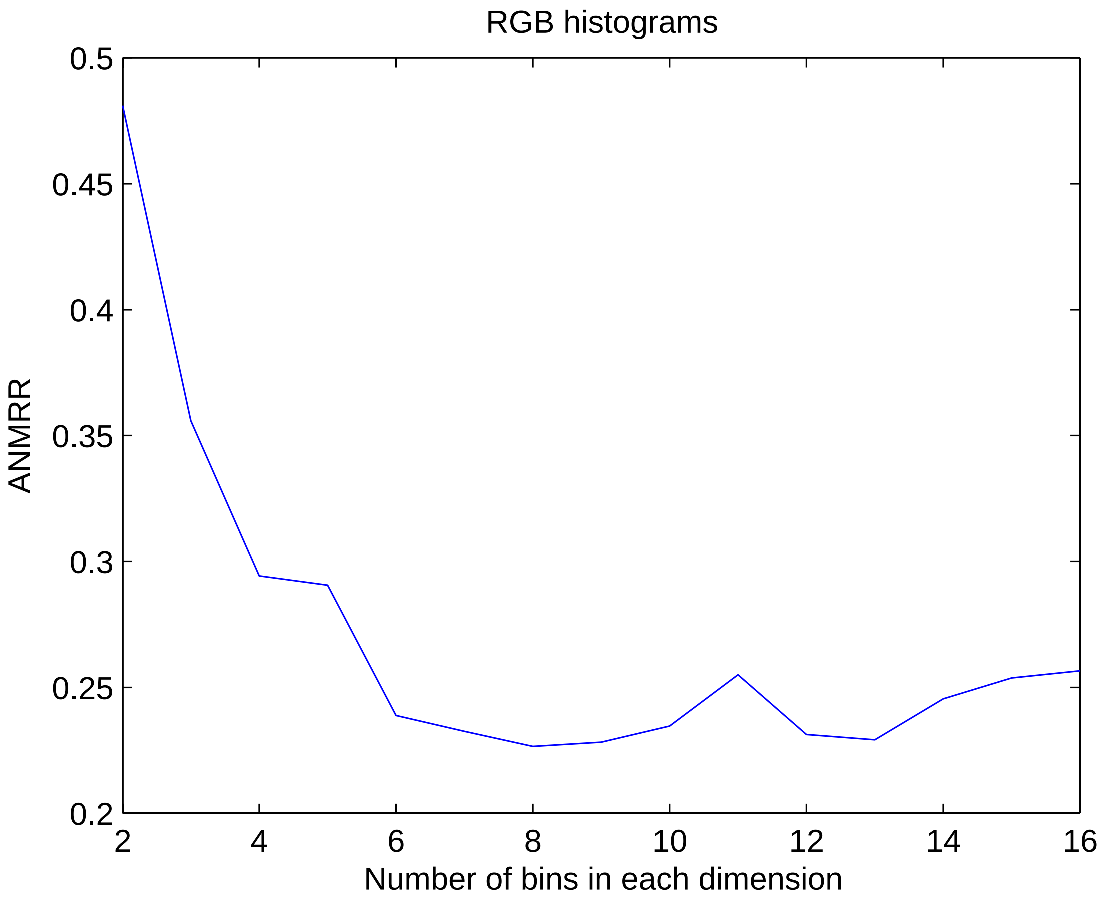


Figure 4.7: Average of ANMRR of 50 standard queries on the MPEG-7 database. Images are described by three-dimensional RGB histograms using different numbers of bins ranging from 2 to 16 in each dimension. 8 seems to be the best value for the number of bins in each dimension of the three-dimensional RGB histograms.

given below shows more reasonable results when applying statistical methods of finding the optimal number of bins of histograms in CBIR applications:

$$nb_{Akaike}^{CBIR} = \sup\left\{\sum_{k}^{nb} N_k \log(\frac{nb \cdot N_k}{n}) + 1 - nb - Penalty(nb)\right\} \tag{4.24}$$

where $Penalty(nb)$ is a penalty function of the number of bins nb. Different penalty functions give different results when optimizing the number of bins. Table 4.7 shows some of the results of our experiments for hue distributions (See the second column of Table 4.6 and Fig.4.5 for comparison). By introducing the penalty function which take into deficiency of using too many bins in CBIR, the number of bins we got from optimization process is closer to the empirical numbers in Fig. 4.5, Fig.4.6, and Fig.4.7.

Penalty function	Optimal bins
$Penalty(nb) = \frac{1}{2} \cdot (nb)^{1.5}$	38
$Penalty(nb) = (nb)^{1.5}$	24
$Penalty(nb) = 2 \cdot (nb)^{1.5}$	17

Table 4.7: Theoretically optimal number of bins using Akaike's method together with a penalty function on the number of bins as described in Eq. 4.24.

4.6 Summary

In color-based image retrieval, images are assumed to be similar in color if their color distributions are similar. However this assumption does not mean that the best estimator of the underlying color distributions always gives the best descriptors for color-based image retrieval. Our experiments show that the histogram method is simple, fast, and outperforms simple kernel-based methods in retrieving similar color images.

In order to improve the retrieval performance of kernel-based methods, two modifications are proposed. They are based on the use of non-orthogonal bases together with a Gram-Schmidt procedure and a method applying the Fourier transform. Experiments were done to confirm the improvements of our proposed methods both in retrieval performance and simplicities in choosing smoothing parameters.

In this chapter we also investigated the differences between parameters that give good density estimators and parameters that result in good retrieval performance. We found that over-smoothed bin-widths of density estimator, for both histogram and kernel-based methods, gives better retrieval performance.

Chapter 5

DIFFERENTIAL GEOMETRY-BASED COLOR DISTRIBUTION DISTANCES

In this chapter, a differential geometry framework is used to describe distance measures between distributions in a family of probability distributions. The way to incorporate important properties of the underlying distributions into the distance measures in the family is also discussed. Examples of simple distances between color distributions of two families of distributions are derived as illustrations of the framework and a theoretical background for the next chapter.

5.1 Measuring Distances Between Color Distributions

Almost all image database retrieval systems provide color as a search attribute. This can be used to search for images in the database which have a color distribution similar to the color distribution of a given query image. In most systems the color histogram is used to represent the color properties of an image.

Once a description of the color distribution has been chosen, the next problem in color-based image retrieval applications is the definition of a distance measure between two such distributions and its computation from their descriptions. Ideally the distance measure between color distributions should have all basic properties mentioned in section 3.4 such as perceptual similarity, efficiency, scalability, robustness, etc.

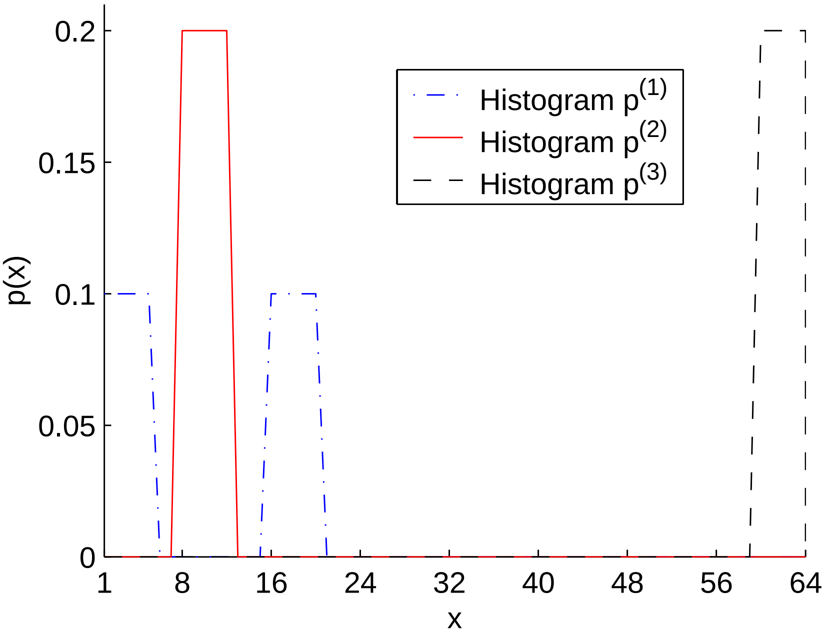


Figure 5.1: Shifted histograms.

Many histogram-based distance measures, however, are derived heuristically and may violate some of these properties. A very simple example, when correlation-based similarity measures give undesirable results, is illustrated in Fig. 5.1. Here many distance measures which do not take into account the color properties of the underlying distributions would assign the same distance to histograms $p^{(1)}$ and $p^{(2)}$ as to histograms $p^{(1)}$ and $p^{(3)}$. Although it seems to be reasonable to require $dist(p^{(1)}, p^{(2)}) < dist(p^{(1)}, p^{(3)})$.

In this chapter we propose a framework to compute the distance between color distributions based on differential geometry. In the framework of Riemannian geometry the distance between points on the manifold is defined as the length of the geodesic curve connecting these points. This is a generalization of the Euclidean distance and has the advantage that it only depends on the geometrical properties of the manifold. It is thus independent of the coordinate system used to describe this geometry. This approach gives a theoretically convincing definition of a distance and many existing distance measurements fall within this framework.

In the next section, the basic idea of a distance measure in a parametric family of distributions is presented briefly together with the connection to some existing distance measures. Some limitations when applying this method in measuring the distance between color distributions are also pointed out. A framework with an example of how to overcome the limitations is introduced in section 5.3. As illustrations for the new framework, distances between distributions are computed for two families of distributions: the family of normal distributions (as a simple example), and the family of linear representations of color distributions (as the theoretical background for the next chapter).

5.2 Differential Geometry-Based Approach

Comparing probability distributions is one of the most basic problems in probability theory and statistics. Many different solutions have been proposed in in the past. One of the, theoretically, most interesting approaches uses methods from differential geometry to define the distance between distributions of a parametric family, all of whose members satisfy certain regular conditions. This approach was introduced by Rao (Rao, 1949) and is described briefly in the following (for detailed descriptions see (Amari, 1985; Amari et al., 1987)).

5.2.1 Rao's Distance Measure

We denote by $\theta = (\theta_1, \theta_2, ..., \theta_r)$ a vector of r $(r \geqslant 1)$ parameters in a parameter space Θ and by $\{p(x \mid \theta), \theta \in \Theta\}$ a family of probability density functions of a random variable X. Each distribution in the family is described by a point in parameter space Θ. We want to measure the distance $d(\theta_1, \theta_2)$ between the distributions which are identified by the parameter values θ_1 and θ_2 in the parameter space Θ.

In order to compute the distance $d(\theta_1, \theta_2)$, the metric at each point in the space Θ should be defined. Considering the metric locally around point $\theta = (\theta_1, \theta_2, ..., \theta_r)$, let $\hat{\theta} = (\theta_1 + d\theta_1, \theta_2 + d\theta_2, ..., \theta_r + d\theta_r)$ be a neighboring point of θ in the parameter space Θ. To the first order, the difference between the density functions corresponding to these parameter points θ and $\hat{\theta}$ is given by

$$p(x \mid \hat{\theta}) - p(x \mid \theta) \approx \sum_{i=1}^{r} \frac{\partial p(x \mid \theta)}{\partial \theta_i} d\theta_i \tag{5.1}$$

and the relative difference by

$$\begin{aligned} dX &= \frac{p(x \mid \hat{\theta}) - p(x \mid \theta)}{p(x \mid \theta)} \\ &\approx [p(x \mid \theta)]^{-1} \sum_{i}^{r} \frac{\partial p(x \mid \theta)}{\partial \theta_i} d\theta_i \\ &\approx \sum_{i=1}^{r} \frac{\partial lnp(x \mid \theta)}{\partial \theta_i} d\theta_i \end{aligned} \tag{5.2}$$

These distributions summarize the effect of replacing the distribution $\theta = (\theta_1, \theta_2, ..., \theta_r)$ by $\hat{\theta} = (\theta_1 + d\theta_1, \theta_2 + d\theta_2, ..., \theta_r + d\theta_r)$. In particular, Rao considers the variance of the relative difference dX in Eq. 5.2 to construct the metric of the space Θ. The distance between the two neighboring distributions is then given by

$$\begin{aligned} ds^2 &= \sum_{i=1}^{r} \sum_{j=1}^{r} E \left\{ \frac{\partial \ln p(X \mid \theta)}{\partial \theta_i} \frac{\partial \ln p(X \mid \theta)}{\partial \theta_j} \right\} d\theta_i d\theta_j \\ &= \sum_{i=1}^{r} \sum_{j=1}^{r} g_{ij}(\theta) d\theta_i d\theta_j \end{aligned} \tag{5.3}$$

This is a positive definite quadratic differential form based on the elements of the information matrix $g_{ij}(\theta)$ for Θ which is defined as the variance-covariance matrix of

$$g_{ij}(\theta) = E \left\{ \frac{\partial \ln p(X \mid \theta)}{\partial \theta_i} \frac{\partial \ln p(X \mid \theta)}{\partial \theta_j} \right\} \quad \text{with } i, j = 1, 2, ...r \tag{5.4}$$

Let

$$\theta(t) : \theta_i = \theta_i(t), \ i = 1, 2, ..., r \tag{5.5}$$

denote an arbitrary parametric curve joining the two points θ_1 and θ_2 in space Θ. Suppose t_1 and t_2 are values of t such that

$$\theta_{1i} = \theta_i(t_1), \theta_{2i} = \theta_i(t_2), \ i = 1, 2, ..., r \tag{5.6}$$

In Riemannian geometry, the length of the curve in Eq. 5.5 between θ_1 and θ_2 is given by

$$s(\theta_1, \theta_2) = \left| \int_{t_1}^{t_2} \sqrt{\sum_{i,j=1}^{r} g_{i,j}(\theta) \frac{\partial \theta_i}{\partial t} \frac{\partial \theta_j}{\partial t}} \ dt \right| \tag{5.7}$$

The distance between the two distributions is then defined as the distance along the shortest curve between the two points θ_1 and θ_2.

$$\begin{aligned} dist(\theta_1, \theta_2) &= \underset{\text{all } \theta(t)}{\text{minimize}}(s(\theta_1, \theta_2)) \\ &= \underset{\text{all } \theta(t)}{\text{minimize}} \left| \int_{t_1}^{t_2} \sqrt{\sum_{i,j=1}^{r} g_{i,j}(\theta) \frac{\partial \theta_i}{\partial t} \frac{\partial \theta_j}{\partial t}} \; dt \right| \end{aligned} \tag{5.8}$$

Such a curve is called a geodesic and is given as the solution to the Euler-Lagrange differential equations (Courant and Hilbert, 1989):

$$\begin{aligned} &\sum_1^n g_{ij}\ddot{\theta} + \sum_1^n \sum_1^n \Gamma_{ijk}\dot{\theta}_i\dot{\theta}_j = 0, \quad \text{with } k = 1..n \\ &\text{where} \\ &\Gamma_{ijk} = \frac{1}{2}\left[\frac{\partial}{\partial \theta_i} g_{jk} + \frac{\partial}{\partial \theta_j} g_{ki} + \frac{\partial}{\partial \theta_k} g_{ij}\right] \\ &\text{and} \\ &\theta(t_1) = \theta_1, \theta(t_2) = \theta_2 \end{aligned} \tag{5.9}$$

5.2.2 Rao's Distance for Well-known Families of Distributions

Although Rao's approach provides a theoretically convincing definition of a distance, its application has been difficult since the differential equations in Eq. 5.9 are generally very difficult to solve analytically. In (Atkinson and Mitchell, 1981) two other methods are described that can be used to derive geodesic distances for a number of well-know distributions. The distances obtained are given below (many of them are used widely in computer vision and image processing)

A simplest example is the case of the family of normal distribution $N(\mu, \sigma)$. The metric of this family is given by

$$ds^2 = \frac{(\partial\mu)^2}{\sigma^2} + \frac{2(\partial\sigma)^2}{\sigma^2}$$

and the distance between two normal distributions is given by:

$$d_{N_1}(N(\mu_1, \sigma_1), N(\mu_2, \sigma_2)) = 2 \times tanh^{-1}\delta \tag{5.10}$$

where δ is the positive square root of

$$\frac{(\mu_1 - \mu_2)^2 + 2(\sigma_1 - \sigma_2)^2}{(\mu_1 - \mu_2)^2 + 2(\sigma_1 + \sigma_2)^2}$$

For n-dimensional independent normal distributions the distance has the same form as in Eq. 5.10

$$d_{N_n}(N(\mu_1,\sigma_1),N(\mu_2,\sigma_2)) = 2\sum_{k=1}^{n} tanh^{-1}\delta_i \tag{5.11}$$

where

$$\begin{aligned}\mu_1 &= (\mu_{11},\mu_{21},...,\mu_{n1})\\ \mu_2 &= (\mu_{12},\mu_{22},...,\mu_{n2})\\ \sigma_1 &= (\sigma_{11},\sigma_{21},...,\sigma_{n1})\\ \sigma_2 &= (\sigma_{12},\sigma_{22},...,\sigma_{n2})\end{aligned}$$

and

$$\delta_i = \sqrt{\frac{(\mu_{i1}-\mu_{i2})^2+2(\sigma_{i1}-\sigma_{i2})^2}{(\mu_{i1}-\mu_{i2})^2+2(\sigma_{i1}+\sigma_{i2})^2}}$$

For the case of two multivariate normal distributions with common covariance matrix, the distance is given by the Mahalanobis distance (for an application in image database search see (Carson et al., 1997))

$$d_M^2(N(\mu_1,\Sigma),N(\mu_2,\Sigma)) = (\mu_1-\mu_2)'\Sigma^{-1}(\mu_1-\mu_2) \tag{5.12}$$

and for multivariate normal distributions with common mean vector it is known as the Jensen distance and given by

$$d_J^2(N(\mu,\Sigma_1)N(\mu,\Sigma_2)) = \sum_i \log\lambda_i^2 \tag{5.13}$$

where λ_i are the roots of the equation $\det(\Sigma_1-\lambda\Sigma_2)=0$.

In the general case when the two normal distributions of the family differ in both mean vectors and correlation matrices, there is no analytical solution. Other measures have to be used in this case. Simple ways are to combine the Mahalanobis and Jensen distances or use the Bhattacharyya distance (Fukunaga, 1990, p.99)

$$\begin{aligned}&d_B^2(N(\mu_1,\Sigma_1),N(\mu_2,\Sigma_2)) =\\ &\frac{1}{8}(\mu_1-\mu_2)'\Sigma^{-1}(\mu_1-\mu_2)+\frac{1}{2}\ln\frac{\det\Sigma}{\sqrt{\det\Sigma_1\det\Sigma_2}}\end{aligned} \tag{5.14}$$

where $\Sigma = 0.5\times(\Sigma_1+\Sigma_2)$.

The intrinsic mathematical difficulties involved in applying the differential geometry framework to a particular family of distributions is not the only problem with this approach. A more fundamental problem is the negligence of the "meaning" of the underlying distributions. In the case of color distributions, for example, it does not consider the properties of the color space and the relation between different colors and their similarities.

As an example consider the application of this method to compute the distance of two color histograms in the space of all histograms of a certain size. Following the above framework, the geodesic distance in this parameter space can be computed analytically and is given by the *arccos* of the scalar product of the histogram entries:

$$d(p^{(1)}, p^{(2)}) = \arccos\left(\sum_i p_i^{(1)} p_i^{(2)}\right) \tag{5.15}$$

This distance is not a good measure between color distributions since it does not take into account the similarity of the colors represented by the bins (Sharing the problem mentioned previously in Fig. 5.1).

5.2.3 Color Distributions and Scale Spaces

One way to improve the distance measure in Eq. 5.15 is by using ideas from the theory of kernel density estimation (Fukunaga, 1990) and scale-space theory (Geusebroek et al., 2000) to define a range of similarity measures.

A kernel-based density estimation describes an unknown probability distribution as the convolution of the data with a suitably chosen kernel $K_s(x)$ of width $s : p_s(x) = p(x) \star K_s(x)$ where p is the histogram. We now define the **similarity** of two histograms $p^{(1)}, p^{(2)}$ at scale s as:

$$\mathcal{S}_s(p^{(1)}, p^{(2)}) = \left\langle p^{(1)}(x) \star K_s(x), p^{(2)}(x) \star K_s(x) \right\rangle \tag{5.16}$$

Using the Parseval identity (Wolf, 1979) we can compute the scalar product in the Fourier domain instead of in the time domain.

$$\begin{aligned} \mathcal{S}_s(p^{(1)}, p^{(2)}) &= \left\langle p^{(1)}(x) \star K_s(x), p^{(2)}(x) \star K_s(x) \right\rangle \\ &= \frac{1}{2\pi} \left\langle \hat{p}^{(1)}(y)\hat{K_s}(y), \hat{p}^{(2)}(y)\hat{K_s}(y) \right\rangle \\ &= \frac{1}{2\pi} \left\langle \hat{p}^{(1)}(y), \hat{p}^{(2)}(y)\hat{K_s}^2(y) \right\rangle \\ &= \left\langle p^{(1)}(x), p^{(2)}(x) \star \bar{K_s}(x) \right\rangle \end{aligned} \tag{5.17}$$

where $\hat{p}(y)$ and $\hat{K_s}(y)$ are the Fourier transforms of $p(x)$ and $K_s(x)$, and $\bar{K_s}(x)$ is the inverse Fourier transform of $\hat{K_s}^2(y)$.

When the kernel $K_s(x)$ is a Gaussian

$$K_s(x) = N(0,\Sigma) = \frac{1}{\sqrt{2\pi s}} e^{-\frac{x^2}{2s}} \tag{5.18}$$

$$\hat{K_s}(y) = e^{-y^2 s/2} \tag{5.19}$$

$$\bar{K_s}^2(x) = \sqrt{\frac{\pi}{s}}\ e^{-x^2/4s} \tag{5.20}$$

and the similarity is then given by

$$\mathcal{S}_s(p^{(1)}, p^{(2)}) = \sqrt{\frac{\pi}{s}} \sum_{l=0}^{N} \sum_{k=0}^{N} e^{\frac{-(k-l)^2}{4s}} p_k^{(1)} p_l^{(2)} \tag{5.21}$$

In an implementation it is important to note that the weight factor of the product $p_k^{(1)} p_l^{(2)}$ (given by $e^{\frac{-(k-l)^2}{4s}}$) depends only on the distance $(l-k)^2$ of the indices. For a fast computation of the distance at different scales it is thus not necessary to store all the products $p_k^{(1)} p_l^{(2)}$ but it is possible to pre-compute the partial sums

$$\pi_\Delta = \sum_k \left(p_k^{(1)} p_{k+\Delta}^{(2)} + p_k^{(1)} p_{k-\Delta}^{(2)} \right) \tag{5.22}$$

which are combined with the weights $e^{\frac{-\Delta^2}{4s}}$ to produce the distance value. This metric is a special case of the histogram techniques described in (Hafner et al., 1995) where fast implementations for image database searches are described.

We used the Vistex[1] database from MIT Lab to test the distance Eq. 5.21 at different scales. Fig. 5.3 shows the search results for the color patch in Fig. 5.2 at three different scale factors. In this extreme example the histogram of the query image consists of only one isolated peak. Smoothing this peak will result in increasing intersection with nearby color regions as shown in Fig. 5.3.

The above method is an improvement for this special case, when color distributions are described as color histograms. For the general case, when a set of r parameters is used to describe a color distribution in Rao's framework, we have to integrate the color information into the distance measures, particularly dealing with equations Eq. 5.1 and Eq. 5.2, where the metric is constructed from the difference between the probabilities of the two distributions.

[1] The VisTex database contains more than 400 images. Most of them contain homogenous textures and some of the images are represented in different resolutions in the database. Detailed information about the database is available at `http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html`.

If we use the absolute difference Eq. 5.1 instead of the relative difference Eq. 5.2, the metric of the new space is given by

$$g_{ij}(\theta) = \iint \frac{\partial p(X \mid \theta)}{\partial \theta_i} \frac{\partial p(Y \mid \theta)}{\partial \theta_j} K(x,y)dxdy \tag{5.26}$$

When the metric of the new space is defined, the geodesic distance between color distributions can be computed by solving the equation systems Eq. 5.9 with the new metric $\{g_{ij}\}$ given in Eq. 5.25 and Eq. 5.26.

In the following we will illustrate the whole framework by two examples: the family of normal distributions, and the family of linear representations of color distributions. The first example is an illustration of the framework, while the result of the second example will be used in the next chapter.

5.3.1 Space of Normal Distributions

An example is the space of normal distributions $N(\mu, \sigma, x)$. Each distribution in this family is described by two parameters μ and σ.

$$\begin{aligned} p(\mu, \sigma, x) &= N(\mu, \sigma, x) \\ &= \frac{\sqrt{2\pi}}{2\pi\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \end{aligned} \tag{5.27}$$

In order to characterize the weights between parameters x and y, we use the Gaussian type kernel $K(x,y)$, which has a form similar to Eq. 3.7

$$K(x,y) = e^{-(x-y)^2} \tag{5.28}$$

The framework in Eq. 5.25 gives us

$$\begin{aligned} \frac{\partial p(\mu, \sigma, x)}{\partial \mu} &= \frac{1}{\sqrt{2\pi}} \frac{x-\mu}{\sigma^3} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\ \frac{\partial p(\mu, \sigma, x)}{\partial \sigma} &= \frac{1}{\sqrt{2\pi}} \frac{(x-\mu)^2}{\sigma^4} e^{-\frac{(x-\mu)^2}{2\sigma^2}} - \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \end{aligned} \tag{5.29}$$

and the metric $\{g_{ij}\}$ in the new parameter space can be computed as follows:

$$\begin{aligned}
g_{11}(\mu,\sigma) &= \iint \frac{\partial p(\mu,\sigma,x)}{\partial \mu}\frac{\partial p(\mu,\sigma,y)}{\partial \mu} e^{-(x-y)^2} dxdy \\
&= \iint \frac{(x-\mu)(y-\mu)}{2\pi\sigma^6} e^{-\frac{(x-\mu)^2}{2\sigma^2}-\frac{(y-\mu)^2}{2\sigma^2}-(x-y)^2} dxdy \\
&= \frac{2}{(1+4\sigma^2)^{3/2}} \\
\\
g_{12}(\mu,\sigma) &= \iint \frac{\partial p(\mu,\sigma,x)}{\partial \mu}\frac{\partial p(\mu,\sigma,y)}{\partial \sigma} e^{-(x-y)^2} dxdy \\
&= 0 \qquad\qquad (5.30) \\
\\
g_{21}(\mu,\sigma) &= \iint \frac{\partial p(\mu,\sigma,x)}{\partial \sigma}\frac{\partial p(\mu,\sigma,y)}{\partial \mu} e^{-(x-y)^2} dxdy \\
&= 0 \\
\\
g_{22}(\mu,\sigma) &= \iint \frac{\partial p(\mu,\sigma,x)}{\partial \sigma}\frac{\partial p(\mu,\sigma,y)}{\partial \sigma} e^{-(x-y)^2} dxdy \\
&= \frac{12\sigma^2}{(1+4\sigma^2)^{5/2}}
\end{aligned}$$

Eq. 5.30 gives the metric in this space as

$$G_{Norm} = \left[g_{ij}\right] = \begin{bmatrix} \frac{2}{(1+4\sigma^2)^{3/2}} & 0 \\ 0 & \frac{12\sigma^2}{(1+4\sigma^2)^{5/2}} \end{bmatrix} \tag{5.31}$$

This leads to the distance $ds(\mu,\sigma)$ at point $\theta(\mu,\sigma)$ as

$$\begin{aligned}
ds^2(\mu,\sigma) &= \sum_{i,j=1}^{r} g_{ij}(\theta)\partial\theta_i\partial\theta_j \\
&= \frac{2(d\mu)^2}{(1+4\sigma^2)^{3/2}} + \frac{12\sigma^2(d\sigma)^2}{(1+4\sigma^2)^{5/2}} \qquad (5.32) \\
&= \left(\frac{2(\mu')^2}{(1+4\sigma^2)^{3/2}} + \frac{12\sigma^2(\sigma')^2}{(1+4\sigma^2)^{5/2}}\right) dtdt
\end{aligned}$$

Suppose now we have two color distributions which are represented by the two points $\theta_1(\mu_1,\sigma_1)$ and $\theta_2(\mu_2,\sigma_2)$. Let $\theta(t)$ be an arbitrary curve connecting $\theta(t_1) = \theta_1(\mu_1,\sigma_1)$ and $\theta(t_2) = \theta_2(\mu_2,\sigma_2)$. In Riemannian geometry, the geodesic distance $d(\theta_1,\theta_2)$ is given by

$$
\begin{aligned}
d(\theta_1,\theta_2) &= \underset{\text{all } \theta(t)}{\text{minimize}}\left(\int_{t_1}^{t_2} ds(\theta(t))\right) \\
&= \underset{\text{all } \theta(t)}{\text{minimize}}\left(\int_{t_1}^{t_2} \sqrt{\frac{2(\mu')^2}{(1+4\sigma^2)^{3/2}} + \frac{12\sigma^2(\sigma')^2}{(1+4\sigma^2)^{5/2}}}\ dt\right) \\
&= \underset{\text{all } \theta(t)}{\text{minimize}}\left(\int_{t_1}^{t_2} F(t)dt\right)
\end{aligned}
\tag{5.33}
$$

where

$$
F(t) = \sqrt{\frac{2(\mu')^2}{(1+4\sigma^2)^{3/2}} + \frac{12\sigma^2(\sigma')^2}{(1+4\sigma^2)^{5/2}}}
$$

From the Calculus of Variation (see in (Courant and Hilbert, 1989, p.202)) the minimization problem in Eq. 5.33 is equivalent to the systems

$$
\begin{aligned}
F - \mu'\frac{\partial F}{\partial \mu'} &= \text{const, say } C_a \\
F - \sigma'\frac{\partial F}{\partial \sigma'} &= \text{const, say } C_b
\end{aligned}
\tag{5.34}
$$

or

$$
\begin{aligned}
F - \frac{2\mu'^2}{F(1+4\sigma^2)^{3/2}} &= C_a \\
F - \frac{12\sigma^2\sigma'^2}{F(1+4\sigma^2)^{5/2}} &= C_b
\end{aligned}
\tag{5.35}
$$

where

$$
\begin{aligned}
F &= \sqrt{\frac{2(\mu')^2}{(1+4\sigma^2)^{3/2}} + \frac{12\sigma^2(\sigma')^2}{(1+4\sigma^2)^{5/2}}} \\
&= C_a + C_b
\end{aligned}
$$

First we reduce Eq. 5.35 to

$$
\begin{aligned}
\frac{2(\mu')^2}{(1+4(\sigma)^2)^(3/2} &= (C_a + C_b)C_b = C_1 \\
\frac{12(\sigma)^2(\sigma')^2}{(1+4(\sigma)^2)^{5/2}} &= (C_a + C_b)C_a = C_2
\end{aligned}
\tag{5.36}
$$

Solving Eq. 5.36 gives us the solution of the geodesic curve

$$\begin{aligned}\sigma &= \frac{3}{2}\sqrt{\frac{1}{C_1^2(t-C_3)^4}-1}\\ \mu &= \frac{3C_1\sqrt{3C_1C_2}}{2\sqrt{2}}(t-C_3)^2+C_4\end{aligned} \tag{5.37}$$

and

$$\begin{aligned}&d(\theta_1(\mu_1,\sigma_1),\theta_2(\mu_2,\sigma_2))\\ &= \sqrt{3[(1+4\sigma_1^2)^{-1/4}-(1+4\sigma_2^2)^{-1/4}]^2+2\sqrt[4]{2}[\sqrt[4]{\mu_1}-\sqrt[4]{\mu_2}]^2}\end{aligned} \tag{5.38}$$

which is the distance between the two normal distributions $\theta_1(\mu_1,\sigma_1)$ and $\theta_2(\mu_2,\sigma_2)$).

5.3.2 Linear Representations of Color Distributions

The second example investigates linear representations of color distributions. For a given set of N basis vectors b_i, a histogram p can be parameterized by the N parameters $\{\theta_i\}_N$ defined by the description

$$\begin{aligned}p(k,\theta) &= p(k,\theta_1..\theta_N)\\ &= \sum_{i=1}^{N}\theta_i b_i(k)\end{aligned} \tag{5.39}$$

Different ways to compute the basics $b_i(k)$ define different linear representation methods of color distributions.

Applying the framework in the previous section for the new representation in Eq. 5.39 we have

$$\frac{\partial p(k)}{\partial\theta_i} = b_i(k) \tag{5.40}$$

and the metric $\{g_{ij}\}$ of the histogram space in Eq. 5.25 can be computed as follows

$$\begin{aligned}g_{ij} &= \sum_l\sum_m b_i(l)b_j(m)a_{lm}\\ &= b_i'Ab_j\end{aligned} \tag{5.41}$$

where $A=[a_{lm}]$ is a symmetric, positive definite matrix defining the properties of the color space. Each entry a_{lm} captures the perceptual similarity between colors represented by bins l and m as described in section 3.4.

Suppose two color distributions $p^{(1)}$ and $p^{(2)}$ in this general space are represented by two set of N parameters $\{\theta_i^{(1)}\}_N$ and $\{\theta_i^{(2)}\}_N$

$$p^{(1)}(k) = \sum_{i=1}^{N} \theta_i^{(1)} b_i(k)$$

$$p^{(2)}(k) = \sum_{i=1}^{N} \theta_i^{(2)} b_i(k)$$

Then the distance between the two distributions $p^{(1)}$ and $p^{(2)}$ in this space is given by

$$d(p^{(1)}, p^{(2)}) = \sum_{i}^{N} \sum_{j}^{N} g_{ij} \Delta\theta_i \Delta\theta_j \tag{5.42}$$

where

$$\Delta\theta_i = \theta_i^{(1)} - \theta_i^{(2)} \tag{5.43}$$

The distance can be written in matrix form as

$$\begin{aligned} d(p^{(1)}, p^{(2)}) &= (p^{(1)} - p^{(2)})^T G (p^{(1)} - p^{(2)}) \\ &= (\Delta p)^T G (\Delta p) \end{aligned} \tag{5.44}$$

where G

$$G = [g_{ij}] = [b_i A b_j] \tag{5.45}$$

is the metric in the new N dimensional parameter space which is expanded by the basics $\{b_i\}_N$. G can be pre-computed in advance since all the basics $\{b_i\}_N$ and the weight matrix A are pre-defined.

The new distance Eq. 5.44 will be used in the next chapter for a new compact representation of color features.

Chapter 6

KLT-BASED REPRESENTATION OF COLOR DISTRIBUTIONS

In many color-based image retrieval systems the color properties of an image are described by its color histogram. Histogram-based search is, however, often inefficient for large histogram sizes. Therefore we introduce several new, Karhunen-Loève Transform (KLT) based, methods that provide efficient representations of color histograms and differences between two color histograms. The methods are based on the following two observations:

- Ordinary KLT considers color histograms as signals and uses the Euclidian distance for optimization. KLT with generalized color distance measures that take into account both the statistical properties of the image database and the properties of the underlying color space should improve the retrieval performance.
- The goal of compressing features for image retrieval applications is to preserve the topology of feature space as much as possible. It is therefore more important to represent the differences between features than the features of the images themselves. The optimization should be based on minimizing the approximation error in the space of local histogram differences instead of the space of color histograms.

Experiments were performed on three image databases containing more than 130,000 images. Both objective and subjective ground truth queries were used in order to evaluate the proposed methods and to compare them with other existing methods. The results from our experiments show that compression methods based on a combination of the two observations described above provide new powerful and efficient retrieval algorithms for color-based image retrieval.

6.1 Introduction

Color has been widely used for content-based image retrieval, multimedia information systems and digital libraries. In many color-based image retrieval (CBIR) applications, the color properties of an image are characterized by the probability distribution of the colors in the image. The color histogram remains the most popular representation of color distributions since it is insensitive to small object distortions and since it is easy to compute. However, it is not very efficient due to its large memory requirement. For typical applications a color histogram might consist of $N = 512$ bins. With such a large number of bins N (ie. $N \geq 20$), the performance of current indexing techniques is reduced to a sequential scanning (Weber et al., 1998; Rui et al., 1999). To make color histogram-based image retrieval truly scalable to large image databases it is desirable to reduce the number of parameters needed to describe the histogram while still preserving the retrieval performance. Approaches to deal with these problems include the usage of coarser histograms (Pass and Zabih, 1999; Mitra et al., 1997), dominant colors or signature colors (Deng et al., 2001; Androutsos et al., 1999; Rubner et al., 1998; Ma, 1997) and application of signal processing compression techniques such as the Karhunen-Loève Transform, Discrete Cosine Transform, Handamard Transform, Haar Transform, and Wavelets etc. (Hafner et al., 1995; Ng and Tam, 1999; Berens et al., 2000; Manjunath et al., 2001; Albuz et al., 2001). Some of them are also suggested in the context of MPEG-7 standard (Manjunath et al., 2001).

It is well known that the optimal way to map N-dimensional vectors to lower K-dimensional vectors ($K \ll N$) is the Karhunen-Loève Transform (KLT) (Fukunaga, 1990). KLT is optimal in the sense that it minimizes the mean squared error of the Euclidian distance between the original and the approximated vectors. However, a straightforward application of the KLT (as well as other transform-based signal processing compression techniques) to the space of color histograms gives poor retrieval performance since:

- The technique treats the color histogram as an ordinary vector and ignores the properties of the underlying color distribution. The usage of the structure of the color space and the color distributions should improve the retrieval performance.

- The goal of ordinary compression is to describe the original signal by a given number of bits such that the reconstruction error is minimized. The ultimate goal of color-based image retrieval is, however, not to recover the original histogram but to find similar images. Therefore it seems reasonable that the topology of the color histogram space locally around the query image should be preserved as much as possible while reducing the number of bits used to describe the histograms. The optimal representation of the differences between color histograms is therefore much closer to the final aim of image retrieval than the optimal representation of the color histograms themselves.

In this chapter we use KLT together with a generalized color-based distance in two different spaces: the space of color histograms $\mathbb{P}$ and the space of local histogram differences $\mathbb{D}$, in which only pairs of histograms with small differences are considered. Using the KLT-basis computed from the space of local histogram differences $\mathbb{D}$ gives an optimum solution in the sense of minimizing the approximation error of the differences between similar histograms. This solution results in a better estimation of the distances between color histograms, and consequently better retrieval performance in CBIR applications.

The chapter is organized as follow: basic facts from color-based image retrieval, particularly the problem of measuring the distances between color distributions, are reviewed in the next section. Our proposed methods are presented in section 6.3. Section 6.4 describes our experiments in which both objective and subjective ground truth queries are used to evaluate our methods and to compare them with other existing methods.

6.2 Distances between Color Histograms

In color-based image retrieval we want to find all images I which have similar color properties as a given query image Q. In this chapter we describe the color properties of images by their color histograms and we define the similarity between images as the similarity between their color histograms. If the color histograms of the images I and Q are given by h_I and h_Q we represent the two images I and Q by two points h_I and h_Q in the color histogram space $\mathbb{P}$ and define the distance between the images as the distance between the two points h_I and h_Q in this space:

$$d(I, Q) = d(h_I, h_Q) \tag{6.1}$$

Popular choices for computing the distances in the color histogram space are histogram intersection (Swain and Ballard, 1991), L_p norm, Minkowski-form, quadratic forms (Hafner et al., 1995; Ng and Tam, 1999), the Earth Mover Distance (EMD) (Rubner et al., 1998) and other statistical distance measures (Puzixha et al., 1999; Rui et al., 1999) as mentioned in section 3.4. The EMD and the quadratic form methods are of special interest since they take into account the properties of the color space and the underlying color distributions.

The EMD is computationally demanding. Basically it computes the minimal cost to transform one histogram into the other. An optimization problem has to be solved for each distance calculation which makes the EMD less attractive in terms of computational speed.

The quadratic form distance between color histograms is defined as:

$$d^2_M(h_1, h_2) = (h_1 - h_2)^T M(h_1 - h_2) \tag{6.2}$$

where $M = [m_{ij}]$ is a positive semi-definite matrix defining the properties of the color space. Each entry m_{ij} captures the perceptual similarity between colors represented by bins i and j. A reasonable choice of m_{ij} is (Hafner et al., 1995):

$$m_{ij} = 1 - d_{ij}/d_{max} \tag{6.3}$$

Here d_{ij} is the Euclidean distance between color i and j in the CIELAB color space and $d_{max} = \underset{ij}{max}\{d_{ij}\}$. (The CIELAB color space is used since its metrical properties are well adapted to human color difference judgments.)

The quadratic form distance tends to overestimate the mutual similarity of color distributions (Stricker and Orengo, 1996; Rubner, 1999). Several suggestions have been made to reduce the mutual similarity of dissimilar colors. One example is

$$m_{ij} = exp(-\sigma(d_{ij}/d_{max})^k) \tag{6.4}$$

described in (Hafner et al., 1995). It enforces a faster roll-off as a function of d_{ij}, the distance between color bins. Another method uses a threshold for similar colors so that only colors which are similar will be considered in contributing to the distance. For example, m_{ij} in Eq. 6.3 can be redefined as (Manjunath et al., 2001):

$$m_{ij} = \begin{cases} 1 - d_{ij}/d_{max} & \text{if } d_{ij} \leq T_d \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

where T_d is the maximum distance for two colors to be considered similar. The value of d_{max} is redefined as αT_d where α is a constant between 1.0 and 1.5.

The quadratic form-based metric is computationally demanding. In a naive implementation, the complexity of computing one distance is $O(N^2)$ where N is the number of bins. Efficient implementations are, however, as fast as simple bin-by-bin distance methods such as histogram intersection or the L_p norm. It has also been reported that these metrics provide more desirable results than bin-by-bin distance methods (Hafner et al., 1995). The quadratic form-based distances are thus commonly used as distance in content-based image retrieval.

6.3 Optimal Representations of Color Distributions

Using the full histogram to compute the distances in Eq. 6.2 is unrealistic for large image databases because of computational and storage demands. Methods for estimating the distances using fewer parameters are needed in order to speed up the search engine and to minimize storage requirements. Thus compression techniques could be used to compress the description of color histograms. The Karhunen-Loève transform (KLT) provides the optimal way to project signals from high-dimensional space to lower dimensional space.

6.3.1 The Discrete Karhunen-Loève Expansion

Let X be an N-dimensional random vector. Then X can be represented without error by the summation of N linear independent vectors Φ_i as

$$X = \sum_{i=1}^{N} y_i \Phi_i = \Phi Y \tag{6.6}$$

where

$$\Phi = [\Phi_1, ..., \Phi_N] \tag{6.7}$$

and

$$Y = [y_1, ..., y_N]^T \tag{6.8}$$

The matrix Φ is deterministic and consists of N linearly independent column vectors. Thus

$$det(\Phi) \neq 0 \tag{6.9}$$

The columns of Φ span the N-dimensional space containing X and are called *basic vectors*. Furthermore, we may assume that the columns of Φ form an orthonormal set, that is,

$$\Phi_i^T \Phi_j = \begin{cases} 1 & \text{for i=j,} \\ 0 & \text{for i}\neq\text{j} \end{cases} \tag{6.10}$$

If the orthonormality condition is satisfied, the components of Y can be calculated by

$$y_i = \Phi_i^T X \tag{6.11}$$

Therefore Y is simply an orthonormal transformation of the random vector X and is itself a random vector.

Suppose that we choose only K (with $(K < N)$) Φ_i's and that we still want to approximate X well. We can do this by replacing those components of Y, which we do not calculate, with pre-selected constants c_i and form the following approximation:

$$\hat{X}(K) = \sum_{i=1}^{K} y_i \Phi_i + \sum_{i=K+1}^{N} y_i \Phi_i \tag{6.12}$$

We lose no generality in assuming that only the first K y_i's are calculated. The resulting representation error is

$$\begin{aligned}\Delta X(K) &= X - \hat{X}(K)\\ &= X - \left(\sum_{i=1}^{K} y_i\Phi_i - \sum_{i=K+1}^{N} c_i\Phi_i\right)\\ &= \sum_{i=K+1}^{N} (y_i - c_i)\Phi_i\end{aligned} \tag{6.13}$$

Note that both $\hat{X}$ and ΔX are random vectors. We will use the mean squared magnitude of ΔX as a criterion to measure the effectiveness of the subset of K features. We have

$$\begin{aligned}\bar{\epsilon}^2(K) &= E\left\{\| \Delta X(K) \|^2\right\}\\ &= E\left\{\sum_{i=K+1}^{N}\sum_{j=K+1}^{N}(y_i - c_i)(y_j - c_j)\Phi_i^T\Phi_j\right\}\\ &= \sum_{i=K+1}^{N} E\left\{(y_i - c_i)^2\right\}\end{aligned} \tag{6.14}$$

For every choice of basis vectors Φ_i and constant terms c_i, we obtain a value for $\bar{\epsilon}^2(K)$. We want to make the choice which minimizes $\bar{\epsilon}^2(K)$

The optimum choice, in the sense of minimizing the mean squared magnitude of ΔX, is obtained when

$$c_i = E\{y_i\} = \Phi_i^T E\{X\} \tag{6.15}$$

and Φ_i are the first K's eigenvectors Φ_k of

$$\Sigma_X = E\left\{(X - E\{X\})(X - E\{X\})^T\right\} \tag{6.16}$$

corresponding to the first K largest eigenvalues of Σ_X. The minimum $\bar{\epsilon}^2(K)$ is thus equal to the sum of the $N - K$ smallest eigenvalues of Σ_X.

6.3.2 Compact Descriptors for Color-based Image Retrieval

In the following we consider a histogram h as a vector in $N-$dimensional space. Selecting $K-$basic functions $\varphi_k, (k = 1, \ldots, N)$ we describe h by K numbers x_k as follow:

$$\widetilde{h}_K = \sum_{k=1}^{K} x_k\varphi_k \tag{6.17}$$

The approximation error is given by:

$$\varepsilon_K(h) \;=\; h - \widetilde{h}_K \;=\; h - \sum_{k=1}^{K} x_k \varphi_k \;=\; \sum_{k=K+1}^{N} x_k \varphi_k \tag{6.18}$$

Ordinary KLT in the histogram space $\mathbb{P}$ selects the basis functions φ_k such that the mean squared error in the Euclidian norm, $\overline{\varepsilon}_E$, is minimized:

$$\overline{\varepsilon}_E^2 = E\left\{\| \varepsilon_K(h)^2 \|\right\} = E\left\{\varepsilon_K(h)^T \varepsilon_K(h)\right\} \tag{6.19}$$

Instead of using the Euclidian distance, a color-based distance can be used where relations between different regions in color space are taken into account. This results in a better correspondence to human perception.

The basis functions φ_k are then selected such that the mean squared error using the color-based distances, $\overline{\varepsilon}_M$, is minimized:

$$\overline{\varepsilon}_M^2 = E\left\{\| \varepsilon_K(h)^2 \|_M\right\} = E\left\{\varepsilon_K(h)^T M \varepsilon_K(h)\right\} \tag{6.20}$$

The computation of the coefficients and the basis functions in this new metric is done as follows:

The matrix M given above is positive semi-definite and can therefore be factored into

$$M = U^T U \tag{6.21}$$

with an invertible matrix U. Next we introduce the modified scalar product between two vectors as:

$$\langle h_1, h_2 \rangle_M = h_1^T M h_2 = h_1^T U^T U h_2 = (U h_1)^T (U h_2) \tag{6.22}$$

Then we introduce an orthonormal basis φ_k with respect to this new scalar product: $\langle \varphi_i, \varphi_j \rangle_M = \delta_{ij}$. A given histogram can now be approximated using only K numbers:

$$h \approx \tilde{h} = \sum_{k=1}^{K} \langle h, \varphi_k \rangle_M \varphi_k = \sum_{k=1}^{K} f_k \varphi_k \tag{6.23}$$

Once the basis vectors φ_k are given, the coefficients f_k in the expansion of Eq. 6.23 are computed by.

$$f_k = \langle h, \varphi_k \rangle_M = h^T M \varphi_k \tag{6.24}$$

The new basis functions φ_k can be found by imitating the construction for the Euclidean case. The squared norm of the approximation of a histogram h

is given by

$$\begin{aligned}\left\|\tilde{h}\right\|_M^2 &= \left\langle\tilde{h},\tilde{h}\right\rangle_M = \left\langle\left(\sum_{l=1}^{K}\langle h,\varphi_l\rangle_M\varphi_l\right),\left(\sum_{k=1}^{K}\langle h,\varphi_k\rangle_M\varphi_k\right)\right\rangle_M \\ &= \sum_{k=1}^{K}\langle\varphi_k,h\rangle_M\langle h,\varphi_k\rangle_M = (U\varphi_k)^TUhh^TU^T(U\varphi_k) \qquad (6.25)\end{aligned}$$

Computing the mean length and using the notation $\Sigma_M = E(Uhh^TU^T)$ we see that the basis vectors with the smallest approximation error can be found by solving the Euclidean eigenvector problem $\Sigma_M\psi_k = c_k\psi_k$. From them the basis vectors are computed as $\varphi_k = U\psi_k$.

Ordinary KLT technique is a special case where the relations between color bins are ignored (M = identity). When the correlations between the input images in the database are ignored ($E\{hh^T\}$ = identity) the solution is identical to the QBIC approach in (Hafner et al., 1995).

Given two color images I, and Q their histograms can be approximated by using only K coefficients as follows:

$$\begin{aligned}\tilde{h}_I &= \sum_{k=1}^{K}\langle h_I,\varphi_k\rangle_M\varphi_k = \sum_{k=1}^{K} f_k^I\varphi_k \\ \tilde{h}_Q &= \sum_{k=1}^{K}\langle h_Q,\varphi_k\rangle_M\varphi_k = \sum_{k=1}^{K} f_k^Q\varphi_k\end{aligned} \qquad (6.26)$$

The distance between the two histograms is:

$$\begin{aligned}d_M^2(I,Q) &= (h_I-h_Q)^TM(h_I-h_Q) \\ &= \|h_I-h_Q\|_M^2 \approx \left\|\tilde{h}_I-\tilde{h}_Q\right\|_M^2 \\ &= \left\langle\tilde{h}_I-\tilde{h}_Q,\tilde{h}_I-\tilde{h}_Q\right\rangle_M \\ &= \left\|\tilde{h}_I\right\|_M^2 + \left\|\tilde{h}_Q\right\|_M^2 - 2\sum_{k}^{K}\left\langle\tilde{h}_I,\varphi_k\right\rangle_M\left\langle\tilde{h}_Q,\varphi_k\right\rangle_M \\ &= \sum_{k}^{K}(f_k^I)^2 + \sum_{k}^{K}(f_k^Q)^2 - 2\sum_{k}^{K} f_k^I\cdot f_k^Q\end{aligned} \qquad (6.27)$$

The first two terms are computed only once and the distance computation in the retrieval phase involves therefore only K multiplications.

We now have an optimal color histogram compression method in the sense that it minimizes the mean squared error of the color-based distances between the original color histograms and the approximated histograms. Application of the method to color-based image retrieval relies on the assumption that a better reconstruction of color histograms from the compression step implies a better retrieval performance.

The ultimate aim of compressing features in an image retrieval applications is, however, not to reconstruct the feature space but, as mentioned before, to preserve the topology of the feature space locally around the query image feature points. In this sense, image retrieval is not primarily concerned about the features of the images, but more about the (dis-)similarity or the differences between features. In Eq. 6.2 the distance was defined as

$$d_M^2(h_1, h_2) = (h_1 - h_2)^T M (h_1 - h_2)$$

It seems reasonable to expect that a KLT designed to provide the best reconstruction of the differences between color histograms may lead to a better retrieval performance. Since we care only about similar images, only pairs of similar color histograms are taken into account in the compression.

We therefore define for a (small) constant δ the space $\mathbb{D}_\delta$ of local histogram differences as:

$$\mathbb{D}_\delta = \{\Delta h = h_1 - h_2 : h_1, h_2 \in \mathbb{P}, d_M(h_1, h_2) \leq \delta\} \tag{6.28}$$

Another way to define the space of local histogram differences is based on the set of nearest neighbors. For each color histogram h_1, we define the local differences space at every $h_1 \in \mathbb{P}$ as

$$\mathbb{D}_n^{h_1} = \{\Delta h = h_1 - h_2 : h_2 \in \mathbb{P}, d(h_1, h_2) \text{ are the } n \text{ smallest distances}\} \tag{6.29}$$

The space of local histogram differences is then defined as the union of all such $\mathbb{D}_n^{h_1}$ at every $h_1 \in \mathbb{P}$

$$\mathbb{D}_n = \bigcup_{h_1 \in \mathbb{P}} \mathbb{D}_n^{h_1} \tag{6.30}$$

After the construction of the spaces of local histogram differences, KLT-techniques are used as before with the only difference that now they operate on the space $\mathbb{D}_\delta$ given in Eq. 6.28 or the space $\mathbb{D}_n$ given in Eq. 6.30 instead of the histogram space $\mathbb{P}$. The basis obtained from applying KLT on $\mathbb{D}_\delta$ and $\mathbb{D}_n$ are then used for compressing the features in the space of color histograms $\mathbb{P}$.

Summarizing we can say that the KLT-based methods described here are based on the following two observations:

- Statistical properties of the image database and the properties of the underlying color space should be incorporated into the distance measure as well as the optimization process.

- The optimization should be based on minimizing the approximation error in the space of local histogram differences instead of the space of color histograms.

6.4 Experiments

The following methods have been implemented and tested in our experiments:

H_K : Full color histogram with K bins.

D_K : Dominant color-based method (Deng et al., 2001; Ma, 1997).

K_K^{QB} : KLT-based method described in QBIC$^{\mathrm{TM}}$ (Hafner et al., 1995).

K_K : Ordinary KLT in the space of histograms $\mathbb{P}$.

$K_K^{\mathbb{D}}$: KLT in the space of differences of neighboring histograms $\mathbb{D}_n$.

$K_K^{\mathbb{M}}$: KLT in $\mathbb{P}$ with color metric M.

$K_K^{\mathbb{D}M}$: KLT in $\mathbb{D}_n$ with color metric M.

The approximation order (or the dimension of the compressed feature space) used in the experiments is given by the subscript K and this notation will be used in the rest of this section.

The following image databases of totally more than 130,000 images are used in our experiments:

Corel database: 1,000 color images (randomly chosen) from the Corel Gallery

MPEG-7 database: 5,466 color images and 50 standard queries (Zier and Ohm, 1999) designed to be used in the MPEG-7 color core experiments

Matton database: 126,604 color images. These images are low-resolution images of the commercial image database maintained by Matton AB in Stockholm (the average size is 108x120 pixels)

In all our experiments, the retrieval performance is measured based on the Average Normalized Modified Retrieval Rank (ANMRR). The detailed description of ANMRR is described in chapter 3. Here we recall briefly that the lower values of ANMRR indicate better retrieval performance, 0 means that all the ground truth images have been retrieved and 1 that none of the ground truth images has been retrieved.

6.4.1 Properties of color histogram space vs. retrieval performance

The retrieval performance of histogram-based methods using quadratic form distances depends on the construction of the color histogram and the metric M defining the properties of the histogram space. In the first set of experiments, the following four different methods of defining the metric M are evaluated in order to find a good matrix M for the next sets of experiments:

M_1 : The method as described in Eq. 6.3

M_2 : The exponential function as in Eq. 6.4

M_3 : Color threshold T_d as in Eq. 6.5

M_4 : Combination of color threshold and exponential roll-off

There are several parameters in the construction of each method used to define M. Changing these parameters affects the distance measure between color histograms and consequently the retrieval performance of the color-based image retrieval.

For example in Eq. 6.4, increasing σ will reduce the influence of neighboring color bins and vice versa. Fig. 6.1 shows the ANMRR of the 50 standard queries for the MPEG-7 database when the metric is defined as M_4 and σ is varying. For the sake of simplicity in parameterizing M, parameter ρ was introduced as a simple normalized version of σ for the case $k = 2$ as:

$$\rho = \frac{\sigma}{d^2_{max} \times \text{ standard deviation of all histograms}} \tag{6.31}$$

M	HSV 256 bins	RGB 512 bins	Lab 512 bins
M_1	0.237	0.229	0.226
M_2, $k = 2$	0.214	0.174	0.188
M_3	0.215	0.174	0.198
M_4	0.216	0.176	0.183

Table 6.1: Best retrieval performance (measured by ANMRR of 50 standard queries in the MPEG-7 database) of different methods of defining the metric M for the color histogram space in HSV $16x4x4$ bins, RGB $8x8x8$ bins, and CIELAB $8x8x8$ bins.

The experiment is repeated for other methods of defining metric M. Table 6.1 summaries the best retrieval performance of each method for different color spaces.

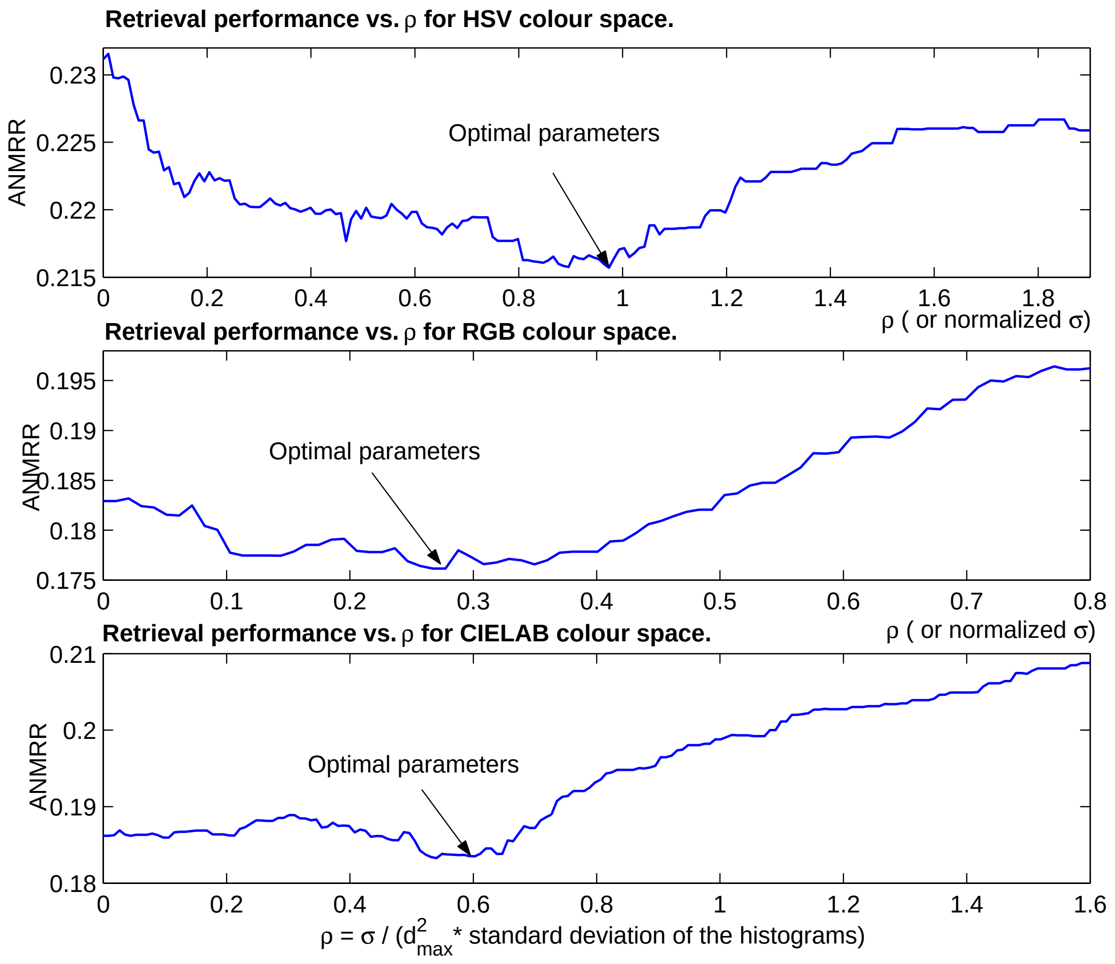


Figure 6.1: Properties of metric M_4 in Eq. 6.4: ANMRR of 50 standard queries from the MPEG-7 database for different color spaces when constants σ and ρ are varying. $T_d = 30$, $\alpha = 1.2$, $d_{max} = 36$.

The results show that the distance measure in Eq. 6.3 overestimates the mutual similarity of dissimilar colors. The retrieval performance is improved using the distance measures in Eq. 6.4 and Eq. 6.5. However when ρ in Eq. 6.4 increases too much and/or the value T_d in Eq. 6.5 decreases too much, the retrieval performance is getting worse. The experimental results show also that the optimum retrieval performance of methods M_2, M_3, and M_4 (which is a combination of both) are comparable.

The optimal parameters depend on both the color perception of the and the application at hand. Finding such an optimal metric M can be done experimentally and its estimation in not discussed here. Instead we ran our experiments (See Fig. 6.1 and Table 6.1) to determine a set of reasonable parameters for the remaining experiments.

6.4.2 Experiments with the Corel database

In the second set of experiments, we estimate the influence of the different approximation methods including the usage of coarser histograms (Pass and Zabih, 1999; Mitra et al., 1997), dominant colors or signature colors (Deng et al., 2001; Androutsos et al., 1999; Rubner et al., 1998; Ma, 1997), the standard KLT, the method used in (Hafner et al., 1995; Ng and Tam, 1999) and the proposed KLT-based methods as presented in the previous section. We compare the retrieval results of the approximation-based methods to the retrieval result achieved when the full histogram is used.

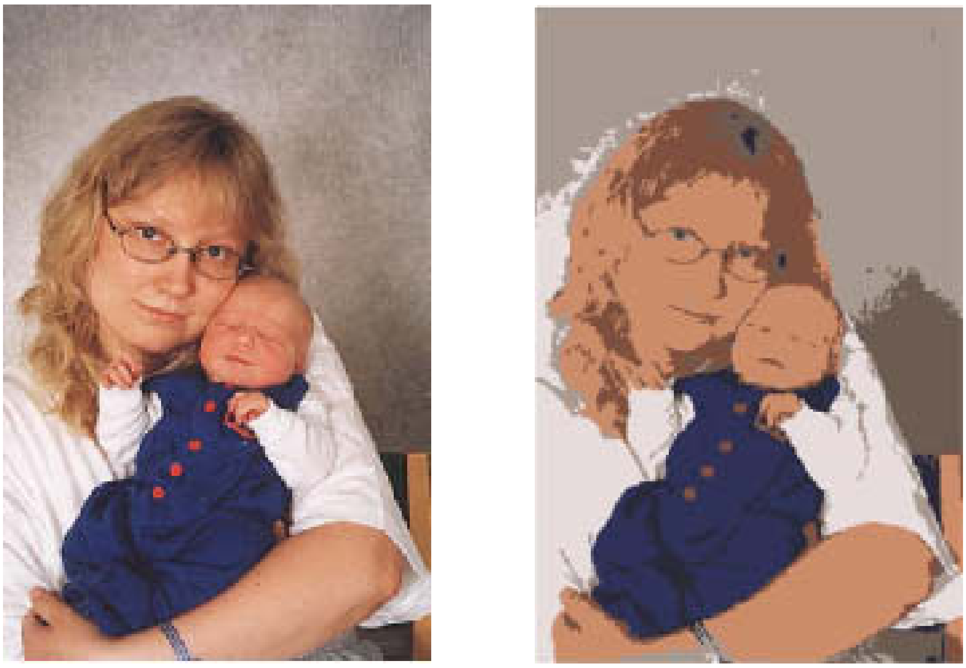


Figure 6.2: A color image and its segmented regions computed by the Mean Shift Algorithm.

A database of 1,000 images randomly chosen from the Corel Gallery was used in the experiments. In the first processing step we compute different descriptions of the color distribution of an image. The CIELAB color space and the distance measure using the metric M_2 as in Eq. 6.4 were chosen for these experiments. In the second step we use these descriptions to approximate the quadratic form-based distance measure from Eq. 6.2. In the retrieval simulation we use every image in the database as a query image and search the whole image database. The result is then compared to the standard method based on the full histogram of 512 bins. This allows us to evaluate the approximation performance of different methods in the context of color-based image retrieval. Again ANMRR is used in the evaluation.

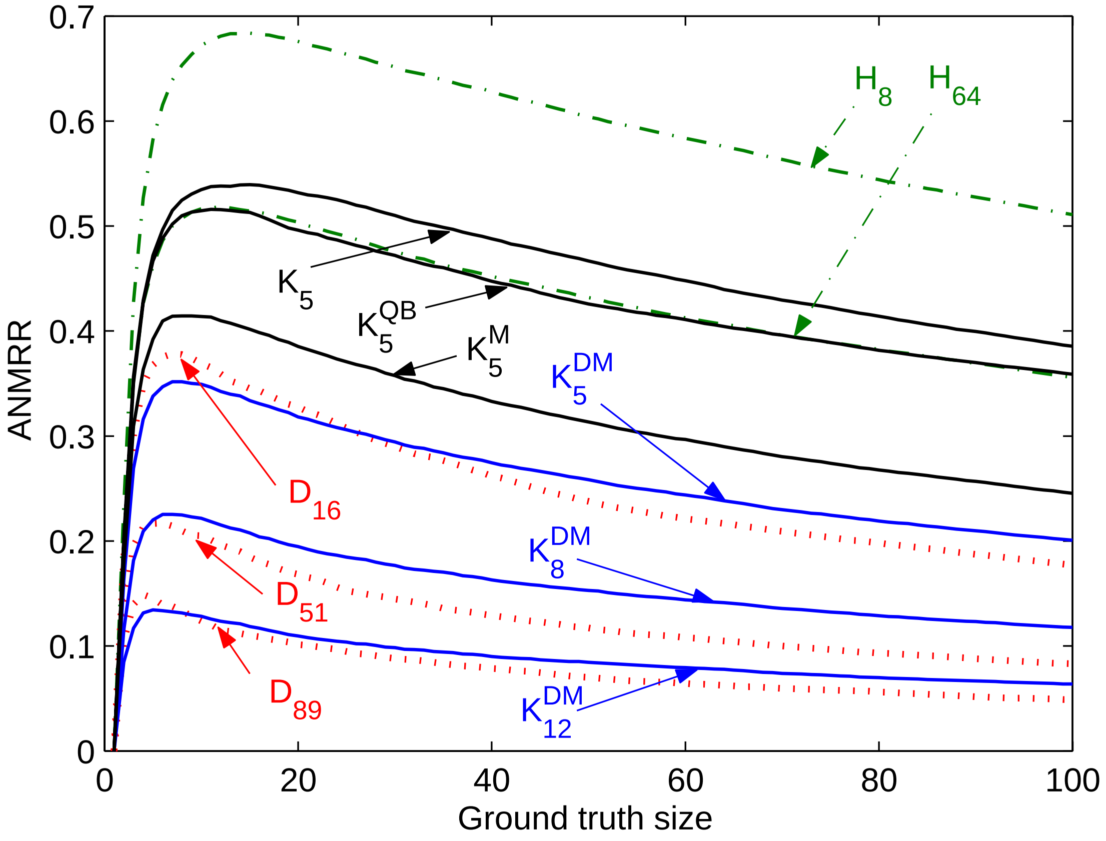


Figure 6.3: ANMRR of 1,000 queries in the Corel database using different histogram compression methods compared to the full histogram-based-method.

In the dominant color-based method, images are segmented into several homogenous regions. The clustering method we used was the Mean Shift Algorithm (Comaniciu and Meer, 1999b). Three different parameter settings were used to cluster each image in the database. The resulting clustered images consisted on average of 8, 25.5 and 44.5 segmented regions. The dominant color of each region is then quantized to one of 512 CIELAB values in the original method in order to speed up the search algorithm. Each region is then described by two parameters: the probability of a pixel lying in this region and the index of the dominant color of the region. An image which is segmented into n dominant color regions is then described by $2 \times n$ parameters. An example of segmented image by the Mean Shift Algorithm[1] is shown in Fig. 6.2.

[1] Implementation of the Mean Shift Algorithm in MatLab can be downloaded from `http://www.itn.liu.se/~lintr/www/demo.html`. Detailed of the algorithm and the original source code can be found at the homepage of the Robust Image Understanding Laboratory, Rutgers University, `http://www.caip.rutgers.edu/riul`

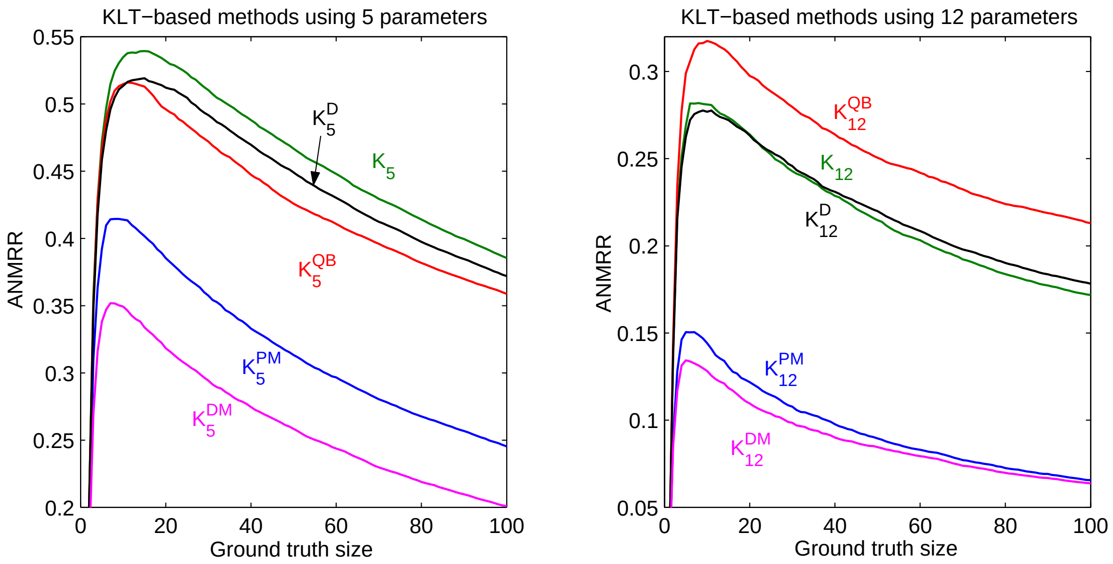


Figure 6.4: ANMRR of 1,000 queries in the Corel database using different KLT-based histogram compression methods compared to the full histogram-based method.

ρ (normalized σ)	K_5^{QB}	K_5	$K_5^{\mathbb{D}}$	$K_5^{\mathbb{M}}$	$K_5^{\mathbb{D}M}$	D_{16}	H_8
0.08	0.418	0.575	0.561	0.154	0.116	0.259	0.640
0.15	0.441	0.542	0.526	0.237	0.204	0.275	0.643
0.3	0.484	0.519	0.500	0.373	0.308	0.310	0.661
0.7	0.545	0.513	0.482	0.441	0.409	0.374	0.693
ρ (normalized σ)	K_{12}^{QB}	K_{12}	$K_{12}^{\mathbb{D}}$	$K_{12}^{\mathbb{M}}$	$K_{12}^{\mathbb{D}M}$	D_{51}	H_{64}
0.08	0.131	0.303	0.336	0.027	0.021	0.123	0.466
0.15	0.203	0.269	0.275	0.055	0.051	0.135	0.471
0.3	0.290	0.254	0.254	0.116	0.106	0.159	0.489
0.7	0.257	0.533	0.248	0.189	0.183	0.208	0.524

Table 6.2: Mean values of ANMRR of 1,000 queries in the Corel database when the ground truth size varies from 10 to 40 for different histogram compression methods compared to the full histogram-based method. Different metrics M were used.

For KLT-based methods operating on space $\mathbb{D}$, we used for every image its 40 nearest neighbors to estimate the space of local histogram differences.

Fig. 6.3 and Fig. 6.4 show some of the comparison results with different lengths of query windows for the case where the metric M_2 is defined as in Eq. 6.4 using $\rho = 0.3$. Different KLT-based methods are compared in Fig. 6.4. Results with other choices of ρ are collected in Table 6.2.

The results from these experiments show that:

- Incorporating information from the structure of the color space and applying KLT in the space of differences between neighboring histograms make the search results in the approximated feature space better correlated to the original full histogram method. The proposed method $K^{\mathbb{D}M}$, which combines the two ideas described above, gives the best performance compared to the other methods in all experiments. For example in Fig. 6.3 $K_5^{\mathbb{D}M}$, using only 5 parameters, gives the same retrieval performance as the dominant color-based method using 16 parameters. It is superior to the full histogram-based method using 64 parameters. $K_{12}^{\mathbb{D}M}$ using only 12 parameters gives about the same retrieval performance as the dominant color-based method using 89 parameters.
- When σ is small, the K^{QB} method described in QBIC (Hafner et al., 1995) is also comparable to the standard full histogram-based method. This is, however, the case when the mutual similarity between dissimilar colors is overestimated. When σ is increased, or the metric M becomes more diagonally dominant, the retrieval performance of the K^{QB} method decreases, compared to other KLT-based methods which are not solely based on the matrix M.
- For large values of $K (K \geq 15)$, results of K^{DM} methods which incorporate the color metric M converged to the standard method faster than K^{QB}.
- The dominant color-based method is fairly good while simple KLT and coarse histogram-based methods show poor results. Performance of the coarse histogram with 64 parameters is inferior to using only 4 parameters in our K_4^{DM} method.

In order to test these conclusions, experiments with the larger databases were carried out.

6.4.3 Experiments with the MPEG-7 database

In the third set of experiments, KLT-based methods are investigated further with the MPEG-7 databases of 5,466 color images. Both objective and subjective queries are used.

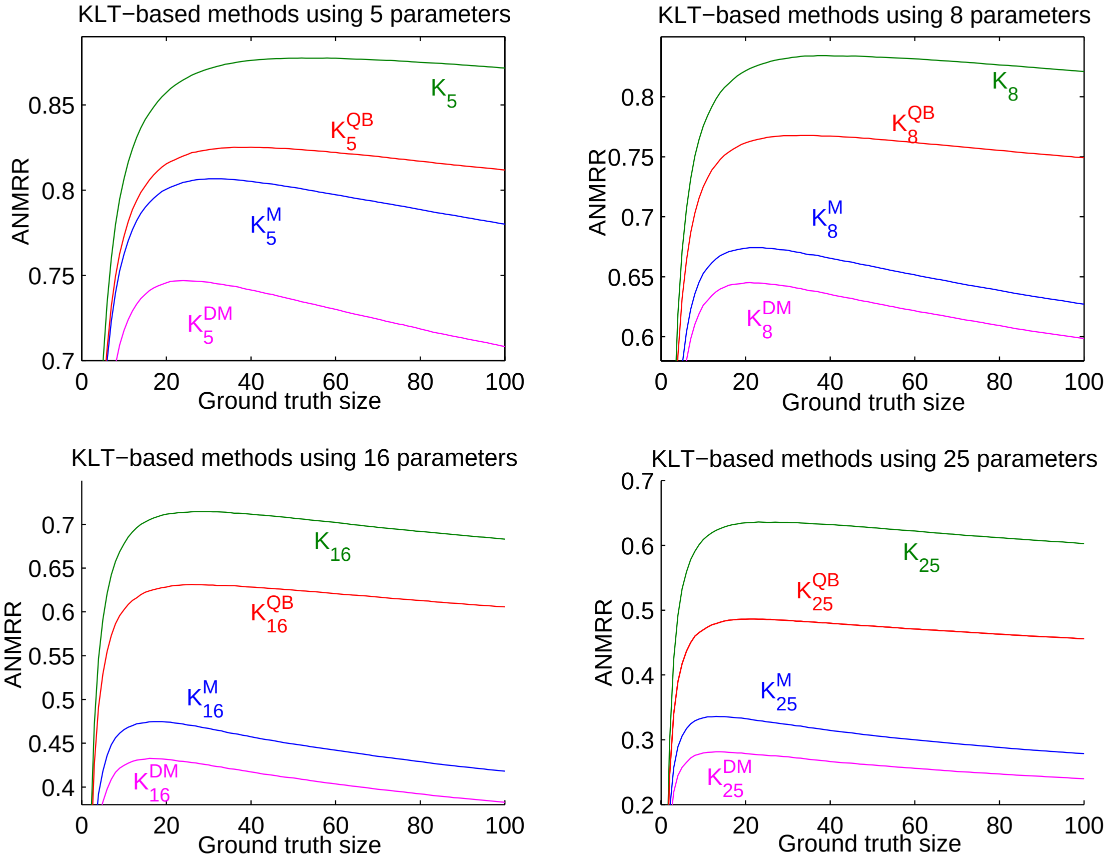


Figure 6.5: ANMRR of 5,466 queries in the MPEG-7 database using different KLT-based histogram compression methods compared to the full histogram-based method.

First the same experiments as in the previous section were performed with the MPEG-7 database. The only different setting was that the number of neighboring images of each image used when constructing the space of local histogram differences is 100 images. Several color spaces, including HSV, RGB and CIELAB, are used in these experiments. Fig. 6.5 and Table 6.3 show the results for different color spaces.

We also used 50 standard queries as subjective search criteria to compare the retrieval performance of these KLT-based methods. The results are shown in Table 6.4.

In another experiment, we select a set of 20 images, where 10 of them are from standard queries, and the other 10 are famous images in image processing such as Lena, Peppers, Mandrill, Parrots, etc. From each of these 20 images a new set of 20 images are generated by adding noise and sub-sampling the images. There are totally 420 images. The parameters that control the generated

Color space and Desc. of the method	K^{QB}	K	K^{M}	K^{MD}
HSV 16x4x4, # of parameters K = 5	0.673	0.628	0.491	0.490
HSV 16x4x4, K = 8	0.544	0.544	0.386	0.365
HSV 16x4x4, K = 16	0.377	0.414	0.197	0.182
HSV 16x4x4, K = 25	0.266	0.314	0.114	0.107
RGB 8x8x8, K = 5	0.775	0.576	0.436	0.419
RGB 8x8x8, K = 8	0.729	0.405	0.268	0.243
RGB 8x8x8, K = 16	0.546	0.227	0.102	0.091
RGB 8x8x8, K = 25	0.450	0.153	0.044	0.041
CIELAB 8x8x8, K = 5	0.558	0.579	0.475	0.455
CIELAB 8x8x8, K = 8	0.505	0.453	0.319	0.292
CIELAB 8x8x8, K = 16	0.425	0.251	0.151	0.137
CIELAB 8x8x8, K = 25	0.345	0.165	0.075	0.072

Table 6.3: Different KLT-based methods compared to the full histogram method. Mean values of ANMRR of 5,466 queries in the MPEG-7 image database when the ground truth size varies from 10 to 40

Color space and Desc. of the method	K^{QB}	K	K^{M}	K^{MD}
HSV 16x4x4, # of parameters = 8	0.422	0.337	0.337	0.333
HSV 16x4x4, K = 16	0.352	0.247	0.257	0.263
HSV 16x4x4, K = 25	0.297	0.238	0.248	0.247
RGB 8x8x8, K = 8	0.487	0.381	0.311	0.316
RGB 8x8x8, K = 16	0.347	0.283	0.232	0.229
RGB 8x8x8, K = 25	0.288	0.275	0.200	0.200
CIELAB 8x8x8, K = 8	0.336	0.383	0.322	0.301
CIELAB 8x8x8, K = 16	0.287	0.298	0.251	0.233
CIELAB 8x8x8, K = 25	0.266	0.256	0.224	0.222

Table 6.4: Different KLT-based methods are compared using the 50 standard queries in the MPEG-7 image database.

images are: P_s = percentage of sampled pixels, P_n = percentage of pixels with added noise, and R_n = the range of the noise magnitudes. Noise is uniformly distributed. Only the RGB color space is used in this experiment. Each set of 20 generated images is intended to have similar color distributions as the original image. We then take these 20 images as the ground truth when retrieving the original image. The average results of 20 different queries are collected in Table 6.5.

The results from the simulation of the search process on both objective and subjective queries of the MPEG-7 database containing 5,466 images, all agreed with the results obtained from the previous section.

P_s	P_n	R_n	# of Dim.	K^{QB}	K	K^M	K^{MD}
20	20	20	5	0.0181	0.0119	0.0111	0.0060
20	20	20	8	0.0098	0.0084	0.0059	0.0049
20	20	20	16	0.0111	0.0051	0.0042	0.0035
20	20	20	25	0.0046	0.0033	0.0032	0.0031
20	20	40	5	0.1225	0.0429	0.0403	0.0346
20	20	40	8	0.0458	0.0200	0.0235	0.0206
20	20	40	16	0.0215	0.0142	0.0181	0.0172
20	20	40	25	0.0139	0.0134	0.0173	0.0172
40	20	20	5	0.0181	0.0116	0.0121	0.0063
40	20	20	8	0.0098	0.0084	0.0060	0.0051
40	20	20	16	0.0111	0.0048	0.0043	0.0035
40	20	20	25	0.0041	0.0031	0.0030	0.0029
60	10	50	5	0.0302	0.0110	0.0144	0.0111
60	10	50	8	0.0192	0.0090	0.0071	0.0068
60	10	50	16	0.0115	0.0045	0.0053	0.0040
60	10	50	25	0.0038	0.0030	0.0029	0.0028

Table 6.5: ANMRR of 20 generated queries for the MPEG-7 image database.

6.4.4 Experiments with the Matton database

Finally we extend the comparison to the large Matton image database containing 126,604 images. The experiment setup is as in the second set of experiments described in Section 4.1. The color histograms were computed in the HSV color space with 16x4x4 bins. A set of 5,000 images was selected randomly, the basis of different KLT-based methods are then computed from this set. For KLT-based methods operating on the space $\mathbb{D}$, we used for every image its 100 nearest neighbors to represent the local histogram differences.

Fig. 6.6 shows the average results when all 5,000 images in the training set were used as query images. We also selected another 5,000 images not in the training set as query images in the image retrieval simulation, the average results for this set are collected in Fig. 6.7

20 queries from the set of 420 generated images as described in section 4.3 are also used to evaluate KLT-based methods in the Matton database. The results are shown in Table 6.6

P_s	P_n	R_n	# of Dim.	K^{QB}	K	K^M	K^{MD}
40	30	60	5	0.317	0.520	0.050	0
40	30	60	8	0.336	0.083	0.014	0.001
40	30	60	16	0.507	0.007	0	0
40	30	60	25	0.174	0.001	0	0
40	30	50	5	0.312	0.445	0.045	0
40	30	50	8	0.305	0.068	0.007	0.001
40	30	50	16	0.442	0.005	0	0
40	30	50	25	0.135	0.001	0	0
40	25	50	5	0.240	0.353	0.032	0
40	25	50	8	0.232	0.054	0.002	0
40	25	50	16	0.332	0.003	0	0
40	25	50	25	0.093	0.0030	0	0

Table 6.6: ANMRR of 20 generated queries for the Matton database.

As we expected, the results done on large database also agreed with earlier results of the small-scale experiments on the Corel database of 1,000 images.

6.5 Summary

We applied KLT-based approximation methods to color-based image retrieval. We presented different strategies combining two ideas: Incorporating information from the structure of the color space and using projection methods in the space of color histograms and the space of differences between neighboring histograms. The experiments with three databases of more than 130,000 images

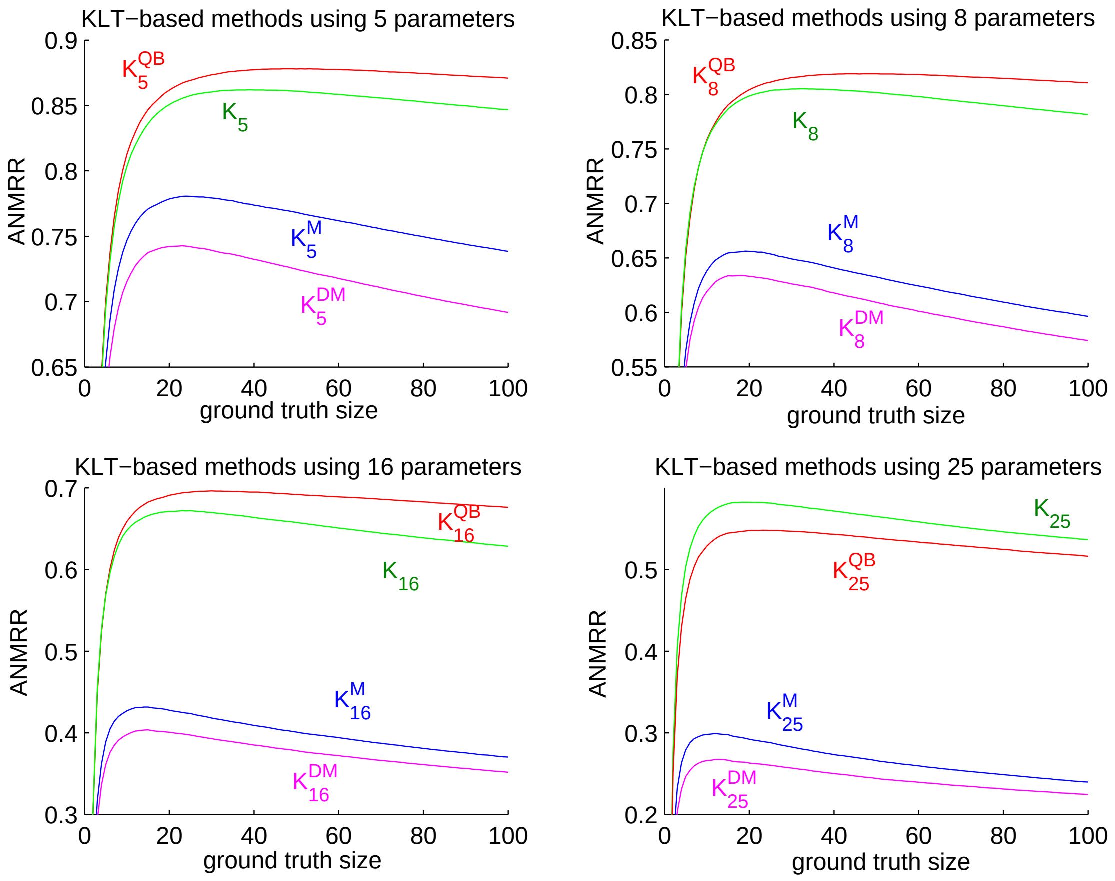


Figure 6.6: ANMRR of 5,000 queries in the Matton database using different KLT-based histogram compression methods compared to the full histogram-based method. 5,000 query images were selected from the training set.

show that the method which combines both the color metric and the difference of histograms space gives very good results compared to other existing methods.

The general strategy of using problem-based distance measures and differences of histograms outlined above is quite general and can be applied for other features used in content-based image retrieval applications.

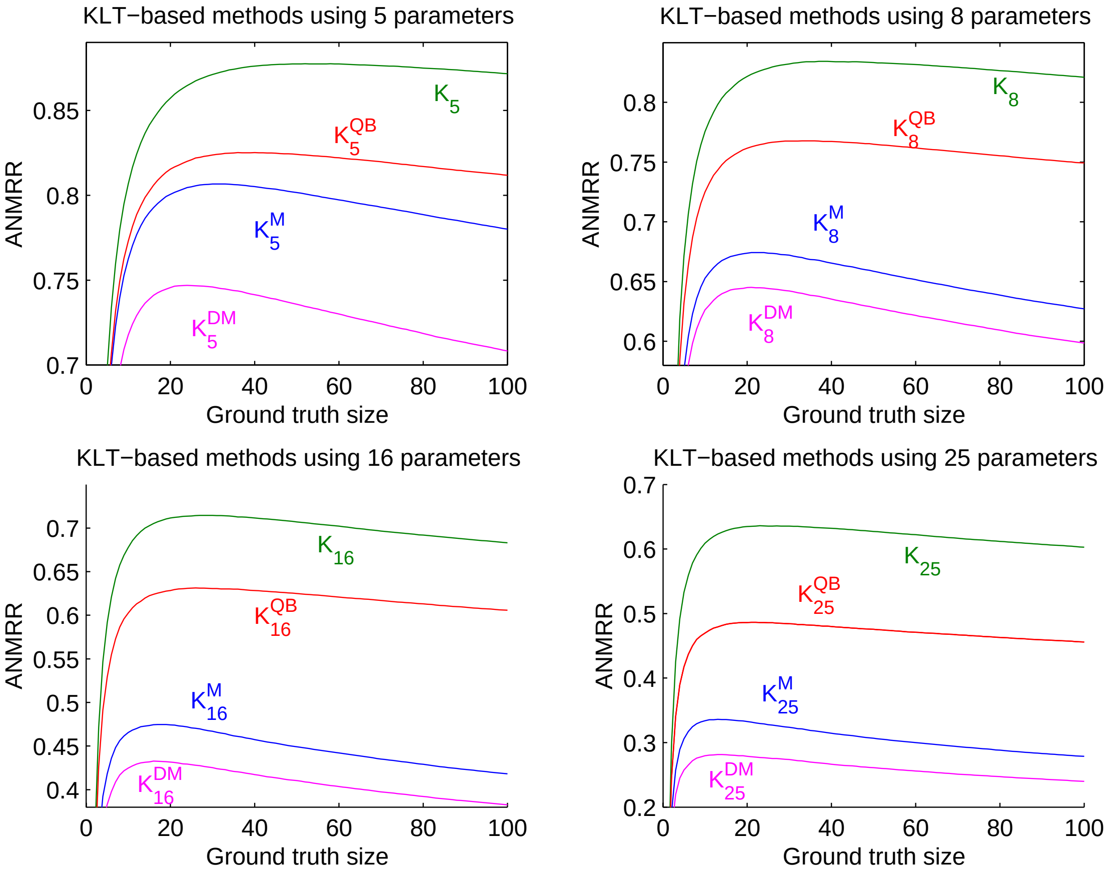


Figure 6.7: ANMRR of 5,000 queries in the Matton database using different KLT-based histogram compression methods compared to the full histogram-based method. 5,000 query images were not selected from the training set.

Chapter 7

PHYSICS-BASED COLOR INVARIANTS

In this chapter we investigate applications of physical models to determine homogeneously colored regions invariant to geometry changes such as surface orientation change, shadows and highlights. Many of the earlier results were derived heuristically, and none of them provide a solution to finding all possible invariants and the dependency between them. Using invariant theory, we can systematically answer such questions. Physical models that have been used are the Kubelka-Munk, the Dichromatic Reflection Model and its extended version. We also propose a robust region-merging algorithm utilizing the proposed color invariant features for color image segmentation applications.

7.1 Introduction

The information in a color image depends on many factors: the scene illumination, the reflectance characteristics of the objects in the scene, and the camera (its position, viewing angle, and sensitivity of sensors). In many applications, for example in color image segmentation or color object recognition, the main interest is however the physical content of the objects in the scene. Deriving features which are robust to image capturing conditions such as illumination changes, viewing angles, and geometry changes of the surface of the objects is a crucial step in such applications.

The interaction between light and objects in the scene is very complicated. Usually intricate models such as the Transfer Radiative Theory or Monte-Carlo simulation methods are needed to describe what happenes when light hits objects. Previous studies of color invariance are, therefore, mostly based on simpler semi-empirical models such as the Dichromatic Reflection Model (Shafer, 1985), or the model proposed by Kubelka and Munk (Kubelka and Munk, 1931). In such methods (Brill, 1990; Klinker, 1993; Gevers and Smeulders, 2000; Stokman, 2000; Finlayson and Schaefer, 2001; Geusebroek et al., 2001; Tran and Lenz, 2002a) invariant features are usually derived heuristically based on assumptions on the physical processes to simplify the form of the underlying physical model. None of them discuss questions such as the dependency between invariance features, or how many of such invariants are really independent, etc. These questions can be answered by using invariant theory (Olver, 1995; Eberly, 1999).

In this chapter, we concentrate on deriving color invariants using different physical models. Invariant theory is used to systematically derive all independent invariants with the help of symbolic mathematical software packages like Maple$^{\mathrm{TM}}$. In the next section, a brief introduction to invariant theory is presented using simple examples for illustration. In section 7.3 the Dichromatic Reflection Model (Shafer, 1985), its extended version, and the application to derive color invariants are described. A review of previous studies summarizing the assumptions that have been used is also included in this section. Section 7.4 investigates the Kubelka-Munk model (Kubelka and Munk, 1931) and its application in deriving color invariants to color invariant problems. We clarify under which assumptions the model is applicable. Based on the analysis of the model, both previous results and our proposed model are derived. Under simplified illumination conditions, we also show that most of the proposed invariant features are also invariant to illumination changes. This is discussed in section 7.5. Color invariant features are then used for color image segmentation. A robust region-merging algorithm is proposed to deal with the noisy feature vectors in section 7.6 before conclusions are drawn in the last section.

7.2 Brief Introduction to Invariant Theory

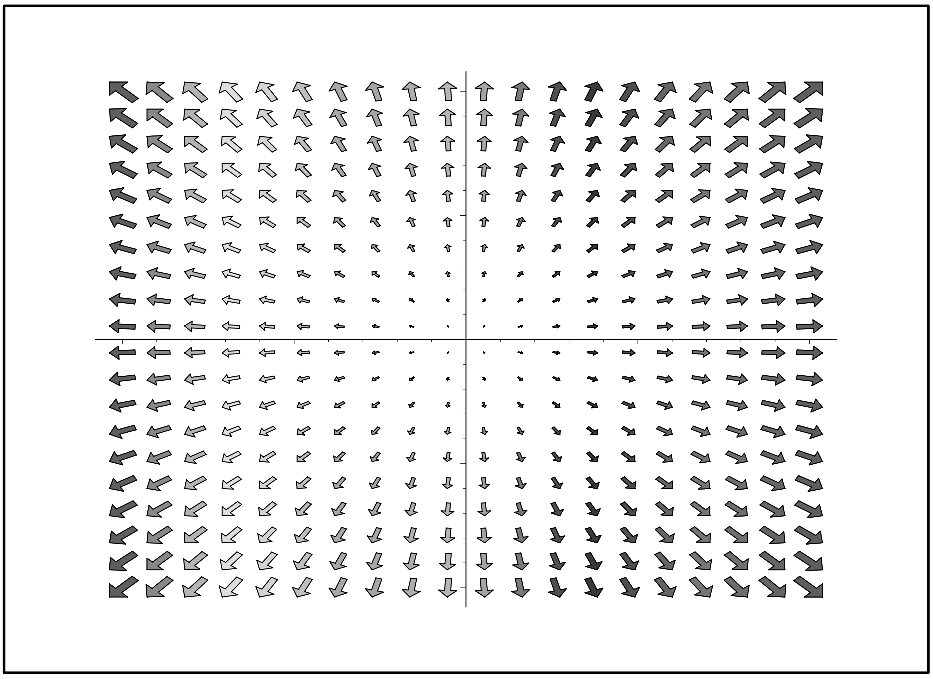


Figure 7.1: Vector field $V = (x, y)$ as in Eq. 7.3

7.2.1 Vector Fields

Let $\mathbb{R}^n$ denote an n-tuple of real numbers. A vector field is defined as a function $V : \mathbb{R}^n \to \mathbb{R}^n$. Denoting the k^{th} component of V as $v_k : \mathbb{R}^n \to \mathbb{R}$, the vector field can be written as an n-tuple:

$$V = (v_1(x), \ldots, v_n(x)), x \in \mathbb{R}^n \tag{7.1}$$

or can be thought of as a column vector when used in matrix calculations. We can also write the vector field as a linear combination as follows:

$$V = \sum_{k=1}^{n} v_k(x) \frac{\partial}{\partial x_k} \tag{7.2}$$

where the symbols $\partial/\partial x_k$ are place-keepers for the components. In this form, a vector field V can be seen as a directional derivative operator and can be applied to functions $f : \mathbb{R}^n \to \mathbb{R}$

An example of vector field $V = (x, y)$ is illustrated in Fig. 7.1

$$V = x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y} \tag{7.3}$$

7.2.2 Invariants

Given a vector field $V : \mathbb{R}^n \rightarrow \mathbb{R}^n$, an invariant is a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ such that the directional derivative satisfies

$$Vf = \sum_{k=1}^{n} v_k(x) \frac{\partial f}{\partial x_k} = 0 \tag{7.4}$$

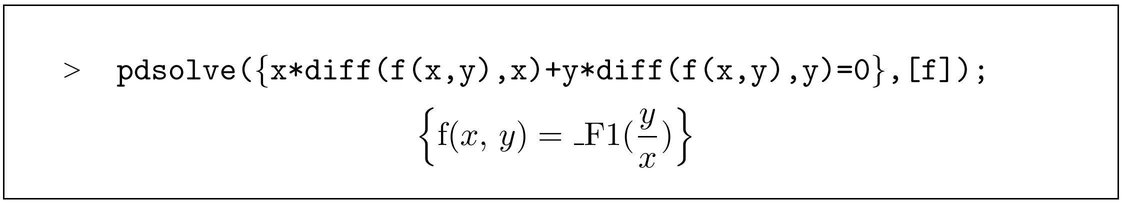
That is f remains constant as you walk in the direction of V.

For example, considering the vector field $V = (x, y)$ in Fig 7.1 and Eq. 7.3, a function $f(x, y)$ is an invariant if

$$0 = Vf = x\frac{\partial f(x,y)}{\partial x} + y\frac{\partial f(x,y)}{\partial y} \tag{7.5}$$

Solving the above differential equation gives us the solution $f(x, y) = F(y/x)$ which means that all functions of y/x have constant value when going along the direction of vector field $V = (x, y)$.

The differential equation Eq. 7.5 can be solved by using a symbolic mathematical software package such as Maple$^{\text{TM}}$. Fig. 7.2 shows a very simple Maple script to solve Eq. 7.5. Maple will be used throughout this chapter to solve differential equations.



```
>  pdsolve({x*diff(f(x,y),x)+y*diff(f(x,y),y)=0},[f]);
```

$$\left\{\mathrm{f}(x,\,y) = \_\mathrm{F1}(\frac{y}{x})\right\}$$

Figure 7.2: A simple Maple script to solve the differential equation Eq. 7.5

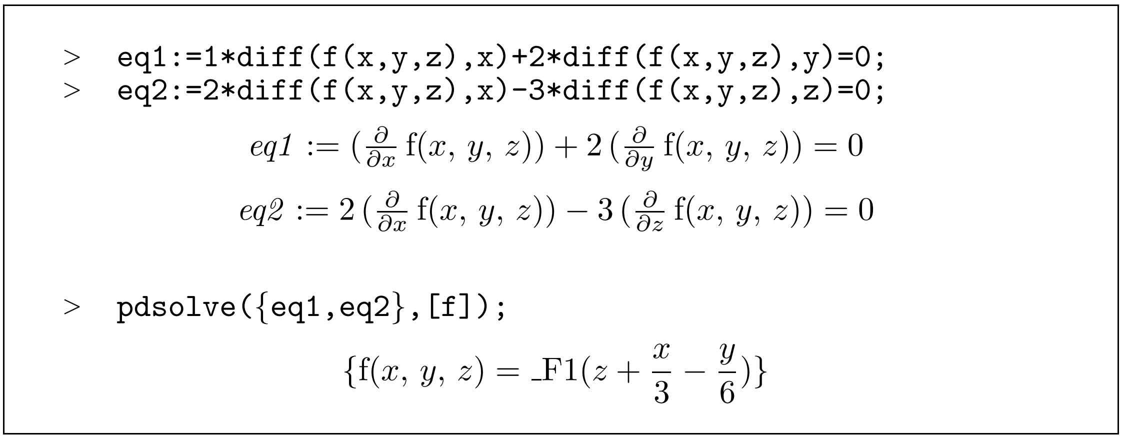
We can also look for invariants with respect to more than one vector field. Let $V_k : \mathbb{R}^n \rightarrow \mathbb{R}^n$ for $k = 1, \ldots, K$ be K vector fields. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is an invariant for the vector fields if $V_k f = 0$ holds for all k. For example, consider the vector fields on $\mathbb{R}^3$,

$$\begin{aligned} V_1 &= \frac{\partial}{\partial x} + 2\frac{\partial}{\partial y} \\ V_2 &= 2\frac{\partial}{\partial x} - 3\frac{\partial}{\partial z} \end{aligned} \tag{7.6}$$

A function $f(x, y, z)$ is an invariant if

$$\begin{aligned} 0 = V_1 f &= \frac{\partial f(x,y,z)}{\partial x} + 2\frac{\partial f(x,y,z)}{\partial y} \\ 0 = V_2 f &= 2\frac{\partial f(x,y,z)}{\partial x} - 3\frac{\partial f(x,y,z)}{\partial z} \end{aligned} \tag{7.7}$$

Fig. 7.3 shows the Maple program to solve the above system of differential equations. The solution is $f(x, y, x) = F(x/3 - y/6 + z)$.



```
> eq1:=1*diff(f(x,y,z),x)+2*diff(f(x,y,z),y)=0;
> eq2:=2*diff(f(x,y,z),x)-3*diff(f(x,y,z),z)=0;
```

$$eq1 := \left(\tfrac{\partial}{\partial x}\,\mathrm{f}(x,\,y,\,z)\right) + 2\left(\tfrac{\partial}{\partial y}\,\mathrm{f}(x,\,y,\,z)\right) = 0$$

$$eq2 := 2\left(\tfrac{\partial}{\partial x}\,\mathrm{f}(x,\,y,\,z)\right) - 3\left(\tfrac{\partial}{\partial z}\,\mathrm{f}(x,\,y,\,z)\right) = 0$$

```
> pdsolve({eq1,eq2},[f]);
```

$$\{\mathrm{f}(x,\,y,\,z) = \_\mathrm{F1}(z + \frac{x}{3} - \frac{y}{6})\}$$

Figure 7.3: A simple Maple script to solve the differential equations Eq. 7.7

7.2.3 Number of Independent Invariants

Given K vector fields $V_k : \mathbb{R}^n \to \mathbb{R}^n$ for $k = 1, \ldots, K$, the previous section discussed how to derive the invariants. They are the solutions of the system of differential equations $V_k f = 0$. The next question is how many functionally independent invariants there are for given vector fields. This question can be answered without solving any differential equation.

Look at the example in Fig. 7.3, the two functions $f_1(x, y) = y/x$ and $f_2(x, y) = 3 + sin((x + y)/x)$ are both invariants of the vector field $V = (x, y)$ in Fig. 7.1. In fact, for any differentiable function $g : \mathbb{R} \to \mathbb{R}$, the function $g(y/x)$ is another invariant of the vector field. But they all depend on the quantity y/x and provide no really new solution.

Let $f_k : \mathbb{R}^n \to \mathbb{R}^n$ for $k = 1, \ldots, K$ (where $K < n$) be K differentiable functions. These functions are said to be functionally independent at $x \in \mathbb{R}^n$ if and only if the $K \times n$ matrix of first derivatives $[\partial f_i/\partial x_j]$ has full rank K.

For the above example

$$\operatorname{rank}\left([\partial f_i/\partial x_j]\right) = \operatorname{rank}\begin{bmatrix} -yx^{-2} & x^{-1} \\ -\cos\left(\frac{x+y}{x}\right) yx^{-2} & \cos\left(\frac{x+y}{x}\right) x^{-1} \end{bmatrix} = 1 \tag{7.8}$$

Therefore the two functions f_1 and f_2 are dependent.

By definition, an invariant f must be the solution of K differential equations: $V_k f = 0$ for $k = 1, \ldots, K$. One might expect that there will be $n - K$ independent invariants since there are only $n-K$ degrees of freedom left. However, this is not always true. Look at the following example with the two vector fields ($K = 2$)

$$\begin{aligned} V_1 &= x_1 \frac{\partial}{\partial x_2} + x_3 \frac{\partial}{\partial x_4} \\ V_2 &= x_2 \frac{\partial}{\partial x_1} + x_4 \frac{\partial}{\partial x_3} \end{aligned} \tag{7.9}$$

acting on the four-dimensional space $\mathbb{R}^4$. We find only one independent invariant as can be seen in the Maple implementation in Fig. 7.4. The reason is that the Lie product of the two vector fields $[V_1, V_2] = V_1V_2 - V_2V_1$ is another vector field

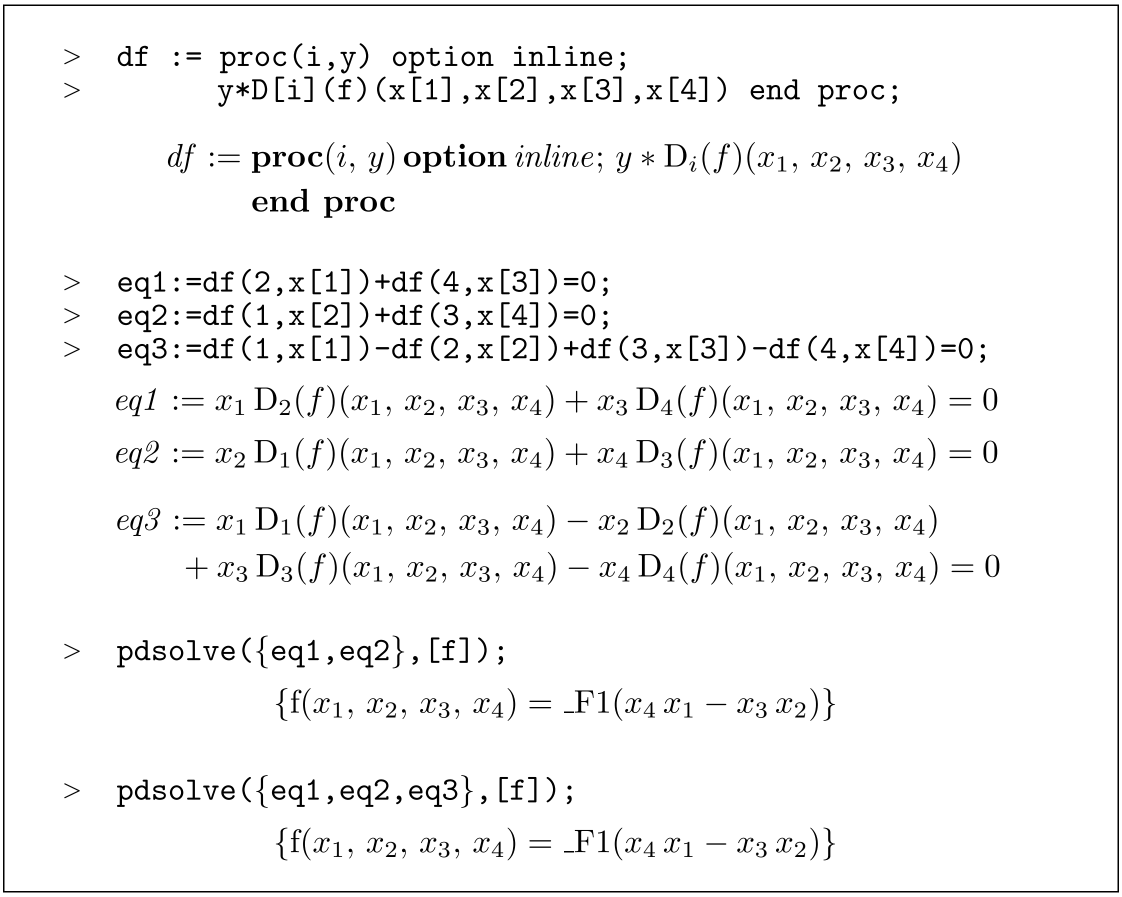
$$[V_1, V_2]f = V_1(V_2 f) - V_2(V_1 f) = 0$$

The Lie product vector field, in this particular case, is independent of both V_1 and V_2.

$$\begin{aligned} [V_1, V2] &= V_1(V_2) - V_2(V_1) \\ &= x_1 \frac{\partial}{\partial x_1} - x_2 \frac{\partial}{\partial x_2} + x_3 \frac{\partial}{\partial x_3} - x_4 \frac{\partial}{\partial x_4} \end{aligned} \tag{7.10}$$

This gives a new differential equations and, therefore, we have only one independent invariant.

Given K vector fields $V_k : \mathbb{R}^n \to \mathbb{R}^n$ for $k = 1, \ldots, K$ with $K < n$, the Lie algebra of the vector fields is obtained by constructing the smallest vector space which contains all sums, scalar multiples, and Lie products of the V_k. We write this vector space $L(V_1, \ldots, V_k)$. The dimensions of this vector space $L(V_1, \ldots, V_k)$ can be different from K. Invariant theory shows that the number of functionally independent invariants is not $n-K$ but n-dim(L), where dim(L) is the number of dimensions of the Lie algebra.



```
>  df := proc(i,y) option inline;
>        y*D[i](f)(x[1],x[2],x[3],x[4]) end proc;
```

$$df := \mathbf{proc}(i,\, y)\,\mathbf{option}\, \mathit{inline};\; y * \mathrm{D}_i(f)(x_1,\, x_2,\, x_3,\, x_4)\;\mathbf{end\ proc}$$

```
>  eq1:=df(2,x[1])+df(4,x[3])=0;
>  eq2:=df(1,x[2])+df(3,x[4])=0;
>  eq3:=df(1,x[1])-df(2,x[2])+df(3,x[3])-df(4,x[4])=0;
```

$$\mathit{eq1} := x_1\,\mathrm{D}_2(f)(x_1,\, x_2,\, x_3,\, x_4) + x_3\,\mathrm{D}_4(f)(x_1,\, x_2,\, x_3,\, x_4) = 0$$

$$\mathit{eq2} := x_2\,\mathrm{D}_1(f)(x_1,\, x_2,\, x_3,\, x_4) + x_4\,\mathrm{D}_3(f)(x_1,\, x_2,\, x_3,\, x_4) = 0$$

$$\mathit{eq3} := x_1\,\mathrm{D}_1(f)(x_1,\, x_2,\, x_3,\, x_4) - x_2\,\mathrm{D}_2(f)(x_1,\, x_2,\, x_3,\, x_4) + x_3\,\mathrm{D}_3(f)(x_1,\, x_2,\, x_3,\, x_4) - x_4\,\mathrm{D}_4(f)(x_1,\, x_2,\, x_3,\, x_4) = 0$$

```
>  pdsolve({eq1,eq2},[f]);
```

$$\{\mathrm{f}(x_1,\, x_2,\, x_3,\, x_4) = \_\mathrm{F1}(x_4\, x_1 - x_3\, x_2)\}$$

```
>  pdsolve({eq1,eq2,eq3},[f]);
```

$$\{\mathrm{f}(x_1,\, x_2,\, x_3,\, x_4) = \_\mathrm{F1}(x_4\, x_1 - x_3\, x_2)\}$$

Figure 7.4: The Maple program to solve the differential equations Eq. 7.9. As in the last line, adding one more vector field into the system does not change the result since the added vector field is the Lie product of the two vector fields.

7.2.4 Examples of One-Parameter Subgroups

We recall that a one-parameter subgroup is a subgroup that depends on only one-parameter. We will only consider cases where the group elements are 2 × 2 matrices and the space on which they operate is the two-dimensional real Euclidean vector space $\mathbb{R}^2$. Particularly, in this section, the following four one-parameter subgroups are addressed: rotation, isotropic scaling, anisotropic scaling, and shearing. Such groups are particularly useful for our derivation in the coming sections.

The one-parameter subgroup of rotations with angle α in two-dimensional space $\mathbb{R}^2$ can be defined in matrix form as:

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{R(\alpha)} R(\alpha)\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \tag{7.11}$$

A function f is an invariant under the group of rotations $R(\alpha)$ if for all angles α we have:

$$f\left(R(\alpha)(x,y)'\right) = f\left(x,y\right) \quad \text{or} \quad \frac{df}{d\alpha}\mid_{\alpha=0}= 0 \tag{7.12}$$

The corresponding vector field V_α and solution for f of this one-parameter subgroup is given by:

$$\begin{aligned} V_\alpha &= y\frac{\partial}{\partial x} - x\frac{\partial}{\partial y} \\ f &= \_\mathrm{F}\left(x^2+y^2\right) \end{aligned} \tag{7.13}$$

The procedure to find an invariant is rather simple. It consists of three steps as shown in Fig. 7.5:

- Define equation(s) describing the underlying process,
- Differentiate the equation(s) with respect to the variable of the given problem at the origin point to find the vector field(s) V_k.
- Solve the differential equations $V_k f = 0$

Describing the underlying process

```
>  roteq:=f(cos(a)*x+sin(a)*y,-sin(a)*x+cos(a)*y);
```

$$\mathit{roteq} := \mathrm{f}(\cos(a)\,x + \sin(a)\,y,\ -\sin(a)\,x + \cos(a)\,y)$$

Deriving the vector field

```
>  rotvf:=map(simplify,eval(subs(a=0,diff(roteq,a))));
```

$$\mathit{rotvf} := \mathrm{D}_1(f)(x,\,y)\,y - \mathrm{D}_2(f)(x,\,y)\,x$$

Solve the differential equation $V_k f = 0$

```
>  pdsolve({rotvf},[f]);
```

$$\{\mathrm{f}(x,\,y) = \_\mathrm{F1}\left(x^2+y^2\right)\}$$

Figure 7.5: The Maple program to find invariants for the rotation one-parameter subgroup.

Invariants of the one-parameter subgroups of scaling and shearing operations can also be solved in a similar way. Their transformations in two-dimensional space $\mathbb{R}^2$ are:

$$\text{Isotropic scaling: } \begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{S_1(s)} S_1(s)\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \exp(s) & 0 \\ 0 & \exp(s) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{7.14}$$

$$\text{Anisotropic scaling: } \begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{S_2(s)} S_2(s)\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \exp(s) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{7.15}$$

$$\text{Shearing: } \begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{S_3(s)} S_3(s)\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{7.16}$$

Their corresponding vector fields and the invariants for each one-parameter subgroup are given by:

$$V_1 = x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}, \qquad f_1 = \_\mathrm{F}\left(\frac{x}{y}\right) \tag{7.17}$$

$$V_2 = y\frac{\partial}{\partial y}, \qquad f_2 = \_\mathrm{F}\left(x\right) \tag{7.18}$$

$$V_3 = y\frac{\partial}{\partial x}, \qquad f_3 = \_\mathrm{F}\left(y\right) \tag{7.19}$$

7.3 Methods using the Dichromatic Reflection Model

When light strikes a surface, it may pass through the interface and the medium. Many complicated interactions will take place. Because the medium's index of refraction differs from that of the air, some of the light will be reflected at the interface producing interface reflection, while another part will transfer through the medium. Transfer of light through a medium includes several fundamental processes such as absorption, scattering, and emission. Absorption is the process by which radiant energy is transformed into another form of energy, e.g. heat or light of different wavelength as in fluorescent materials. Scattering is the process by which the radiant energy is diffused in different directions. Emission is the process by which new radiant energy is created. A result of such processes is that some part of the incoming light will go back from the medium as illustrated in Fig. 7.6. The Dichromatic Reflection Model describes the relation between the incoming light to the interface of the surface and the reflected light which is a mixture of the light reflected at the material surface and the light reflected from the material body.

7.3.1 Dichromatic Reflection Model

The Dichromatic Reflection Model (Shafer, 1985) assumes that the light reflected $L(x, \lambda)$ from a surface of an inhomogeneous object can be decomposed

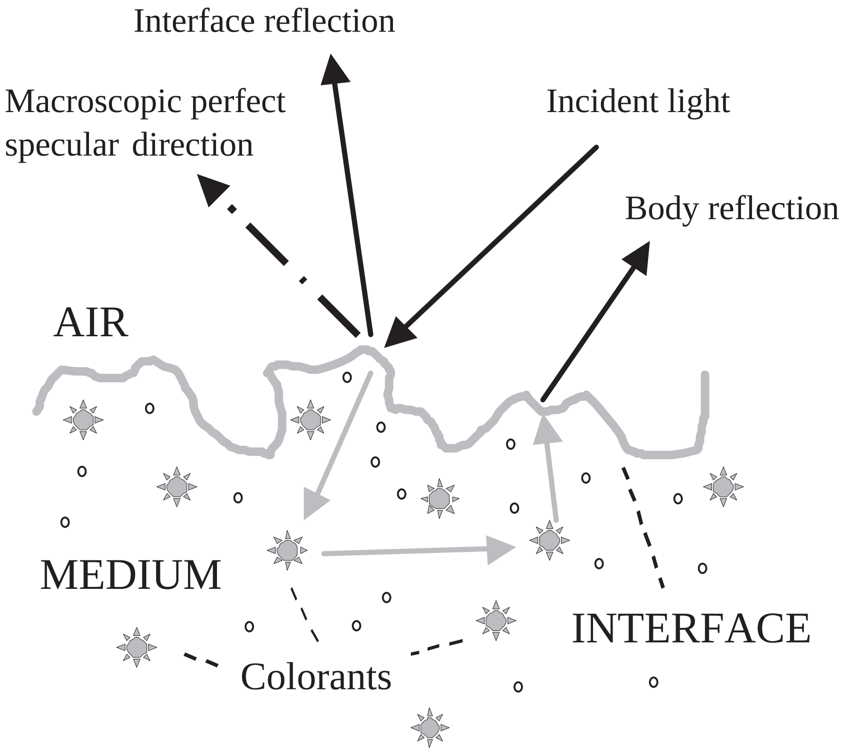


Figure 7.6: The light reflection of inhomogeneous material consists of two parts: interface reflection and body reflection. Note that most materials are optically rough with local surface normals differ from the macroscopic surface normal. The interface reflection will, therefore, be scattered at the macroscopic level as the body reflection part.

into two additive components, an interface (specular) reflectance and a body (diffuse) reflectance under all illumination-camera geometries.

$$L(x,\lambda) = m_S(x)L_S(\lambda) + m_D(x)L_D(\lambda) \tag{7.20}$$

The terms $L_S(\lambda)$ and $L_D(\lambda)$ describe the spectral power distributions of the specular and diffuse components. The subscript S denotes the Specular and D the Diffuse distribution. The parameter x denotes geometry changes including the angle of incidence light, the angle of remittance light, the phase angle, etc.

To express the model in terms of the surface reflectance, let $R_S(\lambda)$ and $R_D(\lambda)$ be the specular and diffuse reflectance respectively, and let $E(\lambda)$ be the spectral power distribution of the incident light. The reflected light is then given by:

$$L(x,\lambda) = m_S(x)R_S(\lambda)E(\lambda) + m_D(x)R_D(\lambda)E(\lambda) \tag{7.21}$$

and, equivalently, the total reflectance is described by

$$R(x, \lambda) = m_S(x) R_S(\lambda) + m_D(x) R_D(\lambda) \tag{7.22}$$

Consider an image of an infinitesimal surface patch, using N filters with spectral sensitivities given by $f_1(\lambda)...f_N(\lambda)$ to obtain an image of the surface patch illuminated by an incident light with spectral power distribution given by $E(\lambda)$. The measured sensor values $C_n(x)$ at pixel x in the image will be given by the following integral over the visible spectrum:

$$\begin{aligned} C_n(x) &= \int f_n(\lambda) \big[m_S(x) R_S(\lambda) E(\lambda) + m_D(x) R_D(\lambda) E(\lambda) \big] d\lambda \\ &= m_S(x) \int f_n(\lambda) E(\lambda) R_S(\lambda) d\lambda + m_D(x) \int f_n(\lambda) E(\lambda) R_D(\lambda) d\lambda \\ &= m_S(x) S_n + m_D(x) D_n \end{aligned} \tag{7.23}$$

If we collect the values $m_S(x), m_D(x)$ in the vector g and the values S_n, D_n in the vector h

$$\begin{aligned} g(x) &= g = (m_S(x), m_D(x)) \\ h_n &= h = (S_n, D_n) \end{aligned} \tag{7.24}$$

and denote the scalar product of two vectors g, h by $\langle g, h \rangle$ then we can write Eq. 7.23 as:

$$C_n(x) = \langle g, h \rangle \tag{7.25}$$

From this equation we see that the dichromatic model factors the measured pixel value $C_n(x)$ into two factors g and h. The factor h depends on the spectral properties of the sensor, the illumination source and the reflectance of the object, whereas the factor g depends only on the geometry features. A geometry invariant must be independent of the values $m_S(x)$ and $m_D(x)$, ie. the vector g. A feature which is invariant to illumination changes must be independent of $E(\lambda)$. The functions $f_n(\lambda), R_S(\lambda)$ and $R_D(\lambda)$ describe the dependency of the color measurement on the characteristics of the sensors and the material of the object in the scene.

The Dichromatic Reflection Model as presented above depends on the assumption that the illumination at any point comes from a single (point or extended) light source. It is more realistic to model the illumination as consisting of a light source plus an ambient or diffuse light $L^A(\lambda)$. Moreover, if the above equations hold locally, we could also extend the model to an illumination changing condition where the illumination $E(x, \lambda)$ is a function in both the spectral and the spatial variables. The extended model is thus given by:

$$\begin{aligned} L(x, \lambda) &= m_S(x) R_S(\lambda) E(x, \lambda) + m_D(x) R_D(\lambda) E(x, \lambda) + L^A(\lambda) \\ &= m_S(x) L_S(x, \lambda) + m_D(x) L_D(x, \lambda) + L^A(\lambda) \end{aligned} \tag{7.26}$$

where both $L_S(x, \lambda)$, $L_D(x, \lambda)$, and $E(x, \lambda)$ are functions of x and λ, the position of the pixel in the scene and the wavelength, respectively.

The measured sensor values $C_n(x)$ at pixel x in the image will be given by the following integral over the visible spectrum:

$$\begin{aligned} C_n(x) &= \int f_n(\lambda)\big[m_S(x)R_S(\lambda)E(x,\lambda) + m_D(x)R_D(\lambda)E(x,\lambda) + L^A(\lambda)\big]d\lambda \\ &= m_S(x)\int f_n(\lambda)E(x,\lambda)R_S(\lambda)d\lambda + m_D(x)\int f_n(\lambda)E(x,\lambda)R_D(\lambda)d\lambda \\ &\qquad\qquad + \int f_n(\lambda)L^A(\lambda)d\lambda \\ &= m_S(x)S_n(x) + m_D(x)D_n(x) + L^A_n \end{aligned} \tag{7.27}$$

The Dichromatic Reflection Model in Eq. 7.23 and its extended version in Eq. 7.27 are more general than typical models used in computer vision and computer graphics, and include most of these models as special cases (Shafer, 1985). Since it is general, consisting of two terms as in the standard model, and three terms in the extended model, it is quite difficult to directly use it for deriving color features which are invariants to either geometric or photometric terms.

Previous investigations used only the standard model and required additional assumptions in order to make Eq. 7.21 and Eq. 7.23 easier to deal with. Often it is reduced to only one term. Some of the assumptions (Klinker, 1993; Gevers and Stokman, 2000; Gevers and Smeulders, 2000; Stokman, 2000; Finlayson and Schaefer, 2001; Tran and Lenz, 2002a) that have been used are:

- Objects in the scene are all matte or dull, ie. there is only a body (diffuse) reflection term, $R_S(\lambda) = 0$ leading to:

$$C_n(x) = m_D(x)\int f_n(\lambda)E(x,\lambda)R_D(\lambda)d\lambda \tag{7.28}$$

- The color distribution has a skewed L or dog-leg shape, meaning that there are only two cases: either $m_D(x) = 0$, or $m_S(x) = 0$.
- The illumination of the scene is white and constant over the scene: $E(x, \lambda) = e = constant.$
- The illumination of the scene is daylight and can be well approximated using the Planck locus of the black-body radiator.
- The surfaces of the objects follow the Natural Interface Reflection (NIR) model, ie. $R_S(\lambda) = r_S$ is independent of the wavelength.

- The filters $f_n(\lambda)$ are narrow band. Then Eq. 7.23 becomes much easier since the integration is eliminated.

$$\begin{aligned} C_n(x) &= \int f_n(\lambda)\big[m_S(x)R_S(\lambda)E(x,\lambda) + m_D(x)R_D(\lambda)E(x,\lambda)\big]d\lambda \\ &= f(\lambda_n)E(x,\lambda_n)\big[m_S(x)R_S(\lambda_n) + m_D(x)R_D(\lambda_n)\big] \end{aligned} \tag{7.29}$$

- The images are assumed to be white-balanced

In the next section, we will relax these assumptions and systematically derive geometric color invariants using the framework presented in the previous section.

7.3.2 Geometric Invariants from the Dichromatic Reflection Model

We first look at the standard dichromatic reflection model in Eq. 7.23. The color values $C_n(x)$ can be measured, the $m_S(x)$ and $m_D(x)$ are unknown geometric terms, S_n and D_n are also unknown but independent of geometric properties. A geometric invariant, therefore, is a function f of color values and should not be dependent on the geometric terms $m_S(x)$ and $m_D(x)$.

We consider first the simplest case when the color information comes from only one pixel x.

$$C_n = C_n(x) = m_S(x)S_n + m_D(x)D_n$$

Each channel has one measurement C_n, but two unknowns S_n and D_n and two variables $m_S(x)$ and $m_D(x)$ from which all the invariants should be independent. All invariants, if they exist, will depend at least either on S_n or D_n. Therefore, this case gives no invariant which is a function of only the measurement C_n. Using information from neighboring pixels is necessary in order to derive geometry invariants based solely on color measurement values.

We consider next the case of using 2 pixels, say x_1 and x_2. Each pixel has N channels. Totally there are $2N$ values $C_n(x_p)$ collected in a system of $2N$ equations. In matrix notation we have

$$\begin{bmatrix} C_n^1 \\ C_n^2 \end{bmatrix} = \begin{bmatrix} C_n(x_1) \\ C_n(x_2) \end{bmatrix} = \begin{bmatrix} m_S(x_1) & m_D(x_1) \\ m_S(x_2) & m_D(x_2) \end{bmatrix} \cdot \begin{bmatrix} S_n \\ D_n \end{bmatrix} = M \cdot \begin{bmatrix} S_n \\ D_n \end{bmatrix} \tag{7.30}$$

The color values $(C_n^1, C_n^2)^T$ are obtained by multiplying the matrix M (containing the geometry terms) with the vector $(S_n, D_n)^T$ which is independent of geometry changes. An invariant function f is a mapping from the $2N$-dimensional space of real numbers to a real number:

$$f : \mathbb{R}^{2N} \to \mathbb{R}$$

This function should be constant under the transformations M. It is well-known that the 2×2 matrix M can be factored into four one-parameter group actions: a rotation with angle α, an isotropic scaling with scale factor s_1, an anisotropic scaling with scale factor s_2, and a shearing with shift s_3. To be invariant under the transformations M, a function f should be invariant along the vector fields of the four one-parameter subgroups described above. The action of each one-parameter group and its invariants have been discussed individually in section 7.2.4. This case is a combination of the four transformations of the above one-parameter subgroups and it can be solved as follows.

The number of independent invariants, as discussed in section 7.2.3, is obtained as the dimension of the space on which the invariants operate minus the dimension of the Lie algebra of the four vector fields. Since these four vector fields are independent, the Lie algebra has at least 4 dimensions leading to the maximum of possible independent invariants as

$$\text{maximum number of invariants} = 2N - 4 = 2(N-2) \tag{7.31}$$

In order to have an invariant, this number should be positive: $2(N-2) > 0$, or the number of channels N should be at least 3.

With 3 channels (as in an RGB image), there will be at most 2 independent invariants. For two pixels x_1 and x_2 and three channels, say R, G, B, we change the notation to

$$\begin{aligned}
\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} &= \begin{bmatrix} R(x_1) \\ R(x_2) \end{bmatrix} = \begin{bmatrix} m_S(x_1) & m_D(x_1) \\ m_S(x_2) & m_D(x_2) \end{bmatrix} \cdot \begin{bmatrix} S_R \\ D_R \end{bmatrix} = M \cdot \begin{bmatrix} S_R \\ D_R \end{bmatrix} \\
\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} &= \begin{bmatrix} G(x_1) \\ G(x_2) \end{bmatrix} = \begin{bmatrix} m_S(x_1) & m_D(x_1) \\ m_S(x_2) & m_D(x_2) \end{bmatrix} \cdot \begin{bmatrix} S_G \\ D_G \end{bmatrix} = M \cdot \begin{bmatrix} S_G \\ D_G \end{bmatrix} \\
\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} &= \begin{bmatrix} B(x_1) \\ B(x_2) \end{bmatrix} = \begin{bmatrix} m_S(x_1) & m_D(x_1) \\ m_S(x_2) & m_D(x_2) \end{bmatrix} \cdot \begin{bmatrix} S_B \\ D_B \end{bmatrix} = M \cdot \begin{bmatrix} S_B \\ D_B \end{bmatrix}
\end{aligned} \tag{7.32}$$

The four vector fields $V_{rot}, V_{isos}, V_{anis}, V_{shear}$ along the directions of the four one-parameter subgroups: rotation, isotropic scaling, anisotropic scaling, and shearing, respectively, are given by

$$\begin{aligned}
V_{rot} &= R_2\frac{\partial}{\partial R_1} - R_1\frac{\partial}{\partial R_2} + G_2\frac{\partial}{\partial G_1} - G_1\frac{\partial}{\partial G_2} + B_2\frac{\partial}{\partial B_1} - B_1\frac{\partial}{\partial B_2}\\
V_{isos} &= R_1\frac{\partial}{\partial R_1} + R_2\frac{\partial}{\partial R_2} + G_1\frac{\partial}{\partial G_1} + G_2\frac{\partial}{\partial G_2} + B_1\frac{\partial}{\partial B_1} + B_2\frac{\partial}{\partial B_2}\\
V_{anis} &= R_2\frac{\partial}{\partial R_2} + G_2\frac{\partial}{\partial G_2} + B_2\frac{\partial}{\partial B_2}\\
V_{shear} &= R_2\frac{\partial}{\partial R_1} + G_2\frac{\partial}{\partial G_1} + B_2\frac{\partial}{\partial B_1} \qquad (7.33)
\end{aligned}$$

It can be shown that the four vector fields are functionally independent and their Lie products do not create any new independent vector field which means that the dimensions of the Lie algebra of the four vector fields equals 4. There must be 2 independent invariants in this case.

Following the framework in the previous section, a function f is an invariant if it satisfies:

$$V_k f = 0 \quad \text{for all } k = \{\text{rotaton, scalings, and shearing}\} \qquad (7.34)$$

Solving the above systems of differential equations gives us the following invariants:

$$f = F\left(\frac{B_1G_2 - G_1B_2}{R_1G_2 - G_1R_2}, \frac{B_1R_2 - R_1B_2}{R_1G_2 - G_1R_2}\right) \qquad (7.35)$$

All the invariants for the dichromatic reflection model in Eq. 7.20 for two pixels of an RGB image are functions of the two invariants. Fig. 7.7 shows how the above analysis can be done automatically in Maple$^{\text{TM}}$.

The result for the case of using two pixels of RGB images can be extended to multichannel images. The four vector fields are, in this case, given by

$$\begin{aligned}
V_{rot} &= \sum_{n=1}^{N} C_n^2\frac{\partial}{\partial C_n^1} - C_n^1\frac{\partial}{\partial C_n^2}\\
V_{isos} &= \sum_{n=1}^{N} C_n^1\frac{\partial}{\partial C_n^1} + C_n^2\frac{\partial}{\partial C_n^2}\\
V_{anis} &= \sum_{n=1}^{N} C_n^2\frac{\partial}{\partial C_n^2}\\
V_{shear} &= \sum_{n=1}^{N} C_n^1\frac{\partial}{\partial C_n^1} \qquad (7.36)
\end{aligned}$$

```
> roteq:=f(cos(x)*R[1]+sin(x)*R[2],-sin(x)*R[1]+cos(x)*R[2],
>         cos(x)*G[1]+sin(x)*G[2],-sin(x)*G[1]+cos(x)*G[2],
>         cos(x)*B[1]+sin(x)*B[2],-sin(x)*B[1]+cos(x)*B[2]);
```

```
> rotvf:=map(simplify,eval(subs(x=0,diff(roteq,x))));
```

$$\begin{aligned} \mathit{req} := \mathrm{f}(&\cos(x)\,R_1+\sin(x)\,R_2,\ -\sin(x)\,R_1+\cos(x)\,R_2,\\ &\cos(x)\,G_1+\sin(x)\,G_2, -\sin(x)\,G_1+\cos(x)\,G_2,\\ &\cos(x)\,B_1+\sin(x)\,B_2,\ -\sin(x)\,B_1+\cos(x)\,B_2)\end{aligned}$$

$$\begin{aligned} &\mathit{rvf} :=\\ &+\mathrm{D}_1(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,R_2-\mathrm{D}_2(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,R_1\\ &+\mathrm{D}_3(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,G_2-\mathrm{D}_4(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,G_1\\ &+\mathrm{D}_5(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,B_2-\mathrm{D}_6(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,B_1\end{aligned}$$

```
> iseq:=f(exp(x)*R[1],exp(x)*R[2],exp(x)*G[1],
>        exp(x)*G[2],exp(x)*B[1],exp(x)*B[2]);
```

```
> isvf:=map(simplify,eval(subs(x=0,diff(iseq,x))));
```

$$\mathit{iseq} := \mathrm{f}(e^x\,R_1,\,e^x\,R_2,\,e^x\,G_1,\,e^x\,G_2,\,e^x\,B_1,\,e^x\,B_2)$$

$$\begin{aligned} &\mathit{isvf} :=\\ &+\mathrm{D}_1(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,R_1+\mathrm{D}_2(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,R_2\\ &+\mathrm{D}_3(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,G_1+\mathrm{D}_4(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,G_2\\ &+\mathrm{D}_5(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,B_1+\mathrm{D}_6(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,B_2\end{aligned}$$

```
> aseq:=f(R[1],exp(x)*R[2],G[1],exp(x)*G[2],B[1],exp(x)*B[2]);
```

```
> asvf:=map(simplify,eval(subs(x=0,diff(aseq,x))));
```

$$\mathit{aseq} := \mathrm{f}(R_1,\,e^x\,R_2,\,G_1,\,e^x\,G_2,\,B_1,\,e^x\,B_2)$$

$$\begin{aligned} &\mathit{asvf} := \mathrm{D}_2(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,R_2\\ &+\mathrm{D}_4(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,G_2+\mathrm{D}_6(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,B_2\end{aligned}$$

```
> sheq:=f(R[1]+x*R[2],R[2],G[1]+x*G[2],G[2],B[1]+x*B[2],B[2]);
```

```
> shvf:=map(simplify,eval(subs(x=0,diff(sheq,x))));
```

$$\mathit{sheq} := \mathrm{f}(R_1+x\,R_2,\,R_2,\,G_1+x\,G_2,\,G_2,\,B_1+x\,B_2,\,B_2)$$

$$\begin{aligned} &\mathit{shvf} := \mathrm{D}_1(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,R_2\\ &+\mathrm{D}_3(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,G_2+\mathrm{D}_5(f)(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)\,B_2\end{aligned}$$

```
> pdsolve({rotvf,isvf,asvf,shvf},[f]);
```

$$\left\{\mathrm{f}(R_1,\,R_2,\,G_1,\,G_2,\,B_1,\,B_2)=\_\mathrm{F1}\left(\frac{-G_2\,B_1+B_2\,G_1}{-R_2\,G_1+R_1\,G_2},\,\frac{B_2\,R_1-R_2\,B_1}{-R_2\,G_1+R_1\,G_2}\right)\right\}$$

Figure 7.7: The Maple script to find the invariants for the dichromatic reflection model in the case of using two pixels of RGB images.

The four vector fields are functionally independent and their Lie products also do not create any new independent vector field. The number of independent invariants in this case is

$$\text{number of invariants} = 2N - 4 = 2(N-2) \tag{7.37}$$

Using the same framework as in the previous section, the following $2(N-2)$ invariants are obtained

$$f = F\left(\left\{\frac{C_n^1 C_j^2 - C_n^2 C_j^1}{C_2^1 C_1^2 - C_2^2 C_1^1}\right\},\ \text{with } n = 3 \ldots N, j = 1,2\right) \tag{7.38}$$

Each added channel will generate two new invariants.

For two pixels x_1 and x_2 of an RGB image, we have the two invariants, see Eq. 7.35 of the previous case. If we substitute the RGB values in Eq. 7.32 into the invariants, it can be shown that their values are independent of both the geometry terms m_S, m_D and the spatial terms.

$$\begin{aligned} f_1 &= \frac{B_1G_2 - G_1B_2}{R_1G_2 - G_1R_2} = \frac{S_BD_G - S_GD_B}{S_RD_G - S_GD_R} \\ f_2 &= \frac{B_1R_2 - R_1B_2}{R_1G_2 - G_1R_2} = \frac{S_BD_R - S_RD_B}{S_RD_G - S_GD_R} \end{aligned} \tag{7.39}$$

Using derivatives instead of color values

All the derivations also work if, instead of using two color pixel values as in Eq. 7.30, we use one color pixel value C_n^k at pixel x_k and its derivative in a given direction $\frac{d(C_n(x))}{dx}|_{x=x_k}$, or the derivatives at two different pixels, or even for only one pixel using its derivative in two different directions. Eq. 7.30 then has one of the following forms

$$\begin{bmatrix} C_n(x_k) \\ \frac{d(C_n(x))}{dx}|_{x=x_k} \end{bmatrix} = \begin{bmatrix} m_S(x_k) & m_D(x_k) \\ \frac{d(m_S(x))}{dx}|_{x=x_k} & \frac{d(m_D(x))}{dx}|_{x=x_k} \end{bmatrix} \cdot \begin{bmatrix} S_n \\ D_n \end{bmatrix} \tag{7.40}$$

and

$$\begin{bmatrix} \frac{d(C_n(x))}{dx}|_{x=x_1} \\ \frac{d(C_n(x))}{dx}|_{x=x_2} \end{bmatrix} = \begin{bmatrix} \frac{d(m_S(x))}{dx}|_{x=x_1} & \frac{d(m_D(x))}{dx}|_{x=x_1} \\ \frac{d(m_S(x))}{dx}|_{x=x_2} & \frac{d(m_D(x))}{dx}|_{x=x_2} \end{bmatrix} \cdot \begin{bmatrix} S_n \\ D_n \end{bmatrix} \tag{7.41}$$

The extended dichromatic reflection model

We now consider the extended dichromatic reflection model. For a set of neighboring pixels, it is reasonable to assume that there is no illumination change locally:

$$E(x_1, \lambda) = E(x_2, \lambda) = E(x, \lambda)$$

where x_1 and x_2 are the two neighboring pixels. This leads to $S_n(x) = S_n(x_1) = S_n(x_2)$ and $D_n(x) = D_n(x_1) = D_n(x_2)$ and we have the following model

$$C_n(x_p) = m_S(x_p)S_n(x) + m_D(x_p)D_n(x) + L_n^A \tag{7.42}$$

The difference is only that there is another term $L_n^A = \int f_n(\lambda)L^a(\lambda)d\lambda$ which is, however, independent of both geometric and illumination factors. Considering the color values at two neighboring pixels x_1 and x_2, taking the difference of the color values we get

$$C_n(x_1) - C_n(x_2) = (m_S(x_1) - m_S(x_2))S_n(x) + (m_D(x_1) - m_D(x_2))D_n(x) \tag{7.43}$$

It has a form similar to the standard model in Eq. 7.23. Thus all the above derivations will still be valid if we take the differences between color values instead of its value. This, however, requires another pixel in the process. For example the first invariant in Eq. 7.35 becomes

$$\frac{(B_1 - B_2)(G_2 - G_3) - (G_1 - G_2)(B_2 - B_3)}{(R_1 - R_2)(G_2 - G_3) - (G_1 - G_2)(R_2 - R_3)} \tag{7.44}$$

We used the invariant derived above in a segmentation application in which we want to segment an object having difficult geometry changes from the background. We computed the color invariant feature

$$I = \frac{B_1G_2 - B_2G_1}{R_1G_2 - R_2G_1}$$

for the image of a paprika as shown in Fig. 7.8. The original image is on the left side, on the right is the computed invariant feature image, and the bottom image is the result of a simple thresholding of the feature image. The paprika can be distinguished from the background, especially in the shadow region where even the human eye has difficulty in recognizing the real border. Here the invariant feature value of a pixel x is estimated as the median value of the feature computed between the pixel x and its 8 connected neighbors. At the border between the background and the paprika, the assumption that the two neighboring pixels should come from the same material does not hold. Therefore we see a noisy border around the paprika in the feature image. Also in the feature image there are some errors in the highlight regions, where the color values are cut because of quantization error. This cutting error in highlight regions was not taken into account when we derived the model.

7.4 Methods using the Kubelka-Munk Model

The dichromatic reflection model as described in the previous section is a general model and it does not consider physical processes once the light enters

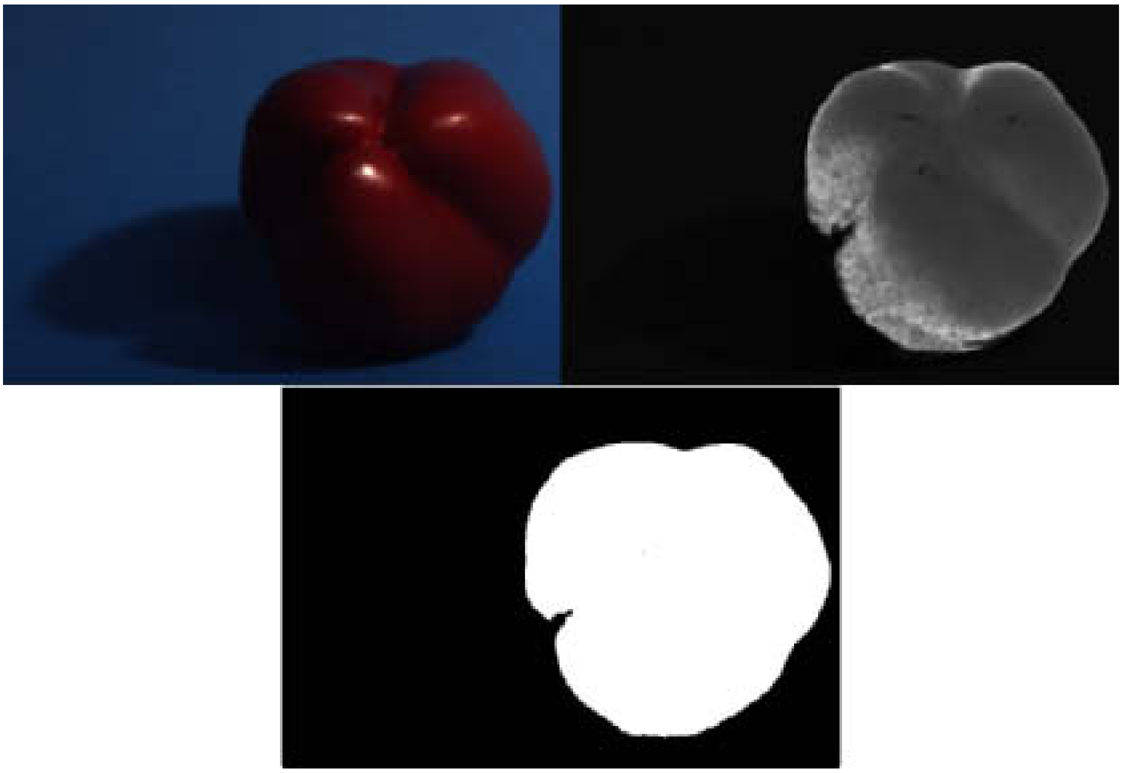


Figure 7.8: An RGB image (top left), the color invariant feature using $I = (B_1G_2 - B_2G_1)/(R_1G_2 - R_2G_1)$ (top right), and the segmented image (bottom) resulting from a simple threshold of the top right image. The color version of the original image is in Fig. 5.4 on page 83.

into the medium. These processes include absorption, scattering, and emission. Radiative Transfer Theory (Chandrasekhar, 1950) can be used to describe the propagation of the light inside the medium. However solving the integro differential equations which describe light propagations in a medium is very difficult. It has been shown that there is no analytic solution except for a few simple cases. Many methods are proposed to solve the problem numerically. For example one can divide the direction of incoming light into sub spaces (called channels) and have much simpler equations of light propagating in such small channels as in the Discrete-Ordinate-Method Radiative Transfer or the Multiflux Radiative Transfer Method. The Kubelka-Munk model is a special case assuming that the light propagation inside the medium is uniformly diffused and the properties of the medium such as scattering and absorption coefficient are isotropic. Under such assumptions, only two fluxes of light propagation inside the medium are enough to approximately describe the whole process.

7.4.1 Kubelka-Munk Model

As mentioned earlier, the Kubenka-Munk model deals only with two fluxes as illustrated in Fig. 7.9, one proceeding downward and the other upward. Consider the downward proceeding flux i during its propagation through an elementary layer with thickness dx at x. As seen in Fig. 7.9, the downward flux will be decreased by an amount of $Kidx$ because of absorption and another amount of $Sidx$ because of scattering where K and S are the fraction of the downward flux lost by absorption and scattering, respectively, in the elementary layer. K and S are known as the absorption and scattering coefficients of the material.

Similar for the upward flux j, it is reduced by an amount of $Kjdx$ because of absorption and $Sjdx$ because of scattering. The total change, dj, of the upward flux thus consists of two parts: the loss because of absorption and scattering of the upward flux and the amount added back to the upward flux because of scattering of the downward flux:

$$-dj = -(S+K)jdx + Sidx \tag{7.45}$$

The total change, di, of the downward flux is

$$di = -(S+K)idx + Sjdx \tag{7.46}$$

If the medium has optical contact with a backing of reflectance R_g, we have the following boundary condition at $x = 0$:

$$j_0 = R_g i_0 \tag{7.47}$$

If the external and internal surface reflectance at the interface of the medium is denoted as r_0 and r_1, respectively (see Fig. 7.10), and I_0 denotes the incoming light to the interface, then the following boundary conditions can be obtained at the interface, $x = D$.

$$i_D = I_0(1-r_0) + j_D r_1 \tag{7.48}$$

$$I_0 R = I_0 r_0 + j_D(1-r_1) \tag{7.49}$$

Solving the differential equations Eq. 7.45 and Eq. 7.46 with the boundary conditions Eq. 7.47, Eq. 7.48, and Eq. 7.49, we obtain the reflectance of the medium

$$R = r_0 + \frac{(1-r_0)(1-r_1)\big[(1-R_gR_\infty)R_\infty + (R_g - R_\infty)\exp(-AD)\big]}{(1-R_gR_\infty)(1-r_1R_\infty) - (R_\infty - r_1)(R_\infty - R_g)\exp(-AD)} \tag{7.50}$$

where

$$R_\infty = 1 + \frac{K}{S} - \sqrt{\frac{K^2}{S^2} + 2\frac{K}{S}} \tag{7.51}$$

$$A = \frac{2S(1-R_\infty^2)}{R_\infty} \tag{7.52}$$

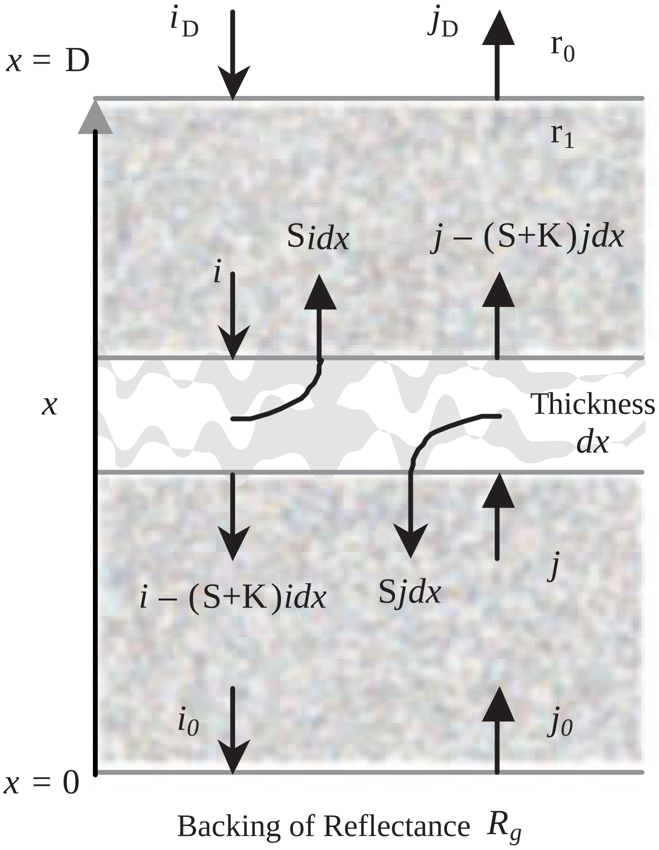


Figure 7.9: Basic of the Kubelka-Munk model.

A is a positive constant and if the medium is thick enough, i.e. $D \to \infty$ then

$$\tilde{R} = r_0 + \frac{(1 - r_0)(1 - r_1)R_\infty}{(1 - r_1 R_\infty)} \tag{7.53}$$

clearly, $\tilde{R}$ is equal to R_∞ when the interface reflections are zeros, $r_0 = r_1 = 0$. R_∞ is the reflectance of the medium layer when the surface reflection is omitted (Nobbs, 1985).

The external and internal surface reflectance at the interface of the medium r_0 and r_1 describe how much of the incident light is reflected at the surface. They depend on many factors: the incident angle of the light, the geometric properties of the surface, the reflective indices of the media, the polarization state of the light beam, and also the wavelength (Judd and Wyszecki, 1975). However, its dependency on the wavelength is very small and can be neglected.

The Kubelka-Munk coefficients K and S are the absorption and scattering coefficients of the medium along the direction in which the model is developed;

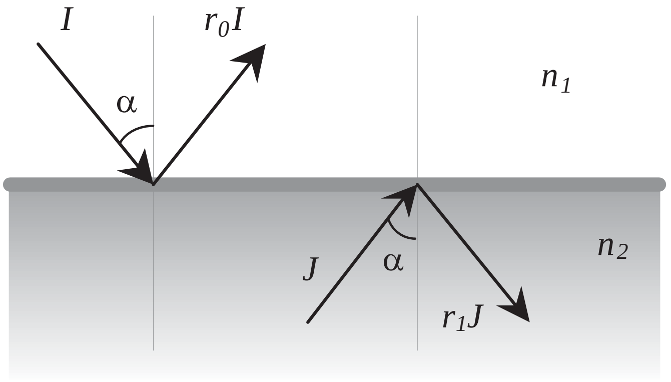


Figure 7.10: The internal reflection r_1 and the external reflection r_0.

we call this the normal direction. When a light beam travels inside the medium in a direction different from the normal direction (which is used by Kubelka-Munk model), it will be absorbed and scattered more in each elementary layer dx since it has to travel a longer distance. Let α denote the angle between the direction of light propagation and the normal direction. Instead of travelling dx the light has to pass a path of length $dx/cos(\alpha)$. Therefore K and S in this direction will be $1/cos(\alpha)$ times larger than in the normal direction and they depend on the angle of the light beam to the normal direction.

Their ratio K/S, however, does not depend on the angle α of the light beam to the normal direction, but only on the absorption and the scattering coefficients per unit path length of the medium. Thus R_∞ as in Eq. 7.51 depends only on the material, but not on the direction of the light beam.

Summarizing, the Kubelka-Munk model shows that the reflectance of the medium can be estimated as in Eq. 7.53 in which R_∞ is the reflectance of the medium layer when the surface reflection is omitted and r_0 and r_1 are the external and internal surface reflectance at the interface of the medium. R_∞ is independent of geometric properties while r_0 and r_1 are not.

7.4.2 Approximation Models for Color Invariants

Geusebroek and his colleagues (Geusebroek et al., 2001; Geusebroek et al., 2002) used the Kubelka-Munk model and proposed a number of color invariants. All their derivations are based on the formula

$$R = \rho + (1-\rho)^2 R_\infty \tag{7.54}$$

which can be derived directly from Eq. 7.53 using the assumptions:

$$\begin{aligned} r_1 &\approx r_0 && (7.55)\\ r_1 R_\infty &\approx 0 && (7.56) \end{aligned}$$

Eq. 7.55 holds only for small incident angle α and small ratio n_2/n_1 of the reflection index between the two media. As we can see in Fig. 7.11 the difference $r_1 - r_0$ is rather big in most cases violating Eq. 7.55. The assumption in Eq. 7.56 is also unrealistic since it holds only for materials which have very high absorption, and low scattering so that R_∞ is small.

Eq. 7.54, however, is still difficult to work with. Aiming to simplify the form of Eq. 7.51 (mainly reducing from two terms to one term), Geusebroek et al. use several other assumptions and consider separately several different cases such as:

- Invariants for equal energy but uneven illumination
- Invariants for equal energy but uneven illumination and matte, dull surfaces
- Invariants for equal energy and uniform illumination and matte, dull surfaces, and planar objects
- Invariants for colored but uneven illumination
- Invariants for a uniform object

It can be shown that most of the above assumptions could be relaxed. Look at Eq. 7.53. If we assume that

$$1 - r_1 R_\infty \approx 1 - r_1 \tag{7.57}$$

or in case this is unrealistic, we could have some compensation factor which could be a constant or even a function of R_∞

$$1 - r_1 R_\infty \approx (1 - r_1)\bar{g}(R_\infty) \tag{7.58}$$

Then Eq. 7.53 become

$$\tilde{R} = r_0 + (1 - r_0)g(R_\infty) \tag{7.59}$$

The color value at pixel x under illumination $E(x, \lambda)$ measured by the camera

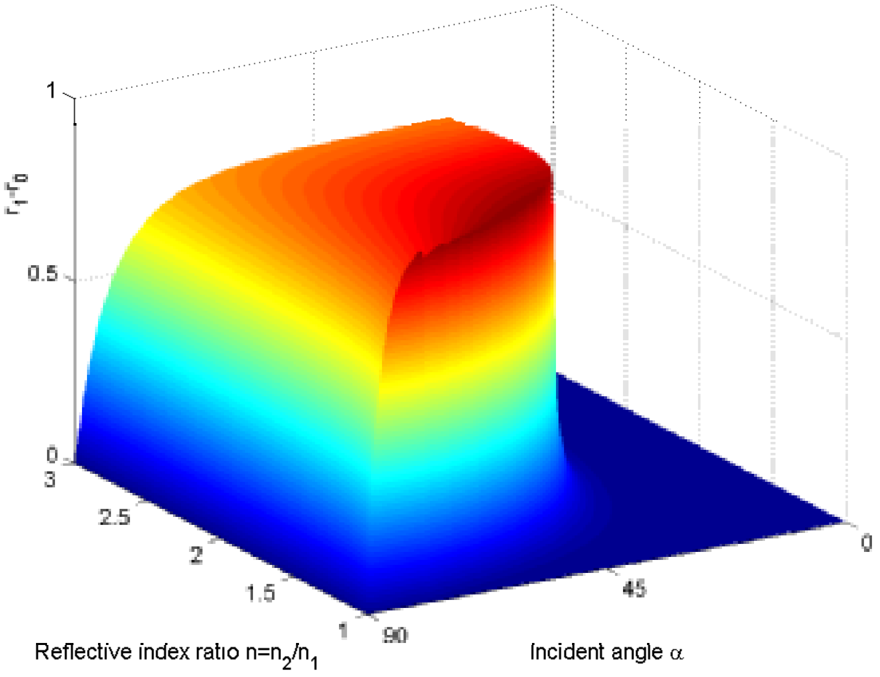


Figure 7.11: The theoretical differences between the internal reflection r_1 and the external reflection r_0 against the angle of the incident light α and the ratio of the reflection index between the two media $n = n_2/n_1$ according to Fresnel's equations (Chandrasekhar, 1950)

.

having sensitivity function $f_n(\lambda)$ can be computed as

$$\begin{aligned} C_n(x) &= \int f_n(\lambda)E(x,\lambda)\big[r_0(x) + (1 - r_0(x))g(R_\infty)\big]d\lambda \\ &= r_0(x)\int f_n(\lambda)E(x,\lambda)d\lambda + (1 - r_0(x))\int f_n(\lambda)E(x,\lambda)g(R_\infty)d\lambda \\ &= r_0(x)S_n(x) + (1 - r_0(x))D_n(x) \\ &= D_n(x) + r_0(x)(S_n(x) - D_n(x)) \end{aligned} \tag{7.60}$$

where $r_0(x)$ depends on geometric factors but $S_n(x)$ and $D_n(x)$ do not. This approximation model will be investigated in the next section.

Although the form of Eq. 7.60 looks very similar to the form of Eq. 7.23, it is not correct to say that Eq. 7.60 is a special case of Eq. 7.23 as in (Geusebroek et al., 2001; Geusebroek et al., 2002). The reason is that in the dichromatic reflection model (Shafer, 1985), Shafer assumed that the geometric terms $m_S(x)$ and $m_D(x)$ are independent of the material properties while this assumption does not hold in the Kubelka-Munk model since both r_0 and r_1 are dependent on the material properties.

7.4.3 Geometric Invariants Using the Kubelka-Munk Model

A geometric invariant feature is a function of the color values $C_n(x)$ and it should be independent of $r_0(x)$. From Eq. 7.60 we find that the n^{th} channel color value of pixel x is given by:

$$\begin{aligned} C_n(x) &= r_0(x)S_n(x) + (1 - r_0(x))D_n(x) \\ &= D_n(x) + r_0(x)(S_n(x) - D_n(x)) \\ &= D_n(x) + r_0(x)O_n(x) \end{aligned} \tag{7.61}$$

We consider P neighboring pixels $x_1, x_2 \ldots x_P$, each pixel with N channels. Totally there are $P \times N$ values $C_n(x_p)$ from $P \times N$ equations. Since all the pixels are neighbors, it is reasonable to assume that there is no illumination change locally around these pixels. Thus the $D_n(x)$ and $O_n(x)$ terms for each channel are identical.

$$\begin{aligned} C_n^p &= C_n(x_p) = D_n(x_p) + r_0(x_p) \cdot O_n(x_p) \\ &= D_n + r_0^p \cdot O_n \qquad \text{with } n = 1 \ldots N, p = 1 \ldots P \end{aligned} \tag{7.62}$$

We use the same strategy as in the previous section to solve the invariant problem for the Kubelka-Munk model.

For one pixel the situation is similar to the previous section. Since there is only one pixel to consider, each channel has only one measurement $C_n(x)$, but two unknowns S_n and D_n. All invariants, if they exist, will depend on at least either S_n or D_n. It can be seen easily from the following example of using two channels. We have two equations to describe color values C_1 and C_2 of pixel x:

$$\begin{aligned} C_1 &= D_1 + r_0(x)O_1 \\ C_2 &= D_2 + r_0(x)O_2 \end{aligned} \tag{7.63}$$

There is only one invariant

$$f = \frac{C_1 - D_1}{C_2 - D_2}$$

but it depends on the unknown D_1, D_2. Therefore, using information from neighboring pixels is necessary.

We consider next the case of using 2 pixels, say x_1 and x_2. Each pixel has N channels. Totally there are $2N$ values $C_n(x_p)$ collected in a system of $2N$ equations as in Eq. 7.62. We change to a shorter notation and compute the differences between the color values between two pixels in the same channel:

$$\begin{aligned} C_n^1 &= C_n(x_1) &&= D_n(x) + r_0(x_1)O_n(x) &&= D_n + \rho_1 O_n \\ C_n^{12} &= C_n(x_1) - C_n(x_2) &&= (r_0(x_1) - r_0(x_2))O_n(x) &&= \rho_2 O_n \end{aligned} \tag{7.64}$$

or in matrix form

$$\begin{bmatrix} C_n^1 \\ C_n^{12} \end{bmatrix} = \begin{bmatrix} 1 & \rho_1 \\ 0 & \rho_2 \end{bmatrix} \begin{bmatrix} D_n \\ O_n \end{bmatrix} = M \begin{bmatrix} D_n \\ O_n \end{bmatrix} \tag{7.65}$$

The color values $(C_n^1, C_n^2)^T$ are obtained by multiplying the matrix M (containing geometry terms) with the vector $(D_n, O_n)^T$ which is independent of geometry changes. An invariant function f in this case is a mapping from the $2N$-dimensional space of real numbers to a real number:

$$f : \mathbb{R}^{2N} \to \mathbb{R}$$

This function should be independent under the transformation M. The transformation matrix M is can be seen as a combination of an anisotropic scaling with scale factor ρ_2 with a shearing action ρ_1. To be invariant under the transformation M, a function f should be invariant along the vector fields of the two anisotropic scaling and shearing one-parameter subgroups described above.

The number of independent invariants, as discussed in section 7.2.3, is obtained as the dimension of the space on which the invariants operate minus the dimension of the Lie algebra of the vector fields. Since in this case, the two vector fields are functionally independent, the Lie algebra has at least 2 dimensions leading to the maximum of possible independent invariants

$$\text{maximum number of invariants} = 2N - 2 = 2(N-1) \tag{7.66}$$

In order to have an invariant, this number should be positive: $2(N-1) > 0$, or the number of channels N should be at least 2.

With 2 channels such as Red and Green channels in an RGB image, there will be at most 2 independent invariants. For two pixels x_1 and x_2 we change the notation to

$$\begin{aligned} \begin{bmatrix} R_1 \\ R_{12} \end{bmatrix} &= \begin{bmatrix} 1 & \rho_1 \\ 0 & \rho_2 \end{bmatrix} \begin{bmatrix} D_R \\ O_R \end{bmatrix} \\ \begin{bmatrix} G_1 \\ G_{12} \end{bmatrix} &= \begin{bmatrix} 1 & \rho_1 \\ 0 & \rho_2 \end{bmatrix} \begin{bmatrix} D_G \\ O_G \end{bmatrix} \end{aligned} \tag{7.67}$$

The two vector fields $V_{aniscale}, V_{shear}$ along the directions of the anisotropic scaling, and shearing one-parameter subgroups are given by

$$V_{aniscale} = R_{12}\frac{\partial}{\partial R_{12}} + G_{12}\frac{\partial}{\partial G_{12}} \tag{7.68}$$

$$V_{shear} = R_{12}\frac{\partial}{\partial R_1} + G_{12}\frac{\partial}{\partial G_1} \tag{7.69}$$

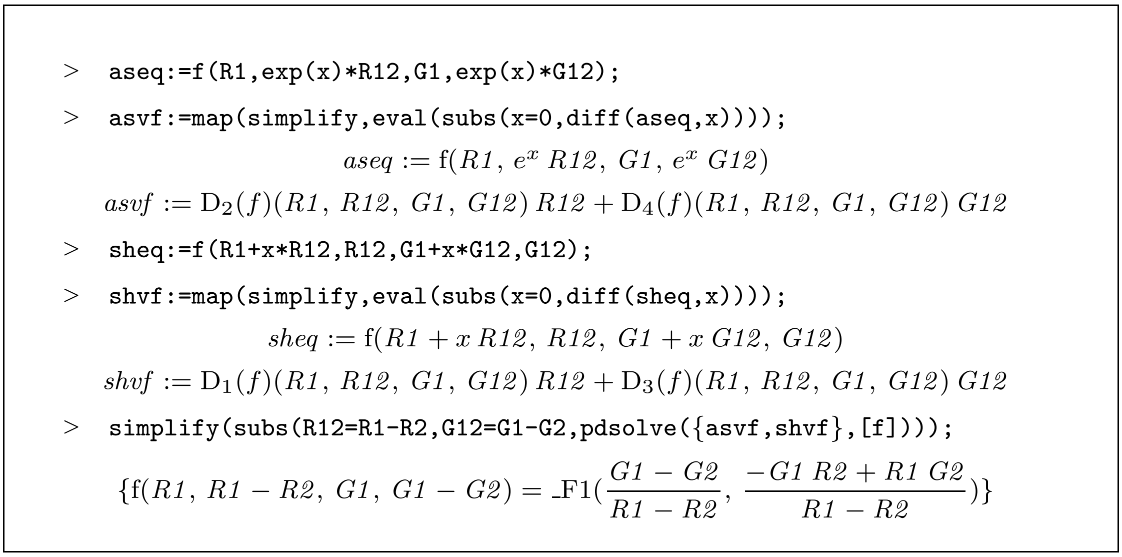
Following the framework in the previous section, a function f is an invariant if it satisfies:

$$V_k f = 0 \qquad \text{for all } k = \{\text{anisotropic scaling and shearing}\} \tag{7.70}$$

Solving the above system of differential equations gives us the following the invariants.

$$\mathrm{f} = \mathrm{F}\left(\frac{G1 - G2}{R1 - R2}, \frac{G1R2 - G2R1}{R1 - R2}\right) \tag{7.71}$$

All the invariants for the Kubelka-Munk model in Eq. 7.59 for two pixels of an RGB image are a function of the two invariants described above. Fig. 7.12 shows how the above analysis can be done automatically in Maple$^{\mathrm{TM}}$.



```
> aseq:=f(R1,exp(x)*R12,G1,exp(x)*G12);
> asvf:=map(simplify,eval(subs(x=0,diff(aseq,x))));
```

$$aseq := \mathrm{f}(R1,\, e^x\, R12,\, G1,\, e^x\, G12)$$

$$asvf := \mathrm{D}_2(f)(R1,\, R12,\, G1,\, G12)\, R12 + \mathrm{D}_4(f)(R1,\, R12,\, G1,\, G12)\, G12$$

```
> sheq:=f(R1+x*R12,R12,G1+x*G12,G12);
> shvf:=map(simplify,eval(subs(x=0,diff(sheq,x))));
```

$$sheq := \mathrm{f}(R1 + x\, R12,\, R12,\, G1 + x\, G12,\, G12)$$

$$shvf := \mathrm{D}_1(f)(R1,\, R12,\, G1,\, G12)\, R12 + \mathrm{D}_3(f)(R1,\, R12,\, G1,\, G12)\, G12$$

```
> simplify(subs(R12=R1-R2,G12=G1-G2,pdsolve({asvf,shvf},[f])));
```

$$\{\mathrm{f}(R1,\, R1 - R2,\, G1,\, G1 - G2) = \_\mathrm{F1}(\frac{G1 - G2}{R1 - R2},\, \frac{-G1\, R2 + R1\, G2}{R1 - R2})\}$$

Figure 7.12: The Maple script to find the invariants for the Kubelka Munk model in the case of using two channels of two pixels.

The above result can be extended to the case of having more than two channels, for example as in RGB or multichannel images. The two vector fields are, in this case, given by

$$\begin{aligned} V_{aniscale} &= \sum_{n=1}^{N} C_n^2 \frac{\partial}{\partial C_n^2} \\ V_{shear} &= \sum_{n=1}^{N} C_n^1 \frac{\partial}{\partial C_n^1} \end{aligned} \tag{7.72}$$

The two vector fields are functionally independent and their Lie product does not create any new independent vector field. The number of independent invariants in this case is

$$\text{number of invariants} = 2N - 2 = 2(N-1) \tag{7.73}$$

Using the same framework as in the previous section, the following $2(N-1)$ invariants are obtained

$$f = F\left(\left\{\frac{C_n^1 - C_n^2}{C_1^1 - C_1^2}, \frac{C_n^1 C_1^2 - C_n^2 C_1^1}{C_1^1 - C_1^2}\right\} \quad \text{with } n = 2 \dots N\right) \tag{7.74}$$

It is very similar to the dichromatic reflection model described in the previous section that the values of the invariants derived for any arbitrary pixels x_1 and x_2 of the same material are independent of both the geometry and the spatial terms.

$$f_1 = \frac{C_n^1 - C_n^2}{C_1^1 - C_1^2} \qquad = \frac{O_n}{O_1} \tag{7.75}$$

$$f_2 = \frac{C_n^1 C_1^2 - C_n^2 C_1^1}{C_1^1 - C_1^2} = \frac{O_n D_1 - D_n O_1}{O_1} \tag{7.76}$$

In this case, each added channel will generate two new invariants.

7.5 Illumination Invariants

It is interesting to observe that most of the invariants proposed in the above framework are also invariant to illuminations under certain conditions.

It has been shown that many illuminants can be well described as linear combinations of a low-dimension basis set (Hernández-Andrés et al., 2001; Judd et al., 1964).

$$E(x, \lambda) = \sum_{k=1}^{K} e_k(x) E_k(\lambda) \tag{7.77}$$

where $E_k(\lambda)$ is a basis vector and $e(x)$ is a K-dimensional vector of weights parameterizing the illumination at x.

For a normal scene where there is a dominant light source (such as outdoor illuminations) or when the spectral properties of the illuminations are mainly caused by intensity changes, the illumination $E(x, \lambda)$ can be described by only one basis function ($K = 1$) as

$$E(x, \lambda) = e_1(x) E_1(\lambda) \tag{7.78}$$

This assumption is generally unrealistic, but for color image segmentation applications where we want to segment an image into regions, it is quite reasonable

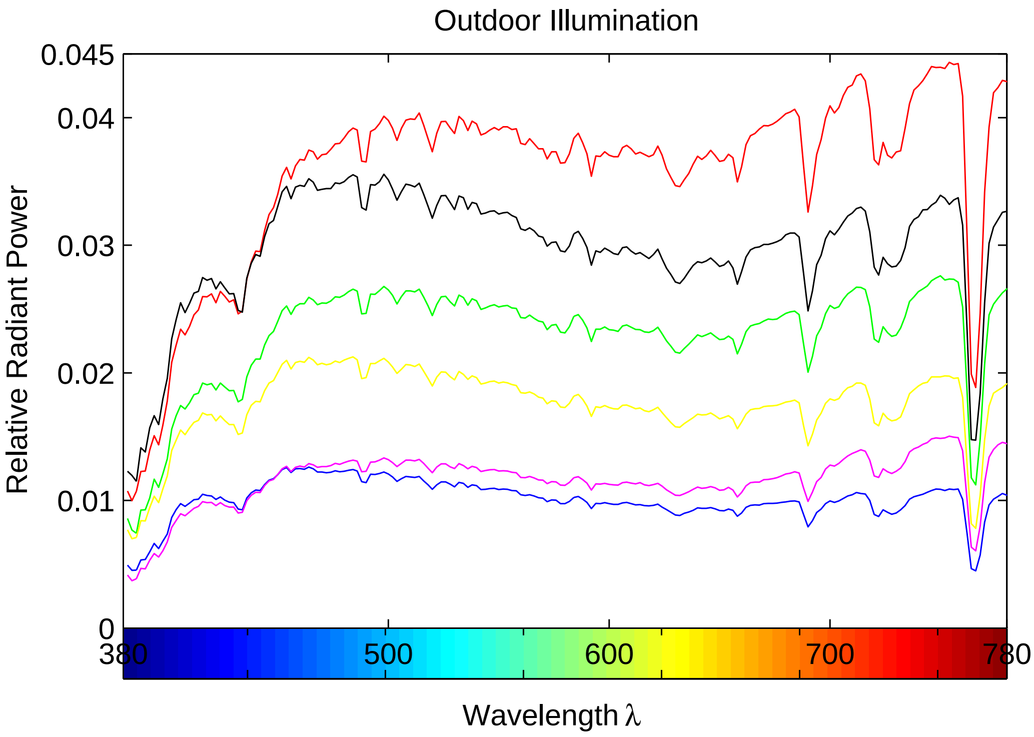


Figure 7.13: Outdoor illuminations measured at different places on campus during a short period of time.

to assume that inside that small region, illumination changes can be described by one-parameter. Under such an assumption, all the invariants which are based on a ratio (such as angle, ratio of length, ratio of area, etc.) are also invariant to illumination since $e_1(x)$ will cancel in the ratio-based invariants.

In order to examine the assumption which has been made in Eq. 7.78, experiments were carried out with a spectrometer SpectraScan PR 705. Fig. 7.13 shows some of the spectra of outdoor illuminations we have measured at different places (direct sunlight, shadow, close to different objects) on our campus during a short period of time. The PCA of this data set shows that 99.84 % of the energy of the spectral data is in the first principal component. In another set of data, in which we measure illuminations at different places in an office room illuminated by six lamps, two PC monitors, and daylight from two windows, 98.68 % of energy is in the first principal component. These examples illustrate that Eq. 7.78 is a reasonable assumption for many normal illuminations.

This is the simplest example where the illumination spectra can be described by one-parameter, in this case the intensity of the illumination source. In another investigation we showed that also the chromaticity properties of illumination sources can (to a large extend) be described by a single parameter (Lenz et al., 2003a). Together with the intensity changes this gives a transformation group with two parameters and invariants can be derived using the framework described above.

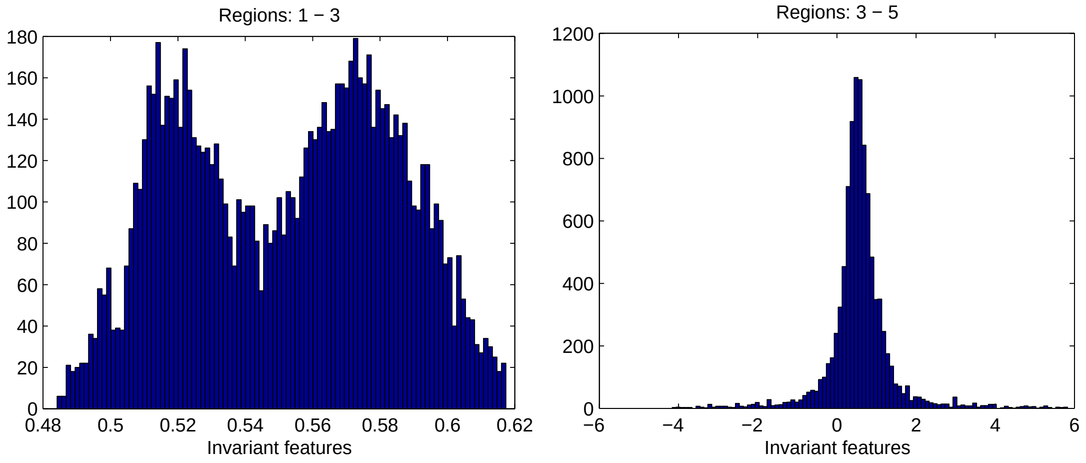


Figure 7.14: Analysis the invariant feature distribution for each pair of regions. Regions are numbered as in Fig. 7.15.

7.6 Robust Region-Merging Algorithm

In the previous sections, we saw that physics-based color image understanding using physical models require quite unrealistic assumptions. This explains why features computed using physical models are noisy. Also most of the invariants have the form of a ratio of two small numbers, for example

$$\frac{R_1 - R_2}{G_1 - G_2}, \frac{R_1 G_2 - R_2 G_1}{G_1 - G_2}, \text{ or } \frac{R_1 B_2 - R_2 B_1}{G_1 B_2 - G_2 B_1}$$

The invariant feature is therefore sensitive to noise, especially when each channel has only 8 bits, or 256 different levels. Robust methods are needed to deal with this situation. In this section, we propose a robust region-merging algorithm for color image segmentation applications using physics-based models described in the previous sections.

The basic idea of the proposed algorithm is that instead of a point-wise clustering decision, we will first over-segment the input color image into homogenous regions, then try to merge them based on the similarity between the feature distributions of regions. The algorithm works as follows:

1. Over-segment the input image into homogenous color regions $R_1, R_2, \dots, R_N$

2. Compute invariant features for a number of pixels in each region using one of the invariants described above.

3. Estimate the distributions of the invariant features $f_1, f_2, \dots, f_N$ for each region based on the above computed samples.

4. For all pairs of regions R_i and R_j, compute the distance between their feature distributions $d_{ij} = \mathrm{dist}(f_i, f_j)$

5. Merge the two regions which have most similar feature distributions: Sorting all the computed distances and merge the two regions corresponding to the smallest distance d. This gives a new region R_{ij}

6. Update the new region R_{ij} instead of the two regions R_i and R_j

7. If the number of remaining regions is still greater than a predefined number, continue with step 4. Otherwise stop.

An example of the algorithm is illustrated in Fig. 7.15, where we first use the Mean Shift Algorithm (Comaniciu and Meer, 1999a) to over-segment the paprika image into seven homogenous color regions. The original image is shown in the left-lower part of Fig. 7.15. For each region, a fixed number of pairs of pixels are randomly selected. The invariant feature

$$I = \frac{B_1 G_2 - B_2 G_1}{R_1 G_2 - R_2 G_1}$$

is then computed for all the pairs. Based on these computed invariant feature values, we estimate the feature distributions of the seven regions. Fig. 7.14 shows two examples of joint distributions of regions $(1, 3)$ and $(3, 5)$. Clearly regions 3 and 5 come from the same material, therefore their joint distribution has only one peak. The joint distribution of regions 1 and 3 has two peak because the two regions belong to different material. Which regions should be merged first is decided on the basis of the similarity between feature distribution of the regions. Distances between these distributions are compared using L_2 metric. The result of the merging process is shown in the right part of Fig. 7.15.

Another more complicated example is done with the color image in Fig. 7.16. The left image is the original image and the right one is the result after over-segmenting the image. Fig. 7.17 presents the result of our proposed algorithm after 160 steps. Most of the regions coming from the same material have been merged. However, the shadow of the cup could not be merged.

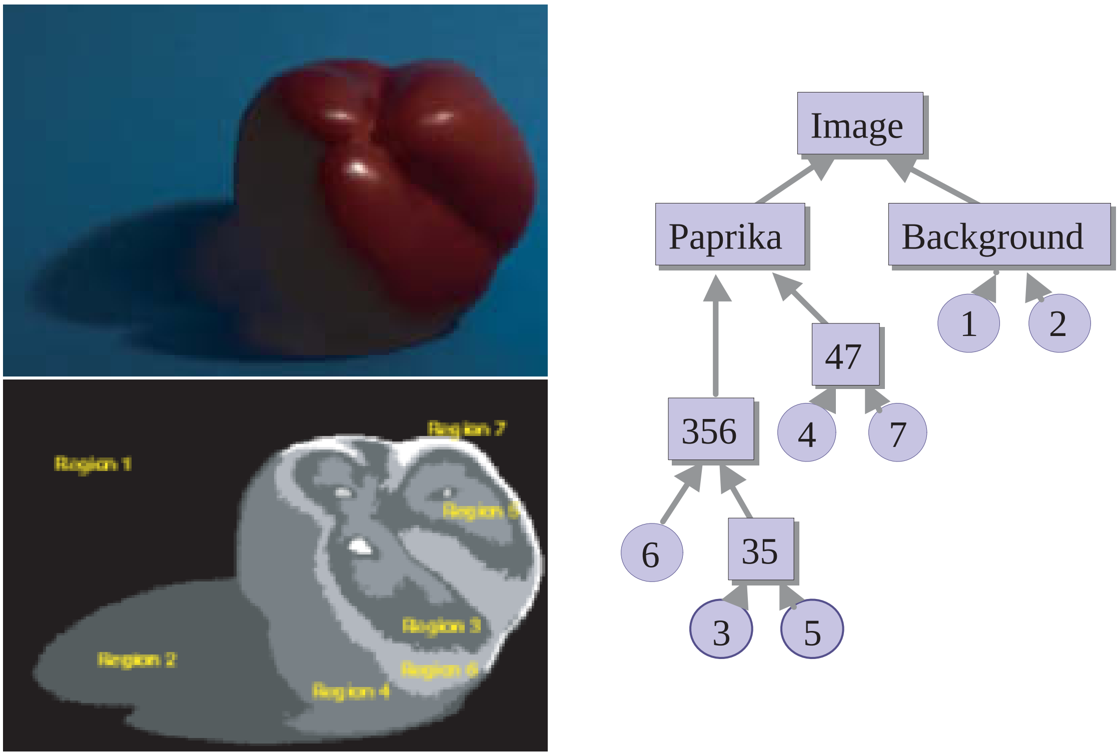


Figure 7.15: The left side shows the original paprika image and its over-segmented image. The right side shows the steps of the robust region-merging algorithm applied to the left images. A color version of the original image is presented in Fig.5.4 on page 83.

7.7 Summary

In this chapter we applied the invariant theory to derive geometry color invariants using different physical reflection models. We concentrated on the problem of how to systematically construct all the independent invariants for a given model. We showed that using the framework all the independent invariants of a given physical process can be constructed. Most of the work can be done by few lines of coding with the help of symbolic mathematical software packages like Maple^{TM}. The dichromatic reflection model, its extended version, and the Kubelka-Munk model were then investigated within the framework. Experiments were done and illustrated that the invariants provide useful information to discriminate between shadow and object points in the scene. For more realistic applications further analysis of the underlying physical processes and an error analysis of the models are needed.

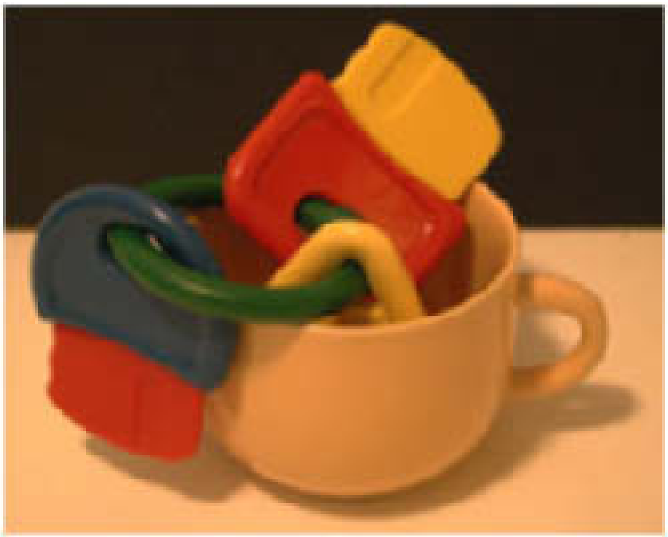


Figure 7.16: Original cup image. A color version of this image is presented in Fig.5.4 on page 83.

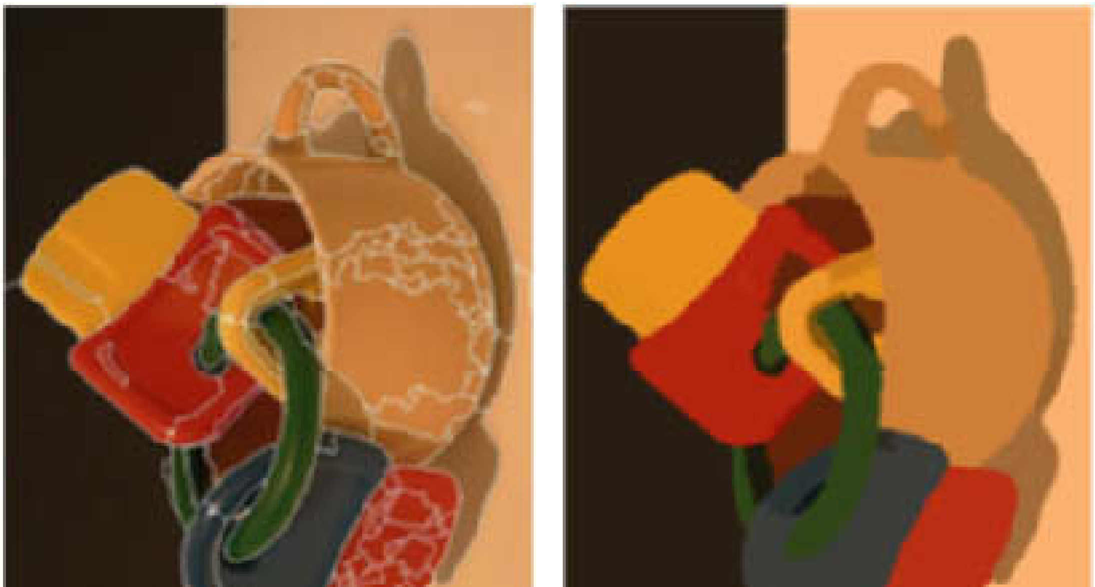


Figure 7.17: The left image shows the result of over-segmenting the image in Fig. 7.16. The right image shows the result of the robust region-merging on the left image after 160 steps. A color version of the two images is presented in Fig.1.3 on page 7.

Chapter 8

MOMENT-BASED NORMALIZATION OF COLOR IMAGES

Many conventional computational color constancy methods assume that the effect of an illumination change can be described by a matrix multiplication with a diagonal matrix. In this chapter we introduce a color normalization algorithm which computes the unique color transformation matrix which normalizes a given set of moments computed from the color distribution of an image. This normalization procedure is a generalization of the independent channel color constancy methods since general matrix transformations are considered. We compare the performance of this normalization method with conventional color constancy methods in color correction and illuminant color object recognition applications. The experiments show that diagonal transformation matrices provide a better illumination compensation. This shows that the color moments also contain significant information about the color distributions of the objects in the image which is independent of the illumination characteristics.

In another set of experiments we use the unique transformation matrix as a descriptor of the set of moments which describe the global color distribution in the image. Combining the matrices computed from two such images describes the color differences between them. We then use this as a tool for color-dependent search in image databases. This matrix-based color search is computationally less demanding than histogram-based color search tools.

This work was done before we started our investigation on general color-based methods. The method is therefore only compared with the histogram intersection method.

8.1 Introduction

It is often assumed that the effect of a change in illumination on an RGB image can be described by a linear transformation, i.e. a 3×3 matrix, see (Finlayson et al., 1994; Kondepudy and Healey, 1994; Drew et al., 1998) for some applications in image processing and computer vision and section 5.12 in (Wyszecki and Stiles, 1982) for a related discussion of color adaptation. Studies of the human visual system suggest that color adaptation is obtained by independent adjustments of the sensitivities of the sensors. This corresponds to a diagonal transformation matrix and is known as von-Kries adaptation. In this paper we will assume that the general model involving a full $3 \times\ 3$ matrix is approximately correct and we will describe a method to compute a unique color transformation matrix which normalizes the probability distribution of the color image. Two examples in which such a normalization is useful are image database search and color mapping. In image database applications it is useful to separate the illumination from the scene properties to allow search for objects independent of the properties of the imaging process. In color mapping the transformation matrix is used to characterize the overall color distribution of an image. Combinations of these matrices can then be used to simulate different color effects.

If we denote by $\mathbf{x}_0$ the color vector (usually containing RGB-values) produced by an object point under illuminant L_0 and by $\mathbf{x}_1$ the color vector produced by the same object point under illuminant L_1 then the linear transformation model assumes that there is a 3×3 matrix $\mathbf{T}$ such that

$$\mathbf{x}_1 = \mathbf{T}\mathbf{x}_0. \tag{8.1}$$

Here we will not assume that the relation in Eq. 8.1 is known for each image point separately. Instead we will only require that it holds in the following statistical sense:

Denote by $p_i(\mathbf{x})$ the probability distribution of a scene under illumination L_i. Then we assume that the color distributions are connected by a linear transformation as follows:

$$p_1(\mathbf{x}) = p_0\left(\mathbf{T}\mathbf{x}\right) \tag{8.2}$$

In this setting the equation incorporates three components:

1. The illumination, represented by $\mathbf{T}$
2. the sensors producing the $\mathbf{x}$-vector and
3. the object, or rather the statistical properties of the scene.

Whether the relation in Eq. 8.1 or Eq. 8.2 is valid depends of course on all of the factors involved.

In the following we assume that we have a pair of images. The goal is to compute for this pair of images the transformation matrix $\mathbf{T}$ such that Eq. 8.2 holds. Here we will not solve the problem directly but we will use a two-step procedure instead. In the first step we will compute for every image I a unique matrix $\mathbf{T}_I$ such that the probability distribution transformed according to Eq. 8.2 has a unique set of moments. The required transformation matrix which maps image I_0 to image I_1 is then given by

$$\mathbf{T} = \mathbf{T}_1^{-1}\mathbf{T}_0 \tag{8.3}$$

Note that the role of $\mathbf{T}$ as a description of the illumination effect was mainly to motivate the approach. In general the matrix $\mathbf{T}$ will depend on both the illumination characteristics and the scene properties. The same illumination change (say from daylight to indoor illumination) will lead to different matrices $\mathbf{T}$ depending on the scene from which it is computed.

8.2 Moments of Color Image

Let now $\mathbf{x} = \big(x_1, x_2, x_3\big)$ be a vector and $p(\mathbf{x})$ be a probability distribution. For a multiindex $i = \big(i_1, i_2, i_3\big)$ we define the moment as:

$$m_i = \int x_1^{i_1} x_2^{i_2} x_3^{i_3} p(\mathbf{x})\, dx \tag{8.4}$$

and call $i_1 + i_2 + i_3$ the order of the moment. First order moments are expectations. We will denote the expectation of component k by η_k :

$$\eta_k = \int x_k p(\mathbf{x})\, dx \tag{8.5}$$

Second-order moments will be denoted by σ_{ij} :

$$\sigma_{ij} = \int x_i x_j p(\mathbf{x})\, dx \tag{8.6}$$

The matrix consisting of the second-order moments is denoted by $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{pmatrix} \tag{8.7}$$

We also need third-order moments in the variables x_2 and x_3 and write:

$$\tau_j = \int x_2^j x_3^{3-j} p(\mathbf{x})\, d\mathbf{x} \qquad (j = 0 \ldots 3) \tag{8.8}$$

In the following series of theorems we will investigate expectations and second-order moments. Assume $p(\mathbf{x})$ is a probability density of the random variable $\mathbf{x} = (x_1, x_2, x_3)'$ with correlation matrix $\mathbf{\Sigma}$ and suppose the correlation matrix $\mathbf{\Sigma}$ has full rank. Then:

Theorem 1 There is a 3×3 matrix $\mathbf{T}$ such that the transformed random variable $\mathbf{Tx}$ has second-order moment matrix $\mathbf{\Sigma}_T = \mathbf{E}$, the identity matrix.

The second-order moment matrix of the transformed variable $\mathbf{Tx}$ is given by $\mathbf{\Sigma}_T = \mathbf{T\Sigma T}'$. Using the singular value decomposition $\mathbf{\Sigma} = \mathbf{V}'\tilde{\mathbf{D}}\mathbf{V}$ with an orthonormal matrix $\mathbf{V}$ and a diagonal matrix $\tilde{\mathbf{D}}$ (with positive entries in the diagonal) we get $\mathbf{\Sigma}_T = \mathbf{T\Sigma T}' = \mathbf{TV}'\tilde{\mathbf{D}}\mathbf{VT}'$. Since the correlation matrix $\mathbf{\Sigma}$ has full rank, we can always define $\mathbf{D}$ through the relation:

$$\mathbf{D} \times \mathbf{D} = \tilde{\mathbf{D}}^{\mathbf{-1}} \tag{8.9}$$

The required solution, which is defined as $\mathbf{T} = \mathbf{DV}$, will normalize the second-order moment matrix to the identity matrix $\mathbf{E}$.

In the following theorem we will normalize the expectation vector:

Theorem 2 There is a 3×3 matrix $\mathbf{T}$ such that the transformed random variable $\mathbf{Tx}$ has second-order moment matrix $\mathbf{\Sigma}_T = \mathbf{E}$ and the expectation vector $(r, 0, 0)'$.

Using the last theorem we can assume that the matrix of second-order moments is the unit matrix: $\mathbf{\Sigma} = \mathbf{E}$. Since the moment matrix of $\mathbf{Tx}$ is equal to $\mathbf{T\Sigma T}' = \mathbf{E}$ we find that the transformation $\mathbf{T}$ has to be a three-dimensional rotation or a reflection. From geometry it is clear that given any vector $\mathbf{y}$ there is a three-dimensional rotation $\mathbf{T}$ such that $\mathbf{Ty} = (r, 0, 0)'$ where r is the length of $\mathbf{y}$. Using $E(\mathbf{Tx}) = \mathbf{T}E(\mathbf{x})$ and $\mathbf{y}$ as the expectation vector $\mathbf{y} = E(\mathbf{x})$ proves the theorem.

The last two theorems ensure that we can find a linear transformation $\mathbf{T}$ such that the expectation vector points in the x-direction and the matrix of second-order moments is the unit matrix. In the next theorem we will investigate to what extent these properties determine $\mathbf{T}$:

Theorem 3 Assume the random processes $\mathbf{x}$ and $\mathbf{Tx}$ have expectation vectors $(r_x, 0, 0)'$ and $(r_T, 0, 0)'$ respectively. Assume further that the matrix of second-order moments is the unit matrix for both processes. Then the matrix $\mathbf{T}$ must be either a 3-D rotation matrix around the x-axis or a reflection matrix of the form:

$$\begin{pmatrix} \delta_1 & 0 & 0 \\ 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 \end{pmatrix} \tag{8.10}$$

where δ_k is either 1 or -1.

From the requirement that the second-order moment matrices of both processes are the unit matrix we get: $\mathbf{E} = \mathbf{TET}' = \mathbf{TT}'$ from which we conclude that $\mathbf{T}$ must be an orthonormal matrix. $\mathbf{T}$ is not necessarily a rotation, it can also be a reflection or a combination of both.

Writing the matrix $\mathbf{T}$ as a product of a rotation followed by a reflection it can be seen that the requirement that the expectation vectors are given by $(r_x, 0, 0)'$ and $(r_T, 0, 0)' = \mathbf{T}\,(r_x, 0, 0)'$ shows that $\mathbf{T}$ has the x-axis as fixed axis. Therefore it must be a rotation around the x-axis or a reflection or a combination of the two. If $r_x > 0$ and $r_T > 0$ then $\delta_1 = 1$.

From the last theorem it follows that the requirement that the transformed process has uncorrelated components with unit variance determines the transformation matrix up to one continuous parameter, the rotation angle around the x-axis. We could therefore add one more constraint, for example in the form of the annihilation of another third-order moment, and fix the value of the rotation angle by the solution of the constraining equation. We will not follow this approach since it does not give a hint on how to find the additional constraint. Instead we will follow a more systematic, group theoretically motivated solution. The group theoretical background is described in (Tran, 1999).

Theorem 4 Consider a two-dimensional stochastic process with variables y, z. Define the third-order moments τ_k as in Eq. 8.8 where we use y, z instead of x_2, x_3. Combine them to the complex third-order moment:

$$t(y, z) = \tau_3 + i\tau_2 + \tau_1 + i\tau_0 \tag{8.11}$$

From the original process compute a new process by applying a 2-D rotation with an angle α to the independent variables y, z resulting in the new variables y', z'. We define the corresponding third-order moments τ_k' and the complex moment $t(y', z')$ correspondingly and get for the complex third-order moments the relation:

$$t(y', z') = e^{i\alpha} t(y, z) \tag{8.12}$$

From this we find the following normalization procedure for the rotation angle.

Theorem 5 For a two-dimensional process with components y, z there is a unique rotation with rotation angle α such that $t(y', z') \in \mathbb{R}$ and $t(y', z') > 0$.

It now remains to investigate the influence of the reflections. Reflections on the first coordinate axis are not possible since we normalized the expectation of x_1 to a positive number. From the definition of the complex moment $t(y, z)$ we get the following effects of reflections on the coordinate axis:

Theorem 6 The complex moment function $t(y,z)$ transforms as follows under reflections:

$$\begin{aligned} t(-y,z) &= -t(y,z) \\ t(y,-z) &= \overline{t(y,z)} \\ t(-y,-z) &= -\overline{t(y,z)} \end{aligned} \tag{8.13}$$

If two stochastic processes given by (y,z) and (y',z') are related by a reflection and if they satisfy $t(y,z) \in \mathbb{R}, t(y',z') \in \mathbb{R}, t(y,z) > 0$ and $t(y',z') > 0$ then the reflection is around the z-axis: $z' = \pm z$.

Summarizing, the normalization procedure works as follows:

1. Use principal component analysis to compute the rotation matrix $\mathbf{T}_1$ such that the matrix of second-order moments is diagonal.
2. Compute the diagonal scaling matrix $\mathbf{T}_2$ such that the transformed variables have unit variance.
3. Apply the rotation matrix $\mathbf{T}_3$ such that the expectation vector points in the positive x-direction.
4. Rotate the last two components with the 2-D rotation matrix $\mathbf{T}_4$ such that the complex third-order moment is real and positive.
5. Finally use a reflection $\mathbf{T}_5$ on the third component to make the lowest odd-order moment positive.
6. The product $\mathbf{T} = \mathbf{T}_5\mathbf{T}_4\mathbf{T}_3\mathbf{T}_2\mathbf{T}_1$ normalizes the moments of the color distributions as described above.

When the matrix of second-order moments $\mathbf{\Sigma_x}$ is singular the matrices $\mathbf{T}_1$, $\mathbf{T}_2$ which normalize the correlation matrix are no longer unique. In this case we select from the whole class of allowable transformation matrices one element. Specifically we assign 1 for all undefined elements on the diagonal of $\mathbf{T}_2$. Each color image defines then a unique transformation matrix but the same transformation matrix may characterize different color distributions. For singular correlation matrices the normalization algorithm is as follows. When

Rank$(\Sigma) = 2$: or the eigenvalues of the second-order moment matrix $\mathbf{\Sigma_x}$ are $\lambda_1 \geq \lambda_2 > 0,\ \lambda_3 \approx 0$. We choose the rotation $\mathbf{T}_3$ as a rotation around the third axis such that the transformed process has correlation matrix $\mathbf{\Sigma_T} = diag(1,1,\lambda_3)$ and expectation vector $(r_1^+, 0, r_3^+)'$ with $r_1^+, d^+ \in \Re^+$. The other matrices $\mathbf{T}_4 = \mathbf{T}_5 = E$

Rank$(\Sigma) < 2$:
In this case we choose the transformation matrices $\mathbf{T}_3 = \mathbf{T}_4 = \mathbf{T}_5 = E$. Important examples are monochrome images.

8.3 Implementation and Experiments

This section describes the application of the proposed normalization algorithm in three difference applications: color correction, illumination-invariant color object recognition, and a color indexing application.

8.3.1 Input databases

The experiments in this chapter used an image database[1] from the Computer Science Laboratory, Simon Fraser University, Vancouver, Canada. We refer to this database as the SFU-database

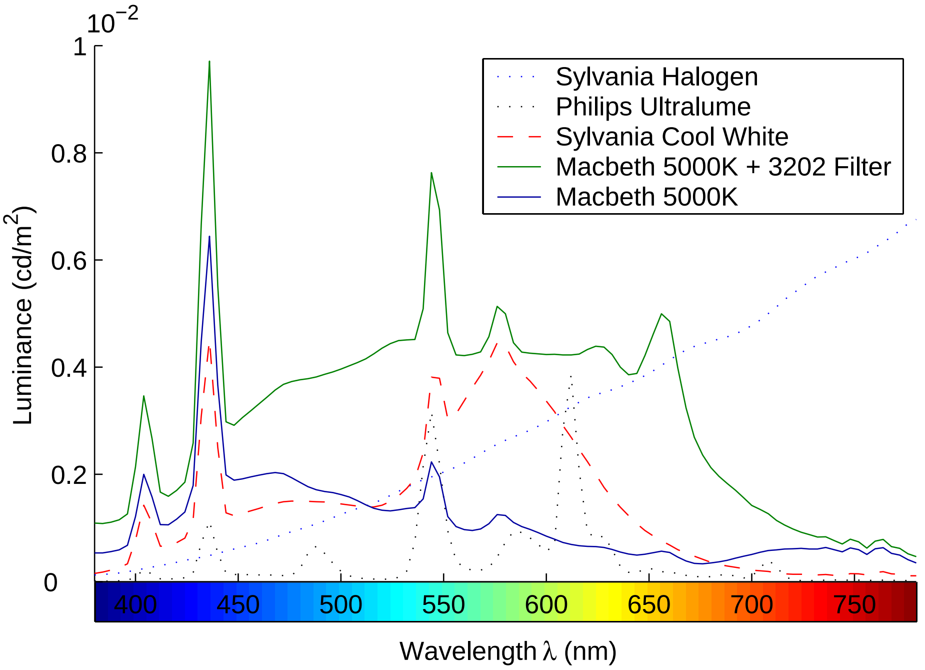


Figure 8.1: Spectra of five test illuminants

The images in the SFU-database show eleven different, relatively colorful objects (Fig. 8.3 shows the objects). The pictures were taken with a Sony DXC-930 3-CCD color video camera balanced for 3200K lighting with the gamma correction turned off so that its response is essentially a linear function of luminance. The RGB response of the camera was calibrated against a Photoresearch 650 spectroradiometer. The aperture was set so that no pixels were clipped in any of the three bands (i.e. $R, G, B \leq 255$).

[1] More information about the data set is available at the website of Computer Science Laboratory, Simon Fraser University, Vancouver, Canada `http://www.cs.sfu.ca`

The images are taken under five different illuminants using the top section (the part where the lights are mounted) of a Macbeth Judge II light booth. The illuminants were the Macbeth Judge II illuminant A, a Sylvania Cool White Fluorescent, a Philips Ultralume Fluorescent, the Macbeth Judge II 5000 Fluorescent, and the Macbeth Judge II 5000 Fluorescent together with a Roscolux 3202 full blue filter, which produced an illuminant similar in color temperature to a very deep blue sky. The effect created by changing between these illuminants can be seen in Fig. 8.2 where the same ball is seen under the different illuminants. The illuminant spectra are plotted in Fig. 8.1.

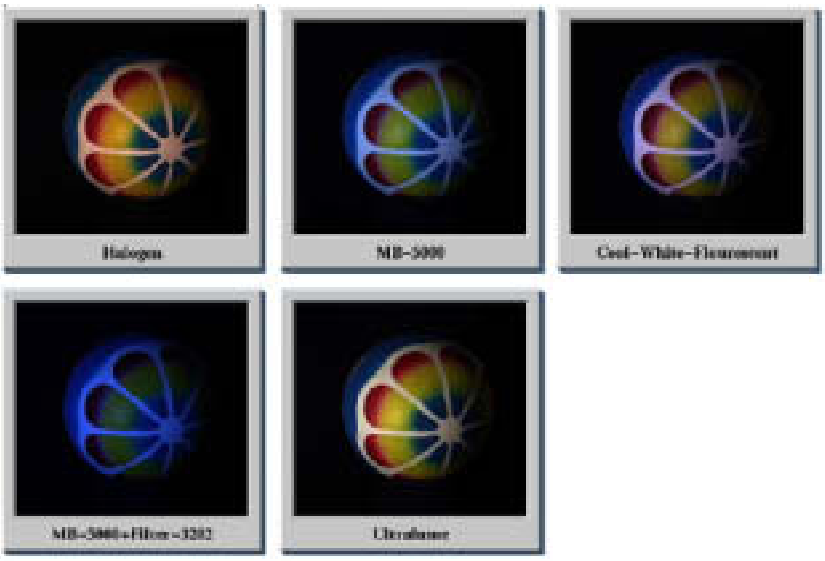


Figure 8.2: Object Ball-2 as seen under 5 different illuminants.

Two sets of images were taken. For the "model" set, images of each object were taken under each of the five illuminants, without moving the object. This gave eleven groups of five registered images. The "test" set is similar, except that the object moved before taking each image. In total, 110 images were used in the database. These two sets of images are used to evaluate color indexing under different scene illuminants with and without changes in object position.

We also used the VisTex (see chapter 5) database in moment-based search.

8.3.2 Color Correction

In our first set of experiments we compared the color mapping properties of the moment-based normalization method with conventional color constancy methods. For this experiment we use the registered images in the SFU database. The object points are in pointwise correspondence and the color mapping depends only on the changing illumination conditions. We implemented and

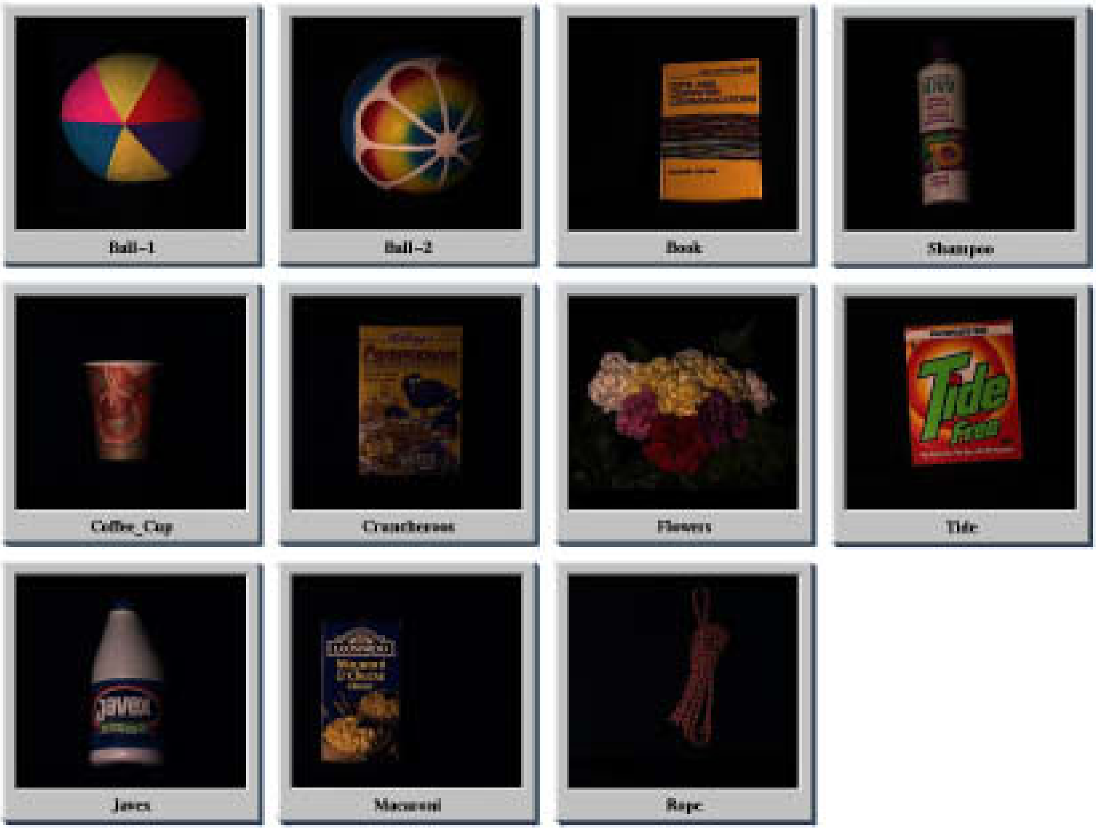


Figure 8.3: The 11 objects in the image database as seen under a single illuminant

tested the performance of the following color constancy methods (the methods and implementation details are described in (Tran, 1999)).

- **NO**: No algorithm applied
- **BT**: Best linear transform by using a full matrix which gives minimum least squared error (BT)
- **BD**: Best diagonal transform by using a diagonal transform which gives minimum least squared error
- **GW**: Grey world algorithm
  using all pixels in the image (GW1) and
  ignoring background points in the image (GW2)
- **RET**: Retinex
  using all pixels in the image (RET1) and
  ignoring background points in the image (RET2)

- **GM**: Gamut mapping
  solution chosen by hull points average (GM1)
  centroid of the hull (GM2)
  maximum volume heuristic (GM3)
- **MB**: Moment-based with different outlier values
  outlier = 0 (MB1)
  outlier = 0.5% (MB2)
  outlier = 1% (MB3)
  outlier = 2% (MB4) and
  Outlier = 5% (MB5)

The implementation of the moment-based method has to take into account that the matrix multiplication model is only an approximation and that the matrix elements must be computed from the moments which are estimated from the image data as described. For real images neither of them is completely fulfilled: the matrix model is only a linear approximation of the true transformation and the moments have to be estimated from the image data. The third-order moments in particularly are highly sensitive to statistical deviations such as outliers (Rousseeuw and Leroy, 1987). This was confirmed in our experiments and we include therefore a preprocessing step in which extreme points are ignored in the third-order moment computations. We did several experiments with different threshold values for outlier detection.

For each object in the database, we have five different images of this object under five illuminants in identical position. We computed for each of the five images the transformation matrix T to transform those images to descriptors which are independent of illuminants. Combining two of them provides the linear color mapping between the two images.

For example, Fig. 8.3.2 shows the images of the ball-2 object, which are corrected by the moment-based method. Five balls in the diagonal are copied from the original images of the object taken from the database, see Fig. 8.3.2. The other balls are results of color constancy corrections. The ball at column i, row j say $B(i,j)$ is the result of mapping ball image j to the illuminant of ball image i.

In order to measure the performance of different color constancy algorithms we use the root mean square (RMS) difference between the mapped image and the registered target image on a pixel-by-pixel basis taken across the entire image . Table 8.1 summarizes the RMS error of the algorithms for the cases were the sampling value is equal 1 (all pixels), 5, 10 and 20.

Sampling is used here to test the effect of downsizing the image. For example sampling = 5 means that not all pixels in the image but only one of 5 x 5 pixels are used. The motivation of using sampling is to test the algorithms in different resolutions of the image database.

Method	$R_1(HI)$	$R_2(HI)$	$R_3(HI)$	$R_1(KL)$	$R_2(KL)$	$R_3(KL)$
Nothing	38.6	7.1	4.8	37.3	11.9	9.0
Perfect	100	0	0	100	0	0
GW1	88.1	4.8	2.7	86.7	5.8	2.6
GW2	95.2	2.8	0.9	95.9	2.8	0.7
RET1	80.2	7.3	1.9	81.1	8.0	2.7
RET2	80.2	7.3	1.9	81.1	8.0	2.7
GM1	82.3	6.6	1.2	85.1	5.5	1.7
GM2	80.3	4.8	3.4	82.8	5.6	3.9
GM3	81.9	4.1	2.6	83.0	5.2	2.9
MB1	65.6	10.2	5.4	64.9	10.8	5.1
MB2	67.7	12.5	5.5	64.4	11.2	6.1
MB3	67.5	14.2	7.7	60.2	14.4	8.0
MB4	79.5	8.0	3.5	69.2	11.1	6.8
MB5	71.3	10.8	5.4	66.3	9.2	5.5

Table 8.2: Color indexing results using OPP axes (Rank k matches for Histogram intersection (HI) and Kullback-Leibler (KL) distance)

We found in these experiments that the results depend significantly on the procedure chosen to compute the histograms. This includes the way to define the bins and to select the number of bins used in histograming. Also in this experiments the diagonal matrix-based methods like gray-world and retinex color constancy algorithms provided better search results than the moment-based method.

An interpretation of these results is that given the three response functions of the human eye, or camera sensor, only the general model is sufficient to map accurately color observations to descriptors. However if a visual systems sensors are narrow band then the diagonal model is all that is required. In our experiments, the images in the database are special. The images were taken carefully under controlled conditions and the camera sensors are quite sharp. That may be one reason explaining why in our experiments, the diagonal model, which is the model of almost color constancy algorithms (Gray world, Retinex, Gamut mapping) worked well. The moment-based method is based on a general model. It estimates the full 3x3 transformation matrix, which has 9 parameters. It is thus more complicated than the diagonal model. One of the reasons why the moment-based method actually is not as efficient might be that it is a normalization algorithm, not a color constancy algorithm. It normalizes the input images to the descriptors which have the same statistical properties (first, second and some third-order moments) by multiplying the input image with a full 3 by 3 matrix M. In this process, both the information coming from illumination as well as sensors and reflectance is normalized. But the goal of color constancy is only to normalize the illumination.

To improve the result of this method when applying it to color constancy, we have to somehow find a way to separate M, which has 9 parameters, into two parts: one part depending on the illumination and the other part independent of the illumination.

8.3.4 Color Indexing

In the last set of experiments we used the transformation matrices as descriptors of the moments of the color distributions. Similar transformation matrices are assumed to originate in similar color distributions and we can therefore use the similarity $\mathrm{dist}(\mathbf{T_1}, \mathbf{T_2})$ between the transformation matrices $\mathbf{T_1}$ and $\mathbf{T_2}$ as a measure of similarity between the underlying images. Here $\mathrm{dist}(\mathbf{T_1}, \mathbf{T_2})$ can be taken as one of the matrixnorms. In our experiments we combined $\mathbf{T_1}, \mathbf{T_2}$ to $\mathbf{T} = \mathbf{T_1}\mathbf{T_2}^{-1}$ and compared $\mathbf{T}$ to the unit matrix by defining $\mathrm{dist}(\mathbf{T_1}, \mathbf{T_2}) = \min_{ij} |t_{ij} - \delta_{ij}|$ where t_{ij} are the elements of $\mathbf{T}$ and δ_{ij} is the Kronecker symbol.

The image in Fig. 8.5 shows a simple example in which mainly green images are retrieved. In this example the first image is the template image and the other images are sorted using the similarity to the template image

An advantage of this matching algorithm is speed: computation of this similarity measure is much faster than the histogram-based methods since it involved only multiplication of two 3x3 matrices. This was implemented and tested on the images in the VisTex database.

8.4 Summary

The goal of this work was to implement and compare color constancy algorithms with emphasis on the moment-based method. Comparisons were performed under both RMS error and performance of color-based object recognition. Two method, Color Indexing and Kullback-Leibler distance were used in object recognition, in which Kullback-Leibler distance performed slightly better.

The work also showed that color constancy pre-processing did a significant improvement in object recognition performance over doing no pre-processing. But it seems that it was not enough for object recognition although the results of color constancy under human vision was quite good.

The moment-based method is actually a normalization algorithm, but when apply it to solve color constancy, it showed quite good results. To apply it in color constancy more efficiency, we have to find a way to separate the illumination information in the normalization process.

Thus to a reasonable extent, the original goal has been achieved. But it is worth pointing out that color constancy processing on image data is not enough for color-based object recognition. We have to find more efficient color constancy algorithm, probably based on a combination of existing methods.

Chapter 9

APPLICATION: BABYIMAGE PROJECT

9.1 Overview

Most of the images used in newspaper production today will sooner or later be converted into digital format. The images are of different quality and in many cases some processing is needed before they can be printed. In this project we investigated a special class of images, family images as shown in Fig. 9.1, that fill approximately one page every week in the regional ”Östgöta Correspondenten” newspaper.

The images on the page are scanned from the original pictures and (after a routine pre-processing step) printed in the order they are scanned. This may lead to a page layout in which pictures of different color characteristics are printed side by side and it may also lead to situations in which images with severe color distortions are printed.

In this project we tested first if standard color correction methods could be used in this application. These studies showed that these methods might lead to unacceptable results if they do not take into account the structure of the images. In our experiments we found that color correction methods should be based on information about the color distribution of the large background area and the face/skin pixels. We then developed a sorting algorithm based on statistical methods that tries to put similar images near to each other. Next we defined a quality function that takes into account the color appearance of a whole page. This quality function can then be used to color correct the sorted images.

We also experimented with an automatic segmentation process that extracts the background and the skin pixels from an image. This basic color segmentation method is then combined with the geometrical information about the location of the background area to divide the image into three regions, the background, the skin areas and the remaining image points. Based on the statistical properties of the background and the skin pixels a two-step color correction method is then applied to decrease the color differences between adjoining images on a page.

9.2 Current Printing Process

In the experiments we used two different databases consisting of 30 and 38 images respectively. The 30 images in the first database were published in one week in the year 1999 and the 38 images in the second set of images were published on one page in the year 2000. Each of the images consisted of approximately 350 x 540 pixels.

Currently the images come as paper prints from the photographer. These paper copies are then scanned (not individually but in larger batches simultaneously) and automatically color corrected with a standard program. This program does not analyze the images but applies the same transformation to all the images. The control parameters for the color transformation are chosen in a way that the average image looks good in print. Together with an image comes a text that describes the family on the image. Since it is important that the right image is combined with the right text it is currently not possible to change the order in which the images are printed on the page.

In the current production process it is possible that an image with a severe distortion of the color distribution (very red faces for example) is printed as it is. It is also possible that images with very different color distributions are printed side by side. Human vision has the ability to compensate automatically for illumination changes. We know that the color of an object usually does not change and we tend to remember colors rather than to perceive them consciously. When we see several images side by side on one page, then the colors of background and face regions, which we usually ignore when we look at the images one by one, are seen as they really are. Consequently we will perceive the page as inhomogeneous and inferior to a more homogeneous layout. When a dark image is surrounded by light images, the page appears to have a dark hole. On the other hand, a page will look homogeneous, and consequently more pleasant, if images of similar color appearance are located near each other.

The two examples in Fig. 9.3 and Fig. 9.4 illustrate the difference between a homogeneous page and a page with very different images located side-by-side. In the middle region of the inhomogeneous image there is a dark image surrounded by light images which makes this page layout clearly inferior to the first, homogeneous page.

9.3 The Proposed Methods

We have performed the following experiments:

1. Manual segmentation of the images in different regions of interest and calculation of statistical properties of the extracted regions.

2. Development of a statistics-based quality function describing the appearance of the page layout.

3. Investigation and implementation of statistics-based sorting strategies to optimize the page layout.

4. Design of context sensitive, global, statistics based color correction methods to improve the appearance of the sorted page layout.

5. Application of automatic color segmentation and clustering techniques to detect background and skin regions in the images.

6. Implementation of context sensitive color screening and mapping algorithms.

9.3.1 Application of Conventional methods

In our first studies we tested conventional color constancy and color normalization methods on the images in the first set of images. These tests showed that a successful processing method required some form of analysis of the image content. Since the main purpose of the study was the development of color correction and automatic page layout methods, we decided to start with a rough manual segmentation of the images.

In the manual segmentation process we identified several regions in each image:

1. One region for every face and

2. One region for the highlighted background points originating in the illumination source

3. One region for the remaining background pixels
4. The remaining region consisting mainly of clothes but also of other skin regions like arms.

For each such region we computed a number of statistical parameters describing the statistical properties of the color distribution in this region such as mean values and correlations between the color vectors in the color coordinate systems like RGB, CIE-LAB and polar coordinates in CIE-LAB.

In a first series of experiments we tested whether conventional color correction methods could be used to improve the color appearance of the final page layout. We tested global methods in which all color pixels undergo the same transformation. The transformations tested included:

1. Standard methods such as the "Grey World" approach and other von-Kries type transformations in which the R-, G- and B-channels are scaled with different scaling factors and
2. Our own color normalization method based on third order moments of the color distributions as described in the previous chapter.
3. CIE-LAB based "Grey World"-type normalization

The transformation parameters were computed from the statistical properties of

1. The complete image
2. The complete background and
3. The background without the highlighted region.

None of these experiments produced acceptable results for all the images in the database. Some of the problems with this method are illustrated in Fig. 9.2. All these images are obtained by using a conventional grey-world algorithm. The color correction parameters are computed from all pixels in the image (left image), the skin-tone pixels (middle image) and the background points (right image). In the correction step the color of all pixels in the image are changed based on these parameters. As a result the global statistics, the skin areas and the background are similar but the resulting image is far from optimal.

As a result of these experiments we decided to experiment with two different normalization strategies:

1. We still use global color mappings that transform all color pixels in an image in the same way but we modify the distance measure between the color properties of two images in two ways:

- We compute the distance between the images as a linear combination of the distance between the background color distributions and the distance between the face-distributions
- We describe the color properties in polar coordinates in CIE-LAB. In this system the L-component represents intensity, the radius in the (ab)-plane measures saturation and the angle in the (ab)-plane represents hue. For each property we can introduce a weight factor describing the cost of changing this property in the face or the background region. We thus constrain the amount of color changes possible in this step to eliminate the risk of extreme color changes

2. In this approach we give up the global mappings and apply two different color mappings to the background region and to the rest of the image. In a transition region the mapping is obtained by blending the two mappings linearly. The transformation parameters are computed from the pixel statistics of the background and the skin regions in the image.

The first approach is simpler since it only requires the computation of the statistical parameters of the background and the face- or skin-regions. The second approach is more complex since it has to compute the statistical parameters of the background and the skin regions and it also has to find the background region and the transition region between the background and the rest of the image.

9.3.2 Optimizing Page Layout

Analyzing the appearance of different arrangements of the images on a page we concluded that the overall impression of a page depended mainly on the color of the large background regions in the images. A visually pleasant arrangement was mainly characterized by small perceptual differences between neighboring images. A quality function capturing this homogeneity property must therefore be based on a measure of the difference between two statistical distributions of color vectors. The definition of a measure that takes into account both the statistical properties of the color vectors and the perceptual relations between the colors in the distributions is still an unsolved problem. In our application we decided that it was sufficient to incorporate only the statistical properties since the colors in the two relevant distributions are always in the same region of color space.

In our first series of experiments we decided to use only globally-defined color transformations where all pixels in an image are treated in the same way. We first used the statistical parameters of the background regions and computed for a pair of images the intensity-based distance between the two distributions.

Statistics-based layout quality function

Among the many possible distance measures between two probability distributions we selected the Bhattacharya (3.17) distance and the differential geometry-based measure for normal distributions presented in chapter 5.

In the following we denote by $dist_{BI}(I_k, I_l)$ the Bhattacharya- and by $dist_{AI}(I_k, I_l)$ the Amari-distance between image I_k and image I_l in the database computed from the distribution of intensity values of all the pixels in the background. As a measure of the intensity of a color we use the $L-$part in the CIE-LAB color co-ordinate system. For the case where the highlight pixels in the background are ignored we get the corresponding distance measures $dist_{BIH}(I_k, I_l)$ and $dist_{AIH}(I_k, I_l)$. For a complete page layout we define the combined distance measures:

$$dist_P = \sum_l \sum_k dist(I_{k,l}, I_{k+1,l}) + \sum_l \sum_k dist(I_{k,l}, I_{k,l+1}) \tag{9.1}$$

where $dist$ is one of the distance measures $dist_{AI}$, $dist_{BI}$, $dist_{AIH}$, or $dist_{BIH}$. The first sum measures the accumulated distances computed over all neighboring images in columns and the second sum is the corresponding measure computed over all neighboring images in rows. If we want to emphasize that the value of $dist_P$ depends on the arrangement A of the images on that page, we write $dist_P(A)$.

Following the general rule to change the original images as little as possible we improve the quality of a page (or decrease the value of the distance measure $dist_P$) by sorting alone. The images on the page are thus only rearranged but their colors are unchanged.

Finding an optimal arrangement A_{opt} with $dist_P(A_{opt}) \leq dist_P(A)$ for all arrangements A is a difficult combinatorial optimization problem. We did not attempt to solve this in general. Instead we start with a random arrangement and improve the page layout by using the following trial-and-error procedure:

- In each iteration we select randomly a pair of images on the page
- Then we compute for each image the contribution of this image pair to the general $dist_P$ value and the contribution when these two images are exchanged.
- If the combined contributions from the two images in the swapped positions is lower than the contributions when they are located in the current positions, we exchange their positions. Otherwise we leave the arrangement as it is.

Such an iteration is very fast since it only involves the computation of 16 distance values (4 distances between the center image and its four neighbors, for each of the two images in each of the two positions). Usually we used 5,000 such checks and found that the process had stabilized in an acceptable rearrangement. Reversing the decision and exchanging the images when the $dist_P$ value is increased by such a change gives a way to find optimally bad pages. These optimization processes were used to obtain the images shown in Fig. 9.3 and Fig. 9.4.

Optimizing page layout using statistical color correction

The page obtained after the sorting consists of the original images as produced by the scanner. After the sorting step we experimented with different techniques to improve the quality of the resulting page further. As mentioned above, we use polar CIE-LAB coordinates at this processing stage. In the optimization procedure the color transformation matrix is modified by three operations:

- Multiplication of the L-component (resulting in an increasing or decreasing intensity value)
- Multiplication of the radial ab-coordinate (modifying the saturation properties)
- Shifting the hue variable

The quality of a given color transformation is then measured by a quality function which incorporates the following factors:

- Cost of changing the initial distributions
- Distance between the background distributions
- Distance between the face distributions

In these experiments the distance between two distributions of color vectors is measured by their Bhattacharya distance since the differential geometry-based method is less well understood for higher dimensional stochastic variables. The final distance between two images is the weighted sum of the three factors mentioned above. Given the quality of a given page layout (as measured by this combination of distance measures) we can optimize it by changing the intensity- and saturation scaling parameters and the hue-shift. Finding a good page layout is an optimization problem that was solved with the help of the MATLAB optimization toolbox. Note that this optimization process does not actually transform the images involved, it operates only on the values of the

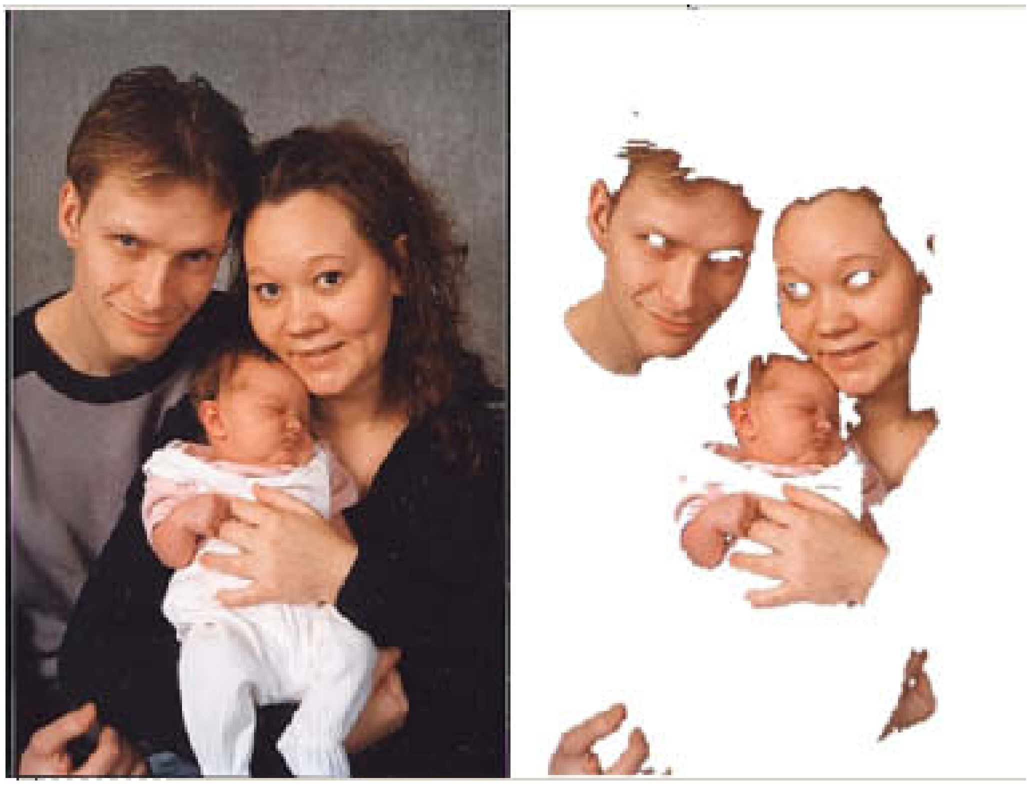


Figure 9.6: A color image and its segmented skin tone area.

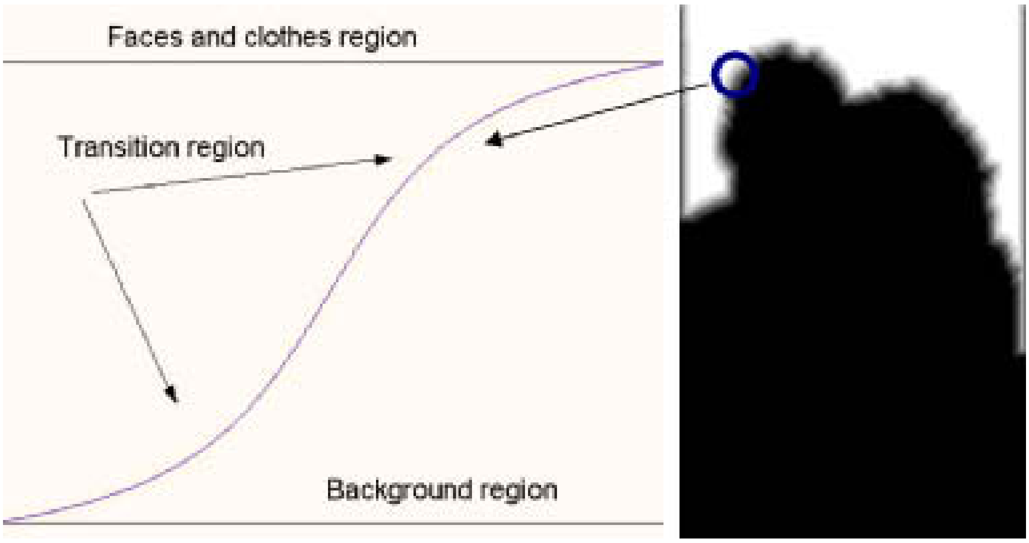


Figure 9.7: Transition area between the background and the rest of the image.

statistical parameters. The colors in an image are only changed after the optimization program stabilized and the final transformation matrix for the image is found.

An example of the results obtained with this technique is shown in Fig. 9.5.

9.3.3 Automated Color Segmentation Techniques

Manual segmentation of the images is very time-consuming and error-prone. We therefore experimented with automatic segmentation techniques to avoid operator intervention. We use first a clustering technique to extract the background and the skin regions. This method classifies regions according to their color properties. It turns out that both background and skin pixels can be automatically extracted with sufficient accuracy. In contrast to the first manual segmentation this method will only extract the skin regions in the faces, it will therefore not select the hair and eye regions for example. It will also detect skin regions outside the faces, such as bare arms. It turns out that the statistical properties of the face regions extracted with the first, manual segmentation and the corresponding skin-regions found by the second method differ significantly.

We find the background and skin regions in an image, by first using the mean shift cluster algorithm to segment the image into several color regions (about 20 regions for each image). The color properties of each region are then used to decide if the region belongs to the background or the skin tone area. Simple thresholding of the intensity, hue, and saturation gives quite robust clustering results. Fig. 9.6 and Fig. 9.7 show the segmented skin tone region of the right image in Fig. 9.1. We also utilized the fact that the background region is the large homogeneous region on top of the image. Therefore it is easy to divide the image into two regions: the background region and the rest. Once the background is identified, it is easy to define two color transformations: one for the background and one for the rest of the image.

As an example we show:

- First two images as they are scanned from the original pictures (Fig. 9.1)
- We modify the left image so that its global color properties become similar to the right image.
- In the next example we modify the left image so that its skin tone pixels become similar to the skin pixels in the right image.
- Then we modify the left image so that its background becomes similar in color appearance to the background of the right image
- In the fourth example both the background and the skin pixels in the left image are modified so that they have similar color appearance as

the corresponding regions in the right image. Since two transformations are used in this method, there will be a border effect on the corrected image, especially when the two transformations are very different. We therefore define a transition area between the two regions of 20 pixels width (Fig. 9.7), and color properties of pixels in this transition area are smoothed so that the border effect is reduced. The result of the experiment is summarized in Fig. 9.8.

Another example is shown in Fig. 9.9. Similar to the previous experiment one can use the results of the automatic segmentation of the skin and background regions to define color transformations of the images that optimize a quality function describing the properties of a page layout.

9.4 Result and Discussion

We developed and investigated two strategies to optimize the color appearance of a printed page consisting of a collection of similar images. The first method uses only global color transformations, which transform all pixels in an image in the same way. Finding the parameters that define the transformation requires however an extraction of the background and the skin regions in the image.

The normalization used in the other method extracts first the skin and the background regions in an image. The skin regions are to a large extend identical to the face regions used in the first method but the also include other regions like arms and they do not include non-skin face regions like hair and eyes. After the color-based segmentation, the geometrical information about the location of the background is used together with the color information to automatically extract the background region. Finally the background and the remaining part of the image are transformed with two different color transformations.

Finally we want to point out that the result of the automatic skin detection process used in the second method cannot only be used for color normalization. It can also be used for pre-screening, i.e. it could, for example, be used to point out for an operator skin regions which have a color distribution which is significantly different from the color properties of typical skin regions. In this way a form of quality control of the analog photo-graphical process and the scanning could be incorporated into the page layout process.

Chapter 10

CONCLUSIONS AND FUTURE WORK

10.1 Conclusions

In the thesis we investigated a number of statistical methods for color-based image retrieval and color correction, color normalization applications.

In the color-based image retrieval applications we first investigated the application of different non-parametric density estimators in estimating color distributions for color-based image retrieval applications. Our experiments show that there is a difference between the best estimator and the best descriptor for image retrieval. We showed that a histogram-based method based on a simple estimator gave better retrieval performance compared to the straightforward application of a kernel density estimator for image retrieval. In order to improve the retrieval performance of kernel-based methods, two modifications were introduced. They are based on the use of non-orthogonal bases together with a Gram-Schmidt procedure and a method applying the Fourier transform. Experiments were performed that confirmed the improvements of our proposed methods both in retrieval performance and simplicity in choosing the smoothing parameters. The affect of different smoothing parameters on retrieval performance was also investigated in the thesis.

Next we derived new, compact descriptors for probability distributions of the colors in images. These new descriptors are based on the modification of the traditional Karhunen-Loève Transform (KLT). The modification is based on the following two important aspects: the geometry of the underlying color space is integrated into the principal component analysis and the principal component analysis operates on the space of local histogram differences and not on the space of all histograms.

We also investigated new distance measures between these descriptors that take into account both the probability distribution and the geometry of the underlying color space. These distance measures are based on a differential geometrical approach which is of interest since many existing dis/similarity methods fall into this framework. The general framework was illustrated with two examples: the family of normal distributions and the family of linear representations of color distributions.

Our experiments with color-based image retrieval methods utilized several image databases containing more than 1,300,000 color images. The experiments show that the proposed method (combining both the color-based distance measures and the principal component analysis based on local histogram differences), is very fast and has very good retrieval performance compared to other existing methods.

In the thesis we also investigated color features which are independent of geometry and illumination changes. Such invariant features are useful in many applications where the main interest in the physical contents of objects such as object recognition. Both statistics- and physics-based approaches were used. For physics-based approaches, we concentrated on geometry invariants and used the theory of transformation groups to find all invariants of a given variation. Detailed descriptions were given for the dichromatic reflection model and the Kubelka-Munk model.

Apart from the image database retrieval methods we investigated color normalization, color correction and color constancy methods. Here we investigated an algorithm to normalize color images which uses a full 3x3 matrix for color mapping. The transformation matrix is computed from the moments of the color distributions of the images of interest. We compared the method to color constancy methods in color correction and illuminant invariant color object recognition. Experiments show that simple methods such as retinex and gray-world methods performed better than more complicated methods such as gamut mapping and our proposed moment-based method. Moreover none of the methods gave perfect recognition of the objects under different illuminations. False alarm rates in the recognition of eleven objects ranged from 5% to 30%. Experiments on color correction provide a reasonably good result under controlled image-capturing conditions.

Using conventional, global color correction methods in a real color correction application produced unacceptable results. We therefore developed an algorithm to re-arrange the layout of a printed newspaper page and a local color correction algorithm that was specially tuned to this application.

Summarizing, we conclude that statistical methods are useful in color-based applications, especially in applications where human perception is involved. Combining color information and statistical methods usually improves the performance of the method.

10.2 Future work

Following the investigations described in this thesis, a number of problems could be investigated further.

We have shown that kernel density estimators provide a new efficient way to describe color distributions in content-based image retrieval. The Gram-Schmidt procedure and the method applying the Fourier transform described in the thesis are examples that use kernel-based methods for image retrieval. The method proposed in chapter 6 could clearly also be used in connection with kernel density estimators.

The general strategy of using problem-based distance measures and differences of histograms is quite general and can be applied to other features used in content-based image retrieval applications such as texture. Applying this strategy to kernel-based descriptors is also another example that may improve retrieval performance.

The Karhunen-Loève Transform is a linear approximation method which projects the signal onto a priori given subspace. However, better approximations can be obtained by choosing the basis vectors depending on the signal or at least over collections of signals. Color histograms which contain isolated singularities can be well approximated with this non-linear procedure.

Color invariants have been investigated and applied to several color-based applications in the thesis. However, future work still requires improving the performance in such applications. This includes a better understanding of the underlying physical processes when light interacts with materials to be able to decouple the influence of the physical properties of the objects, the illumination and the sensor properties.

Bibliography

Akaike, H. (1974). New look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19(6):716–723.

Albuz, E., Kocalar, E., and Khokhar, A. (2001). Scalable color image indexing and retrieval using vector wavelets. *IEEE Trans. on Knowledge and Data Engineering*, 13(5):851–861.

Amari, S.-I. (1985). *Differential Geometrical Methods in Statistics.* Springer.

Amari, S.-I., Barndorff-Nielsen, O. E., Kass, R. E., Lauritzen, S. L., and Rao, C. R. (1987). *Differential Geometry in Statistical Inference.* Institute of Mathematical Statistics, Hayward, California.

Androutsos, D., Plataniotis, K. N., and Venetsanopoulos, A. N. (1999). A novel vector-based approach to color image retrieval using a vector angular-based distance measure. *Computer Vision and Image Understanding*, 75(1/2):46–58.

Atkinson, C. and Mitchell, A. (1981). Rao's distance measure. *Sankhya*, 43:345–365.

Bach, J. R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R., and Shu, C. F. (1996). The Virage image search engine: An open framework for image management. In *Proc. of SPIE Storage and Retrieval for Image and Video Databases.*

Baxter, M. J., Beardah, C. C., and Westwood, S. (2000). Sample size and related issues in the analysis of lead isotope ratio data. *Journal of Archaeological Science*, 27:973–980.

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. (1990). The R*Tree: An efficient and robust access method for points and rectangles. In *Proc. of ACM SIGMOD.*

Benchathlon (2003). The benchathlon network, http://www.benchathlon.net/.

Berens, J., Finlayson, G. D., and Gu, G. (2000). Image indexing using compressed colour histogram. In *IEE Proc.-Vis. Image Signal Processing*, pages 349–353.

Birge, L. and Rozenholc, Y. (2002). How many bins should be put in a regular histogram. Technical Report PMA-721, CNRS-UMR 7599, University Paris VI.

Brill, M. H. (1990). Image segmentation by object color: A unifying framework and connection to color constancy. *Journal Optical Society of America*, 10:2041–2047.

Brunelli, R. and Mich, O. (2001). Histogram analysis for image retrieval. *Pattern Recognition*, 34:1625–1637.

Carson, C., Belongie, S., Greenspan, H., and Malik, J. (1997). Region-based image querying. In *Proc. of CVPR Workshop on Content-Based Access of Image and Video Libraries.*

Chandrasekhar, S. (1950). *Radiative Transfer.* Oxford University Press, Oxford, UK.

Comaniciu, D. and Meer, P. (1999a). Distribution free decomposition of multivariate data. *Pattern Analysis and Applications*, 2(1):22–30.

Comaniciu, D. and Meer, P. (1999b). Mean shift analysis and applications. In *Proc. of IEEE Int'l. Conf. on Computer Vision*, pages 1197–1203.

Courant, R. and Hilbert, D. (1989). *Methods of Mathematical Physics.* John Wiley & Son.

Deng, Y., Manjunath, B. S., Kenney, C., Moore, M. S., and Shin, H. (2001). An efficient color representation for image retrieval. *IEEE Trans. on Image Processing*, 10(1):140–147.

Devroye, L. and Gyorfi, L. (1985). *Nonparametric Density Estimation: The L_1 view.* John Wiley & Sons, New York.

Dow, J. (1993). Content-based retrieval in multimedia imaging. In *Proc. of SPIE Storage and Retrieval for Image and Video Databases.*

Drew, M. S., Wei, J., and Li, Z.-N. (1998). On illumination invariance in color object recognition. *Pattern Recognition*, 31(8):1077–1087.

Eberly, D. (1999). Geometric invariance. Technical report, Magic Software, `http://www.magic-sofware.com`.

Equitz, W. and Niblack, W. (1994). Retrieving images from a database using texture alogrithms from the QBIC system. Technical Report RJ 9805, Computer Science, IBM Research.

Fairchild, M. D. (1997). *Color Appearance Models*. Addison-Wesley.

Faloutsos, C., Equitz, W., Flickner, M., Niblack, W., Petrovic, D., and Barber, R. (1994). Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262.

Faloutsos, C., Flickner, M., Niblack, W., Petkovic, D., Equitz, W., and R.Barber (1993). Efficient and effective querying by image content. Technical report, IBM Research.

Finlayson, G. D., Drew, M. S., and Funt, B. V. (1994). Color constancy: generalized diagonal transforms suffice. *Journal Optical Society of America*, 11(11):3011–3019.

Finlayson, G. D. and Schaefer, G. (2001). Solving for colour constancy using a constrained dichromatic reflection model. *International Journal of Computer Vision*, 42(3):127–144.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. (1995). Query by image and video content: The QBIC project. *IEEE Computer*, 28(9).

Forsyth, D. A. (1997). Finding pictures of objects in large collections of images. *Digital Image Access and Retrieval.*

Freedman, D. and Diaconis, P. (1981). On the histogram as a density estimator: L_2 theory. *Zeit. Wahrscheinlichkeitstheor Verw. Geb.*, 57:453–476.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press.

Funt, B. V., Barnard, K., and Martin, L. (1998). Is machine colour constancy good enough. In *Proc. of European Conf. on Computer Vision*, pages 445–459.

German, D. (1990). Boundary detection by constrained optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(7).

Geusebroek, J. M., Boomgaard, R., Smeulders, A. W. M., and Dev, A. (2000). Color and scale: The spatial structure of color images. In *Proc. of European Conference on Computer Vision.*

Geusebroek, J. M., Gevers, T., and Smeulders, A. W. M. (2002). Kubelka-munk theory for color image invariant properties. In *Proc. European Conf. on Colour Graphics, Imaging, and Vision.*

Geusebroek, J. M., van den Boomgaard, R., Smeulders, A. W. M., and Geerts, H. (2001). Color invariance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350.

Gevers, T. (2001). Robust histogram construction from color invariants. In *Proc. of IEEE Intl. Conf. on Computer Vision.*

Gevers, T. and Smeulders, A. W. M. (1999). Color based object recognition. *Pattern Recognition*, 32:453–464.

Gevers, T. and Smeulders, A. W. M. (2000). PicToSeek: Combining color and shape invariant features for image retrieval. *IEEE Trans. on Image Processing*, 9(1):102–119.

Gevers, T. and Stokman, H. M. G. (2000). Classifying color transitions into shadow-geometry, illumination highlight or material edges. In *Proc. of IEEE Int'l Conf. on Image Processing*, pages 521–525.

Greene, D. (1989). An implementation and performance analysis of spatial data access. In *Proc. of ACM SIGMOD.*

Gunther, N. and Beretta, G. (2001). A benchmark for image retrieval using distributed systems over the internet: BIRDS-I. In *Proc. of Internet Imaging II, Electronic Imaging Conf.*

Gupta, A. and Jain, R. (1997). Visual information retrieval. *Comm. ACM*, 40(5).

Guttman, A. (1984). R-Tree: A dynamic index structure for spatial searching. In *Proc. of ACM SIGMOD.*

Hafner, J., Sawhney, H. S., Equitz, W., Flickner, M., and Niblack, W. (1995). Efficient color histogram indexing for quadratic form. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(7):729–736.

Hermes, T. (1995). Image retrieval for information systems. In *Proc. of SPIE 2420 Conf. on Storage and Retrieval for Image and Video Databases III.*

Hernández-Andrés, J., Romero, J., Nieves, J. L., and Lee Jr., R. L. (2001). Color and spectral analysis of daylight in southern europe. *Journal of the Optical Society of America A*, 18(6):1325–1335.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W., and Zabih, R. (1997). Image indexing using color correlogram. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition.*

Jones, K. S. and Willett, P. (1977). *Reading in Information Retrieval.* Morgan Kaufmann Pub. Inc.

Judd, D. B., MacAdam, D. L., and Wyszecki, G. (1964). Spectral distribution of typical daylight as a function of correlated color temperature. *Journal Optical Society of America*, 54(10):1031–1040.

Judd, D. B. and Wyszecki, G. (1975). *Color in Business, Science and Industry, 3rd Ed.* Wiley, New York.

Kanazawa, Y. (1993). Hellinger distance and Akaike's information criterion for the histogram. *Statistics and Probability Letters*, 17:293–298.

Klinker, G. J. (1993). *A Physical Approach to Color Image Understanding.* A. K. Peters Ltd.

Kondepudy, R. and Healey, G. (1994). Use of invariants for recognition of three-dimensional color textures. *Journal Optical Society of America*, 11(11):3037–3049.

Kubelka, P. and Munk, F. (1931). Ein Beitrag zur Optik der Farbanstriche. *Zeitschrift für Technische Physik*, 11a:593–601.

Lee, D., Barber, R., Niblack, W., Flickner, M., Hafner, J., and Petkovic, D. (1994). Indexing for complex queries on a query-by-content image database. In *Proc. of IEEE Int'l Conf. on Image Processing.*

Lennie, P. and D'Zmura, M. (1988). Mechanisms of color vision. *CRC Critical Reviews in Neurobiology.*

Lenz, R., Bui, T. H., and Hernández-Andrés, J. (2003a). One-parameter subgroups and the chromaticity properties of time-changing illumination spectra. In *Proc. of SPIE Electronics Imaging.*

Lenz, R. and Tran, L. V. (1999). Statistical methods for automated colour normalization and colour correction. In *Advances in Digital Printing. IARIGAI Int. Ass. Res. Inst. for the Printing, Information and Communication Industries, Munich, Germany.*

Lenz, R. and Tran, L. V. (2000). Measuring distances between color distributions. In *Proc. of Int'l Conf. on Color in Graphics and Image Processing, Saint-Etienne, France.*

Lenz, R., Tran, L. V., and Bui, T. H. (2003b). Group theoretical invariants in color image processing. *Submitted to IS&T/SID's 11th Color Imaging Conference, Scottdale, USA.*

Lenz, R., Tran, L. V., and Meer, P. (1999). Moment based normalization of color images. In *Proc. of IEEE Workshop on Multimedia Signal Processing, Copenhagen, Denmark.*

Li, B. and Ma, S. D. (1995). On the relation between region and contour representation. In *Proc. of IEEE Int'l Conf. on Image Processing.*

Luo, M. R. (1999). Color science: past, present and future. In *Color Imaging. Vison and Technology. Ed L.W. MacDonald and M.R. Luo.*

Ma, W.-Y. (1997). *Netra: A Toolbox for Navigating Large Image Databases.* PhD thesis, Dept. of Electrical and Computer Engineering, University of California at Santa Barbara.

Ma, W.-Y. and Manjunath, B. S. (1995). A comparision of wavelet transform features for texture image annotation. In *Proc. of IEEE Int'l Conf. on Image Processing.*

Ma, W.-Y. and Manjunath, B. S. (1997). Netra: A toolbox for navigating large image databases. In *Proc. of IEEE Int. Conf. on Image Processing.*

Manjunath, B. S. and Ma, W.-Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(18).

Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., and Yamada, A. (2001). Color and texture descriptors. *IEEE Tran. on Circuits and Systems for Video Technology*, 11(6):703–715.

Maître, H., Schmitt, F., Crettez, J.-P., Wu, Y., and Hardeberg, J. (1996). Spectrophotometric image analysis of fine art painting. In *Proc. of IS&T/SID's 4th Color Imaging Conference, Scottsdale, USA*, pages 50–53.

Mehtre, B. N., Kankanhalli, M., and Lee, W. F. (1997). Shape measures for content based image retrieval: A comparison. In *Proc. of IEEE Int'l Conf. on Multimedia Computing and Systems.*

Mitra, M., Huang, J., and Kumar, S. R. (1997). Combining supervised learning with color correlograms for content-based image retrieval. In *Proc. of 5th ACM Multimedia Conf.*

Ng, R. T. and Tam, D. (1999). Multilevel filtering for high-dimensional image data: Why and how. *IEEE Tran. on Knowledge and Data Engineering*, 11(6):916–928.

Nobbs, J. H. (1985). Kubelka-munk theory and the prediction of reflectance. *Rev.Prog.Coloration*, 15:66–75.

Notes, G. R. (2002). Search engine statistics: Database total size estimates, http://www.searchengineshowdown.com/stats/sizeest.shtml, 31 dec. 2002.

Ohanian, P. P. and Dubes, R. C. (1992). Performance evaluation for four classes of texture features. *Pattern Recognition*, 25(8):819–833.

Oliva, A. (1997). Real-world scene categorization by a self-organizing neural network. *Perception*, supp 26(19).

Olver, P. (1995). *Equivalence, Invariants and Symmetry.* Cambridge University Press.

Parkkinen, J., Jaaskelainen, T., and Kuittinen, M. (1988). Spectral representation of color images. In *Proc. of IEEE 9th Int'l Conf. on Pattern Recognition.*

Pass, G. and Zabih, R. (1999). Comparing images using joint histograms. *Multimedia Systems*, 7(3):234–240.

Pentland, A., Picard, R. W., and Sclaroff, S. (1996). Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision.*

Plataniotis, K. N. and Venetsanopoulos, A. N. (2000). *Color Image Proccessing and Applications.* Springer.

Puzixha, J., Buhmann, J. M., Rubner, Y., and Tomasi, C. (1999). Empirical evaluation of dissimilarity measures for color and texture. In *Proc. of IEEE Int'l. Conf. on Computer Vision.*

Randen, T. and Husoy, J. H. (1999). Filtering for texture classification: a comparative study. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):291–310.

Rao, C. R. (1949). On the distance between two populations. *Sankhya*, 9:246–248.

Ratan, A. L. and Grimson, W. E. L. (1997). Traning templates for scene classification using a few examples. In *Proc. of IEEE workshop on CBA of Image and Video Lib.*

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection.* Wiley.

Rubner, Y. (1999). *Perceptual Metrics for Image Database Navigation.* PhD thesis, Stanford University.

Rubner, Y., Tomasi, C., and Guibas, L. J. (1998). A metric for distributions with applications to image databases. In *in Proc. of IEEE Int'l Conf. on Pattern Recognition.*

Rudemo, M. (1982). Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9:65–78.

Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62.

Scassellati, B., Alexopoulos, S., and Flickner, M. (1994). Retrieving images by 2D shape:acomparison of computation methods with human perceptual judgments. In *Proc. of SPIE Storage and Retrieval for Image and Video Databases.*

Schettini, R., Ciocca, G., and Zuffi, S. (2000). Color in databases: Indexation and similarity. In *Proc. of Int'l Conf. on Color in Graphics and Image Processing*, pages 244–249.

Schettini, R., Ciocca, G., and Zuffi, S. (2001). *Color Imaging Science: Exploiting Digital Media, Ed. R. Luo and L. MacDonald*, chapter A Survey on Methods for Colour Image Indexing and Retrieval in Image Database. John Wiley.

Scott, D. W. (1979). On optimal and data-based histograms. *Biometrika*, 66:605–610.

Scott, D. W. (1985). Average shifted histograms: Effective non-parametric density estimotors in several dimensions. *Ann. Statist.*, 13:1024–1040.

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization.* Wiley, New York.

Sellis, T., Roussopoulos, N., and Faloutsos, C. (1987). The R^+-Tree: A dynamic index for multi-demensional objects. In *Proc. of Int'l Conf. on Very Large Databases.*

Shafer, S. A. (1985). Using color to separate reflection components. *Color Research and Application*, 10(4):210–218.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London.

Simonoff, J. S. and Udina, F. (1997). Measuring the stability of histogram appearance when the anchor position is changed. *Computational Statistics and Data Analysis*, 23:335–353.

Smith, J. R. and Chang, S.-F. (1996). *Intelligent Multimedia Information Retrieval, Ed. M. T. Maybury*, chapter Querying by color regions using the VisualSeek content-based visual query system. MIT Press.

Smith, J. R. and Chang, S.-F. (1997). Visually searching the web for content. *IEEE Multimedia Magazine*, 4(3).

Squire, D. M. and Pun, T. (1997). A comparison of human and machine assessments of image similarity for the organization of image databases. In *Proc. of Scandinavian Conf. on Image Analysis.*

Süsstrunk, S., Buckley, R., and Sven, S. (1999). Standard rgb color spaces. In *Proc. of IS&T/SID's 7th Color Imaging Conference*, pages 127–134.

Stokes, M., Nielsen, M., and Zimmerman, D. (2000). What is srgb? *http://www.srgb.com.*

Stokman, H. M. G. (2000). *Robust Photometric Invariance in Machine Color Vision.* PhD thesis, Intelligent Sensory Information Systems group, University of Amsterdam.

Stricker, M. and Orengo, M. (1996). Similarity of color images. In *Proc. of SPIE Storage and Retrieval for Image and Video Databases.*

Sturges, H. A. (1926). The choice of a class interval. *Journal of American Statistical Association*, 21:65–66.

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1):11–32.

Tamura, H., Mori, S., and Yamawaki, T. (1978). Texture features corresponding to visual perception. *IEEE Trans. on Sys., Man. and Cyb.*, 8(6).

Tran, L. V. (1999). Computational color constancy. Master's thesis, Department of Signal and Systems, School of Electrical and Computer Engineering, Chalmers University of Technology, Gothenburg, Sweden.

Tran, L. V. (2001). *Statistical Tools for Color Based Image Retrieval.* Licentiate's thesis, LiU-TEK-LIC-2001:41, Dept. of Science and Technology, Linköping University. ISBN 91-7373-121-8.

Tran, L. V. and Lenz, R. (1999). Color constancy algorithms and search in image databases. In *"In Term of Design" Workshop, NIMRES2, Helsinki, Finland.*

Tran, L. V. and Lenz, R. (2000). Metric structures in probability spaces: Application in color based search. In *Proc. of Swedish Society for Automated Image Analysis, Halmstad, Sweden.*

Tran, L. V. and Lenz, R. (2001a). Comparison of quadratic form based color indexing methods. In *Proc. of Swedish Society for Automated Image Analysis, Norrköping, Sweden.*

Tran, L. V. and Lenz, R. (2001b). PCA based representation for color based image retrieval. In *Proc. of IEEE Int'l Conf. on Image Processing, Greece.*

Tran, L. V. and Lenz, R. (2001c). Spaces of probability distributions and their applications to color based image database search. In *Proc. of 9th Congress of the International Colour Association, Rochester, USA.*

Tran, L. V. and Lenz, R. (2002a). Color invariant features for dielectric materials. In *Proc. of Swedish Society for Automated Image Analysis, Lund, Sweden.*

Tran, L. V. and Lenz, R. (2002b). Compact colour descriptors for color based image retrieval. *Submitted to Signal Processing.* `http://www.itn.liu.se/~lintr/papers/sp03`.

Tran, L. V. and Lenz, R. (2003a). Characterization of color distributions with histograms and kernel density estimators. In *Proc. of SPIE Internet Imaging IV Conf., Electronics Imaging, Santa Clara, USA.*

Tran, L. V. and Lenz, R. (2003b). Differential geometry based color distribution distances. *Submitted to Pattern Recognition Letter.*

Tran, L. V. and Lenz, R. (2003c). Estimating color distributions for image retrieval. *to be submitted to IEEE Trans. on Pattern Analysis and Machine Intelligence.* `http://www.itn.liu.se/~lintr/papers/pami03`.

Tran, L. V. and Lenz, R. (2003d). Geometric invariance in describing color features. In *Proc. of SPIE Color Imaging VIII: Processing, Hardcopy, and Applications Conf., Electronics Imaging, Santa Clara, USA.*

TREC (2002). Text retrieval conference, http://trec.nist.gov.

Wand, M. P. (1996). Data-based choice of histogram bin width. *Journal of American Statistical Association*, 51:59–64.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing.* Chapman and Hall.

Weber, R., Schek, H., and Blott, S. (1998). A quantitative analysis and performance study for similarity search methods in high-demensional spaces. In *Proc. of Int'l Conf. on Very Large Databases*, pages 194–205.

Weszka, J., Dyer, C., and Rosenfeld, A. (1976). A comperative study of tecture measures for terrain classification. *IEEE Trans. on Sys., Man. and Cyb.*, 6(4).

Willshaw, D. (2001). *Special issue: Natural stimulus statistics. Network Computation in Neural Systems*, volume 12. Institute of Physics and IOP Publishing Limited 2001.

Wolf, K. B. (1979). *Integral Transforms in Science and Engineering.* Plenum Publ. Corp, New York.

Wyszecki, G. and Stiles, W. S. (1982). *Color Science.* Wiley & Sons, London, England, 2 edition.

Zeki, S. (1999). *Inner Vision.* Oxford University Press.

Zier, D. and Ohm, J. R. (1999). Common datasets and queries in MPEG-7 color core experiments. Technical Report Doc. MPEG99/M5060, ISO/IEC JTC1/SC29/WG11.

List of Figures

List of Tables

Citation Index

Văn Miếu (1070) - Quốc Tử Giám (1076)