بسم الله الرحمن الرحيم

الجامعة الإسلامية للتكنولوجيا

UNIVERSITE ISLAMIQUE DE TECHNOLOGIE
ISLAMIC UNIVERSITY OF TECHNOLOGY
DHAKA, BANGLADESH
THE ORGANISATION OF THE ISLAMIC CONFERENCE

Content Based Image Retrieval with Combined Color and Texture Features

By:

H. M. Raine Ahmed (094446)

Md. Faishal Yousuf (094448)

Supervised by:
Dr. Md. Hasanul Kabir
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology

Co-Supervised by:
Faisal Ahmed
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Academic Year: 2012-2013

Department of Computer Science and Engineering
Islamic University of Technology.
A Subsidiary Organ of the Organization of Islamic Cooperation.
Dhaka, Bangladesh.
October 2013

Declaration of Authorship

We, H. M. Raine Ahmed and Md. Faishal Yousuf, declare that this thesis titled, ”Content Based Image Retrieval with Combined Color and Texture Features” and the work presented in it are our own. We confirm that:

- This work was done wholly while in candidature for a Bachelor degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.

Submitted By:

H. M. Raine Ahmed (094446)

Md. Faishal Yousuf (094448)

Content Based Image Retrieval with Combined Color and Texture Features

Approved By:

Prof. Dr. M.A. Mottalib
Head of the Department,
Department of Computer Science and Engineering,
Islamic University of Technology.

Dr. Md. Hasanul Kabir
Thesis Supervisor,
Assistant Professor,
Department of Computer Science and Engineering,
Islamic University of Technology.

Faisal Ahmed
Thesis Co-Supervisor,
Lecturer,
Department of Computer Science and Engineering,
Islamic University of Technology.

Abstract The number of digital image and video databases in the Internet and other information sources are growing rapidly. Indexing these huge database by name is a very laborious job. To serve this purpose the concept of Content Based Image Retrieval (CBIR) has been introduced which uses visual contents to search images from large scale image databases according to users' interests.

We proposed a new image content descriptor that considers both texture and color information of an image and works on each color plane of a RGB image at the same time to reduce the feature vector size. In a $n \times n$ neighborhood of an image we find the dissimilarity of the center pixel with all other neighboring pixels using the Euclidean Dissimilarly method and Cosine similarity method. Then based on a threshold value we set some binary values to the dissimilarity values and finally get the feature vector. In this process all the three color planes (R, G and B) are considered. That's why our feature descriptor contains both color and texture information. Using this feature descriptor we train the system. When a query image comes we measure the distance between the query image and the database images using the Chi-square method and determine the most similar images of the query image. The experiments show that our method gives better image retrieval rate while using a small feature vector which is computationally very efficient.

Keywords CBIR, Image Content Descriptor, Combined Color and Texture Features

Acknowledgements

Foremost, we would like to express our sincere gratitude to our thesis supervisor Dr. Md. Hasanul Kabir for the continuous support in our research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped us in all the time of research and writing of this thesis.

Besides our supervisor, we would like to thank our co-supervisor Mr. Faisal Ahmed for helping us tirelessly. We must admit that his motivation was the key to our success. His friendly supervision, advice, constructive criticism and continuous support made it possible for us to complete the research work properly.

Contents

List of Figures

Chapter 1

Introduction

Interest in the potential of digital images has increased enormously over the last few years, fuelled at least in part by the rapid growth of imaging on the World-Wide. Users in many professional fields are exploiting the opportunities offered by the ability to access and manipulate remotely-stored images in all kinds of new and exciting. However, they are also discovering that the process of locating a desired image in a large and varied collection can be a source of considerable frustration. The problems of image retrieval are becoming widely recognized, and the search for solutions an increasingly active area for research and development. Some indication of the rate of increase can be gained from the number of journal articles appearing each year on the subject, growing from 4 in 1991 to 12 in 1994, and 45 in 1998.

1.1 What is Content Based Image Retrieval (CBIR)?

Problems with traditional methods of image indexing have led to the rise of interest in techniques for retrieving images on the basis of automatically-derived features such as color, texture and shape - a technology now generally referred to as Content-Based Image Retrieval (CBIR).

Content-based image retrieval, a technique which uses visual contents to search images from large scale image databases according to users' interests, has been an active and fast advancing research area since the 1990s. During the past decade, remarkable progress has been made in both theoretical research and system development. However, there remain many challenging research problems that continue to attract researchers from multiple disciplines.

After a decade of intensive research, CBIR technology is now beginning to move out of the laboratory and into the marketplace, in the form of commercial products like QBIC [Flickner et al, 1995] and Virage [Gupta et al, 1996]. However, the technology still lacks maturity, and is not yet being used on a significant scale. In the absence of hard evidence on the effectiveness of CBIR techniques in practice, opinion is still sharply divided about their usefulness in handling real-life queries in large and diverse image collections. Nor is it yet obvious how and where CBIR techniques can most profitably be used [Sutcliffe et al, 1997].

1.2 Motivation

1.2.1 The growth of digital imaging

The use of images in human communication is hardly new - our cave-dwelling ancestors painted pictures on the walls of their caves, and the use of maps and building plans to convey information almost certainly dates back to pre-Roman times. But the twentieth century has witnessed unparalleled growth in the number, availability and importance of images in all walks of life. Images now play a crucial role in fields as diverse as medicine, journalism, advertising, design, education and entertainment.

Technology, in the form of inventions such as photography and television, has played a major role in facilitating the capture and communication of image data. But the real engine of the imaging revolution has been the computer, bringing with it a range of techniques for digital image capture, processing, storage and transmission which would surely have startled even pioneers like John Logie Baird. The involvement of computers in imaging can be dated back to 1965, with Ivan Sutherland's Sketchpad project, which demonstrated the feasibility of computerized creation, manipulation and storage of images, though the high cost of hardware limited their use until the mid-1980s. Once computerized imaging became affordable (thanks largely to the development of a mass market for computer games), it soon penetrated into areas traditionally depending heavily on images for communication, such as engineering, architecture and medicine. Photograph libraries, art galleries and museums, too, began to see the advantages of making their collections available in electronic form. The creation of the World-Wide Web in the early 1990s, enabling users to access data in a variety of media from anywhere on the planet, has provided a further massive stimulus to the exploitation of digital images. The number of images available on the Web was recently estimated to be between 10 and 30 million [Sclaroff et al, 1997] - a figure which some observers consider to be a significant underestimate.

1.2.2 The need for image data management

The process of digitization does not in itself make image collections easier to manage. Some form of cataloguing and indexing is still necessary - the only difference being that much of the required information can now potentially be derived automatically from the images themselves. The extent to which this potential is currently being realized is discussed below.

The need for efficient storage and retrieval of images - recognized by managers of large image collections such as picture libraries and design archives for many years - was reinforced by a workshop sponsored by the USA's National Science Foundation in 1992 [Jain, 1993]. After examining the issues involved in managing visual information in some depth, the participants concluded that images were indeed likely to play an increasingly important role in electronically-mediated communication. However, significant research advances, involving collaboration between a numbers of disciplines, would be needed before image providers could take full advantage of the opportunities offered. They identified a number of critical areas where research was needed, including data representation, feature extractions and indexing, image query matching and user interfacing.

1.3 Application Domain

A wide range of possible applications for CBIR technology has been identified. Potentially fruitful areas include:

1.3.1 Crime prevention

Law enforcement agencies typically maintain large archives of visual evidence, including past suspects' facial photographs (generally known as mugshots), fingerprints, tyre treads and shoeprints. Whenever a serious crime is committed, they can compare evidence from the scene of the crime for its similarity to records in their archives. Strictly speaking, this is an example of identity rather than similarity matching, though since all such images vary naturally over time, the distinction is of little practical significance. Of more relevance is the distinction between systems designed for verifying the identity of a known individual, and those capable of searching an entire database to find the closest matching records. CBIR can do this job very efficiently.

1.3.2 Intellectual property

Trademark image registration, where a new candidate mark is compared with existing marks to ensure that there is no risk of confusion, has long been recognized as a prime application area for CBIR. Copyright protection is also a potentially important application area. Enforcing image copyright when electronic versions of the images can easily be transmitted over the Internet in a variety of formats is an increasingly difficult task. There is a growing need for copyright owners to be able to seek out and identify unauthorized copies of images, particularly if they have been altered in some way.

1.3.3 Architectural and engineering design

Architectural and engineering design share a number of common features - the use of stylized 2- and 3-D models to represent design objects, the need to visualize designs for the benefit of non-technical clients, and the need to work within externally-imposed constraints, often financial. Such constraints mean that the designer needs to be aware of previous designs, particularly if these can be adapted to the problem at hand. Hence the ability to search design archives for previous examples which are in some way similar, or meet specified suitability criteria, can be valuable.

1.3.4 Fashion and interior design

Similarities can also be observed in the design process in other fields, including fashion and interior design. Here again, the designer has to work within externally-imposed constraints, such as choice of materials. The ability to search a collection of fabrics to find a particular combination of color or texture is increasingly being recognized as a useful aid to the design process.

1.3.5 Journalism and advertising

Both newspapers and stock shot agencies maintain archives of still photographs to illustrate articles or advertising copy. These archives can often be extremely large (running into millions of images), and dauntingly expensive to maintain if detailed keyword indexing is provided. Broadcasting corporations are faced with an even bigger problem, having to deal with millions of hours of archive video footage, which are almost impossible to annotate without some degree of automatic assistance.

1.3.6 Medical diagnosis

The increasing reliance of modern medicine on diagnostic techniques such as radiology, histopathology, and computerized tomography has resulted in an explosion in the number and importance of medical images now stored by most hospitals. While the prime requirement for medical imaging systems is to be able to display images relating to a named patient, there is increasing interest in the use of CBIR techniques to aid diagnosis by identifying similar past cases.

1.3.7 Geographical information systems (GIS) and remote sensing

Although not strictly a case of image retrieval, managers responsible for planning marketing and distribution in large corporations need to be able to search by spatial attribute (e.g. to find the 10 retail outlets closest to a given warehouse). And the military are not the only group interested in analyzing satellite images. Agriculturalists and physical geographers use such images extensively, both in research and for more practical purposes, such as identifying areas where crops are diseased or lacking in nutrients - or alerting governments to farmers growing crops on land they have been paid to leave lying fallow.

1.3.8 Cultural heritage

Museums and art galleries deal in inherently visual objects. The ability to identify objects sharing some aspect of visual similarity can be useful both to researchers trying to trace historical influences, and to art lovers looking for further examples of paintings or sculptures appealing to their taste. However, many of the image queries put to art libraries are at levels 2 or 3 as defined in section 2.3 above, well beyond the capabilities of the current generation of CBIR systems.

1.3.9 Education and training

It is often difficult to identify good teaching material to illustrate key points in a lecture or self-study module. The availability of searchable collections of video clips providing examples of (say) avalanches for a lecture on mountain safety, or traffic congestion for a course on urban planning, could reduce preparation time and lead to improved teaching quality. In some cases (complex diagnostic and repair procedures) such videos might even replace a human tutor.

Reports of the application of CBIR technology to education and training have so far been sparse - though Carnegie-Mellon University's Info media system is being trialed at a number of universities, including the Open University in the UK [van der Zwan et al, 1999]. It appears to be too early to form any definite conclusions about the system's effectiveness in practice.

1.3.10 Home entertainment

Much home entertainment is image or video-based, including holiday snapshots, home videos and scenes from favorite TV programs or films. This is one of the few areas where a mass market for CBIR technology could develop. Possible applications could include management of family photo albums ('find that photo of Aunt Sue on the beach at Brighton') or clips from commercial films ('play me all the car chases from James Bond movies').

1.3.11 Web searching

Cutting across many of the above application areas is the need for effective location of both text and images on the Web, which has developed over the last five years into an indispensable source of both information and entertainment. Text-based search engines have grown rapidly in usage as the Web has expanded; the well-publicized difficulty of locating images on the Web [Jain, 1995] indicates that there is a clear need for image search tools of similar power. Paradoxically, there is also a need for software to prevent access to images which are deemed pornographic.

1.4 Architecture of CBIR

1.4.1 Selecting an Image Content Descriptor

Generally speaking, image content may include both visual and semantic content. Visual content can be very general or domain specific. General visual content include color, texture, shape, spatial relationship, etc. Domain specific visual content, like human faces, is application dependent and may involve domain knowledge. Semantic content is obtained either by textual annotation or by complex inference procedures based on visual content. This chapter concentrates on general visual contents descriptions. Later chapters discuss domain specific and semantic contents. A good visual content descriptor should be invariant to the accidental variance introduced by the imaging process (e.g., the variation of the illuminant of the scene). // However, there is a tradeoff between the invariance and the discriminative power of visual features, since a very wide class of invariance loses the ability to discriminate between essential differences. Invariant description has been largely investigated in computer vision (like object recognition), but is relatively new in image retrieval.

1.4.1.1 Colour retrieval

Several methods for retrieving images on the basis of colour similarity have been described in the literature, but most are variations on the same basic idea. Each image added to the collection is analysed to compute a colour histogram which shows the proportion of pixels of each colour within the image. The colour histogram for each image is then stored in the database. At search time, the user can either specify the desired proportion of each colour (75% olive green and 25% red, for example), or submit an example image from which a colour histogram is calculated. Either way, the matching process then retrieves those images whose colour histograms match those of the query most closely. The matching technique most commonly used, histogram intersection, was first developed by Swain and Ballard [1991]. Variants of this technique are now used in a high proportion of current CBIR systems. Methods of improving on Swain and Ballard's original technique include the use of cumulative colour histograms [Stricker and Orengo, 1995], combining histogram intersection with some element of spatial matching [Stricker and Dimai, 1996], and the use of region-based colour querying [Carson et al, 1997]. The results from some of these systems can look quite impressive.

1.4.1.2 Texture retrieval

The ability to retrieve images on the basis of texture similarity may not seem very useful. But the ability to match on texture similarity can often be useful in distinguishing between areas of images with similar colour (such as sky and sea, or leaves and grass). A variety of techniques has been used for measuring texture similarity; the best-established rely on comparing values of what are known as second-order statistics calculated from query and stored images. Essentially, these calculate the relative brightness of selected pairs of pixels from each image. From these it is possible to calculate measures of image texture such as the degree of contrast, coarseness, directionality and regularity [Tamura et al, 1978], or periodicity, directionality and randomness [Liu and Picard, 1996]. Alternative methods of texture analysis for retrieval include the use of Gabor filters [Manjunath and Ma, 1996] and fractals [Kaplan et al, 1998]. Texture queries can be formulated in a similar manner to colour queries, by selecting examples of desired textures from a palette, or by supplying an example query image. The system then retrieves images with texture measures most similar in value to the query. A recent extension of the technique is the texture thesaurus developed by Ma and Manjunath [1998], which retrieves textured regions in images on the basis of similarity to automatically-derived codewords representing important classes of texture within the collection.

1.4.1.3 Shape retrieval

The ability to retrieve by shape is perhaps the most obvious requirement at the primitive level. Unlike texture, shape is a fairly well-defined concept - and there is considerable evidence that natural objects are primarily recognized by their shape [Biederman, 1987]. A number of features characteristic of object shape (but independent of size or orientation) are computed for every object identified within each stored image. Queries are then answered by computing the same set of features for the query image, and retrieving those stored images whose features most closely match those of the query. Two main types of shape feature are commonly used - global features such as aspect ratio, circularity and moment invariants [Niblack et al, 1993] and local features such as sets of consecutive boundary segments [Mehrotra and Gary, 1995]. Alternative methods proposed for shape matching have included elastic deformation of templates (Pentland et al [1996], del Bimbo et al [1996]), comparison of directional histograms of edges extracted from the image

(Jain and Vailaya [1996], Androutsas et al [1998]), and shocks, skeletal representations of object shape that can be compared using graph matching techniques (Kimia et al [1997], Tirthapura et al [1998]). Queries to shape retrieval systems are formulated either by identifying an example image to act as the query, or as a user-drawn sketch (Hirata and Kato [1992], Chan and Kung [1997]).

1.4.2 Similarity Measures and Indexing

1.4.2.1 Similarity Measures

Instead of exact matching, content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities with the query image. Many similarity measures have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different similarity/distance measures will affect retrieval performances of an image retrieval system significantly. The most commonly used similarity measures are

- Minkowski-Form Distance
- Quadratic Form (QF) Distance
- Mahalanobis Distance
- Kullback-Leibler (KL) Divergence and Jeffrey-Divergence (JD)
- Chi square method.

1.4.2.2 Indexing Schemes

Another important issue in content-based image retrieval is effective indexing and fast searching of images based on visual features. Because the feature vectors of images tend to have high dimensionality and therefore are not well suited to traditional indexing structures, dimension reduction is usually used before setting up an efficient indexing scheme.

1.4.3 User Interaction

For content-based image retrieval, user interaction with the retrieval system is crucial since flexible formation and modification of queries can only be obtained

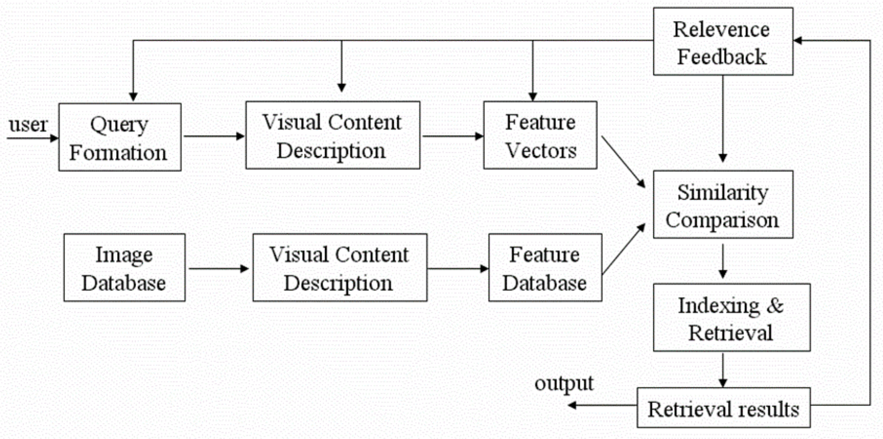


FIGURE 1.1: Diagram for content-based image retrieval system.

by involving the user in the retrieval procedure. User interfaces in image retrieval systems typically consist of a query formulation part and a result presentation part.

1.4.3.1 Query Specification

Specifying what kind of images a user wishes to retrieve from the database can be done in many ways. Commonly used query formations are: category browsing, query by concept, query by sketch, and query by example. Category browsing is to browse through the database according to the category of the image. For this purpose, images in the database are classified into different categories according to their semantic or visual content. Query by concept is to retrieve images according to the conceptual description associated with each image in the database. Query by sketch and query by example is to draw a sketch or provide an example image from which images with similar visual features will be extracted from the database.

1.4.3.2 Relevance Feedback

Relevance feedback is a supervised active learning technique used to improve the effectiveness of information systems. The main idea is to use positive and negative examples from the user to improve system performance. For a given query, the system first retrieves a list of ranked images according to a predefined similarity metrics. Then, the user marks the retrieved images as relevant (positive examples) to the query or not relevant (negative examples). The system will refine the retrieval results based on the feedback and present a new list of images to the user. Hence, the key issue in relevance feedback is how to incorporate positive and negative examples to refine the query and/or to adjust the similarity measure.

1.4.4 Performance Evaluation

To evaluate the performance of retrieval system, two measurements, namely, recall and precision are borrowed from traditional information retrieval. For a query q, the data set of images in the database that are relevant to the query q is denoted as R(q), and the retrieval result of the query q is denoted as Q(q). The precision of the retrieval is defined as the fraction of the retrieved images that are indeed relevant for the query (1.1)

$$precision = \frac{\left|Q\left(q\right) \bigcap R\left(q\right)\right|}{\left|Q\left(q\right)\right|} \tag{1.1}$$

The recall is the fraction of relevant images that is returned by the query (1.2)

$$recall = \frac{\left|Q\left(q\right) \bigcap R\left(q\right)\right|}{\left|R\left(q\right)\right|} \tag{1.2}$$

Usually, a tradeoff must be made between these two measures since improving one will sacrifice the other. In typical retrieval systems, recall tends to increase as the number of retrieved items increases; while at the same time the precision is likely to decrease. In addition, selecting a relevant data set R (q) is much less stable due to various interpretations of the images. Further, when the number of relevant images is greater than the number of the retrieved images, recall is meaningless. As a result, precision and recall are only rough descriptions of the performance of the retrieval system.

1.5 Challenges

The challenges faced today by the researchers in the field of face recognition is to design a system which can classify images in an uncontrolled environment with a high accuracy. In real world there may be same image with different illumination, viewing angle, rotation etc. Moreover every system adds some noise in the image while they are being taken. Objects in the image can be in occluded form which is difficult to differentiate. So an ideal feature representation should be stable in all these environment and have high discrimination information which minimized the within class variations while maximizing the between class variation. A descriptor which is robust in changing environment but requires a large time for classification is not suitable and is of little use in real time systems. On the other hand discarding discriminatory information in order to reduce the feature size would reduce the accuracy and reliability of the system. So the major challenge is to extract an image representation which is robust in uncontrolled environment and at the same time having a small size while still holding the essential discriminatory information.

1.6 Objective

Our goal was to design an efficient image content descriptor which will be

- Invariant to rotation and scaling
- Robust in Illumination changes
- Able to give better results for noisy images
- Can perform better in case of occlusion
- Computationally Efficient
- Can give better performance than existing methods

1.7 Research Contribution

In this field of research, we have introduced a new color image feature descriptor. We have combined the color and texture information in our method to get better result than the existing methods. Our proposed method considers the three color components at once while calculating the feature vector. This combine each color plane along with the texture information in the descriptor but do not increase the feature vector size.

Chapter 2

Literature Review

2.1 Feature Extraction

2.1.1 Color histograms

Color histograms are widely used in image indexing. A color histogram of an image describes the frequency of each color level in the image in pixel domain. To extract a histogram, an image is quantized into n sets of colors $C = c_1, ..., c_n$ if necessary. A histogram H is a vector $H = (h_1, ..., h_n)$, with each bin $h_i(1 \leq i \leq n)$ as the frequency in color c_i. An example in Figure 2.1 shows an image, the corresponding grey-level histogram, and one-dimensional histograms in the R, G, and B channels.

The color histogram is easy to compute and effective in characterizing both the global and local distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle.
Since any pixel in the image can be described by three components in a certain color space (for instance, red, green, and blue components in RGB space, or hue, saturation, and value in HSV space), a histogram, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains, the more discrimination power it has.

Limitations of Color Histogram
There are however, several difficulties associated with the color histogram (CH). CH is sensitive to noisy interferences such as illumination changes and quantization errors. Moreover large dimension of CH involves large computation on indexing.

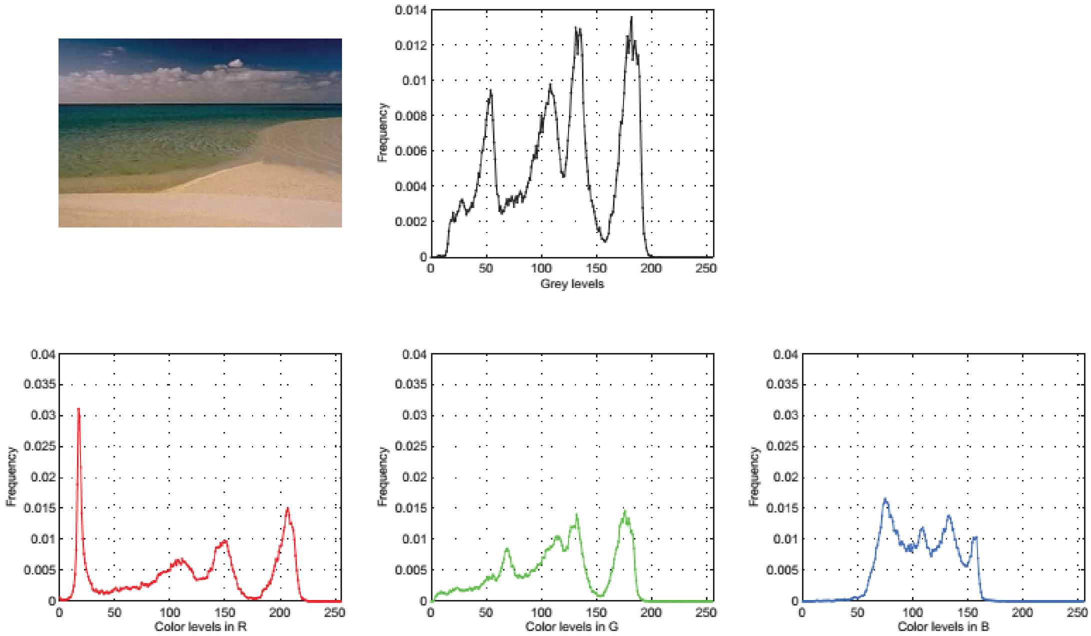


FIGURE 2.1: An image, its grey level histogram, and the histograms in R, G, and B.

It does not take into consideration color similarity across different bins and Information about object location, shape, and texture is discarded. Image retrieved by using global color histogram may not be semantically related even though they share similar color distribution.

2.1.2 Dominant Color Descriptor

The Dominant Color Descriptor (DCD) provides a compact description of the representative colors in an image or image region. Its main target applications are similarity retrieval in image databases and browsing of image databases based on single or several color values. Unlike the traditional histogram based descriptors, the representative colors are computed from each image instead of being fixed in the color space, thus allowing the color representation to be accurate and compact.

Dominant color descriptor extracts the features from an image by clustering the colors in an image into a small number of colors and is defined as

$$F = \{c_i, p_i\}, (i = 1, 2, ..., N)$$

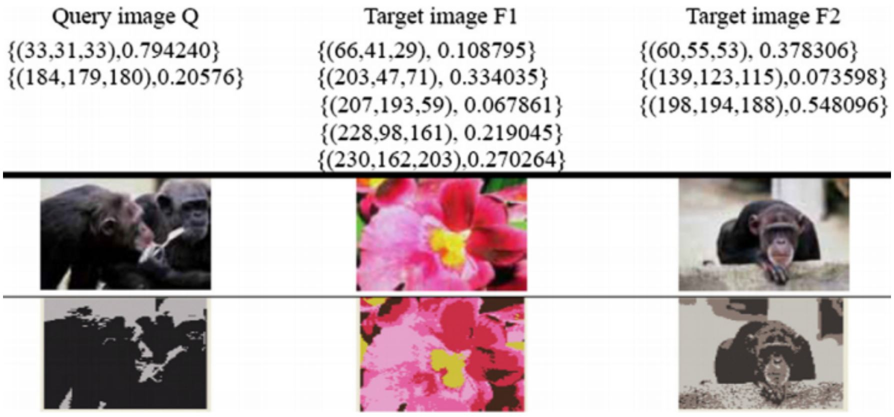


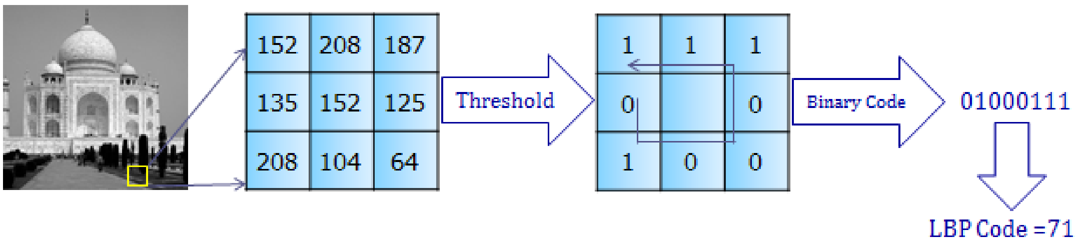
FIGURE 2.2: Examples with the dominant colors and their percentage values.

The descriptor consists of the representative colors ci and their percentages pi. The quadratic histogram distance measure (QHDM) is used for similarity measure for DCD. With this simple and compact representation, DCD allows efficient indexing for similarity retrieval while sacrificing retrieval accuracy due to lack of spatial information of the description compared to other color descriptors.
For dominant color extraction, the generalized Lloyd algorithm described in is used for color clustering.

2.1.3 Local Binary Pattern

Ojala et al. introduced the Local Binary Pattern operator in 1996 as a means of summarizing local gray-level structure. The operator takes a local neighborhood around each pixel, thresholds the pixels of the neighborhood at the value of the central pixel and uses the resulting binary-valued image patch as a local image descriptor. It was originally defined for 33 neighborhoods, giving 8 bit codes based on the 8 pixels around the central one.

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P\text{-}1} s(i_p - i_c)2^P \qquad s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



152	208	187
135	152	125
208	104	64

1	1	1
0		0
1	0	0

FIGURE 2.3: Calculation procedure of LBP (here, P=8).

LBP operates with p neighboring pixels using the center as a threshold. The final LBP code is then produced by multiplying the thresholded values by weights given by powers of 2 and adding the results in a way described in the figure 2.3. Where, i_c is the gray value of the center pixel, i_p is the gray value of its neighbors P is the number of neighbors and R is the radius of the neighborhood.

Advantages

- It is invariant to monotonic changes in gray-scale.
- Fast to calculate.

Limitation

- Poor performance for noisy image and large change in illumination.

2.1.4 Local Triplet Pattern (LTP)

The LTP(He and Cercone (2009)) feature of an image is a histogram which contains spatial information among neighboring pixels in the image. An LTP level is extracted from each 3 3 pixel block. The color levels of the eight surrounding pixels are compared with the color level of the center pixel. The comparison returns one of the triplet codes: 0, 1, or 2 to represent the three conditions:
The color level of a neighboring pixel is smaller than, equal to, or larger than the color level of the center pixel. The eight triplet codes from the eight surrounding pixels are then transformed to an LTP level.

$$comp_{LTP}(a,b) = \begin{cases} 2,\ a > b; \\ 1,\ a = b; \\ 0,\ a < b. \end{cases} \qquad ltp = \sum_{i=0}^{7} comp_{LTP}(C_i, C) \times 3^i$$

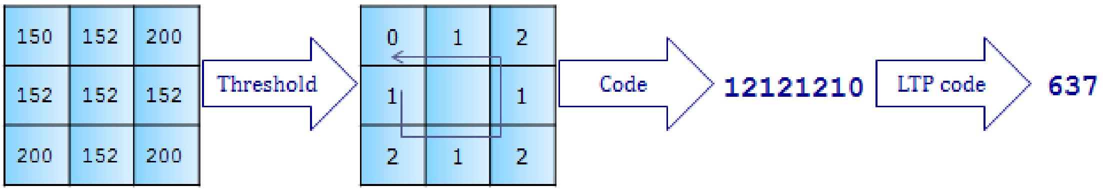


Figure 2.4: Calculation procedure of LTP.

After LTP levels are generated from each 3 3 pixel block in an image, an LTP histogram is extracted as the feature. The LTP histogram is a vector T = $t_0, ..., t_6560$, where t_i is the number of pixels whose LTP level is equal to i in the image. The function *compLTP* puts two consequences on the LTP histogram feature. The first is that the LTP histogram is very sparse. The function *compLTP* results 1 if and only if two color levels are the same. Natural images, however, usually do not contain color blocks with all the same color levels. Patterns with 1 (equal) in the triplet codes are always far less frequent than patterns with 0 (less) and 2 (larger). A scaling parameter S to the LTP feature is introduced in this method. An image is scaled from all original color levels into a quantized color space with fewer color levels, which implies all color levels are clustered into several groups. The number of the groups is the scaling parameter S. After the scaling operation, compLTP (a, b) returns 0 if the color levels a and b belong to the same group, which is very close to how humans perceive the same situation.

The second consequence is that the LTP feature size (6561) is much larger than the LBP feature size (256). In order to provide more ef?cient and ?exible features, a neighboring parameter N is introduced into the LTP feature. The parameter N is the number of the neighboring pixels which are included to generate an LTP level. N is an integer between 1 and 8. The size of an LTP histogram with a neighboring parameter N is 3N. N is a tradeoff between feature capacity and ef?ciency. In order to cover all the possible combination between pixel pairs, He and Cercone suggested that N is at least 4. Under the conditions that ef?ciency permits, a larger N is better.

Although N neighboring color levels are taken, LTP still captures most joint information of a color level with the other neighboring color levels. For example when N is 4, the LTP level is calculated from the center pixel C and neighboring pixels C0 to C3 . The pixels C4 and C7 are not included. However, when the C4 is the center pixel, the previous center pixel C is the neighboring C0 in this block. Thus, N neighboring color levels capture major patterns in an image.

Advantages

- It is invariant to monotonic changes in gray-scale.
- Gives better result than LBP.

Limitation

- The histogram is too sparse and large to be effectively applied and to provide efficient retrievals.
- Poor performance if the image has noise change in illumination.

2.1.5 Directional Local Extrema Patterns (DLEP)

DLEP extracts the directional edge information based on local extrema in 0°,45 deg,90 deg, and 135 deg directions in an image. In proposed DLEP for a given image the local extrema in 0?,45?,90?, and 135? directions are obtained by computing local difference between the center pixel and its neighbors as shown below:

$$I'(g_i) = I(g_c) - I(g_i); i = 1, 2, 3, ..., 8 \tag{2.1}$$

The local extremas are obtained by the following equation

$$\hat{I_\alpha}(g_c) = f_3(I'(g_j), I'(g_{j+4})); j = (1 + \alpha/45) \tag{2.2}$$

$$\forall\alpha = 0°, 45°, 90°, 135°$$

$$f_3(I'(g_j), I'(g_{j+4})) = \begin{cases} 1 & \text{if } I'(g_j) \times I'(g_{j+4}) \geq 0 \\ 0 & \text{else} \end{cases} \tag{2.3}$$

The DLEP is defined ($\alpha = 0°, 45°, 90°, 135°$) as follows:

$$DLEP(I(g_c))|_\alpha = \left\{\hat{I_\alpha}(g_c); \hat{I_\alpha}(g_1); \hat{I_\alpha}(g_2); ...\hat{I_\alpha}(g_8)\right\} \tag{2.4}$$

The detailed representation of DLEP can be seen in the figure 2.5.

Eventually, the given image is converted to DLEP images with values ranging from 0 to 511. After calculation of DLEP, the whole image is represented by building a histogram supported by the following equation

$$H_{DLEP|_\alpha}(l) = \sum_{j=1}^{N_1}\sum_{k=1}^{N_2} f_2(DLEP(j,k)|_\alpha, l); l\varepsilon[0, 511] \tag{2.5}$$

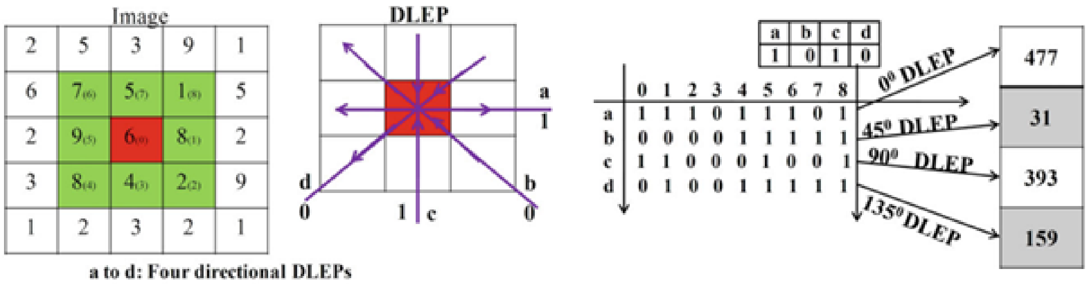


Figure 2.5: Example of obtaining DLEP for the 3*3 pattern.

Advantages

- Directional features are very valuable for image retrieval applications.
- DLEP captures more spatial information as compared with LBP.
- It is invariant to monotonic changes in gray-scale and fast to calculate.

Limitation

- Poor performance if the image has Noise and large change in illumination.

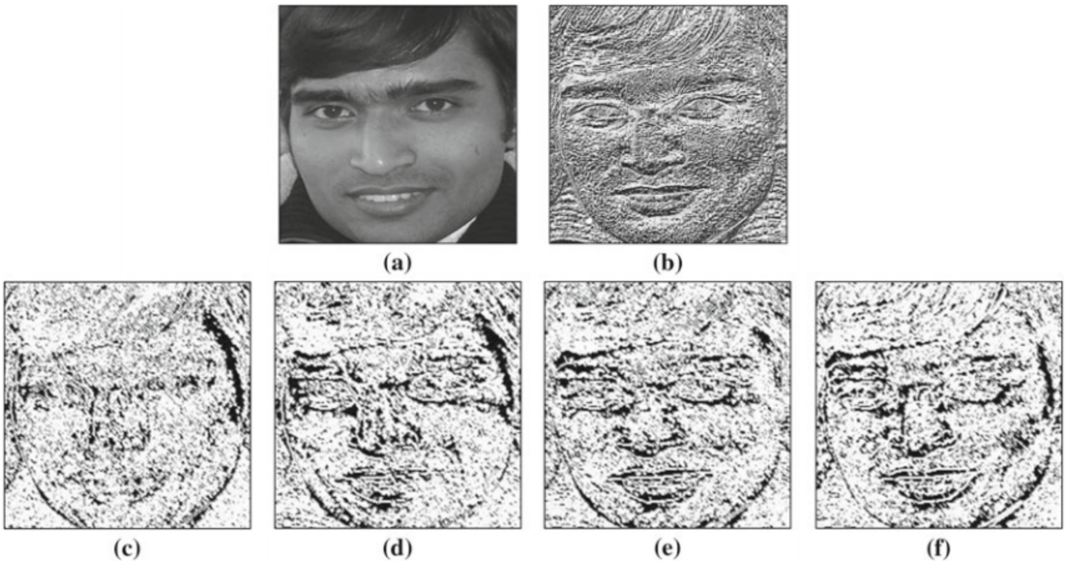


Figure 2.6: Example of LBP and DLEP feature maps, a sample image LBP feature map, c DLEP feature map in 0° direction, d in 45° direction, e in 90° direction, and f in 135° direction.

Chapter 3

Proposed Method

3.1 Overview

As we have seen that most of the methods works on gray image. This causes a loss of color information. There are some methods that work with color but they consider each plane (RGB) separately. As a result size of the feature vector becomes large. Now in this section we propose a method that considers both texture and color information of an image and works on each color plane at the same time to reduce the feature vector size.

3.2 Proposed Image Descriptor

An RGB color image is an $M \times N \times 3$ array of color pixels , where each color pixel is a triplet corresponding to the red, green, and blue components of an RGB image at a specific spatial location (see figure 3.1).

In RGB model, each color appears in its primary spectral components of red green and blue which is based on a Cartesian coordinate system. Figure 3.2 shows the schematic of the RGB color cube.

In our proposed method at first a 3×3 neighborhood is defined as a cell around each pixel of a color image. Then we find the dissimilarity of the center pixel with all other neighboring pixels. When determining the dissimilarity we considered each of the three color components (R, G and B) of the image. To find the dissimilarity we have used Euclidean Dissimilarly method (3.1). This method gives the dissimilarity between two pixels using the three color components at

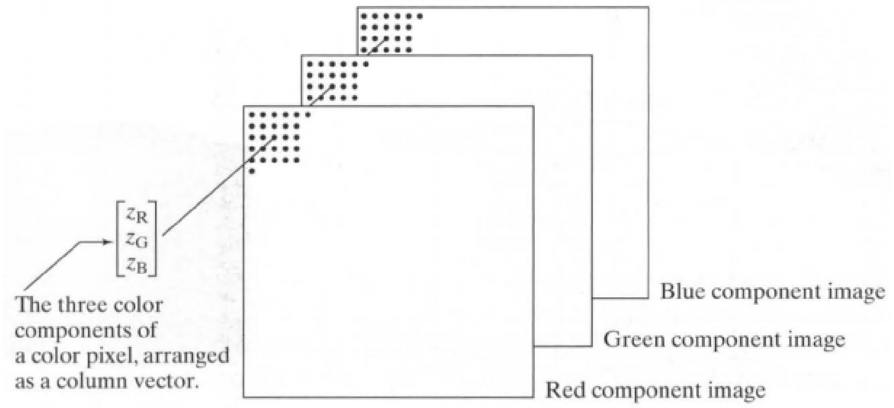

$\begin{bmatrix} z_R \\ z_G \\ z_B \end{bmatrix}$
The three color components of a color pixel, arranged as a column vector.
Blue component image
Green component image
Red component image

FIGURE 3.1: Schematic showing how pixels of an RGB color image is formed from the corresponding pixels of the three component images.

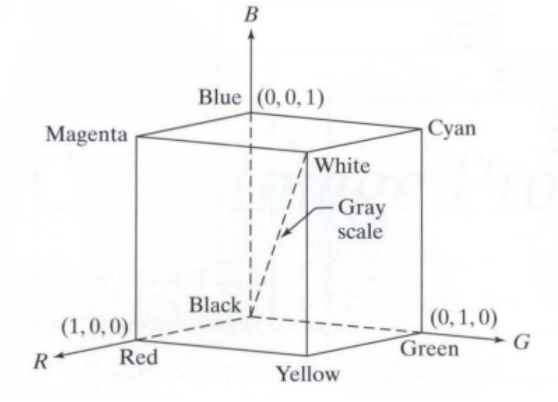


FIGURE 3.2: Schematic of the RGB color cube.

once. Unlike most other methods our method didn't require to calculate the dissimilarity for each of the color planes separately. This helps us to keep all color information in our feature descriptor along with less calculation.

$$dissimilarity = \sqrt{(red_c - red)^2 + (green_c - green)^2 + (blue_c - blue)^2} \qquad (3.1)$$

In our next step we replace the dissimilarity values with either 0 or 1 depending on a threshold value T using the function g(x) in equation 3.2.

$$g\left(x\right) = \begin{cases} 1 & \text{if } distance \leq T \\ 0 & \text{if } distance > T \end{cases} \qquad (3.2)$$

The threshold T is determined by inspection (Figure 3.5). Some visually uniform colored regions were considered and we calculated the intensity difference between

240, 184, 134	144, 64, 16	140, 65, 20
240, 179, 130	235, 179, 131	234, 185, 131
140, 60, 29	143, 65, 20	236, 180, 131

(a)

7.86	186.36	185.32
5.10		6.08
183.27	183.80	1.41

(b)

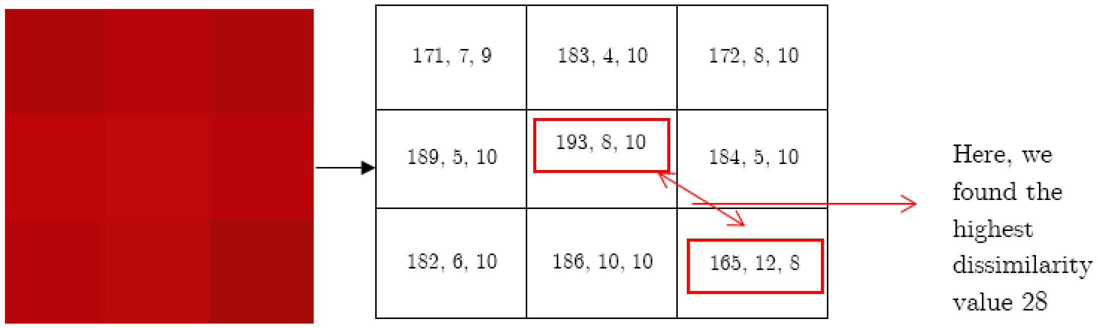
Figure 3.3: (a) 3 × 3 neighborhood. (b) Dissimilarity measure using equation (3.1).

7.86	186.36	185.32
5.10	C	6.08
183.27	183.80	1.41

1	0	0
1	C	1
0	0	1

Figure 3.4: Replacing the dissimilarity values with either 0 or 1 using equation 3.2

the center pixel and other neighboring pixels. We take the highest difference between two neighboring visually uniform colored pixels as the threshold T. Any dissimilarity value below or equal to this threshold is considered as similar colored region.



171, 7, 9	183, 4, 10	172, 8, 10
189, 5, 10	193, 8, 10	184, 5, 10
182, 6, 10	186, 10, 10	165, 12, 8

Here, we found the highest dissimilarity value 28

Figure 3.5: Calculation process of Threshold T.

In our observation we have found that the highest value is 28. We have experimented with other values around 28 and found the best result for threshold value 35.

Alternatively To find the dissimilarity we have used Cosine Similarly method. Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude.

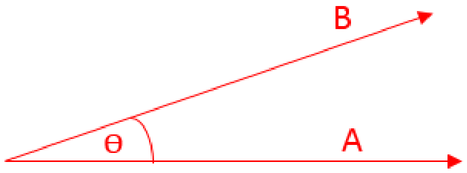


Figure 3.6: Two vectors A and B, θ is the angel between them.

Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0, 1]. Note that these bounds apply for any number of dimensions, and Cosine similarity is most commonly used in high-dimensional positive spaces. Here as we are using an image with 3 dimensions (R, G, B) we can use this method to find the dissimilarity between two pixels. To perform this task we can use equation (3.3).

$$similarity = \cos\left(\Theta\right) = \frac{A.B}{|A|\,|B|} = \frac{\sum_{i=0}^{n} A_i \times B_i}{\sqrt{\sum_{i=0}^{n}\left(A_i\right)^2} \times \sqrt{\sum_{i=0}^{n}\left(B_i\right)^2}} \tag{3.3}$$

Here, $A = [R_c G_c B_c]$ and $B = [RGB]$

In equation 3.3 vector A holds the Red, Green and Blue components of the center pixel and vector B holds the three components of the neighboring pixel.
Now we need a threshold value T up to which we can consider two pixels as similar. Like previous method the threshold T is determined by inspection. We have considered two pixels as same with a maximum orientation variation of 30°.

Like previous method we replace the dissimilarity values with either 0 or 1 depending on the threshold value $T = cos(30°)$ using the equation 3.2.

The replacement of the three component pixel values with either 0 or 1 depending on the threshold value T, gives a 3×3 neighborhood with only binary values.

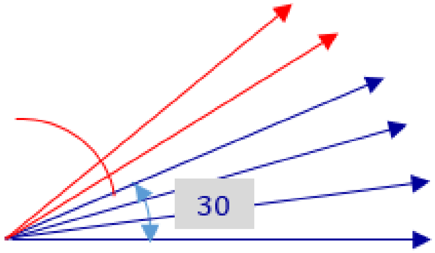
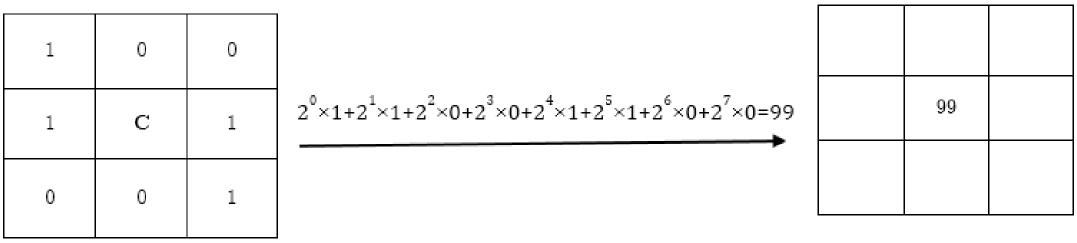


FIGURE 3.7: Pixel similarity depending on Threshold

From this binary value we calculate the decimal value. We take the lowest possible decimal value to make it rotation invariant and place it in central pixel C.



1	0	0
1	C	1
0	0	1

$$2^0\times1+2^1\times1+2^2\times0+2^3\times0+2^4\times1+2^5\times1+2^6\times0+2^7\times0=99$$

	99	

FIGURE 3.8: Calculation process of final central pixel value.

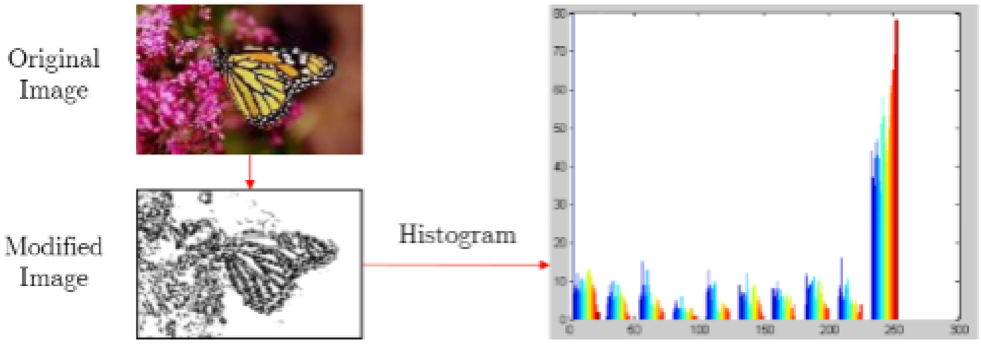


FIGURE 3.9: Feature vector generation.

Then we shift the 3 × 3 window to the next pixel and do the same. When the window completes traversing the whole image, each pixel contains a decimal value within 0 to 255 (2^8).

Now we generate a histogram of our modified image. This histogram is our final feature vector.

3.3 Image Classification

Applying our proposed method on an image we get the feature vector for that image. Thus we find the feature vector of all the images in a particular class of training images. Then we average these values to generate the average feature vector for that particular class of images. This average feature vector is stored for classifying any new image.

For classifying a new image we first compute its feature vector and then compare it with the stored average feature vectors of all the classes. Two feature vectors are compared using Chi-square dissimilarity method. The Chi-square dissimilarity is measured using the formula in equation 3.4.

$$x^2 = \sum \frac{(H_1 - H_2)^2}{(H_1 + H_2)} \tag{3.4}$$

The unknown image is assigned a class label among the known images whose feature vector gives the minimum dissimilarity with the feature vector of the unknown image.

3.4 Image Retrieval

If we want to find images similar to a given query image, we can do it in two ways. We can compare feature vector of all images in the database with the feature vector of query image using Chi-square dissimilarity method. This will give all the similar images in a ranked order of similarity. Since we are comparing all images in the database, it take more time.

Alternately we can first classify the image and then compare with the images of that class only. This will save comparing time. Since our method can classify images quite efficiently so there is less chance of getting irrelevant images.

While retrieving images, our method put those images first which are similar in color along with similarity of objects in the images.

3.5 Properties of our Descriptor

Here we discuss the good properties of our proposed image descriptor. For each pixel the proposed descriptor contains information about neighboring pixels along with each of the three color component information. Some properties of the method are:
The use of cells captures local information around every pixel and calculating the dissimilarity between neighboring cells captures texture information between image regions. Thus the pattern generated for every pixel generated by our method contains both local information as well as more global structure and this make it more robust to pose and expression variation and occlusion.

Use of Euclidean distance for finding the dissimilarity between the pixels reduces the size of the feature vector while considering the all the three components of color.
The dissimilarity response for a threshold value T helps us to generate a LBP code which makes our method rotation invariant.

Chapter 4

Experimental Result

4.1 Experimental Setup

We have used Corel dataset for the implementation purpose. It's a huge dataset with different types of images. It contains 10,800 images. These images are divided into 80 categories. We took 50 categories of them each containing 100 images e.g., autumn, aviation, bonsai, castle, cloud, dog, elephant, iceberg, primates, ship, steam-engine, tiger, train, and waterfall and so on. The images in a category are taken from a different viewpoint, different illumination conditions and also from different objects of similar types.

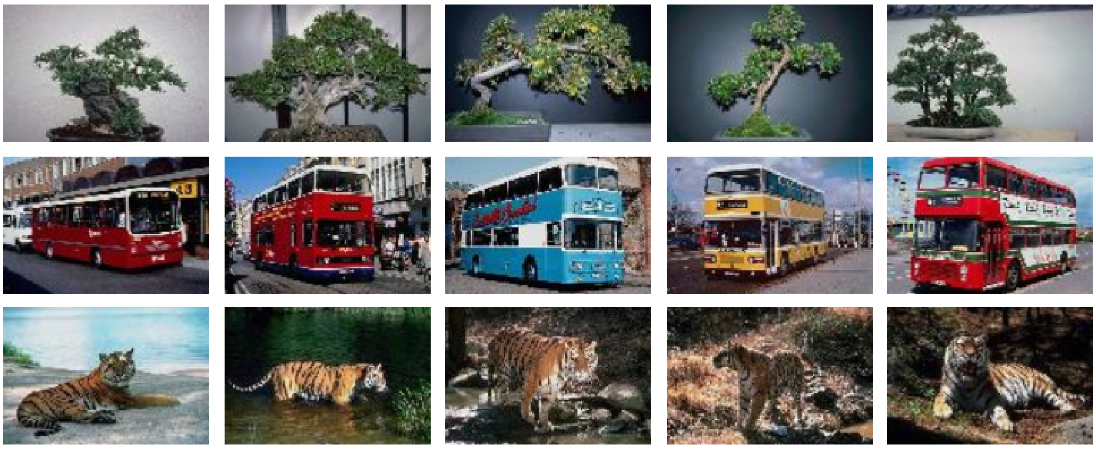
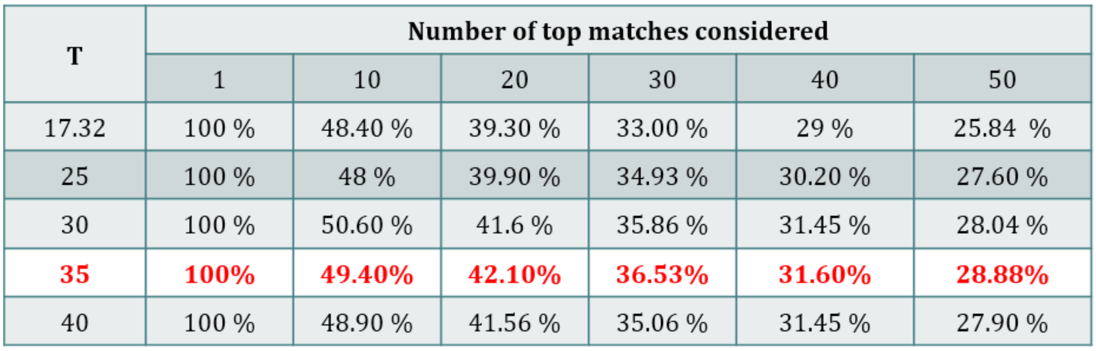


FIGURE 4.1: Example of images from corel database. Only images from bonsai, bus and tiger categories are shown.

4.2 Parameter Tuning

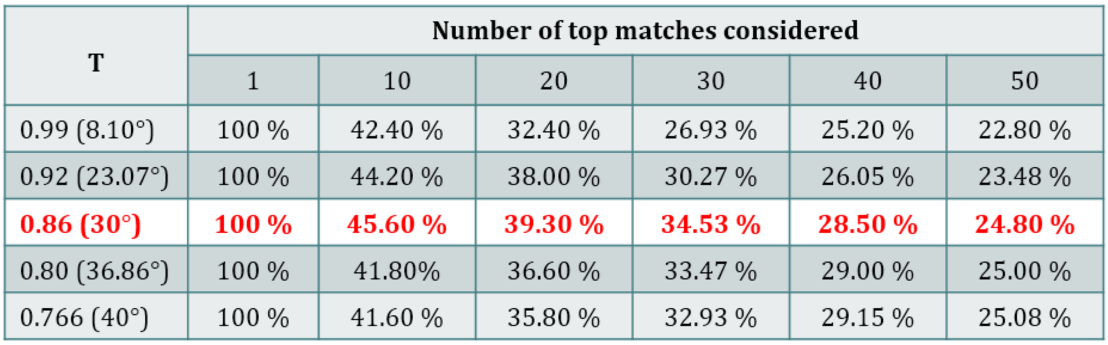
We examined the behavior of our system using different threshold values. Figure 4.2 shows the performance analysis of our system for different threshold values. Undoubtedly threshold 35 gave the best result among them when using Euclidean Distance method.



T	Number of top matches considered					
	1	10	20	30	40	50
17.32	100 %	48.40 %	39.30 %	33.00 %	29 %	25.84 %
25	100 %	48 %	39.90 %	34.93 %	30.20 %	27.60 %
30	100 %	50.60 %	41.6 %	35.86 %	31.45 %	28.04 %
35	**100%**	**49.40%**	**42.10%**	**36.53%**	**31.60%**	**28.88%**
40	100 %	48.90 %	41.56 %	35.06 %	31.45 %	27.90 %

FIGURE 4.2: Parameter tuning for Euclidean Distance method.

In case of Cosine Similarity Method we get the best result for $T = Cos(30°) = 0.86$



T	Number of top matches considered					
	1	10	20	30	40	50
0.99 (8.10°)	100 %	42.40 %	32.40 %	26.93 %	25.20 %	22.80 %
0.92 (23.07°)	100 %	44.20 %	38.00 %	30.27 %	26.05 %	23.48 %
0.86 (30°)	**100 %**	**45.60 %**	**39.30 %**	**34.53 %**	**28.50 %**	**24.80 %**
0.80 (36.86°)	100 %	41.80%	36.60 %	33.47 %	29.00 %	25.00 %
0.766 (40°)	100 %	41.60 %	35.80 %	32.93 %	29.15 %	25.08 %

FIGURE 4.3: Parameter tuning for Cosine Similarity method.

4.3 Example of Image Retrieval

An example of the image retrieval process (Figure 4.4) is shown the image. In this example the tick mark represents correct retrieval and cross mark represents wrong retrieval. The first five retrieved images were shown only.

4.4 Result Comparison

The retrieval efficiency is measured by calculating the precision rate (4.1)

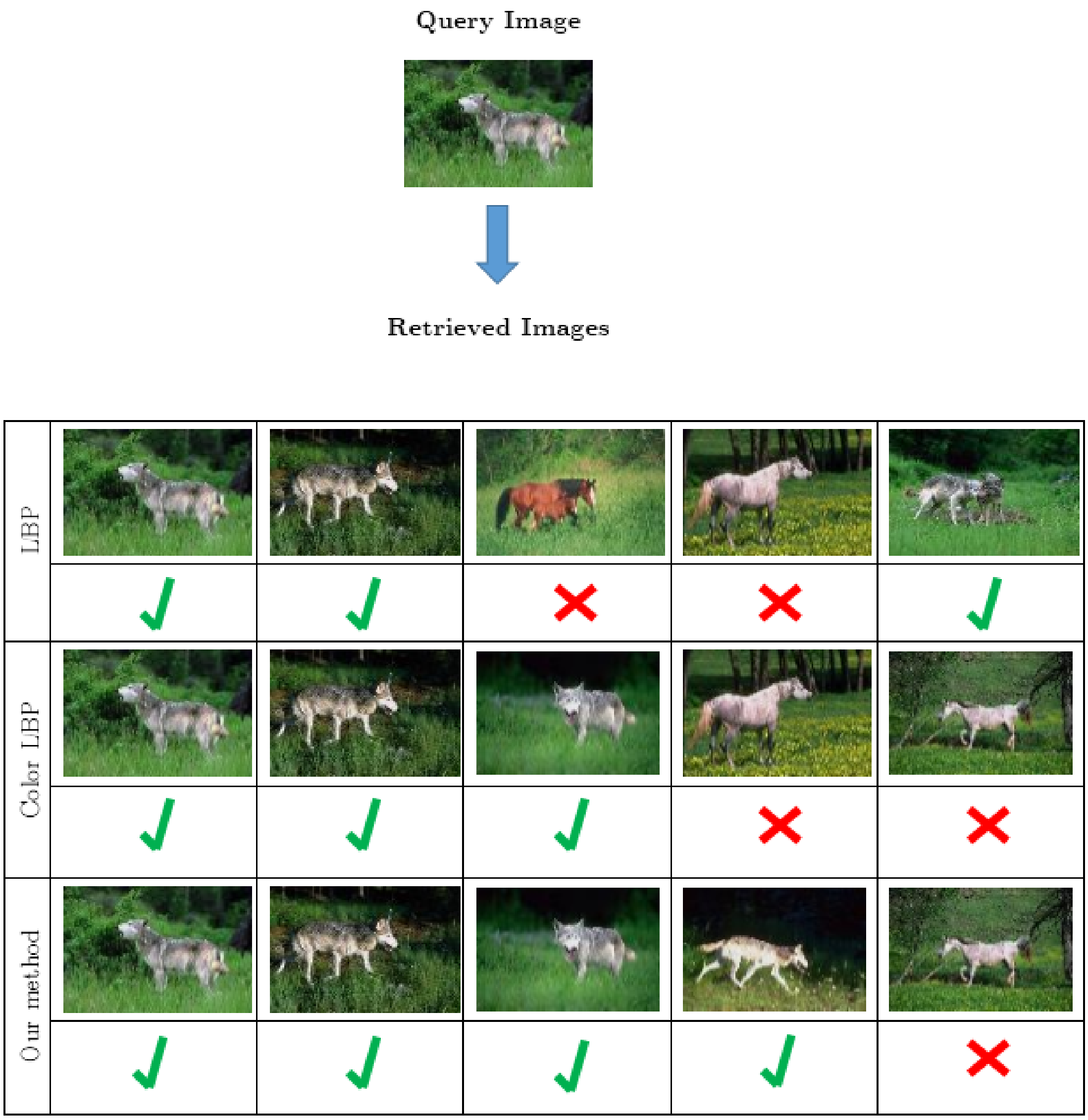
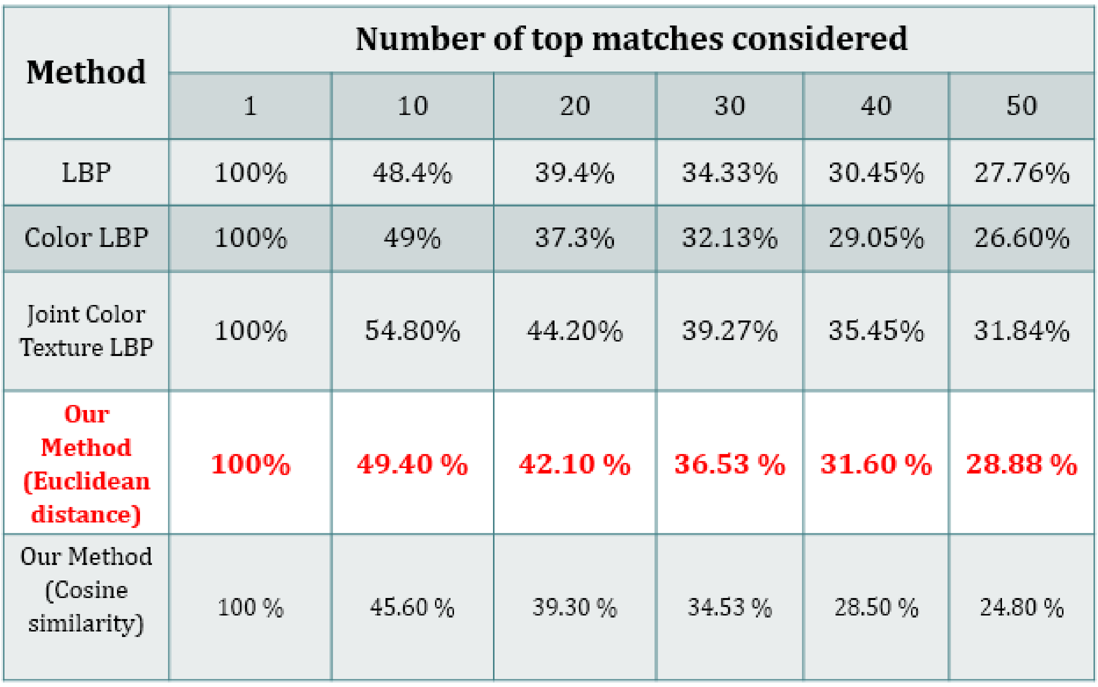


FIGURE 4.4: Example of image retrieval of LBP, Color LBP and our method.

$$\text{Precision rate} = \frac{\text{Number of relavent images retrieved}}{\text{Number of images retrieved}} \tag{4.1}$$

We compared our method with some of the existing methods and found the result shown in table 2. We have considered both texture based and color based methods. From the result we have found that our method performs better than the LBP and Color LBP method.

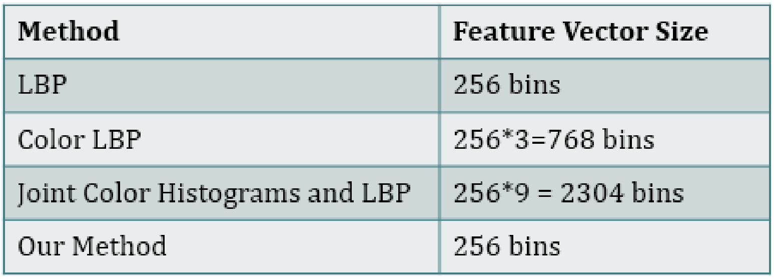


Method	Number of top matches considered					
	1	10	20	30	40	50
LBP	100%	48.4%	39.4%	34.33%	30.45%	27.76%
Color LBP	100%	49%	37.3%	32.13%	29.05%	26.60%
Joint Color Texture LBP	100%	54.80%	44.20%	39.27%	35.45%	31.84%
Our Method (Euclidean distance)	**100%**	**49.40 %**	**42.10 %**	**36.53 %**	**31.60 %**	**28.88 %**
Our Method (Cosine similarity)	100 %	45.60 %	39.30 %	34.53 %	28.50 %	24.80 %

FIGURE 4.5: Performance Comparison.

4.5 Observation

Our method and LBP has the same feature vector size (256 bins) but it contains additional color information and hence performs better.
Color LBP has three times more feature vector size (256*3=768) than our method but still our method performs better.



Method	Feature Vector Size
LBP	256 bins
Color LBP	256*3=768 bins
Joint Color Histograms and LBP	256*9 = 2304 bins
Our Method	256 bins

FIGURE 4.6: Feature vector size comparison of different methods.

And finally the Joint Color Texture LBP has a feature size of 256*9 = 2304 bins which is much higher than ours (256) and also it requires a huge runtime. With low feature vector size and low time complexity our method performs considerably well.

Chapter 5

Conclusion

Color based operators are more powerful than texture operators based on gray scale information in stable illumination conditions.

At the same time, texture features especially the Local Binary Pattern (LBP) distributions provide fairly robust performance irrespective of illumination.

This thesis proposes a new color image descriptor which is robust in illumination changes, Invariant to rotation and computationally efficient.

As a further processing step we want to apply this algorithm using a 55 window. But this will increase feature vector size. So, we will apply a dimension reduction method to make our method computationally efficient.

Chapter 6

Bibliography

[1] Swain M., Ballard D.: Color Indexing. In: Third International Conference on Computer Vision. (1990) 11-32

[2] Manjunath B. S., Ohm J.-R., Vasudevan V., Yamada A.: Color and Texture Descriptors. IEEE Transactions on Circuits and Systems for Video Technology 11 (2001) 703-715

[3] 3. Ojala T., Pietikainen M., Hardwood D.: A Comparative Study of Texture Measures with Classification Based on Feature Distribution. Pattern Recognition 29 (1996) 51-59

[4] Valtteri Takala, Timo Ahonen, and Matti Pietikainen: Block-Based Methods for Image Retrieval Using Local Binary Patterns. SCIA 2005, LNCS 3540, pp. 882-891, 2005.

[5] Daan He, Nick Cercone: Local Triplet Pattern for Content-Based Image Retrieval. ICIAR 2009, LNCS 5627, pp. 229-238, 2009.

[6] Subrahmanyam Murala, R. P. Maheshwari, R. Balasubramanian: Directional local extrema patterns: a new descriptor for content based image retrieval. International Journal of Multimedia Information Retrieval 2012.

[7] Huang J., Kumar S. R., Mitra M., Zhu W.-J., Zabih R.: Image Indexing Using Color Correlograms. In: IEEE Conference on Computer Vision and Pattern Recognition (1997) 762-768

[8] Jae Young Choi, Konstantinos N. Plataniotis, Yong Man Ro: Using colour local binary pattern features for face recognition, ICIP 2010

[9] Matti Pietikinen, Topi Maenpaa and Jaakko Viertola: Color Texture Classifcation with Color Histograms and Local Binary Patterns. (2002)

[10] Topi Maenpaa, Matti Pietikinen and Jaakko Viertola : Separating Color and Pattern Information for Color Texture Discrimination (2002)

[11] Darshak G. Thakore, A. I. Trivedi : Content based image retrieval techniques Issues, analysis and the state of the art.

[12] Neetu Sharma, Paresh Rawat, and jaikaran Singh : Efficient CBIR Using Color Histogram Processing (2011)

[13] Topi Maenpaa, Matti Pietikinen : Classification with color and texture: jointly or separately? (2004)

[14] Dong Kwon Park, Yoon Seok Jeon, Chee Sun Won : Efficient Use of Local Edge Histogram Descriptor

[15] SHI Dongcheng, XU Lan, HAN Ungyan : Image retrieval using both color and texture features (2007)

[16] Ricardo da Silva Torres, Alexandre Xavier Falco : Content-Based Image Retrieval: Theory and Applications (2006)

[17] Subrahmanyam Murala, R. P. Maheshwari, Member, IEEE, and R. Balasubramanian, Member, IEEE : Local Tetra Patterns: A New Feature Descriptor for

Content-Based Image Retrieval (2012)

[18] C.-H. Lin, C.-W. Liu, H.-Y. Chen : Image retrieval and classification using adaptive local binary patterns based on texture features (2012)

[19] Xiaoli Yuan, Jing Yu, Zengchang Qin, and Tao Wan : A SIFT-LBP image retrieval model based on bag-of-features (2011)

[20] Mohammad Saberi, Farshad Tajeripour, and Shervan Fekri-Ershad : Content Based Image Retrieval Approach Based on Top-Hat Transform And Modified Local Binary Patterns (2012)

[21] P.V. N. Reddy and K. Satya Prasad : Inter Color Local Ternary Patterns for Image Indexing and Retrieval [ICACCT-2011]