COLOUR-BASED IMAGE RETRIEVAL ALGORITHMS BASED ON COMPACT COLOUR DESCRIPTORS AND DOMINANT COLOUR-BASED INDEXING METHODS

AHMED TALIB ABDULAMEER

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2014

Permission to Use

Abstrak

Capaian imej berasaskan kandungan (CBIR) dilaporkan sebagai salah satu bidang penyelidikan yang paling aktif dalam dua dekad lalu. Tiga masalah prestasi CBIR ialah ketidaktepatan dapatan semula imej, kerumitan yang tinggi ciri pengekstrakan, dan penurunan semula imej selepas pengindeksan pangkalan data, menyebabkan CBIR tidak sesuai digunakan pada peranti mudah alih. Objektif utama tesis ini adalah untuk meningkatkan prestasi CBIR. Untuk mencapai objektif ini, tiga kaedah telah digunakan. Kaedah pertama menggunakan imej warna dominan (DC) dipilih sebagai penyumbang utama untuk tujuan ini kerana ciri yang kompak dan keserasian dengan sistem visual manusia. Capaian semula imej berasaskan semantik adalah dicadangkan untuk menyelesaikan masalah capaian yang tidak tepat dengan menumpukan pada objek imej. Kesan latar belakang imej dikurangkan untuk memberi tumpuan lebih kepada objek dengan memberikan pemberat untuk objek dan latar belakang DC. Nisbah peningkatan ketepatan ditingkatkan berbanding kaedah yang dibandingkan. Rangka kerja DC pemberat adalah dicadangkan untuk mengitlak teknik ini di mana ianya ditunjukkan dengan menggunakannya pada perihalan warna. Manakala untuk mengurangkan kerumitan yang tinggi pada warna Correlogram daripada segi pengiraan dan ruang memori, kaedah kedua perwakilan padat Correlogram dicadangkan. Langkah persamaan yang sedia ada berasaskan DC Correlogram disesuaikan untuk meningkatkan ketepatannya. Kedua-dua kaedah digabungkan untuk menghasilkan pemerihal warna yang baik dari segi masa dan memori kerumitan ruang. Hasilnya, ketepatan telah ditingkatkan berbanding kaedah yang sedia ada dan ruang memori dikurangkan 10% kurang daripada ruang asalnya. Peralihan warna ke dalam beberapa rangka kerja DC dicadangkan untuk mengitlak konsep DC. Selain itu, kedua teknik pengindeksan berasaskan DC dicadangkan untuk mengatasi masalah pengindeksan dengan menggunakan RGB dan ruang warna persepsi LUV. Kajian ini menyumbang kepada pengurangan ruang carian pangkalan data serta pada masa yang sama memelihara ketepatan yang sama capaian imej berasakan kandungan.

Kata kunci: Capaian imej berasaskan kandungan, Correlogram warna dominan, Pengindeksan berasaskan warna, Perihalan warna padat.

Abstract

Content based image retrieval (CBIR) is reported as one of the most active research areas in the last two decades, but it is still young. Three CBIR's performance problem in this study is inaccuracy of image retrieval, high complexity of feature extraction, and degradation of image retrieval after database indexing. This situation led to discrepancies to be applied on limited-resources devices (such as mobile devices). Therefore, the main objective of this thesis is to improve performance of CBIR. Images' Dominant Colours (DCs) is selected as the key contributor for this purpose due to its compact property and its compatibility with the human visual system. Semantic image retrieval is proposed to solve retrieval inaccuracy problem by concentrating on the images' objects. The effect of image background is reduced to provide more focus on the object by setting weights to the object and the background DCs. The accuracy improvement ratio is raised up to 50% over the compared methods. Weighting DCs framework is proposed to generalize this technique where it is demonstrated by applying it on many colour descriptors. For reducing high complexity of colour Correlogram in terms of computations and memory space, compact representation of Correlogram is proposed. Additionally, similarity measure of an existing DC-based Correlogram is adapted to improve its accuracy. Both methods are incorporated to produce promising colour descriptor in terms of time and memory space complexity. As a result, the accuracy is increased up to 30% over the existing methods and the memory space is decreased to less than 10% of its original space. Converting the abundance of colours into a few DCs framework is proposed to generalize DCs concept. In addition, two DC-based indexing techniques are proposed to overcome time problem, by using RGB and perceptual LUV colour spaces. Both methods reduce the search space to less than 25% of the database size with preserving the same accuracy.

Keywords: Content-based image retrieval, Dominant colour correlogram, Colour-based indexing, Compact colour descriptors.

Acknowledgement

To begin, the praise be to Allah (S.W.T). It is impossible for me to accomplish this work without the blessing of Allah and contribution from so many gracious people who have given me all sorts of assistance. To these people, I am indeed grateful.

First of all, I would like to express my deepest gratitude to my supervisors, Dr. Massudi Mahmuddin, Dr. Husniza Husni and Associate Professor Dr. Loay E. George for their guidance and support throughout my study. I cannot fully express my appreciation to their exceptional efforts every time I need them. So, thank you very much and I really appreciate your help.

I believe that words are not enough to express my appreciation to my parents, brothers, sisters and relatives for their love and continuous pray. What I know is that I am indebted to them for being patient, inspiring and supportive of all time.

Grateful appreciation is also offered to all friends who have given valuable advice and support to me for completing this research, my wife Mrs. Dalia Abdulhadi, Dr. Naseer Ali, Ghassan Nashat, Ayad Hameed, Rafid Sagban and Mohammed Fadhil. Special thanks are also extended to Mr. Abdulhadi Abdulameer, Mr. Mohammed Sahib, Dr. Nashat Jassim and all other individual friends who have kept my cheerful along this journey. To those who I couldn't name them specifically here because there are simply so many of them, I offer my apology. But I am deeply indebted to them for their kindness and generosity.

Last but not least, heartiest thank you to my wife Dalia for her love, patience, everlasting support and constant motivation. Without her nothing would be possible. My deepest gratitude goes to my beloved son Yousif, for his love, patience and for being so wonderful. Although she knew nothing about it, my youngest daughter Tara has been the cutest piece in my life.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ANMRR	Average Normalized Modified Retrieval Rank
AP	Average Precision
ARR	Average Retrieval Rate
BIC	Border/Interior Pixel Classification
BOVW	Bag of Visual Words
CBIR	Content-based Image Retrieval
CBIRS	Content-based Image Retrieval System
CCV	Colour Coherence Vectors
CLEF	Cross Language Evaluation Forum
ColGrm	Colour Correlogram
CSD	Colour Structure Descriptor
DC	Dominant Colour
DCBC	Dominant Colour-based Correlogram
DCD	Dominant Colour Descriptor
DQM	Dynamic Quantization Method
EMD	Earth Mover's Distance
FV	Feature Vector
GB	Giga bytes
GCD	Global Colour Descriptor
GLA	Generalized Lloyd Algorithm
HSV	Hue, Saturation, Value Colour Space
k-NN	k-Nearest Neighbours
LBA	Linear Block Algorithm
LUV	Adams Chromatic Valence Colour Space
MAP	Mean Average Precision
MB	Mega bytes
MP7DCD	MPEG-7 Dominant Colour Descriptor
MPHSM	merging palette histogram for similarity measure
NMRR	Normalized Modified Retrieval Rank

NN	Neural Networks
P	Precision
QBE	Query by Example
QBS	Query by Sketch
QSM	Quadratic Similarity Measure
R	Recall
RGB	Red, Green, Blue Colour Space
RF	Relevance Feedback
SCD	Spatial Colour Descriptor
SM	Similarity Measure
SOM	Self-Organization Map
SVM	Support Vector Machines
WDCD	Weighted Dominant Colour Descriptor

CHAPTER ONE

INTRODUCTION

1.1 Introduction

The widespread of digital images and availability of huge storage space led to emergence of massive image collections, which are called *digital image libraries* (Attig, Copeland & Pelikan, 2004). These digital libraries spread on the Internet through the progress of transmission techniques. The wealth of available digital data, especially images, has introduced a problem to those who are seeking information in the digital libraries. This problem lays in managing and organizing these digital image libraries (databases). Therefore, *Indexing and Retrieval* concepts were introduced; *Indexing* relates to "how to store images in database and to retrieve them later (through querying) efficiently", whereas *Retrieval* relates to "how to retrieve images that are relevant to the query from images' database". Both concepts relate with the question of "how to speed up retrieval of the relevant images from databases?"

There are two methods to retrieve images from digital library according to Torres and Falcao (2006). These methods are generally known as image retrieval methods. The first method is Annotation-Based Image Retrieval (ABIR) that depends on metadata associated with each image and use traditional query techniques to retrieve images from database by a keyword (Mehyar & Atoum, 2012; Torres & Falcão, 2006). There are two disadvantages in this method (Chang, Tsai & Chou, 2013; Eitz, Hildebrand, Boubekeur & Alexa, 2010; Poursistani, Nezamabadi-pour, Askari Moghadam & Saeed, 2013). Firstly, it requires annotation of all images in the database which is a laborious and time consuming process. Secondly, annotation process usually is inefficient process because the user does this process in an unsystematic way, where different users use different words to describe the same picture (subjectivity of the users). This will reduce the efficiency of text-based image search (Pavlidis, 2008; Torres & Falcao, 2006; Zagoris, Chatzichristofis, Papamarkos & Boutalis, 2009). Hence, the second method emerged, which is searching and retrieving image based on the image content where many features can be extracted. This method is called *Content-based Image Retrieval (CBIR)*, which was introduced in the 1990s. Image visual content can be analyzed by extracting image features such as colour, texture and shape that are called low level features (Liu & Yang, 2013; Rui, Huang & Chang, 1999; Smeulders, Worring, Santini, Gupta & Jain, 2000; Torres & Falcao, 2006).

Designing generic CBIR applications requires advanced algorithms in image understanding field as well as advances in computer hardware, where both of them are unrealized yet (Pavlidis, 2008; Penatti, Valle & Torres, 2012). Therefore, most efforts nowadays are focused on CBIR for specific applications (Grubinger, 2007; Pavlidis, 2008; Penatti et al., 2012). As a result, many CBIR applications have been developed. CBIR applications varied from personal to commercial, medical, academic, military applications and many other areas like shopping sites on Internet, TV channels, libraries and government archive applications (Carson, Belongie, Greenspan & Malik, 2002; Li & Wang, 2008; Ma & Manjunath, 1999; Smith & Chang, 1996; Torres & Falcão, 2006; Wang, Li & Wiederhold, 2001). Due to increasing CBIR applications, large image collections have emerged. Nevertheless, the advent of large image collections also causes problem. Utilizing simple visual features with large databases leads to inaccurate CBIR results. Therefore, the latter is still behind text-based search because these simple features cannot express the semantic information of an image.

Using low level features for large databases will affect the overall performance of CBIR in terms of decreasing retrieval accuracy and increasing computational complexity, memory and disk space and retrieval time (Guldogan, 2008; Zhang, 2011). This will impede the deployment of CBIR, especially on limited-sources devices such as mobile devices. This will influence the success of CBIR in general. Therefore, improving CBIR performance represents an essential role for providing successful CBIR for various devices. The performance of CBIR systems (*CBIRSs*) depends on two key factors: *accuracy* and *time*. There are three issues that are related to these factors, which are depicted in Figure 1.1 and illustrated as follows (Guldogan, 2008; Howarth, 2007; Zhang, 2011):

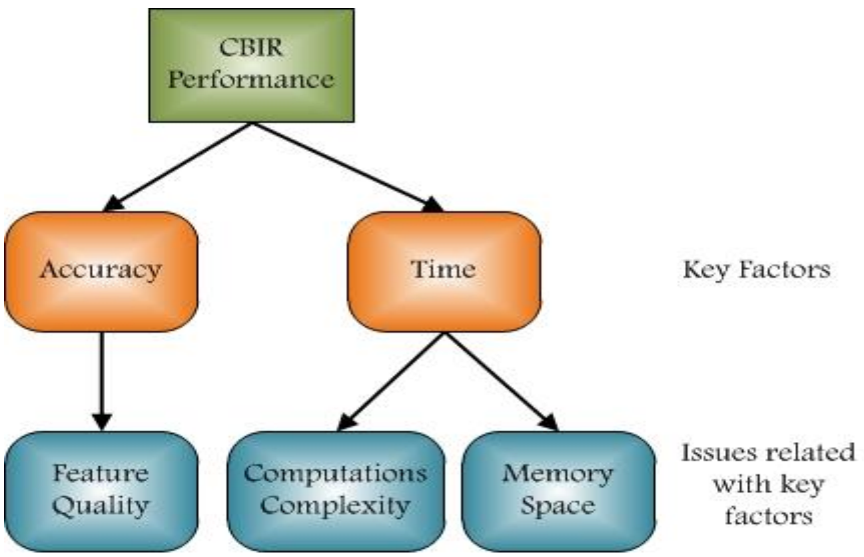


Figure 1.1: CBIR performance influencing factors and their related issues

- *Quality of features* that is affected by the type of extracted features. It influences the semantic image retrieval; in other words, retrieving images similar to human perception.

- *Complexity of computations* that is influenced by feature extraction and similarity measure processes. It affects the speed of the search.

- *Memory and disk space* that is also influenced by feature extraction process. It has impact on the storage space (main memory or hard disk) that are required to store database's features. Additionally, it affects the time of image matching during retrieval process and in turn on the speed of the search.

An improvement of feature quality with reduc the computation complexity is the most challenging issue in CBIR. Moreover, indexing of images' database is one of the important techniques that have an impact on the overall CBIR performance (Goldgun, 2008; Howarth, 2007). Indexing process helps in reducing search space of CBIR during the query. This leads to increase accuracy of the *CBIRS*, by narrowing the search and make it within the scope of relevant images only. Indexing also allows query to be performed on relevant sectors of the database which leads to faster searching thus addressing the search delay.

1.2 Motivations

Due to the rapid growth of digital images, automatic image retrieval becomes inevitable. Generally, image retrieval can be based on metadata (keywords) or content. Most recent image retrieval methods depend on metadata. However, these metadata-based techniques suffered from several problems (Pavlidis, 2008; Torres & Falcao, 2006). Time consuming, inaccurate and user subjective-based process represent the main problems of metadata-based methods. These problems inspire the research of CBIR.

Although CBIR is an active research field, it still needs more research (Datta et al., 2008; Penatti et al., 2012). Therefore, there are number of problems and challenges that need to be overcome (Zhang, 2011). Some of these problems that motivate this work include the following. Due to extracting simple visual features, CBIR is still behind text-based search because these simple features cannot express the semantic information of an image. In other words, semantic information (high-level features) of an image is different than the actual features that can extracted by machine (low-level features). This is known as *semantic gap* problem (Datta, Joshi, Li & Wang, 2008; Hanjalic, Lienhart, Ma & Smith, 2008; Rahman, 2008; Rasiwasia, Moreno & Vasconcelos, 2007; Wang, Zhang & Zhang, 2008). Additionally, expensive computations and memory space that are needed for extracting good visual features represent another problem of applying CBIR especially on limited-resources devices such as mobile devices. Moreover, the emergence of large databases led to appearing additional problem which is slowing down the image retrieval process. These problems caused increase an overall complexity of CBIR. Accordingly, the basic motivation of this thesis is to decrease complexity of this young field.

To reduce the semantic gap problem, extracting an object from the image represents one of the promising solutions (Aboulmagd, El-Gayar & Onsi, 2008; Dobrescu, Stoian & Leoveanu, 2010; Eakins & Graham, 1999). Since colour feature represents the most distinguishable feature among other visual features (Grubinger, 2007; Penatti et al., 2012; Yang et al., 2008; Zhang, 2011), hence, colour feature is used to extract and retrieve images' objects. The selection of colour feature and the need for improving the CBIR performance motivate this work to use image's Dominant Colours (DCs) feature to be the main pivot. This is because, its compact property as well as its compatibility with the human colour perception. Most DCs are extracted from perceptual LUV colour space because it has perceptual property. This is the motivation of using LUV colour space in this research due to this property that make it close to human colour perception (Kiranyaz, Birinci, & Gabbouj, 2012).

1.3 Problem Statement

Although CBIR is reported as one of the most active research areas in the last two decades (Datta, Joshi, Li & Wang, 2008; Datta, Li & Wang, 2005; Rafiee, Dlay & Woo, 2010; Zhang, 2011), it is still young (Datta, Joshi, Li & Wang, 2008; Zhang, 2011). Therefore, it is still far away from user satisfaction (Guldogan, 2008; Pavlidis, 2008; Zhang, 2011). Accordingly, the problem of CBIR is that its accuracy needs to be improved and computations complexity and memory space need to be reduced to obtain user satisfaction and to make CBIR suitable for every platform (computer or mobile) (Guldogan, 2008; Zhang, 2011).

CBIR depends on low level visual features, such as colour, texture and shape, in analyzing the image content. Since colour feature considers as most distinguishable features among other visual features, colour feature is selected as the main feature of this research. Accordingly, addressing the problems of colour-based methods must be more focused. Problems of colour-based CBIR performance can be classified into several categories.

The first category relates to retrieval accuracy where retrieving images that are semantically similar to the query is required. This is the most prominent problem in CBIR in general due to the *semantic gap* problem (Datta, Joshi, Li & Wang, 2008; Hanjalic, Lienhart, Ma & Smith, 2008; Liu & Yang, 2013; Rahman, 2008; Rasiwasia, Moreno & Vasconcelos, 2007; Wang, Zhang & Zhang, 2008). This gap occurs because the computer representation (extracted features) of the image is different than the human interpretation of the same image. Therefore, quality of features needs to be improved. Extracting the object of the image represents one of the promising solutions to reduce the semantic gap problem (Aboulmagd, El-Gayar & Onsi, 2008; Dobrescu, Stoian & Leoveanu, 2010; Eakins & Graham, 1999). Non-discrimination between background and object colours is the main reason of semantic gap problem in colour descriptors (Krishnan, Banu & Christiyana, 2007; Renato, Mario & Alexandre, 2002).

The second category is related to the image retrieval time. Most good colour descriptors, which incorporate the spatial relations of the colours such as *Correlogram*, have high computations complexity and memory space (Kiranyaz, Birinci & Gabbouj, 2010, 2012; Wong, Po & Cheung, 2007). Therefore, this high complexity represents the second problem of colour descriptors that is discussed in this thesis.

The third category relates with the indexing methods that are used to increase the speed of image retrieval process because it narrows down the search space of the database. Colour indexing methods, which are indexed by vector quantization methods, have main problem which is “approximate values of the clusters representatives” problem (Mejdoub, Fonteles, BenAmar & Antonini, 2009; Yildizer, Balci, Jarada & Alhajj, 2012). This is the third problem of colour descriptors.

Therefore, the questions that can be identified in this research are:

1. How can the accuracy of colour-based image retrieval be increased in terms of retrieving images that are semantically similar to the query?

2. How to reduce the computation complexity and memory space for colour descriptors?

3. How to reduce the search space of colour-based CBIR, which can positively affect accuracy and speed of retrieval?

1.4 Research Objectives

The main objective of this research is to reduce the complexity of colour-based CBIR methods. Dominant colours are used to achieve this aim due to their compact property and their matching to the colour perception of the human visual system. From the given main objective, specific objectives of this research can be listed as follows:

1. To propose semantic feature for dominant colour descriptor.

2. To design compact representation of colour *Correlogram* descriptor.

3. To develop an enhanced DC-based indexing methods for colour descriptors.

4. To evaluate performance of the proposed methods.

For the first specific objective, an improved dominant colour descriptor is proposed to achieve semantic image retrieval where it can retrieve images that have similar objects of the query. This is accomplished by reducing the effect of large background colour using feature level and similarity measure level -based algorithms.

For the second specific objective, compact representation of colour *Correlogram* is designed to reduce computation complexity and memory space; this reduces the image retrieval time. Additionally, an existing DC-based *Correlogram* is adapted where a new similarity measure is proposed to enhance the retrieval accuracy. The two methods are integrated to achieve maximum compactness and speed.

For the third specific objective, two DC-based indexing methods are developed to reduce the search space within the database. This speeds up the image retrieval process as well as increasing the accuracy. These methods can be applied to all colour descriptors not only DC-based descriptors.

Lastly, the fourth specific objective can be achieved by evaluating all the previous specific objectives in terms of different criteria depending on the nature of each objective.

1.5 Significance of Research

This section presents a quick description about the significance of the research where benefits that can be provided to the image retrieval society are discussed. Improving CBIR methods, especially DC-based CBIR, have significant positive effects on image retrieval performance. The potential benefits of this research can be summarized as follows:

- Although the benefits of this research to the wide range of image retrieval applications, the direct beneficiary is colour-based image retrieval applications such as trademarks, flags, logos and video retrieval applications. The potential applications that can benefit from this research are identified in Section 7.3. The performance of the colour-based methods are improved by applying the concept of DCs and its related contributions of this research where the retrieval accuracy becomes more semantic and accurate than before, computations complexity and memory space are decreased and the speed of retrieving images is increased.

- CBIR applications that use multi-features will also be positively affected from this research. This is because, colour is one of the main features in most CBIR applications. Therefore, enhancing the colour features in these applications will increase the overall performance of these applications.

- The reduction of consuming the devices' resources in this research provides good step towards applying *CBIRS* on limited-resources devices such as mobile devices. In other words, the contributions of this work will help increasing the practicality of *CBIRS* on different machines (computers and mobile devices).

1.6 Scope and Limitations

The scope of this research can be identified in three domains as follows:

- ***Image Types:*** among the different visual features, colour feature is focused in this research due to its importance in CBIR. Additionally, extracting object from the image can reduce the well-known semantic gap problem of CBIR. Therefore, images of fixed-colour object are used as images type of this research. Examples of this type of images are cartoon images because cartoon characters (objects) are recognized by their colours, where the colours of cartoon character in most images are the same (Jiebo & Crandall, 2006; Khan et al., 2012). Additionally, some natural images also have fixed-colour objects such as flags, trademarks and animals images.

- ***Visual Features:*** Many low level visual features have been used in different CBIR applications such as colour, texture and shape. According to the importance of colour feature in CBIR in general and especially in object recognition (Gevers & Smeulders, 1999; Khan et al., 2012; Sande, Gevers & Snoek, 2008), this thesis concentrates more on colour than other features.

- ***Database Size:*** the size of database in the CBIR applications, which use only the image content without textual information and use distance-based similarity instead of classification-based similarity, is restricted with range from 1,000 to 20,000 images (Clough, Grubinger, Deselaers, Hanbury & Muller, 2007; Leung & Ip, 2000; Vailaya, Figueiredo, Jain & Zhang, 2001; Zhang, 2011). This is conducted because the feature extraction and image retrieval processes in such case will require long time (up to 10 minutes) for each query (Grubinger, 2007; Leung & Hibler, 1991). Therefore, the evaluation databases that opted for this thesis are within this medium-size range.

1.7 Research Summary

To summarize all research questions and objectives, Table 1.1 refers to all questions of the research and its corresponding objectives. Additionally, methods that are used to achieve these objectives and outcomes for each objective are also mentioned.

Table 1.1: *Research Questions, Objectives, Methods and Outcomes Table.*

Questions	Main Objective	Specific Objectives	Methods	Deliverables
How can the accuracy of colour-based image retrieval be increased in terms of retrieving images that are semantically similar to the query?	**To reduce the complexity of colour-based CBIR methods**	To propose semantic feature for dominant colour descriptor	- Literature Analysis - Design Feature level-based Algorithm - Design Similarity Measure level-based Algorithm	Weighted Dominant Colour Descriptor (WDCD) Generic Framework of Weighting Dominant Colours
How to reduce the computation complexity and memory space for colour descriptors?		To design compact representation of colour *Correlogram* descriptor	- Analysis complexity of existing colour descriptors - Design the compact representations of colour descriptor - Design an Algorithm that converts a large colour-based descriptor into few colour-based DCs with new Similarity measure	Compact Representation of *Correlogram* Descriptor Generic Framework of converting large number of colours into few DCs

How to reduce search space of colour-based CBIR, which can positively affect accuracy and speed of retrieval?		To develop an enhanced DC-based indexing methods for colour descriptors	- Literature Analysis of indexing methods - Design RGB Indexing algorithm - Design LUV indexing algorithm	An Indexing Method for RGB space An Indexing Method for Perceptual LUV space
		To evaluate performance of the proposed methods	- Accuracy metrics - Time (speed) comparison - Memory space comparison	Validated Approaches

1.8 Thesis Organization

The thesis is organized into seven chapters, including chapter one; the other chapters can be summarized as follows:

Chapter 2 extensively reviews the literature that is related to this research. Firstly, the components of CBIR are illustrated where these components provide the base to understand this research. In addition, the works that are closely related to the contributions also are discussed. Finally, this chapter is summarized by literature review diagram where the flow and components of this chapter are depicted.

The methodology of this research is explained in the beginning of Chapter 3. Subsequently, all phases of the methodology are discussed in detail. This includes four phases: research clarification, descriptive study I, prescriptive study and descriptive study II. Moreover, settings of the proposed *CBIRS* are identified as well as the reasons of selecting these settings are justified.

The remaining three chapters are contribution chapters, where each chapter starts with a brief introduction that highlights the chapter contribution. Subsequently, there is a section to explain the current problem of the topic that relates with each chapter contribution. Each chapter contribution is discussed; the algorithms and techniques of contributions are detailed in the next sections. Lastly, experimental results that show the comparison with the competing methods are presented.

Chapter 4 is dedicated to the proposed weighted dominant colour descriptor whereas *Chapter 5* is devoted to the compact representation of colour *Correlogram* and to converting large number of colours-based methods into few DCs-based methods.

Additionally, generic frameworks of the weighting of DCs and the conversion into few DCs are designed and verified. Indexing methods represent the contribution of *Chapter 6* where two indexing methods are proposed. First method uses the conventional RGB colour space while the second method uses perceptual LUV colour space.

Chapter 7 presents the conclusions of this thesis. Firstly, the contributions of this work are described. Afterward, the significance of the research is presented. In addition, some possible future recommendations are outlined. Lastly, future applications that can be benefit from this thesis are identified.

CHAPTER TWO

LITERATURE REVIEW

This chapter introduces content-based image retrieval and reviews some of its related issues to provide the base for this thesis. Research problems, which are addressed in this thesis, are related to colour feature of CBIR field. These problems include large background dominance problem, inapplicability problem of good colour descriptor in large database and increasing retrieval time problem in large image databases. Section 2.1 describes the core and non-core components of *CBIRS* as well as identifying some questions that lead to the following sections. Sections 2.2, 2.3 and 2.4 review and justify the problems of this research. Finally Section 2.5 summarizes the chapter.

2.1 Content-Based Image Retrieval

CBIR is searching and retrieving images by analyzing their contents where image contents can be described by visual features such as colour, texture and shape. In a typical CBIR system, as shown in Figure 2.1, there are two processes that can be differentiated. The first one is an offline process; it represents a preprocessing operation in image retrieval system. This process is achieved without user interaction thus it does not affect the real time of image retrieval system. It involves database images' feature extraction and database indexing operations. Feature extraction operation includes extracting visual contents of images then represents them by Feature Vectors (FVs) whereas database indexing provides an efficient way to search the image in the database (Long, Zhang & Feng, 2003). The second process is an online process; it represents the essential operation of retrieval system. This process is conducted when the user provides a query to the image retrieval system and the response to this query is needed within a specified time. Therefore, time is very important in this stage because it affects the efficiency of the CBIR system. This online process involves the query's feature extraction and similarity measure operations. In this context, query's FV is extracted first and then it is compared to all database FVs using similarity measure operation (Celebi & Alpkocak, 2000; Duaimi, 2006). According to the distance between the query image and each image in the database, the top *N* "closest" images will be retrieved. In other words, *N* images that have the smallest distance to the query image are retrieved with the aid of indexing method (Zhang, 2002). Accordingly, *CBIRS* comprises of three core components, which are visual features extraction, similarity measure functions and indexing methods (Cai, Song & Feng, 2012; Datta, et al., 2008; Howarth, 2007).

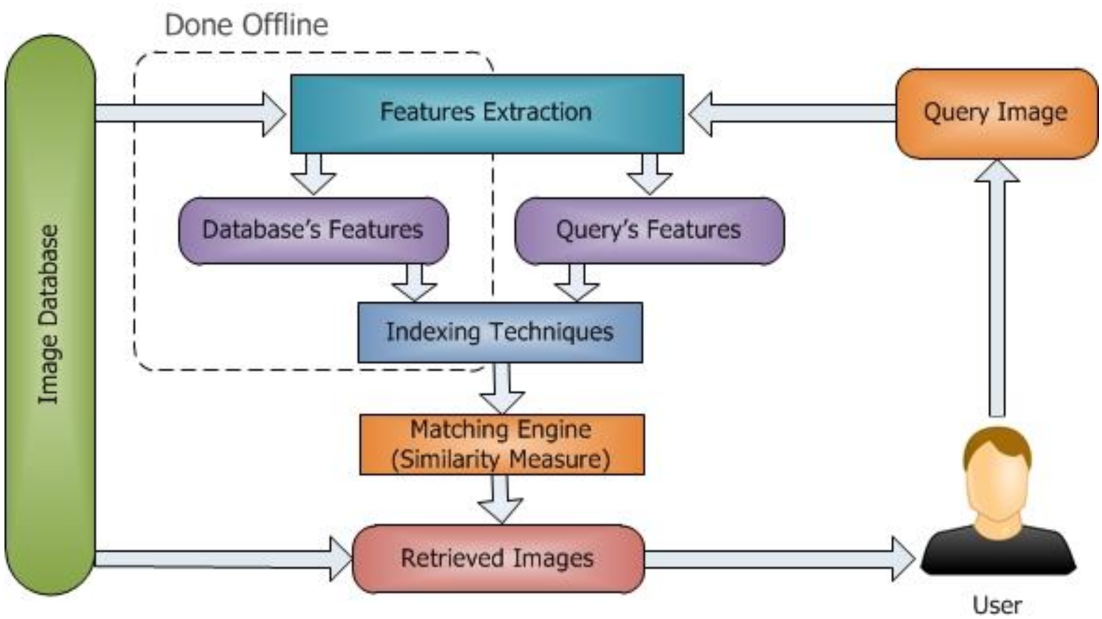


Figure 2.1: Components of CBIR System (adopted from Tran, 2003).

CBIR has two key performance factors, which are *retrieval accuracy* and *retrieval time*, the time needed for retrieving the results (Aulia, 2005; Datta, et al., 2008; Howarth, 2007; Pavlidis, 2008). The former is related with the quality of the extracted feature (that depends on image representation method). The latter is related with computational complexity of feature extraction whereas indexing methods are related with both factors. This research focuses on how to increase performance of CBIR through investigation of the issues and solutions that relate to these two key factors. The following subsections illustrate components of content-based image retrieval to provide complete understanding of CBIR.

2.1.1 Query Formulation

CBIR is a task that searches an image in a large data collection where the retrieved images must match the query image. Query formulation is used to describe the users' needs. There are two types of queries in CBIR which are Query by Sketch and Query by Example (see Figure 2.2). Both queries can be explained as follows (Jacobs, Finkelstein & Salesin, 1995):

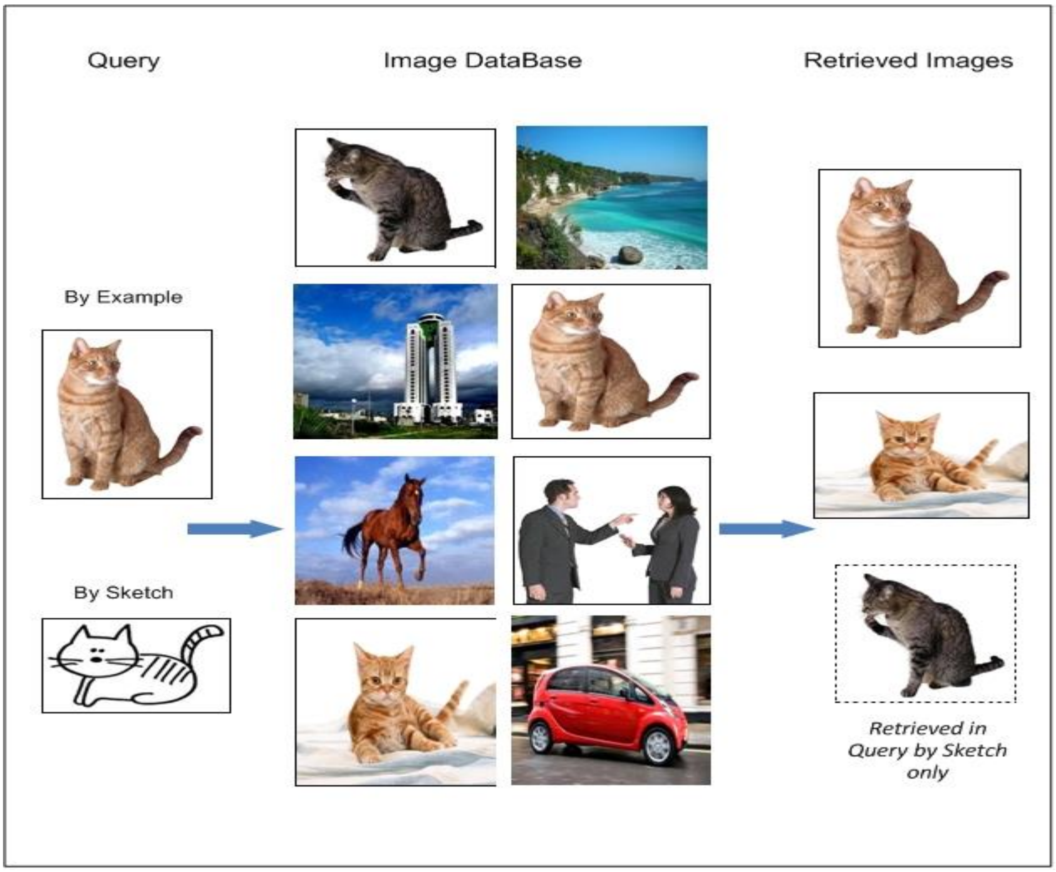


Figure 2.2: Types of query in CBIR (adopted from Siggelkow, 2002)

- Query by Sketch (QBS) – this type of query enables the user to use a sketch to query the retrieval system. In this type of query, the system already knows some features such as the geometric shapes that are used to compose the sketch. Therefore, only a few features from the sketch are needed to be extracted, such as the colour.

- Query by Example (QBE) – in this query, the user can provide an image example to be queried. Using the Query by Example, all image features must be extracted by CBIR system. Thus, feature extraction process plays a vital role to extract FVs that will be used in similarity measure between the query and database images.

2.1.2 Visual Feature Extraction

When the user query the CBIR system, an important question must be asked, which is “how to represent an image” or “how to describe the image content”? In CBIR, normally the image can be represented or described by visual features but what are the visual features that are suitable to be extracted? *The type of CBIR application* is the key to correctly determine the features that can describe the image content (Penatti et al., 2012; Zhang, 2011).

Colour, texture and shape features are the extensively studied visual features in CBIR field (Rui, Huang & Chang, 1999; Smeulders, Worring, Santini, Gupta & Jain, 2000). Colour is the widely used feature in CBIR where 3D representation of colour outperformed on the one dimensional representation (gray-level images) in discrimination power (Smeulders et al., 2000). Different applications can represent colour feature in various colour spaces such as RGB, HSV and LUV. Texture feature is also considered as a powerful feature in CBIR. It represents the granularity and repetitive patterns in the image and it is widely used in specific applications such as medical and aerial images. Shape features are not used broadly as colour and texture features because they depend on segmentation algorithms that already suffered from inaccuracy in separating the similar regions (Zhang, 2011). As mentioned before, the selection of visual features is dependent on the application that the CBIR designed for. Fixed-colour object-based image retrieval is the application of this thesis. Therefore, colour feature is the suitable visual feature for this application. Additionally, texture features are not useful in this type of images because they are applied to specific type of images that their implicit semantics are closely related to repetitive property of texture features such as aerial and medical images (Datta, Joshi, Li & Wang, 2008). Moreover, inaccuracy of image segmentation makes shape features less useful in pure image retrieval applications. Shape features are widely used in classification-based image retrieval applications that utilize the predefined samples to learn the classifier then use this classifier to recognize the query shape (Zhang, 2011).

MPEG-7 visual descriptors that are explained by Manjunath, Ohm, Vasudevan and Yamada (2001) and Sikora (2001) have emerged to be well known descriptors in CBIR field where they are used to describe content of multimedia items. The relevant descriptors to this thesis are certainly the MPEG-7 colour descriptors that include scalable colour descriptor, colour layout descriptor and dominant colour descriptor (Chang, Sikora & Purl, 2001; Manjunath, Ohm, Vasudevan & Yamada, 2001). Dominant colour descriptor is one of the compact colour descriptors that describe the few prominent colours in the image and their percentages. Such compact property of DCs can be used to reduce complexity of many expensive computations of colour descriptors as in perceptual colour descriptor (Kiranyaz, Birinci & Gabbouj, 2012) and dominant colour structure descriptor (Wong, Po & Cheung, 2007). Additionally, it matches the human colour perception. This is because there is a fact that humans cannot perceive more than eight colours (Mojsilovic, Hu & Soljanin, 2002) or just few prominent colours in the image (Broek, Kisters & Vuurpijl, 2004; Kiranyaz et al., 2012; Mojsilovic et al., 2000). Therefore, DCs are considered as the main pivot for the contributions of this thesis.

Some other effective visual features have been proposed for CBIR (Smeulders et al., 2000), include salient points, regions or objects and spatial location or relationship where from these features, some semantic information can be obtained from the image (Zhang, 2011). Using multiple visual features instead of single feature is attractive to many research recently (Zhang, 2011). Combination or integration of multiple features is necessary for general purpose CBIR to improve retrieval performance (Datta et al., 2008; Smeulders et al., 2000). The critical problems lie in two issues: 1) how to integrate these multiple features? 2) how to measure image similarity by using multiple features? Integration of salient object with DCs methods is investigated in Chapter 4 in this thesis.

2.1.3 Indexing Methods

In most CBIR systems, visual content (colour, texture and shape) descriptors are extracted from all images and are represented as multidimensional feature vectors. These FVs can be saved in the form that is called *feature index scheme*, which can be built in as offline to speed up the retrieval process for user query.

In the early image retrieval systems, simple files or database files were used to store FVs such as VIPER that used mySQL (Luoni, 2000) to save the extracted features instead of normal inverted files. Both methods (inverted or database files) are computationally inefficient because they used simple linear search and they are inefficient in variable-size features (Grubinger, 2007).

Although high performance search techniques exist in modern database management systems such as Oracle interMedia (Mauro, 2008), some studies tend to use similarity-based storage techniques such as tree-based and logarithmic performance indexing techniques (Lew, Sebe, Djeraba & Jain, 2006). Examples of these similarity-based methods are R-Tree, R+-Tree, R*-Tree, TV trees, grid files, Linear Quad-Tree, k-d tree, K-d-B tree, priority k-d tree and SS+ tree. Most of these indexing methods have one common problem. They have good performance in 20 or less dimensions (Faloutsos et al., 1994). The second problem is that they use Euclidean distance for visual features comparison, which is not simulating the human perception (Rui et al., 1999).

Solutions for the aforementioned problems can be summarized as the following: For the first problem, features vectors tend to be high dimensional vectors (up to 100) (Grubinger, 2007). This will make these descriptors infeasible in large image databases in CBIR field. Therefore, dimensional reduction methods can be executed before performing efficient indexing technique. Efficient dimensional reduction methods can reduce dimensions of FVs without significant degradation of image retrieval accuracy (White & Jain, 1996). Most popular dimensional reduction methods are Principal Component Analysis (PCA) and Karhunen-Loeve transform (KLT) (Grubinger, 2007). Other approaches also reduce the features dimensions such as neural network-based (Catalan & Jin, 2000) and clustering-based (Salton & McGill, 1986) approaches. For the second problem, one attempt is carried out to solve this problem by using hierarchical indexing method which depends on self-organization map (SOM) (Zhang & Zhong, 1995). With respect to this thesis, only indexing methods that depend on 3-dimensional colour feature are focused, as illustrated in Section 2.4. This is because, this research focuses only on colour feature and the proposed methods are colour-based indexing methods, specifically they are DC-based indexing methods.

2.1.4 Similarity Measure

In CBIR, Similarity Measure (SM) is an important operation to determine the similarity of image in human perceptual perspective. The simplest form of similarity measure is distance measure for single visual feature. This simple distance-based SM needs to be improved, as well as integration of multiple visual features need a careful design for SM, to be consistent with the human perceptual similarity.

A. Distance-based Similarity Measure

There are several distance measures used in CBIR to compute similarity of images (using FVs). Examples of these distance measures are Minkowski metric, Earth Mover's Distance (EMD), K-L divergence and Hausdorff distance. The famous Manhattan distance (L_1 distance) and Euclidean distance (L_2 distance) are special cases of Minkowski metric. L_1 distance is used by many studies such as MPEG-7 colour structure descriptor (Manjunath et al., 2001), Border/Interior Pixel Classification (BIC) method (Renato, Mario & Alexandre, 2002) and many others; it computes the distance between two feature vectors as in Eq. 2.1 (Deza & Deza, 2009).

$$L_1(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.1}$$

L_2 also widely used to measure the similarity in image retrieval field. Similarity between two FVs can be computed using L_2 distance as depicted in Eq. 2.2 (Deza & Deza, 2009). Additionally, some improvements were made to Euclidean distance to increase its effectiveness (Li & Lu, 2009; Liwei, Yan & Jufu, 2005). Moreover, both of L_1 and L_2 distances are broadly used to find the distance between colours in the colour space.

$$L_2(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.2}$$

The EMD and Hausdorff distances are used in region-based image retrieval (Ko & Byun, 2002; Rubner, Tomasi & Guibas, 2000). Kullback-Leibler (K-L) divergence distance is used to measure the similarity of texture features (Do & Vetterli, 2002). In this type of SM, similarity (or dissimilarity) distances of database images to the query are computed. Respecting to the distances results, images will be sorted and displayed accordingly. Although being accurate, distance measures lack good discrimination in some applications. Therefore, classification algorithms are used as an alternative to distance measure where they consider CBIR as image classification problem (Zhang, Lin & Zhang, 2001; Zhou & Huang, 2003).

B. Classification-based Similarity Measure

Image classification is a process of assigning a label (that belongs to one of predefined image classes) to unlabeled query image. This process can be achieved by supervised machine learning algorithms such as Support Vector Machines (*SVM*), Neural Networks (*NN*), k-Nearest Neighbours (*k-NN*), Decision Trees and others. Two methods can be used to design image classifiers (Zhang, 2011). The first method utilizes recommended distance metrics (such as *k-NN* classifier) where the distance of two images in a feature space is computed using recommended distance metrics. However, there is no guarantee that the recommended distance metrics can achieve good classification performance of visual descriptors. In the second method that does not need recommended distance metric for classification (such as SVM), classifier aims to find a set of maximum-margin hyper plane in a high-dimensional space which is able to well separate the image categories, normally in which Euclidean distance function is used to measure the feature distance.

As mentioned above, some applications treat image retrieval as image classification problem and use a classification-based instead of distance-based methods to measure similarity (Zhang, 2011). For example, CBIR can be considered as a binary classification problem where images in the database are seen as positive and negative classes; positive class is the class that consists of all relevant images to the query whereas negative class is the class of irrelevant images. Some samples from both classes (positive and negative) must be used to train the classifier. Subsequently, the classifier can be used to recognize the query whether it belongs to the positive or negative classes (Tong & Chang, 2001; Zhang et al., 2001). Another type of CBIR problem that use image classification is one-class problem. One-Class SVM (Chen, Zhou & Huang, 2001) was used to solve this type of problem. Moreover, CBIR is treated as multi-class classification problem (Peng, 2003). Zhang (2011) also states that classification algorithm can be used in image retrieval when including user relevance feedback (RF). RF is obtained through involving the user in image retrieval process where this can improve the results. Selecting positive and negative samples from the query results and feed them back to the classifier for many iterations help to train it correctly. Relevance feedback-based image retrieval attracts many attentions in the last decade (Tao, Tang, Li & Wu, 2006; Zhou & Huang, 2003).

Even though image classification can enhance accuracy of CBIR system because the classifier is already trained on all predefined classes, many research have highlighted some potential problems of treating image retrieval as image classification problem. These problems are:

i. Small size of training samples and asymmetric training samples, where both of them lead to inaccurate image classification (Tao, Li & Maybank, 2007; Tao, Tang, Li & Wu, 2006).

ii. In large and variable databases, such as web databases, some classes may be unseen beforehand (Datta et al., 2008). This means that the query that comes from these classes is derived from non-predefined classes. This is called hidden classes problem where the query that come from these classes cannot be handled by conventional approaches (Zhang, 2011).

iii. In highly updated and heterogeneous databases (in other words, the image collections that cannot be organized), classifying images is impossible (Zhang, 2011). This is because these collections cannot be categorized into predefined classes either because there is a large number of images that are added periodically resulting in many new classes emerging, or because many images may belong to one or more classes where the important conditions of classification algorithms are the images must belong to only one class and the classes must be non-overlapped;

iv. Classification algorithms, that do not use distance metric such as SVM, need fixed-size feature vector to be applicable. For example DCDs cannot be used in this type of classification algorithms because each image has different number of DCs.

Due to all the aformentioned problems and especially using DCs as a main pivot of all contributions, this thesis adopts distance-based similarity measure instead of classification-based similarity measure.

2.1.5 Image Databases for CBIR

Image databases can be categorized generally into personal, domain-specific, enterprise, archives and web databases (Datta et al., 2008). The type of image database is mainly dependent on the application that it will be utilized for. Additionally, the type of image database can also effect on the complexity of CBIR design (Zhang, 2011) where its size, storage and data type (homogeneous and heterogeneous) must be taken into consideration when designing CBIRS for specific application. The relationship between the database and other components of CBIRS is depicted in Figure 2.3.

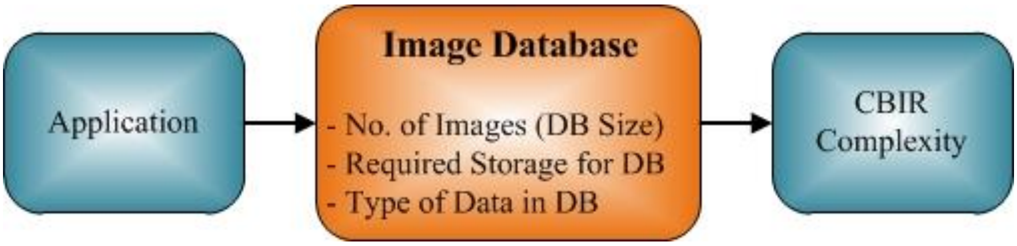


Figure 2.3: Relation of Database with other components of CBIR System

To increase the performance of CBIRS, two issues must be considered. The first issue is using multiple features and the second one is categorization of unstructured database (Zhang, 2011). The first issue is used in this thesis as illustrated in Chpater 4. The second issue is vital in many applications such as web applications. Handling of unstructured database can be achieved by learning-based techniques using two methods, supervised methods such as image classification and unsupervised methods such as image clustering, where both of them can organize an unstructured image database (Datta et al., 2008; ElAlami, 2011).

There are several image datasets that can be used for CBIR evaluation such as Corel dataset, Caltech-101 dataset (Fei-Fei, Fergus & Perona, 2004), TRECVID (TRECVID, 2003), ImageCLEF (ImageCLEF, 2003), as well as other datasets that were designed for specific domains. The evaluation datasets of this thesis are dedicated for colour-based methods, it explained in Section 3.2.5.

2.1.6 Performance Evaluation

In CBIR research, performance evaluation includes three criteria - ground truth, evaluation metrics and number and types of queries (Zhang, 2011). Hence, the following subsection discusses two of these criteria, evaluation metrics and number and types of queries, whereas the third one, ground truth, is explained in Section 3.2.6 (A).

A. Evaluation Metrics

There are many evaluation metrics used in CBIR. The most common metrics are Precision (P) and Recall (R). The first metric (P) measures the accuracy of the image search. In other words, P refers to "the ability of the system to present *only* the relevant images". Its formula is depicted in Eq. 2.3. The second metric (R) measures the completeness of the image search. R refers to "the ability of the system to present *all* relevant images", as shown in Eq. 2.4 (Grubinger, 2007; Zhang, 2011).

$$Precision\ (P) = \frac{The\ \ number\ of\ relevant\ retrieved\ images}{The\ number\ of\ retrieved\ images} \tag{2.3}$$

$$Recall\ (R) = \frac{The\ \ number\ of\ relevant\ retrieved\ images}{The\ total\ number\ of\ relevant\ images\ in\ database} \tag{2.4}$$

Although these metrics are good in measuring retrieval performance, they are inadequate when they are alone. For example, one can obtain R=1 by retrieving all database images, or keeping the precision high when retrieving only few images. Therefore, the solution is either using them together such as precision-recall graph or specifying number of retrieved images such as cut-off value of precision or recall.

Precision at 10 cut-off value, P(10), is the precision in retrieving 10 images only. This metric is easy to interpret and it is preferable by the users in certain tasks such as web searching (Buckley & Voorhees, 2004; Penatti et al., 2012). Therefore, this metric is used to evaluate many retrieval systems such as ImageCLEF (Clough, Grubinger, Deselaers, Hanbury & Muller, 2007) or INEX Multimedia (Westerveld & van Zwol, 2007) and is suitable to evaluate the potential future applications of this thesis. However, P(10) has two drawbacks; firstly it lacks the discrimination power because the changes that can affect on this metric must be achieved on the top 10 ranked images only. Secondly, it is very poor in computing the average accuracy for certain number of queries because there are different numbers of ground truth images of different categories (Grubinger, 2007). This may produce incorrect recall measures within only top 10 ranks (for example, the semantic meaning of P(10) of query that has 6 relevant images is different than the query that has 150 relevant images). Therefore in addition to P(10), other methods must be used in this thesis for performance evaluation.

Precision-Recall graph is another solution for the disadvantages of using either P or R alone. To be more precise and to ease the comparison with others, this graph can be summarized as one value, Mean Average Precision (MAP), which is the most popular evaluation metric (Grubinger, 2007; Zhang, 2011). It can be computed by finding the mean of Average Precision (AP) for all queries, as depicted in Eq. 2.5.

$$MAP = \frac{1}{Q}\sum_{i=1}^{Q} AP_i \tag{2.5}$$

AP can be defined as the mean of precision values for all relevant images of a specific query. AP can be computed as below:

$$AP = \frac{1}{r}\sum_{i=1}^{r} P_i \tag{2.6}$$

P_i represents precision value for all relevant images r, where the precision is computed after retrieving of each relevant image. Precision for specific image can be computed as mentioned in Eq. 2.3. MAP exhibits good discrimination power and stability because it depends on large enough amount of information to make it have lower error rate than precision and recall metrics (Grubinger, 2007; Penatti et al., 2012; Zhang, 2011). Therefore, it is used as evaluation metric in this thesis.

Other metric that have been used widely in CBIR are Average Retrieval Rate (ARR) and Average Normalized Modified Retrieval Rank (ANMRR) (Kiranyaz, et al., 2010; Manjunath et al., 2001; Yang et al., 2008). These metrics are introduced by MPEG-7 standard committee. ARR can be computed via Eq. 2.7.

$$ARR = \frac{1}{N_Q}\sum_{q=1}^{N_Q} RR(q) \qquad \leq 1 \tag{2.7}$$

where N_Q represents the number of queries that are used for the purpose of verifying the descriptor in specific dataset. The abbreviation, RR represents the retrieval rate of a single query (it is similar to Recall metric). It can be calculated using Eq. 2.8.

$$RR(q) = \frac{N_R(\propto, q)}{N_G(q)} \qquad \leq 1 \tag{2.8}$$

$NG(q)$ denotes the number of ground truth images (database relevant images) of a query q. $N_R(\alpha,q)$ indicates the number of the relevant images found in the first $\alpha^*N_G(q)$ images. These images are represented as the retrieved images of the query, where α should be more than or equal to 1. High ARR value (≈1) means a good retrieval rate whereas the low ARR value (≈ 0) indicates a bad retrieval rate.

ANMRR is considered as one of the most accurate metrics used in CBIR. It combines many conventional metrics, which are hit-miss counters, precision-recall and ranking information. Besides, it represents all of them in one value (Kiranyaz, et al., 2010, 2012). To compute ANMRR, one can use the following equations:

$$ANMRR = \frac{\sum_{q=1}^{N_Q} NMRR(q)}{N_Q} \leq 1 \tag{2.9}$$

$$NMRR(q) = \frac{2\,AVR(q) - N_G(q) - 1}{2\,W - N_G(q) + 1} \leq 1 \tag{2.10}$$

$$AVR(q) = \frac{\sum_{k=1}^{N_G(q)} R(k)}{N_G(q)} \tag{2.11}$$

where R(k) is the rank of each ground truth images in the query result window W of the size 2*N_G(q). Any non-relevant images appear within the window W will get R(k)=W+1. The best value of NMRR(q) is 0; it represents all ground truth images that are found in the window W of the query results. The worst case for NMRR(q) is 1; where there are no relevant images retrieved. Therefore, the lower value of ANMRR is better than the higher value. In this work, the selected performance metrics and the reasons behind this will be identified in Section 3.2.6 (B).

B. Number and Type of Queries for Evaluation

To present significance of the quantitative results, retrieval performance must be calculated for a number of queries. Although there is no clear quantitative evidence, some experienced researchers estimate sufficient number of queries to verify performance of image retrieval systems (Voorhees & Buckley, 2002). For example, Jones and van Rijsbergen (1976) showed that 250 queries are usually acceptable while Leung suggests that 20 queries are enough (Leung & Ip, 2000). Voorhees (1998) stated that the results that were obtained from queries less than 25 are relatively unstable. Therefore, TREC identified 25 as the minimum number for queries and 50 is the preferred number of queries (Voorhees & Harman, 2000). Moreover, Zhang (2011) and Chen, Wang and Krovetz (2005) stated that the acceptable range is from 100 to 1000 queries. However, all the above suggestions are not accurate because they did not consider the size of the database which is closely related and directly proportional to the number of queries. Yamada et al. (2001) identify that number of queries is 1% of dataset size. This argument is emphasized by many studies such as Chen, Wang and Krovetz (2005), Po and Wong (2004) and Zhang (2011) where they used the number of queries that ranged from 1% to 2% of the dataset size. Other studies where small databases were used (less than or equal 1000 images), all database images were used as queries (Mustaffa, Ahmad, Mahmod & Doraisamy, 2012).

Diversity of queries is very important to ensure fair and honest results (Grubinger, 2007; Penatti et al., 2012), thus the evaluation queries are selected from all classes of the database in this work.

2.1.7 CBIR Applications and Systems

CBIR is applied in various applications including medical, web, art and culture and personal (Kankanhalli & Rui, 2008). As users have different interests in different applications this leads to different requirements for CBIR techniques. For example, in medical applications such as Kankanhalli and Rui (2008) and Müller, Michoux, Bandon and Geissbuhler (2004) a medical professional may be interested with the dark area in the lung X-ray (as instance) because it may mean a specific disease. However, in the art and culture applications such as Kushki, Androutsos, Plataniotis and Venetsanopoulos (2004), the art objects have distinct colour, texture and shape whereas there are other applications that used only colours for efficient retrieval of object images such as flag, trademarks, manufactured objects, postal stamps and textile patterns (Babu, Mehtre & Kankanhalli, 1995). This thesis focusses on only colour-based applications.

Several CBIR systems were developed in the recent years such as Picsom (Laaksonen, Koskela & Oja, 2002), SIMPLIcity (Wang et al., 2001) and MARS (Ortega-Binderberger & Mehrotra, 2004). Recently, two famous image search engines emerged Google image search[1] and Bing image search[2]. Google search engine offers "search by image" option, where the user can upload an image or enter an image's URL as query to search for similar images on the web whereas Microsoft Bing search engine presents "Similar Images" options to search for similar images for any web image resulted in Bing search page. These search engines combined between content- and text-based techniques in their image retrieval systems (Zhang, 2011). Additionally, there are many CBIR systems introduced for various domains such as family album search, remote sensing, botany, mineralogy and astronomy (Datta et al., 2008).

[1] http://images.google.com/
[2] http://www.bing.com/images/

2.2 Dominance of Large Background in Colour Descriptors

To retrieve colour images from multimedia database, low level features and especially colour feature have been widely used. This is because colour represents the most distinguishable feature compared with other visual features, such as texture and shape (Grubinger, 2007; Penatti et al., 2012; Yang et al., 2008; Zhang, 2011). It represents a basic cue for object and scene recognition (Sande, Gevers & Snoek, 2008, 2010).

From the perspective of feature extraction, colour-based descriptors can be divided into two categories:

i) *Global descriptors* that consider the whole image to obtain their features. There is no partitioning or pre-processing stage during feature extraction process. The resulted descriptors from this approach are simple and fast but lack spatial colour information and high discriminating power. The most famous example of this representation is global colour histogram (Swain & Ballard, 1991);

ii) *Local descriptors* that obtain their features from local regions or partitions of image. This can be achieved by dividing the image into either fixed size or different size regions. The former type is called *fixed partitioning-based approaches* and they have more spatial information about colours in the image. An example of this approach is cell colour histogram (Stehling, Nascimento & Falcao, 2003). The latter type is called *segmentation-based approaches* where the regions of image can be extracted by either segmentation or clustering methods. These descriptors usually have better accuracy than others but introduce more complexity of feature extraction process; examples of this approach are colour-based clustering (Stehling, Nascimento & Falcao, 2001) and dominant colours (Deng, Manjunath, Kenney, Moore & Shin, 2001; Manjunath, Ohm, Vasudevan & Yamada, 2001). Additionally, DCs feature is considered as a global colour descriptor from similarity measure perspective because similarity can be computed from dominant colours and their percentages only like colour histogram (Kiranyas et al., 2010, 2012).

In addition to the global and local approaches of an image's feature extraction, there are other interesting methods that recieved more attention recently which are *local invariant feature-based approaches* (Tuytelaars & Mikolajczyk, 2008). These approaches introduced features that are invariant to different image transformation such as translation, scaling, rotation and affine transformation (Tuytelaars & Mikolajczyk, 2008; Zhang, Yang, Cour, Yu & Metaxas, 2012). Salient edges and regions (saliency map) detection are part of these approaches and they are widely used in many computer vision applications including object recognition (Rutishauser, Walther, Koch & Perona, 2004) and image retrieval (Chen, Cheng, Tan, Shamir & Hu., 2009). Combination of different representation methods (at the feature or rank level) leads to improvement of image retrieval accuracy (Zhang et al., 2012). Therefore, in this thesis, combination of global and local invariant features (at feature level) is proposed to enhance colour-based image retrieval, especially to reduce effect of large background, which is the problem that most colour descriptors suffered from. Therefore, these types of descriptors are detailed in the next sections. Additionally, similarity measure also are used to solve background dominance problem, thus it is discussed in the last subsection.

2.2.1 Global Colour Descriptors

Colour histogram, which is proposed by Swain and Ballard (1991), has been extensively used as global colour descriptor (Alaoui, Ouatik, Alaoui & Meknassi, 2009; Khan et al., 2012; Penatti, Valle & Torres, 2012). It is used to solve translation and rotation invariant problems. Besides, it is characterized by being easily implemented and accurate, particularly with small database size. Many enhancements in histogram-based approaches have been achieved (Gong, Chuan & Xiaoyi, 1996). However, such approaches have several drawbacks; the basic one is its dependence on a static quantization methods. Static quantization is used to reduce colour space and in turn to reduce the storage space and extraction time required for colour histogram. Additionally, it suffers from low discrimination power. This is because many similar colours may be set to different bins; a matter that makes the similarity measure (L_1, L_2 or histogram intersection) between the two histogram inefficient.

To solve the static quantization problem in the colour histogram, a quadratic similarity distance was proposed (Hafner et al., 1995). The proposed method is dedicated to compute the similarity between two images where each one of these images has different histogram bins. That is, if *X* is the colour histogram of the first image with *N* bins and *Y* is the colour histogram of the second image with *M* bins, one can write the histogram of both images as in the following form: $X= \{(C_1, W_1^X), (C_2, W_2^X), ..., (C_N, W_N^X)\}$ and $Y= \{(C_1, W_1^Y), (C_2, W_2^Y), ..., (C_M, W_M^Y)\}$, where *C* represents the colour value and *W* is the weight (frequency or percentage) of each colour in the image. The quadratic distance (D_Q) between these two images can be computed as follows:

$$D_Q(X,Y) = (X-Y)^T A(X-Y) = \sum_{i}^{N}\sum_{j}^{M} a_{ij}\left(W_i^X - W_j^Y\right)\left(W_i^X - W_j^Y\right) \quad (2.12)$$

where A=[a_{ij}] is the colour similarity matrix between the bins C_i and C_j. It has also been noticed that the metric depends on the colour similarity of the bins; however, it has some tolerance to the difference between the colours (Stricker & Orengo, 1996). Po and Wong (2004) show that the quadratic distance has some limitations. For instance, it does not match human's colour perception. Besides, it gives incorrect rank to the retrieved images in some cases.

Due to the limitations of histogram and the fact that humans cannot perceive more than eight colours (Mojsilovic, Hu & Soljanin, 2002), the best solution would be extracting the DCs from the images. Consequently, several DC descriptors have been proposed as the following: MPEG-7 DCD (Yamada, Pickering, Jeannin & Jens, 2001) and other research (Babu, Mehtre & Kankanhalli, 1995; Deng, Kenney, Moore & Manjunath, 1999; Fauqueur & Boujemaa, 2002; Manjunath et al., 2001; Mojsilovic et al., 2002; Wong, Po & Cheung, 2007; Yang, Chang, Kuo & Li, 2008). The DCD, which is extracted using a dynamic quantization, is compact and efficient compared to the other global image descriptors (Kiranyaz, Birinci & Gabbouj, 2010, 2012; Vidal, Cavalcanti, de Moura, da Silva & da Silva Torres, 2012). This is because DCDs require less time and storage consumption compared to the spatial colour descriptors.

In this respect, MPEG-7 Committee have proposed many colour, texture and shape descriptors to be used in image and video retrieval (Deng et al., 1999; Yamada et al., 2001). Kiranyaz, Birinci and Gabbouj (2010), Mojsilovic et al., (2002) and Yang et al. (2008) maintain that human visual system firstly identifies prominent colours in the image and then processes the other details. The whole process resembles the way humans recognize image from its dominant colours without paying any attention to their distribution. MPEG-7's DCD (MP7DCD) provides compact and effective representations for colours in an image or region of interest (Yamada et al., 2001). The prospects of compact dominant colour representation have attracted numerous research recently. Researchers are trying to reduce the size of colour descriptors from several hundred bins (derived from histogram based methods) to only eight colours (as in the MP7DCD) (Kiranyaz et al., 2010, 2012; Wong et al., 2007). This compactness is mandatory in specific applications such as web-based image retrieval (Penatti, Valle & Torres, 2012) and in limited-resources platforms such as mobile devices (Guldogan, 2008). This compactness is necessary to reduce memory consumption and reduce computations in similarity measure process, so as to speed up the retrieval system.

In the simplest form, DCD contains the following form:

$$DCD\,(I) = \{\,(Ci\,,Pi\,)\,,\quad i=1\ldots N\,\}, \tag{2.13}$$

where N is the number of dominant colours in an image I, Ci represents the 3-D value of the dominant colours and Pi represents the percentage of each DC. However, MP7DCD have certain drawbacks, thus it has been undergone some enhancements (Deng et al., 2001; Stehling, Nascimento & Falcao, 2001). Most of the previously conducted studies were dedicated to improve the DC extraction process. This is due to the fact that MPEG-7 uses Generalized Lloyd Algorithm (GLA) for colour quantization (Lloyd, 1982). The latter is characterized by several limitations: 1) It is a time-consuming method; 2) the number of its clusters must be predefined before starting clustering process; 3) using different initial cluster seeds lead to different results. Hence, code book is proposed for colour quantization to reduce the range of colours in an image (Mojsilovic et al., 2002). Moreover, a new quantization method, Linear Block Algorithm (LBA), was proposed to extract DCs faster than MPEG-7 DCD (Yang et al., 2008).

The aforementioned enhancement methods of DCD are mainly used to speed up the process. MP7DCD is accurate but it lacks certain semantic information. That is, the prominent colours and their percentages may only lead to retrieve many dissimilar images that share the same biggest DC. Usually, the dissimilarity occurs when the background colour of an image has the largest percentage. In other words, most of the images retrieved by DCD contain similar background colours if their percentages are high. However, they differ among each other with respect to the semantic of the colour that has the largest percentage (whether it is background or object). This large percentage of colour will affect the similarity measure of images where the low percentage colours will get less consideration. Therefore, giving weight to each dominant colour is used in this work to set importance of DCs. Specifying the important colours (object's colours) from these total dominant colours enhance the accuracy of image retrieval. Additionally, this accuracy improvement will effect positively on the methods that can be integrated with dominant colours such as colour structure (Wong et al., 2007) and colour *Correlogram* (Kiranyaz et al., 2010, 2012).

2.2.2 Local Invariant Descriptors

Due to the inaccuracy of image segmentation process that considers the base of normal local features (Jaswal & Kaul, 2009; Long, Zhang & Feng, 2003). Local invariant features instead are widely used recently for solving a wide variety of problems, from image matching and recognition of specific objects to the recognition of object categories (Arampatzis, Zagoris & Chatzichristofis, 2013; Tuytelaars & Mikolajczyk, 2008). These features are characterized by their invariance to image transformation such as translation, rotation, affine and others. Tuytelaars and Mikolajczyk (2008) explain that the local invariant features are widely used not for locality nor for invariance property but rather for their ability to shift to the form that the researcher prefer to use them in. Recently, representing image content in a robust and flexible way is focused (Rahmani, Goldman, Zhang, Krettek & Fritts, 2005). This is achieved by using local features effectively to compensate the use of semantic-level segmentation where separating object(s) from the background is a very hard problem. Actually, this problem cannot be solved by using low-level features only (Boykov & Jolly, 2001; Cour & Shi, 2007; Ferrari, Tuytelaars & Gool, 2004).

Local invariant feature-based method is a task consisting of higher-level processing steps to extract relevant information or at least to be robust to the outliers in the image (Tuytelaars & Mikolajczyk, 2008). This new way of looking at local features has opened up a whole new range of applications, and moves many steps closer towards cognitive-level image understanding. There are many local feature detectors include corner, blob and region detectors that use different features such as contour-based, edge-based, intensity-based, biologically plausible-based, colour-based or model-based methods. Salient features (regions or edges) represent one of the outcomes of local feature detectors. Saliency idea has been used in many computer vision algorithms (Achanta, Estrada, Wils & Susstrunk, 2008; Achanta, Hemami, Estrada & Susstrunk, 2009; Cheng, Zhang, Mitra, Huang & Hu, 2011). The early approach of using edge detectors was to extract object descriptions where it depends on the idea that the edges are more significant than other parts of the image. More explicit usages of saliency can be divided into those that concentrate on low-level local features (Schmid, Mohr & Bauckhage, 2000), and those that compute salient groupings of low-level features (Sha'ashua & Ullman, 1988). Moreover, some approaches operate at both levels (Milanese, 1993). These salient points are the points on the object which are almost unique. Many methods are used to extract saliency features or map from image as reported by previous research (Achanta, Estrada, Wils & Susstrunk, 2008; Achanta, Hemami, Estrada & Susstrunk, 2009; Cheng, Zhang, Mitra, Huang & Hu, 2011; Goferman, Zelnik-Manor & Tal, 2010). Although the saliency idea originated from local features (regions or edges), there are many attempts to extract it from global contrast of image or combination between them (Achanta et al., 2009; Cheng et al., 2011; Zhai & Shah, 2006).

Due to the useful properties of salient features extraction methods, one of them is selected in this research that conducted by Cheng et al. (2011). It uses to extract salient object from the image to improve dominant colour descriptor. The selection of this method among many salient object detection methods (Achanta et al., 2008; Achanta et al., 2009; Goferman et al., 2010) is due to its properties simple, efficient and more accurate than all mentioned saliency object detection methods.

As a direct reference to the large background dominance problem, there are two modest solutions that have been proposed in the literature to solve this issue. The first one was introduced by Krishnan, Banu and Christiyana (2007) and it was a feature level-based solution. It assumed that the lighter colour in the image represents the object colour and the darker colour is the background but this assumption is uncertain for various image contents. Moreover, it depends only on the largest colour percentage in the object (only one colour) whereas the object may contain many small percentage colours which many researchers reported that the object in the image may be small or large and its percentage in the image is almost 25% (Das, Riseman & Draper, 1997; Kim, Park & Kim, 2003). The second solution was proposed by Renato, Mario and Alexandre (2002) where it was a similarity measure level-based solution using logarithm distance (*dLog*), to solve this problem through computing similarity measure in BIC method. This method still suffers from the large percentage domination problem, as shown in the experiments that were conducted in Section 5.5.4 to prove the generality of the proposed solution.

2.2.3 Similarity Measure of Colour Descriptors

MPEG-7 DCD's quadratic similarity measure (QSM) that is used by Deng et al. (2001) and Yamada et al. (2001), as depicted in Eq. 2.15, has serious drawbacks. For instance, it does not match human colour perception (Po & Wong, 2004; Yang et al., 2008).

$$D_Q(I_1, I_2) = \sum_{i=0}^{N-1} p_i^2 + \sum_{j=0}^{M-1} p_j^2 - \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} 2a_{i,j}\, p_i p_j \qquad (2.15)$$

Therefore, some improvements to QSM have been proposed as shown below:

1) Ma, Deng and Manjunath (1997) proposed the similarity measure as follows.

$$D_{Ma}(I_1, I_2) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left(L_2(C_i, C_j) * |p_{i-}p_j| \right), \qquad (2.16)$$

where L_2 is the Euclidian distance between the colours C_i and C_j; P represents the percentages of the DC C_i and C_j; M and N represent the number of DCs in image I_1 and image I_2, respectively.

2) Mojsilovic, Kovacevic, Hu, Safranek and Ganapathy (2000) proposed a similarity measure as stated below:

$$D_{MojSilovic}(I_1, I_2) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left(L_2(C_i, C_j) + |p_{i-}p_j| \right) \qquad (2.17)$$

3) Po and Wong (2004) proposed a merging palette histogram for similarity measure (MPHSM). Firstly, they merged similar DCs of the two images. This step is done if L_2 distance between them is smaller than a certain threshold. This step helps to produce a common palette (that has N_m colours) of the two images (with N_1 colours and N_2 colours respectively, where $N_m \leq N_1+N_2$). Secondly, two new DC histograms of the two image palettes (each has N_m colours) are generated based on the new common palette. The colours in the latter palette that have a distance smaller than a certain threshold to the original DC will get the same frequency of original bin (DC);

otherwise, a new bin will get zero. Thirdly, a conventional histogram intersection method is applied to the new (equal bins) histogram to find their dissimilarity. To illustrate the histogram intersection considers the following equation:

$$D_{MPHSM}(I_1, I_2) = \sum_{i=0}^{N_m-1} min\,(p_{1i}, p_{2i}), \tag{2.18}$$

where P_1 and P_2 represent the percentages of DC in the two images I_1 and $I_{2,}$ respectively.

4) Yang et al. (2008) proposed a new similarity measure that simulates human colour perception (as present in Eq. 2.19, 2.20 and 2.21). The new similarity measure is proved that it is better than the aforementioned similarity measures (Ma et al., 1997; Mojsilovic et al., 2000; Po & Wong, 2004) and is closer to the human perception.

$$S_{i,j} = \left[\, 1 - \left|p_i(i) - p_j(j)\right| \,\right] \times \min\left(p_i(i), p_j(j)\right) \tag{2.19}$$

$$SIM^{\,yang}(I_1, I_2) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} a_{i,j}\, S_{i,j} \tag{2.20}$$

$$D_{yang}(I_1, I_2) = 1 - SIM^{yang}(I_1, I_2) \tag{2.21}$$

In Eq. 2.19, p represents the percentage of DC in the image, $S_{i,j}$ refers to the similarity between colour percentages. On the other hand, $a_{i,j}$, in Eq. 2.20, represents colour similarity between the two colours C_i and C_j, as indicated in Eq. 2.22. In Eq. 2.20, $SIM^{yang}(I_1,I_2)$ represents the similarity ratio of the two images. Finally, to measure the dissimilarity between the two images, one can use Eq. 2.21.

$$a_{i,j} = \begin{cases} 1 - \frac{d_{i,j}}{d_{max}} & if\ d_{i,j} \le Th_d \\ 0 & if\ d_{i,j} > Th_d \end{cases} \tag{2.22}$$

where $d_{i,j}$ represents Euclidean distance between C_i and C_j, the abbreviation C represents the 3-D colour values (in CIE-Luv colour space), which can be computed as follows:

$$d_{i,j} = \sqrt{(C_i^L - C_j^L)^2 + (C_i^U - C_j^U)^2 + (C_i^V - C_j^V)^2} \qquad (2.23)$$

The threshold Th_d represents the maximum distance whereby the two colours are considered similar, $d_{max} = \alpha\, Th_d$, $\alpha = 1 \; or \; 1.2$.

All the above methods and their similarity measures suffer from a common problem. The problem is represented by the dependence on the largest DCs percentages in the image during image retrieval. To reduce this dependency, a modification is proposed and this modification is applied to more than one of these dissimilarity measures to ensure its generality.

2.3 Applicability of Colour Descriptors in Large Databases

Many studies have been conducted with respect to CBIR (Aboulmagd, El-Gayar & Onsi, 2009; Alaoui et al., 2009; Broek, Kisters & Vuurpijl, 2004; da Silva Torres & Falcão, 2006). In these studies, several visual (low-level) features, such as colour, texture and shape have been used. Colour descriptors play an important role to reduce the gap between low level features, such as colour, texture and shape and high level semantic concepts, such as emotions, events or scenes (Premchaiswadi & Tungkasthan, 2011; Tungkasthan, Intarasema & Premchaiswadi, 2009). Colour is considered as the powerful cue for CBIR (Kiranyaz et al., 2010, 2012) as well as for object and scene recognition (Sande, Gevers & Snoek, 2008). Moreover, it represents an effective feature for image analysis because it is robust for noise, image orientation, scaling and resolution (Guldogan, 2008; Kiranyaz et al., 2012; Lee, Lee & Ha, 2003; Premchaiswadi & Tungkasthan, 2011; Schettini, Ciocca & Zuffi, 2001). Therefore, various colour descriptors have been proposed by many researchers (Alaoui, Ouatik, Alaoui & Meknassi, 2009; Chinlek & Premchaiswade, 2001; Pass & Zabih, 1999; Pass, Zabih & Miler, 1997; Qiu, 2003; Renato et al., 2002; Stricker & Orengo, 1996; Swain & Ballard, 1991; Yi & William, 2007).

Studies varied in their usage of colour descriptors (Kiranyaz et al., 2010). For instance, from similarity measure perspective, two types of colour descriptors can be distinguished. First one is Global Colour Descriptors (GCDs) whereas the second is Spatial Colour Descriptors (SCDs). The former is used to measure the similarity between two images by taking into account both the colours and their percentages in the images, such as the colour histogram (Gong, Chuan & Xiaoyi, 1996; Swain & Ballard, 1991) and the dominant colours (Babu et al., 1995; Deng et al., 1999; Fauqueur & Boujemaa, 2002; Manjunath et al., 2001; Mojsilovic et al., 2002; Wong et al., 2007; Yamada et al., 2001; Yang, Chang, Kuo & Li, 2008). The latter type, measures the similarity between the two images by considering both the existing colours and their distributions or arrangements in the image, such as Colour Coherence Vectors (CCV) (Pass et al., 1997), Border/Interior Pixel Classification (BIC) (Renato et al., 2002) and colour *Correlogram* (Huang, Kumar, Mitra, Zhu & Zabih, 1997; Kunttu, Lepisto, Rauhamaa & Visa, 2003).

2.3.1 Global Colour Descriptors

In GCDs, colour histogram, which was proposed by Swain and Ballard (1991), had been extensively used as a global colour descriptor. Since the original representation of RGB colour is 24-bits, which means 16-million colours will be assigned to each pixel in the image. This imposes infeasibility problem in both time and memory space. Hence, static quantization is used to reduce colour space to make storage and time more realistic. However, histogram-based approaches still have several drawbacks. The first one is its dependence on a static quantization method. The second drawback is that these methods do not match human colour perception (Po & Wong, 2004; Yang et al., 2008). Hence, extracting DCs from the image represented the best solution in this regard. This is due to their compact size and their matching to human colour perception, as mentioned previously in section 2.1.2.

Consequently, several dominant colour descriptors have been proposed where DCs require less time and storage consumption. Although the effectiveness of DCs in colour-based image retrieval, it is still one of the global colour descriptors. These descriptors have a basic problem which is lacking spatial correlations of colours within the image. Accordingly, these descriptors will consider different images, in their colour distribution, as similar images because they have the same colours percentages as depicted in Figure 2.4.

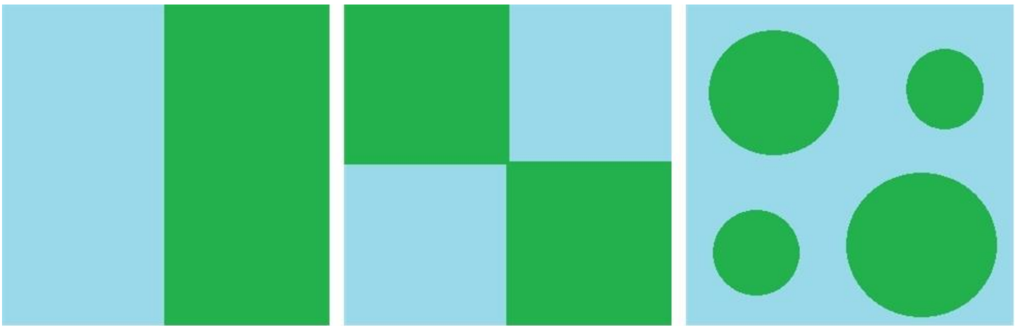


Figure 2.4: Images of the same colour percentages with different colour distributions

2.3.2 Spatial Colour Descriptors

"What are the colours" and "how much percentage of colours" in the compared images are not enough to make a decision about images similarity. The complementary part of image similarity is "where the colours are located" or "how the colours are spatially distributed in the images" (Kiranyaz et al., 2010, 2012). GCDs are working without the complement part; therefore, the results are not satisfactorily presented in the CBIR field. Hence, many methods have been proposed to include this complement part (spatial relationship of colours) (Kiranyaz et al., 2010, 2012; Kunttu et al., 2003; Pass et al., 1997). For example, CCV which divides the colour histogram depending on spatial coherence of the pixels (histogram refinement method) (Pass et al., 1997). Blurring the image is used to remove small colour differences of pixels and produce N discrete colours. A pixel is considered coherent if its colour is similar to the colour of a region that belonged to; otherwise, it is considered as incoherent. The feature vector of such method is represented as N pairs as follows.

$CCV(I) = \{(\alpha(C_1), \beta(C_1)), (\alpha(C_2), \beta(C_2)), \ldots (\alpha(C_N), \beta(C_N))\}$, the pair $(\alpha(C_1), \beta(C_1))$ represents number of coherent and incoherent pixels of colour C_1 respectively. Pairs of all *N* colours are extracted for image *I* then L_1 metric can be used to find a similarity between the feature vectors. This method outperforms colour histogram in image retrieval (Kunttu et al., 2003; Pass et al., 1997). This confirms the assumption that "classifying pixels using colours only without their spatial distribution will not depict the real colour compositions of the image" (Kiranyaz et al., 2010, 2012). Many other approaches also have been proposed to prove effectiveness of spatial relationship among image colours such as Nagasaka and Tanaka (1992) that used concept of colour boundaries and Lee et al. (2003) and Stricker and Orengo (1996) that used colour adjacency concept. The *BIC* method (Renato et al., 2002) also demonstrates that spatial correlations of colours offer enhancement to the accuracy where this method used borders' pixels to identify the objects' shape. From all of these approaches, a simple conclusion can be drawn that the relative distance (inter-distance) of image's colours can capture true or real composition of colours in the image. These SCDs have important properties, i.e. translation and rotation invariant.

A. Colour Correlogram

One of the most active approaches among all SCDs is Colour *Correlogram* (*ColGrm*) (Huang et al., 1997; Kunttu et al., 2003). *ColGrm* demonstrated that it outperforms the other colour descriptors in large database (Pantaii, 2012). *ColGrm* is a table indexed by colour pairs (*Ci*, *Cj*) where k^{th} entry specifies the probability of finding a colour *Ci* at a distance *k* from a colour *Cj* in the image; *i*, *j* are indexes to colours within range of *m* quantized colours and *k* is a distance within maximum distance d. *ColGrm* $\gamma_{c_i,c_j}^{(k)}$ can be expressed as follows:

$$\gamma_{c_i,c_j}^{(k)} = Pr_{P1\,\epsilon\,I<ci>}\left(P_2 \epsilon I < c_j > \middle| |p1 - p2| = k\right) \quad (2.24)$$

$C_i, C_j \epsilon \{C_1, C_2, \dots \dots, C_m\}, k \epsilon \{1, \dots \dots, d\}$ and $|P_1\text{-}P_2|$ are the distance between pixels P_1 and P_2, computed using maximum norm L_∞. *ColGrm* complexity is $O(m^2 d)$, this consumes high CPU time and memory space especially in large databases. To explain this situation, the following example can be considered. Assume the image is of width (W=500) and height (H=400). In such dimensions, a suitable value for *d* will be 40 – 200 corresponding to the formula: $d \approx 10\% - 50\%$ of the smallest dimension in the image (Kiranyaz et al., 2010, 2012; Premchaiswadi & Tungkasthan, 2011). Any value of *d* less than this range will not be suitable to capture true spatial colour distributions of the image, because it will describe colours within a small range only. Even with selection the lower bound of the range of distance *d* (*d=40*), the complexity of *Correlogram* algorithm is still too high and require several processing hours per image on computer. Moreover, it will require large memory space for feature vector even for a small image database. Several possible solutions can be applied to solve this infeasibility problem. The first solution is to reduce the range of distance *d* (for example, let d ≈ 10); this will reduce the complexity by 4 times only (not a significant reduction) and unfortunately the true spatial colour distribution cannot be identified precisely. Another solution is by reducing colour space using quantization algorithm. A typical quantization for RGB colour space is eight partitions for each band, 8*8*8 that equals to 512 total colours. This will speed up the *ColGrm* around 30 times. However, it is still require immense memory space (about 80 MB per image). This will make the *ColGrm* applicable for small database only (for example, storage space required for 1000 images is roughly 80 GB). This space is unsuitable for main memory of the most computers nowadays. Hence, to make it applicable, drastic reduction of colours is needed using coarser quantization (to 4 partitions in each band or less). Therefore, a simplified version of *ColGrm* called *Autocorrelogram* is introduced (Huang et al., 1997). *Autocorrelogram* characterizes spatial colour distribution of the same colours only, each colour with itself without identifying correlations with other colours. The latter case may cause degradation of the colour descriptor and this is actually happened, where many studies reported that *ColGrm* is better in retrieval accuracy than *Autocorrelogram* (Huang et al., 1997; Kiranyaz et al., 2010, 2012; Kunttu et al., 2003; Premchaiswadi & Tungkasthan, 2011; Tungkasthan et al., 2009).

B. Extensions of Colour Correlogram

Even though the problem of *ColGrm* and *Autocorrelogram*, Ma and Zhang (1998), Chun, Kim and Jang (2008) and recently Penatti et al. (2012) show (using extensive experiments) that *ColGrm* and *Autocorrelogram* achieve better performance than other global and spatial colour descriptors such as colour histogram, colour moments, CCV and others. Some extensions have been made to both *ColGrm* and *Autocorrelogram* such as Markov stationary features (Li, Wu, Wang & Zhang, 2008) that are an extension to *Autocorrelogram*. Additionally, other methods were proposed such as wavelet *Correlogram* (Lee, Lee, Ahn & Rhee, 2008), Gabor wavelet *Correlogram* (Moghaddam & Saadatmand-Tarzjan, 2006), joint

Correlogram (Williams & Yoon, 2007) and multi-resolution joint *Autocorrelogram* (Mustaffa, Ahmad, Mahmod & Doraisamy, 2012). All of these approaches perform just slightly better than the original *ColGrm* descriptor with more time complexity (Kiranyaz et al., 2012; Tungkasthan et al., 2009). Moreover, two methods were introduced to reduce the complexity of *Correlogram* (Premchaiswadi & Tungkasthan, 2011; Tungkasthan et al., 2009) from $O(m^2d)$ into $O(3md)$. However, the accuracy of image retrieval is degraded where the proposed methods offered precision similar to *Autocorrelogram* (Tungkasthan et al., 2009). Recent method also introduced to reduce the time complexity of *ColGrm* by approximation of a descriptor (Taranto, Mauro, Ferilli & Esposito, 2010). This method depends on randomization of selection either by the image pixels or the neighbours of the pixels; this certainly will decrease the accuracy compared with the original *ColGrm* but with decreasing time complexity to the half. Drawbacks of this method are the complexity of memory space remained $O(m^2d)$ and the accuracy of such algorithm is not fixed because it depends on randomization of selecting the candidate pixels or neighbours to build *ColGrm* feature vector. Accordingly, the proposed method depends on the original *ColGrm* in adaptation and comparison.

From the early discussion in this section, a conclusion can be drawn that DC concepts can solve both perceptual and infeasibility problems of the colour histogram but DC-based methods still GCDs and lack of spatial colour correlations.

2.3.3 Dominant Colour-based Methods

Many attempts to integrate DCs concept with existing colour descriptors are conducted. The first one is integration of DCs concept with Colour Structure

Descriptor (CSD) (Wong, Po & Cheung, 2007). In this method, similar colours of the two compared images as well as distances between these colours are determined. Moreover, colour structure values are also computed, and then based on Eq. 2.24, the similarity of two dominant colour structure features is calculated. This equation contains some symbols, cd_i represents colour differences of matched colours, sd_i represents the differences of colour structure, β determines the importance of colour distance in the similarity and T_d represents the threshold value to consider the matched colours are similar. The unmatched colours of two images also participate in computing the dissimilarity measure where their colour structure values are added to sd_i and maximum of matched colours differences (cd_{max}) is assigned to all unmatched colours.

$$D(F1, F2) = 1 - \sum_i \left[(1 - sd_i)\left(1 - \frac{cd_i}{\beta T_d}\right)\right] \qquad (2.24)$$

Another attempt was conducted by Zhang and Tai (2008) where this attempt tries to synthesis more than one colour descriptors such as colour histogram, main colour, mean colour, the accumulative histogram, colour pair, dominant colour of partition, colour *Correlogram* and colour moments as well as relevance feedback. They show that the best result is achieved by synthesis colour histogram and *Correlogram* with relevance feedback (Zhang & Tai, 2008). Lifang, XiangLin, Rui and Hui (2012) also divide an image into 4*4 blocks where each block with one dominant colour. Lifang et al. (2012) use Euclidean distance or block distance to find similarity measure between sub-blocks. In this method, the accuracy was similar to MPEG-7 DCD that used quadratic similarity function but with less time. Additionally, Lifang et al.

(2012) used different methods to give weights to these sub-blocks but these weights are fixed, hence this will degrade performance of this method because the objects in the images are located in different positions.

Recent attempt is carried out by integrating DCs with *ColGrm*, to get higher performance than each of them when they are applied separately. Good attempt recently has been conducted to integrate them together in (Kiranyaz et al., 2012). It applies *ColGrm* concepts on the few DCs instead of a large number of colours, but it has some deficiencies in simulating the original *Correlogram* (through imperfect similarity measure). They used penalty trio model to find dissimilarity between the two images by joining global information (extracted from DCs) and spatial information (extracted from *ColGrm*). They used CIE-LUV colour space to extract DCs from the images using an algorithm that was introduced in (Deng et al., 1999). Trio model that measures the dissimilarity between query image Q and database image I can be expressed by the following equation.

$$P_{trio}(Q,I) = P_{\emptyset}(Q,I) + \left(\alpha P_G(Q,I) + (1-\alpha)P_{Corr}(Q,I)\right) \tag{2.25}$$

There are three terms to measure the dissimilarity of the two images. The first term is $P_{\emptyset}$ that measures the amount of mismatching DCs between the two images. Whereas P_G and P_{Corr} measure the differences of matching DCs. P_G represents global difference between similar (matched) DCs of the two images, i.e. dissimilarity of the DCs values themselves and their percentages within images. P_{Corr} represents *Correlogram* differences of matched DCs of the two images. The term α (that has a value of between 0 and 1) represents the weight that can be given to global (P_G) and

ColGrm (P_{Corr}) differences for determining their importance in the trio model. Briefly, the two terms $P_{\emptyset}$ and P_G represent the global colour differences of the two images, which are used in all GCDs. Whilst P_{Corr} represents differences of colours' spatial correlations of the two images, which is used by *ColGrm* (SCD). Combination of the two global and spatial differences needs careful efforts. Global differences between two images is already normalized between 0 and 1 (by original GCDs); but spatial differences that adopted from *ColGrm* is not normalized. Kiranyaz et al. (2012) changed the dissimilarity equation of *ColGrm* because they claimed that it is not efficient. This will lead to serious performance degradation; as addressed later in Chapter 5.

2.4 Reduction of Search Space in Large Databases

Searching a large image database imposes many challenges because the time required for retrieving query image is high. Researchers address some issues that related with image retrieval performance, which can be categorized into two categories (Alexandrov, Ma, Abbadi & Manjunath, 1995; Guldogan, 2008). The first category concerns with retrieval robustness (image retrieval accuracy). It focuses on feature extraction and pattern recognition phases; it ignored the retrieval efficiency where most works in this category rely on sequential search (Hou, Zhao & Shi, 2010; Kiranyaz et al., 2010; Kunttu et al., 2003Yang et al., 2008). The second category concerns with retrieval efficiency (time required for image retrieval). It focuses on multi-dimensional indexing process to speed up retrieval process with same or at least slightly degradation of the accuracy. Actually, the methods of reducing retrieval time (second category) are directly affected by the retrieval methods (first category).

In other words, method that is used to speed up the search, it must depend on the features of image retrieval method. These features will be used as a key to reduce the search space and in turn to speed up the image retrieval. For example, instead of searching the whole database, the search can be focused only on the images that are similar to the query (in terms of features). Therefore, the first category's methods must be discussed first to select what are the suitable features for indexing. Methods of the two categories are discussed in this section where the colour feature only will be focused because it is the scope of this research.

2.4.1 Feature Extraction Methods for CBIR

According to the retrieval accuracy, colour is the salient feature among the low level visual features. As a representative of colour feature, colour histogram and dominant colour descriptors are the widely used methods in content based image retrieval (CBIR) (Kiranyaz et al., 2010). These descriptors capture global colour distribution of the image. Despite of the colour histogram is characterized by simplicity in its implementation but it results large feature vector that is difficult to index. The feature vector length in colour histogram is ranged from few tens to few hundreds. Singular Value Decomposition (SVD) (Hafner et al., 1995) is proposed with regards to feature vector's dimension reduction issue. The problem of this method is an approximation that happens to the original values may leads to degradation of colour descriptors. The high dimensionality of the histogram feature vector make it highly computational cost in similarity measure and inefficient in searching/indexing process.

Various methods have been proposed to cope with these problems. Zhang, Gong, Low and Smoliar (1995) used dominant histogram colours to reduce number of colours that can represent the image. Multi-resolution colour clustering is also proposed to reduce computational cost of similarity measure process (Wan & Kuo, 1998). Moreover, Babu et al. (1995) propose colour indexing method using R-tree spatial indexing method (Guttman, 1984) that is applied on regions dominant colours in flag and trademarks databases. These images actually contain homogenous regions that can be represented by small number of colours. This method is used to retrieve images from database that have only the exact colour of query image because this is required in this application as well as this is appropriate in these databases that consist of homogeneous colours. Moreover, a good method of extracting dominant colours from image regions is proposed (Deng et al., 2001). This method used colours in the image regions with their percentages as features and quadratic similarity measure as matching process in image retrieval. An efficient colour indexing method is used in this work, lattice structure in 3-D colour domain (Deng et al., 2001). They used edge flow segmentation algorithm (Ma & Manjunath, 1997) to segment image into regions then utilizes GLA quantization algorithm (Lloyd, 1982) for colour quantization to extract dominant colours from each region separately, each region has approximately 3.5 dominant colours. This method requires finding all lattice points of a lattice cell within specific radius. Therefore, the radius of the lattice cell must be selected carefully. Thus, a wider range than the normal range of points may be resulted. This requires removing the out-of-range points; thus an overhead is added to this method. Additionally, there is a possibility that the lattice point is located at the border of lattice cell. This is another problem of this method that will be discussed later in this section as one of the popular problems in indexing techniques.

Moreover, MPEG-7 committee proposes dominant colour descriptor (*MP7DCD*) (Yamada et al., 2001) that is similar to the method in (Deng et al., 2001) but MPEG-7 committee used GLA for colour quantization to extract maximum eight dominant colours from the whole image instead of extracting DCs only from images' segments or regions. This depends on the fact that humans cannot perceive more than eight colours (Mojsilovic et al., 2002). *MP7DCD* (Yamada et al., 2001) and its variants (Po & Wong, 2004; Yang et al., 2008) show effectiveness of DCD in CBIR in spite of they capture global colours distribution only from the image. The benefits of DCDs over histogram-like methods is the former finds image's representative colours from image itself instead of making it fixed in the colour space as the latter. This will make it an accurate and compact descriptor. Therefore, DCD is selected to be the base of this research and subsequently indexing method will depend on this descriptor and its features.

2.4.2 Indexing methods for CBIR

In large image databases, indexing is an urgent matter to reduce the search space of the retrieval process and in turn to speed up this process. *MP7DCD* and its variants perform sequential search in their retrieval process; this will impose delay in the time of image retrieval process. Therefore, colour indexing method should be used to reduce search space of *MP7DCD* and thus to speed up retrieval process. Before discussing colour indexing methods, indexing methods of image features in general must be reviewed.

For indexing the image features, there are two main approaches in general: multi-dimensional indexing and vector quantization techniques, as shown in Figure 2.5. Multi-dimensional indexing techniques are divided into two categories, Space Partitioning (SP) and Data-Partitioning (DP) methods. Both of them divide the space or data into small partitions but the difference lies in how the partitioning process is accomplished. SP methods such as kd-tree (Bentley, 1979), Grid files and hB-tree divide the whole space into disjoint partitions without consideration of the data (feature vectors). Whereas in DP methods such as R-tree, SS-tree and SR-tree, feature space is divided depending on features (data) distribution in the database (Bohm, Berchtold & Keim, 2001; Mejdoub, Fonteles, BenAmar & Antonini, 2009; Yildizer, Balci, Jarada & Alhajj, 2012). The advantage of SP method is it performs complete and disjoint partitions of the whole space that means there is no overlapping between these partitions. A disadvantage of this method is that empty partitions may be produced because there are no data to occupy these partitions thus resulting in an increased size of indexing structure. Another disadvantage of SP method occurs when the query point is located at the border of partition; this will lead to degradation of the retrieval performance in two situations. In the first situation, if the search on this point was made within that partition only and there are some similar points in some neighbour partitions. This will decrease the retrieval accuracy due to ignoring some similar points in the search space. In the second situation, if all neighbour partitions are taken into account through the search on query point. This will increase the number of computations required for the retrieval process because the search space will increase to include many partitions instead of one. On the other hand, the advantage of DP method is the size of indexing structure is compatible with the features in the database where there are no empty partitions that will increase the size of indexing structure. The disadvantage of DP method is the overlapping between partitions; this may degrade performance of the search. The common and critical problem of multi-dimensional indexing techniques, as mentioned before, lies when the number of feature dimensions is high, it is called *curse of dimensionality*.

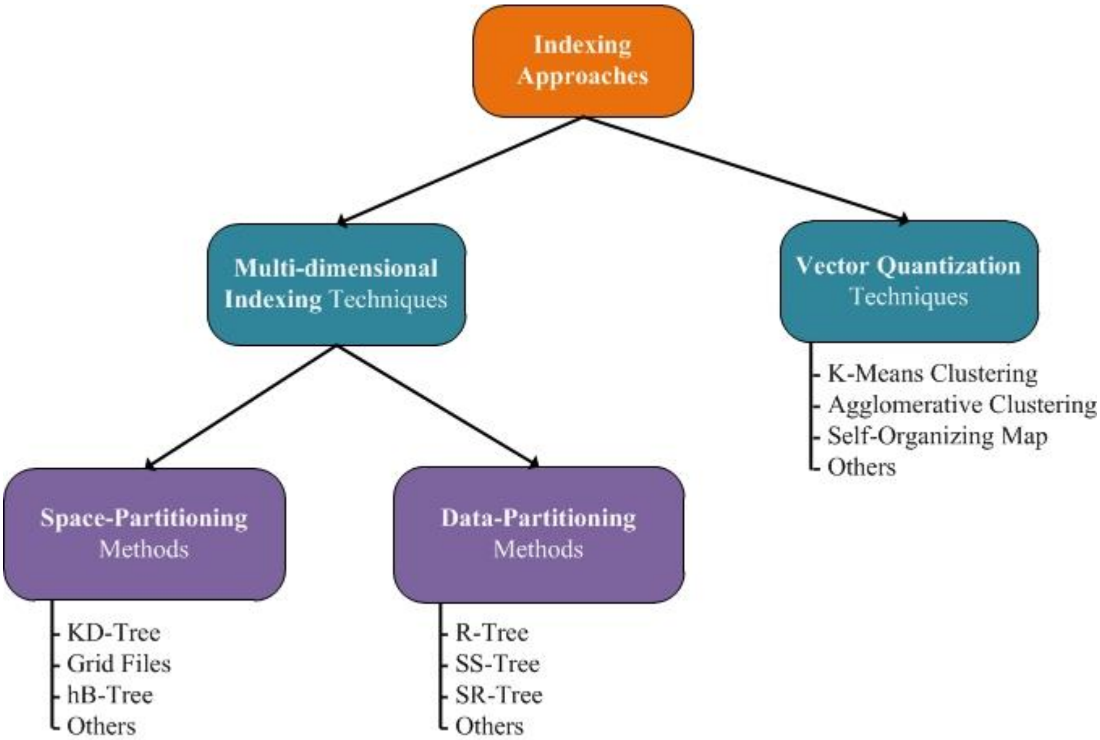


Figure 2.5: Taxonomy of CBIR Indexing Methods

In vector quantization techniques, there are many techniques that have been proposed, which include hierarchical K-means clustering (Nister & Stewenius, 2006), agglomerative clustering (Leibe, Mikolajczyk & Schiele, 2006), randomized tree (Moosmann, Triggs & Jurie, 2006) and self-organizing map (Kaski, Kangas & Kohonen, 1998). In these methods, there is no partitioning of space or data into small parts, instead grouping the data into clusters or groups is achieved. The properties of these clusters (groups) are the distance between the cluster members (intra-distance) should be minimized whereas the distance between the different clusters (inter-distance) should be maximized. Each cluster (group) is represented by a single value called cluster's centroid. Cluster's centroid is computed by averaging all cluster members, thus the query point is compared with cluster's centroid instead of original value of the members. Disadvantages of these methods are the initial number of clusters (K) that need to be known prior to the clustering process. Additionally, majority of these methods do not preserve ordering structure of data space. This will lead to expensive online distance computation, which is needed to compare query point with all clusters' centroids to select the nearest one. Moreover, comparing with clusters' centroids instead of original values lead to inaccurate results because some cluster's members are far from the query points in spite of having suitable distance from cluster's centroid. The latter problem is called "feature approximation problem". Comparison of different indexing is depicted in Table 2.1, where the disadvantages of each method are marked with bullet and the advantages are signed by (√).

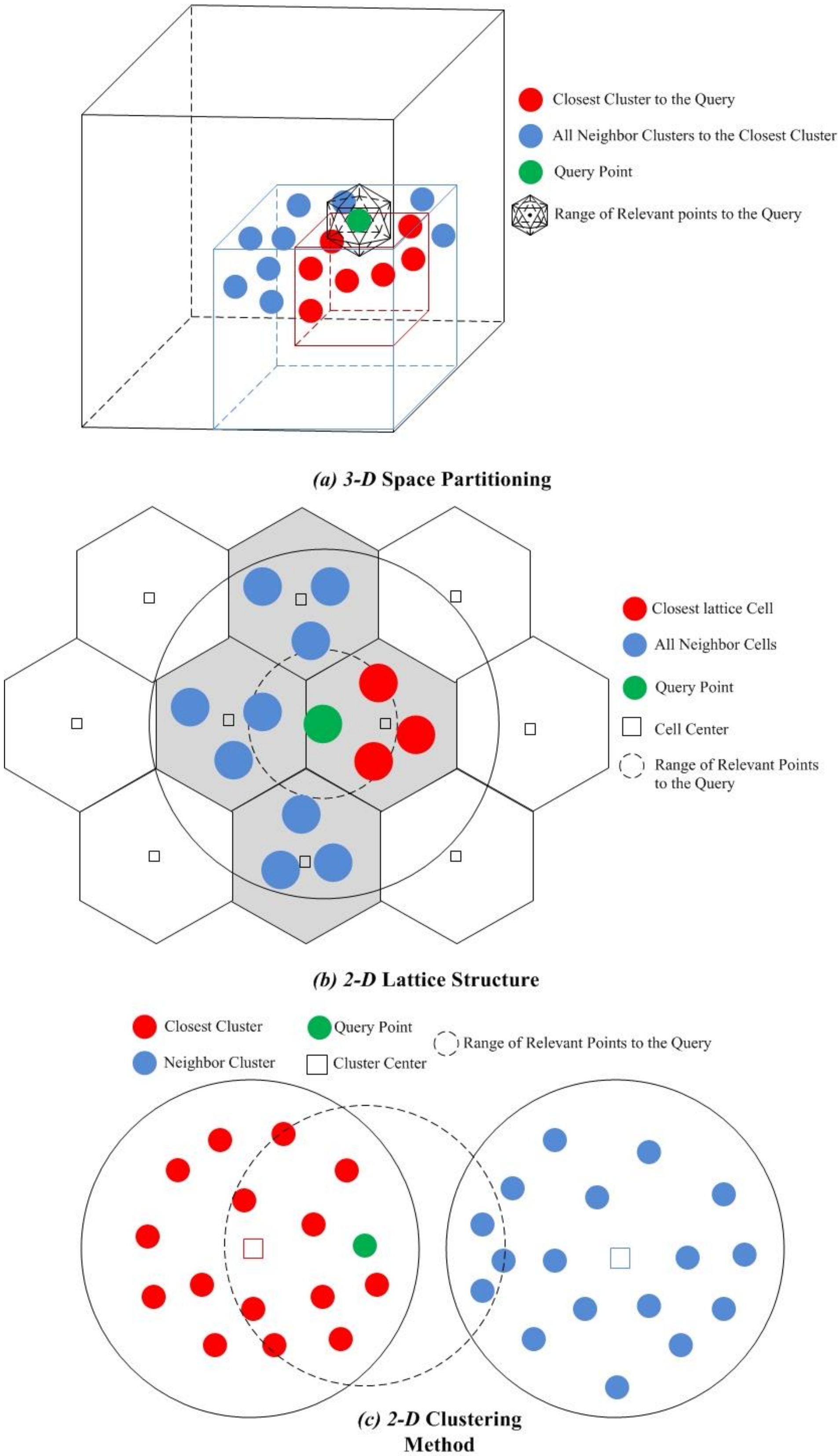
Table 2.1: *Comparison of different Indexing methods*

No.	Index Features or Problems	SP	DP	VQ
1	Overlapping Problem between partitions	✓	●	●
2	Existing of empty partitions that increase size of index structure	●	✓	✓
3	Problem of locating the query on the border of partition or cluster	●	●	●
4	Problem of needing Initialization of parameters of Index scheme	✓	✓	●
5	Problem of Sequential Scan of data to find closest Cluster	✓	✓	●
6	Feature Approximation Problem that lead to Accuracy Degradation	✓	●	●
7	Dynamic property of Index Structure (during insertion and deletion, there is no need for reconstructing the index structure)	✓	●	●

2.4.3 Colour-based Indexing Methods for CBIR

For 3-dimensional colour indexing, several methods have been proposed in CBIR field. High dimensional histogram indexing that was used by Deng et al. (2001) for comparison was considered as the simplest and most expensive indexing method. This method suffers from high dimensional problem, 1024-D of colour histogram bins. Babu et al. (1995) combined colour clustering (using LEADER (Spath, 1980)) and spatial indexing method (R-Tree) for indexing colours of flags and trademarks databases. Sudhamani and Venugopal (2007) also proposed a method for colour clustering and indexing. They used mean shift algorithm for colour clustering (Sudhamani & Venugopal, 2006), R*-Tree for spatial indexing (Beckmann, Kriegel, Schneider & Seeger, 1990) and perceptually uniform LUV colour space instead of RGB. The above two methods depend on clustering that suffered from aforementioned problems of clustering. Ma and Manjunath (1997) proposed NeTra system for image retrieval. They proposed binary colour table for colour indexing that depended on 256 colours codebook extracted using GLA method in RGB colour space (Ma & Manjunath, 1997). Restricting with 256 colours certainly will lead to accuracy degradation as a result of colour approximation (similar to clustering method).

In general, colour-based indexing methods depend on fixed range of colours in similarity measure. Therefore, Spatial indexing methods such as R-Tree and R*-Tree are not necessary and fixed indexing structure is more efficient (Deng, Manjunath, Kenney, Moore & Shin, 2001; Samet, 1990). Accordingly, Lattice structure was proposed in (Deng et al., 2001) that is characterized by efficient finding the nearest neighbours of given point (colour) in 3-dimensional LUV colour space. But this efficiency depends on careful selection of radius in hexagonal lattice cell and this is not a straightforward process, hence there is no comparison (in the literature) has been made with this method. Additionally, it suffers from same problem of SP and clustering methods, which is the query point may locate at the border of lattice cell, as depicted in Figure 2.6. Thus, this method is also suffered from same problems of previous studies except the process of finding query's relevant points is fast; but not all these points are actually related to the query. This may occur due to the selection of large value of radius; hence further computations are required to exclude the irrelevant points. Moreover, lattice structure has better performance in uniform distribution than non-uniform distribution (Pauleve, Jegou & Amsaleg, 2010). Thus, it is recommended for RGB colour space instead of LUV colour space.



(a) 3-D **Space Partitioning**

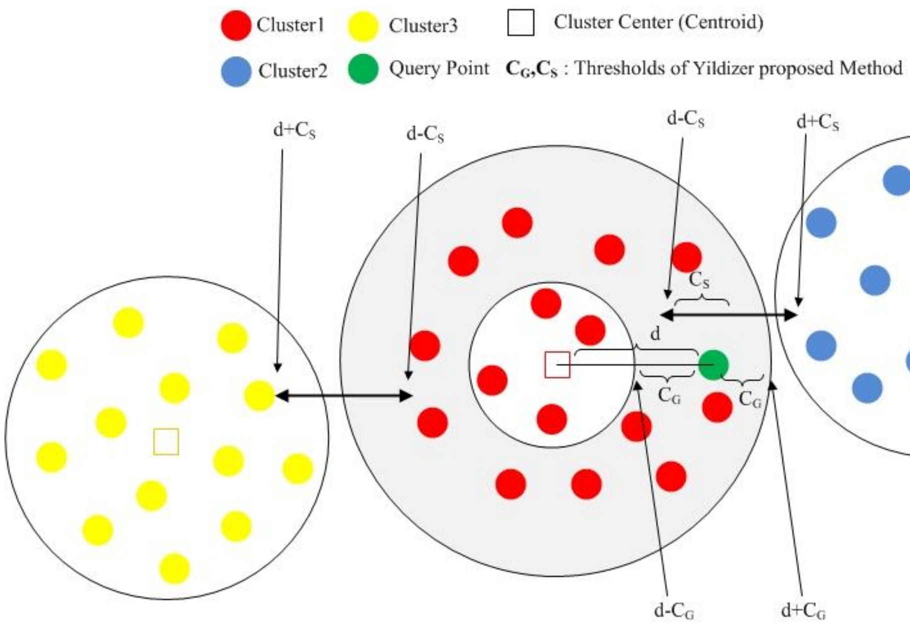
(b) 2-D **Lattice Structure**

(c) 2-D **Clustering Method**

Figure 2.6: Illustration of locating the query image on border of SP, lattice structure and cluster.

In Figure 2.6 (“2-D clustering method”, Figure 2.6 (c), is used as example to clarify the idea), all red points fall into first cluster whereas the blue points are belonging to the second cluster. Query point (image) is denoted by green colour and it is closer to the red cluster’s centroid than blue cluster’s centroid. The optimal search space that must be considered for query image is denoted by dashed line circle. The query image located at the border of first (closest) cluster however it is near to some points in the second cluster, as depicted in a circle of dashed line. Thus, if the closest cluster is only considered then some relevant points will be discarded. Additionally, dashed line circle explains that some points in the closest cluster are not related to the query. This is because, the query is compared with the representative of cluster (the centroid), which its value is approximated to represent all cluster points, instead of comparing with original points’ values. Therefore, retrieval performance will be degraded in the both following cases. In the first instance, when the closest cluster is selected only for matching with the query (as example, selecting first red cluster only in Figure 2.6). In this case, accuracy will be decreased because some relevant images will be ignored as well as some additional computations will be added because some irrelevant images will be compared. In the second case in an instance when all neighbour clusters to the closest cluster are selected for comparison (as example, selecting both clusters in Figure 2.6). Expensive computations will be performed because large number of irrelevant points will be compared.

Recent research is proposed by Yildizer et al. (2012) to solve this problem. The significant contribution of this research is introducing two threshold values C_G and C_S that can be considered as search space parameters. C_G represents the distance from query point that can be searched around it (in the closest cluster) to find similar images instead of considering all cluster members. C_S represents the distance from query point that can be considered to add other clusters to the similarity searching process; both thresholds are explained in Figure 2.7. It is worth mentioning that the distances of all cluster points to the cluster centroid are computed and saved during construction of the model to release online query process from this computations. Additionally, distances are stored in B+Tree structure (Lightstone, Teorey & Nadeau, 2007; Powell, 2006) to reduce the costly I\O operations that are needed for retrieving images from secondary storage in large database.



**Note: Shaded area only will be searched as well as the blue and yellow clusters will be considered in the searching process.*

Figure 2.7: Explanation of the new Threshold values of the method proposed by (Yildizer et al., 2012).

From Figure 2.7, one can notice that C_G threshold represents the distance from query point that will be used for searching the images within. Therefore, searching images of closest cluster centroid have to be in the range from d-C_G to d+C_G (shaded area in red cluster) instead of all cluster members. They identify that cluster members in that area are close to the query. Moreover, C_S threshold determines the distance that can be used to search the other neighbour clusters. Thus, searching on new clusters will be in the range d-C_S to d+C_S from closest cluster centroid.

Threshold values can be initialized by setting C_G to the average of distances of all cluster members to the centroid and C_S to Zero. These values must be iteratively updated until the level of accuracy obtained from the query is a constant. The updating to threshold values can be performed using the simple formula $C \pm \delta$. This will add an expensive computation to the retrieval time of the query. They assume that this process can be computed once only for first query and then the model can be built and used by all other subsequent queries. Actually, this assumption is questionable because the position of the query point can effect on these parameters; hence generalization of these thresholds by single query is unacceptable for large and diverse database.

As conclusion, all vector quantization indexing methods, which most colour-based indexing methods are based on, suffered from colour approximation problem. Precisely, the indexing methods that based on dominant colours suffer from two approximation processes. In the first process, occurs when producing the dominant colours where the dynamic quantization method, such as GLA, achieves first approximation process on colours. Second approximation process is carried out during vector quantization process, such as K-means clustering method, to produce colour centroids. Therefore, this problem will be addressed in Chapter 6 to discuss and propose the solutions.

2.5 Chapter Summary

This research focuses on how to increase performance of CBIR through investigation of the issues and solutions that relate to the CBIR key factors, as shown in Figure 2.8.

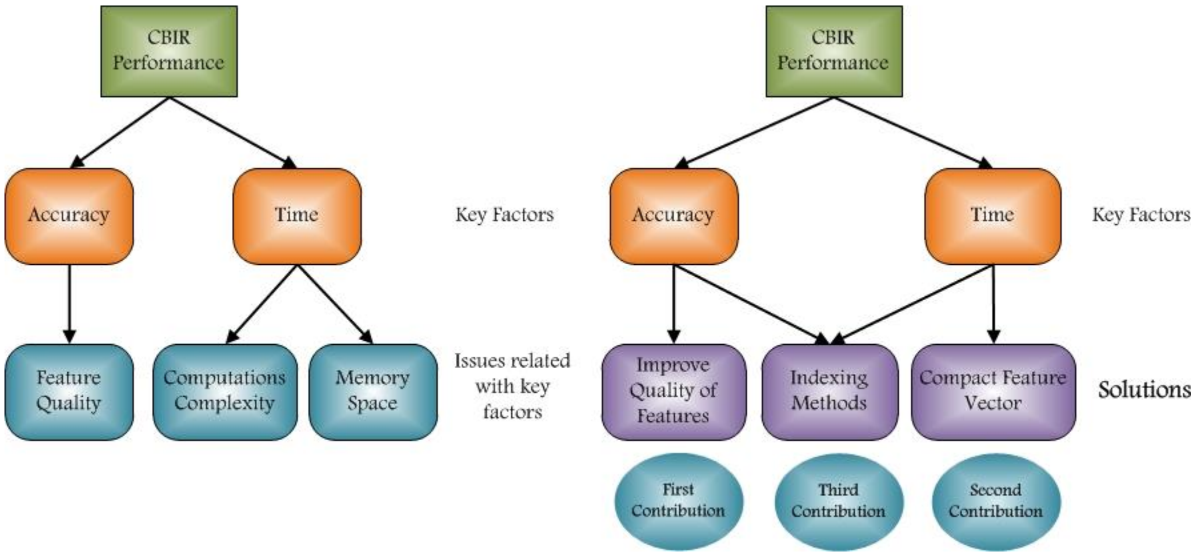


Figure 2.8: Key Factors and issues of the CBIR performance and related Solutions and Contributions in this Research

This chapter starts with a survey about content-based image retrieval. This survey includes the core and non-core components of CBIR. The core components are visual feature extraction, indexing methods, similarity measures whereas the other components are query formulation, image collections, performance evaluation and CBIR applications. Colour represents the main feature that is in the focus of this thesis. Additionally, three colour-related issues are investigated in this chapter.

- The first issue is the dominance of large background on image similarity in colour descriptors. This problem can be noticed in most colour descriptors; however few modest solutions have been proposed to solve this problem.

Therefore, different global and local image descriptors are reviewed to find the way to solve this problem, which leads to the feature-level and similarity measure level-based solutions. These two solutions are explained in Chapter 4.

- The second issue of this research is the applicability of good and well-known descriptor in large image database. Good colour descriptors are identified by reviewing the literature. Additionally, the problem of the good descriptor, colour *Correlogram*, is addressed. The problem occurs due to the high complexity of *Correlogram*. Previous solutions of *Correlogram* complexity are reviewed and analysed. Hence, solutions are proposed in Chapter 5.

- The third issue that is investigated in this thesis is the reduction of search space in large image databases. Several indexing methods have been used in this purpose to reduce search space and in turn to speed up the image retrieval process. Specifically, colour indexing techniques are focused and their drawbacks are identified. Two colour indexing methods are proposed and they are explained in Chapter 6.

Reviewing the literature of CBIR and the issues that relate to this research can be summarized in Figure 2.9; contributions of the thesis are also pointed out. The next chapter, Chapter 3, presents the research methodology extensively.

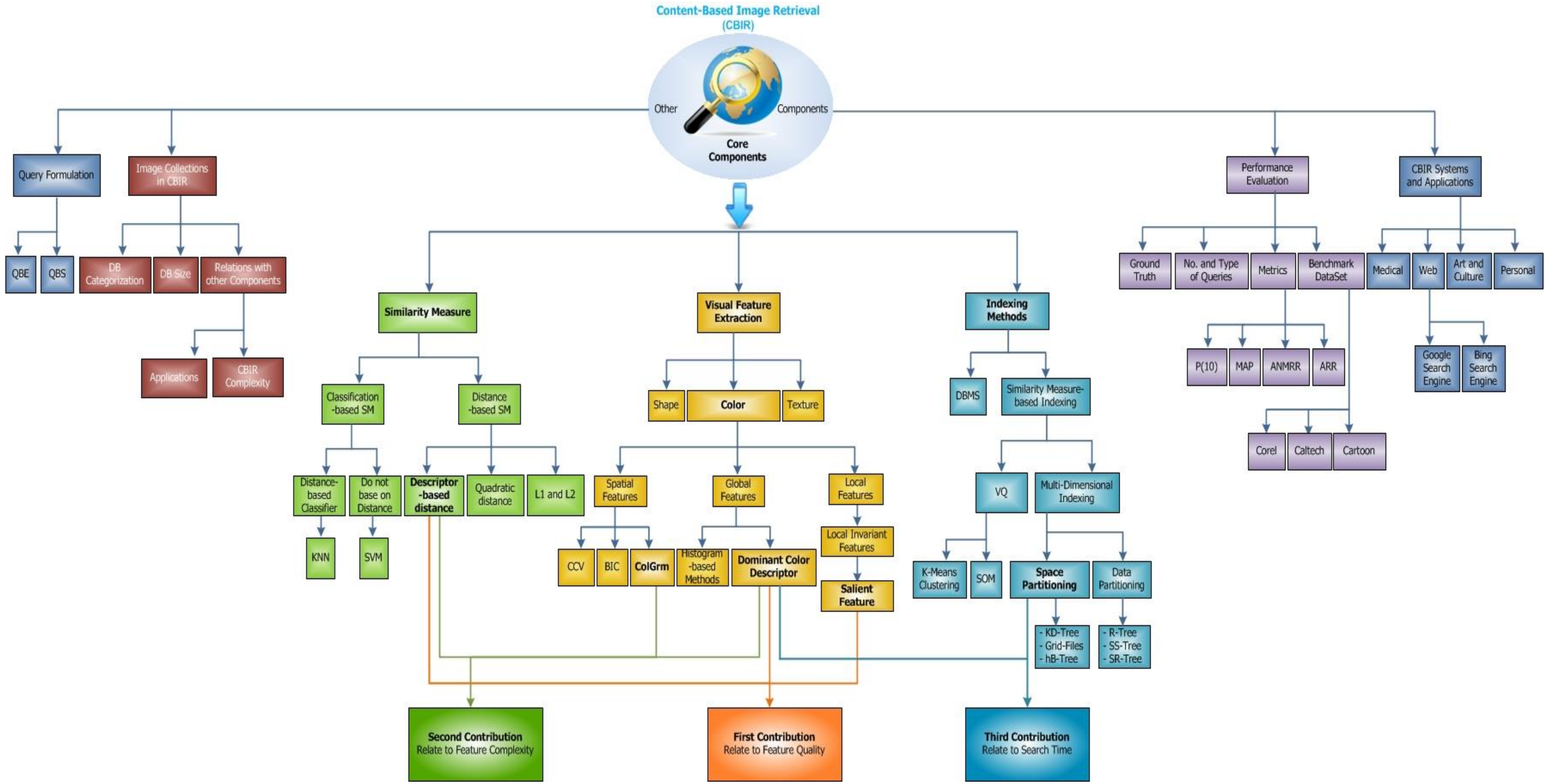


Figure 2.9: Diagram of research approach.

CHAPTER THREE
RESEARCH METHODOLOGY

This chapter presents methods that are used to carry out the research. According to the research objectives, which are listed in Section 1.5, the phases of conducting this research are extensively explained in Section 3.1. Parameters setting of the proposed *CBIRS* of this research are identified and justified in Section 3.2. Finally, a summary of this chapter is presented in Section 3.3.

3.1 Design Research Methodology

In this thesis, Design Research Methodology (DRM) (Blessing & Chakrabarti, 2009) is adapted. DRM aids to identify the criteria of successful research. Additionally, it helps the researchers to design rigorous and efficient research. DRM has four phases: Research Clarification (RC), Descriptive Study I (DS-I), Prescriptive Study (PS) and Descriptive Study II (DS-II) as depicted in Figure 3.1.

Phases	Processes	Methods	Outcomes
Research Clarification	1- Realization of Content-based Image Retrieval Components	Survey of CBIR Systems	Recognize the Criteria that influence CBIR Performance
	2- Understand Color-based Image retrieval Methods	Reviewing Color-based CBIR Methods	Address the problems of Color-based CBIR
Descriptive Study I	3- Identify the key to the semantic color descriptors	Study the related works of color descriptors	Reducing the effect of large background color is the key for semantic (object-based) image retrieval
	4- Reduce the Complexity of Correlogram	Investigate the current solutions of Color Correlogram complexity	Address the factors that effect the correlogram complexity
	5- Study the ability of applying DCs on Color Descriptors	Examine the current Conversion into Dominant Colors Techniques	Specify the imperfection of Similarity Measure of the current Color Conversion methods
	6- Study speed up of Color-based Image Retrieval	Analyze the existing Color and General Indexing Techniques	Identify the gaps of current indexing methods
Prescriptive Study	7- Develop a solution of background dominance problem	Develop Feature level-based and Similarity Measure level-based Algorithms	A Weighted Dominant Color Descriptor (WDCD)
	8- Design Compact representation to Color Correlogram Descriptor	Design color-based and distance-based reduction algorithms	A Compact-Generalized Correlogram Method (CGC)
	9- Design and Implement Conversion Method from Large No. of Colors into few Dominant Colors	Design an Adapted Similarity Measure Algorithm of DCs-based Methods	Color Conversion Method from large no. of colors into few DCs (CCM)
	10- Develop Color Indexing Techniques	Develop two color indexing Algorithms	RGB Indexing Method LUV Indexing Method
Descriptive Study II	11- Evaluate Weighted DCD, Compact Correlogram, and Indexing Methods	Accuracy and Efficiency Metrics	Validated WDCD, CGC, CCM, and RGB & LUV Indexing Methods
	12- Validate and Evaluate Weighting and Conversion of DCs Concepts	Applying the Proposed concepts on more than one color descriptor	Generic Frameworks of Weighting DCs and Conversions into DCs Concepts

Figure 3.1: Proposed Design Research Methodology.

Figure 3.1 presents the flow of phases, the methods that are used in each phase and their outcomes, figure details are as follows:

Research clarification (RC): This phase specifies the research problem by identifying the criteria of success. The success criterion of this research is identifying the factors that influence the performance of colour-based CBIR. This is achieved by realization and reviewing content- and colour-based image retrieval methods. Additionally, problems of colour-based methods are also addressed. This is accomplished in Chapter 1 and 2.

Descriptive Study I (DS-I): This phase is dedicated to study and investigate the related work of the specific problems of colour-based CBIR. The outcome of this phase represents the gaps and drawbacks of the current colour methods that affect the CBIR performance. The methods that are used to find these gaps are study the related work of colour descriptors, investigate the current solutions of Correlogram complexity, examine the current conversion into dominant colours methods and analyze the existing colour indexing methods to address the gaps and drawbacks of these methods. These gaps and drawbacks are discussed in Chapter 2.

Prescriptive Study (PS): This phase represents designing and implementation of methods that solve the gaps and drawbacks of current colour-based methods, which are mentioned in *DS-I*. The resulted algorithms of this phase are (i) Weighted Dominant Colour Descriptor (WDCD) and Mutual Colour Ratio (MCR) that are presented in Chapter 4. (ii) Compact-Generalized *Correlogram* method (CGC), adapting the similarity measure of DC-based methods and general Colour Conversion Method (CCM) to DCs, which are explained in Chapter 5. (iii) Two RGB and LUV indexing methods that are detailed in Chapter 6.

Descriptive Study II (DS-II): This phase includes analysis of the results that are produced from the previous phase. This phase concerns with validation and evaluation of the proposed methods through some evaluation metrics (accuracy and efficiency metrics) and comparison with recent works. These evaluation methods are explained in Chapters 4, 5 and 6 with each proposed method. The outcome of this phase represents the validated methods as well as two generic frameworks. The first one is to apply the weighting of DCs concept whereas the second one is to apply colour conversion methods from a large number of colours into few dominant colours. These frameworks are generalized by applying them on more than one colour descriptor.

3.2 CBIR Settings for this Research

The components of CBIR are explained in Chapter 2, Section 2.1. The core components comprise visual feature extraction, similarity measure functions and indexing methods. The remaining components include query formulation, image collections, performance evaluations and CBIR applications. The core components are used to fulfill the research contributions whereas necessity of setting and tuning the other components become inevitable to accomplish *CBIRS*. Therefore, the following subsections are dedicated to the *CBIRS* settings, which are used to complement the research. The justifications behind selection of these settings are also clarified.

3.2.1 Query Formulations

In this thesis, QBE is adopted because it is considered as the most representative way among other query methods (Datta, Joshi, Li & Wang, 2008; Zhang, 2011) such as QBS and other subsequently emerged methods such as interactive method (Fang, Geman & Boujemaa, 2005) and multiple examples query images method (Tahaghoghi, Thom & Williams, 2001; Zhang, 2011).

3.2.2 Visual Feature Extraction

To provide semantic image retrieval, the focus on image objects is inevitable because the object-based image retrieval offers middle level of features instead of the low level features, which suffered from semantic gap problem (Aboulmagd, El-Gayar & Onsi, 2008; Dobrescu, Stoian & Leoveanu, 2010; Eakins & Graham, 1999; King & Lau, 1999). Since colour is considered as a basic cue for object recognition, colour feature is the focus in this research.

In colour features, there are many descriptors ranged from simple descriptors such as colour histogram to complicated one such as colour *Correlogram*. From this variety of colour descriptors, DCD is selected because of its compact property where its size is smaller compared with other colour descriptors. Additionally, it matches the human colour perception thus, allowing the use of few dominant colours while obtaining higher accuracy than descriptors that uses large quantity of colours, as clarified in Section 2.3 and Section 2.4.

To increase *CBIRS* accuracy, combining features is one of the popular solutions (Zhang, 2011). Therefore, salient object detection methods are used to be combined with DCD to enhance its accuracy. The reasons behind this selection is the problem of DCD, which has been tackled, requires an approximate method to detect the object to overcome large background dominance problem that most colour-based descriptors suffered from. In this work, two of the salient object detection algorithms are used. One for natural images and another simple one for cartoon images because the latter type of images is characterized by the object of cartoons are surrounded by bold dark contours (Sykora, Burianek & Zara, 2003, 2005). These algorithms show the best accuracy among the existing algorithms hence they are adopted in this research, as mentioned in Section 2.2.2.

Moreover, colour *Correlogram* and Border/Interior classification (BIC) methods are considered as the best colour descriptors when applied on large image databases (Pedronette & Torres, 2012; Penatti et al., 2012). Therefore, they are selected and enhanced and the generality of DCs weighting and conversion concepts have also been proven in Section 5.5.

3.2.3 Indexing Methods

Different categories of indexing methods are illustrated in Chapter 2, Section 2.4. Colour-based indexing methods frequently used Vector Quantization (VQ) methods to speed up the search. These methods suffer from colour approximation problem, hence, solutions are required. Space partitioning (SP) methods represent the simplest type of indexing methods. Therefore, this type of indexing methods is selected to improve the VQ methods. Accordingly, two SP-based indexing techniques are proposed and will be detailed in Chapter 6.

3.2.4 Similarity Measure

Two similarity measure (SM) methods of *CBIRSs* are explained in Chapter 2, Section 2.1.4, distance-based SM and classification-based SM. Distance-based SM is used in *target search type* where the search depends on pattern matching through matching visual features of the compared patterns. The result of this type of search will be images that are similar to the query. Classification-based SM is used in *category search type* where the search depends on classification methods that use machine learning techniques. The result of this type of the search will be the category that the query image is belonged to. This research adopts the first type of similarity measure, distance-based SM, for several reasons as explained in Section 2.1.4.

3.2.5 Image Datasets

As previously mentioned in Section 2.1.5, type of dataset is mainly depend on the application that is intended to use. Thus, the images datasets that are supposed to be used in this research are datasets that have images of fixed-colour objects. Depending on the characteristics of these images, CBIR features and complexity are determined. The main dataset in this research is cartoon images collection to evaluate the proposed colour descriptors. This is because the characteristic of the most cartoon characters is appearing with the same colours in all or most images (Jiebo & Crandall, 2006; Khan et al., 2012).

Two new cartoon databases are introduced in this work. First one is of size 5K and the second extended one is of size 11K. Cartoon-5K dataset contains 5128 images. This dataset has 85 classes (cartoon characters); each class has at least 50 images. Cartoon-11K dataset contains 11,120 images. Cartoon dataset has 146 classes (cartoon characters); each one has at least 35 images. Cartoon datasets are collected from Google image search[3]. The downloaded images are in different sizes and converted to JPEG format for unifying the type of images in the database. These databases have diversity in their colours and content to reflect the nature of real world databases to be good for testing and evaluation of the proposed descriptors (Clough, Grubinger, Deselaers, Hanbury & Muller, 2007).

Corel dataset are the most commonly used dataset in CBIR field (Müller, Marchand-Maillet & Pun, 2002; Penatti et al., 2012; Thomée, 2010) where it is a general task dataset and is used for various tasks in CBIR. It has been considered, for a long time, as the standard dataset for image retrieval evaluation, specifically in colour-based CBIR, where it is used by many research (Carneiro & Vasconcelos, 2005; Ghoshal, Ircing & Khudanpur, 2005; He, 2004; Hoi & Lyu, 2004; Jeon, Lavrenko & Manmatha, 2003; Kiranyaz, Birinci & Gabbouj, 2010; Kiranyaz et al., 2012; Li, 2005; Li, Bhanu & Dong, 2005; Lin, Liu & Chen, 2005; Neumann & Gegenfurtner, 2006; Rahmani, Goldman, Zhang, Krettek & Fritts, 2005; Rui & Huang, 2000; Thomée, 2010; Wang, Li & Wiederhold, 2001; Yanai, Shirahatti, Gabbur & Barnard, 2005; Yang, Chang, Kuo & Li, 2008; Zhou, Chen & Dai, 2006). Corel image collections are available in many versions, where the first version contains one thousand images. This collection consists of ten

[3] http://images.google.com/

categories where each category has 100 images. It is available for free and it is called the WANG database, henceforth referred to as “Corel-1K”. The second version of this collection contains ten thousand images. It is made up of 80 categories and each category has a minimum of 100 images. This free collection is referred to as “Corel-10K” hereafter. The images of these two versions do not have any keywords other than category name that represents content of all images in this category. In Corel dataset, a wide range of natural images themes is covered. Other versions of Corel image collections are commercially available where they are expensive and are copyright restricted image collections. Since this collection has many categories that have objects of the same colour such as horses, elephants, buses and others, it will be used in this thesis as a benchmark for measuring performance of colour descriptors.

Caltech datasets include two common datasets. The first one is called Caltech-101[4] which contains 101 categories whereas the second one is extended to 256 image categories. These datasets are designed to evaluate object recognition and classification tasks (Deng, Zhang, Mortensen, Dietterich & Shapiro, 2007; Griffin, Holub & Perona, 2007; Hegazy & Denzler, 2008; Malisiewicz & Efros, 2007; Russell, Freeman, Efros, Sivic & Zisserman, 2006), which are close to the objectives of the proposed descriptors in this thesis. Therefore, some categories of Caltech-101 dataset (which objects have the same colour in most images in their categories) are selected as benchmark in this thesis. The reason of using Corel and Caltech datasets in this research, and especially in colour-based task, is there are a lot of previous studies that used these datasets in such purpose, as

[4] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

mentioned previously in this Section. Datasets that used in this research are summarized in Table 3.1.

Table 3.1: *Summary of the datasets that are used in the research.*

Dataset Name	No. of Classes	No. of images in each class	Total images
Corel-1K[5]	10	100	1000
Caltech-101[6]	26	minimum is 35	2562
Cartoon-5K	85	minimum is 50	5128
Corel-10K[7]	80	minimum is 100	10,800
Cartoon-11K	146	minimum is 35	11,120

The size of the database that should be used to evaluate image descriptors should not be too small to be useful as a benchmark where the collection should have a significant number of images. The benchmark database should consist of at least 1000 images (Leung & Ip, 2000). On the other hand, a database that is too large would also be impractical because content-based retrieval will require time for indexing (feature extraction) and consequently an indexing of query image and similarity measure with all database images will require long time (some minutes per query) (Leung & Hibler, 1991). Thus, images collections with more than 20,000 images are considered as infeasible for applications that depend on the content only (Leung & Ip, 2000). Other studies confirm that the database size is ranged from 1,000 to 20,000 (ImageCLEF, 2003; Vailaya, Figueiredo, Jain & Zhang, 2001; Zhang, 2011). Therefore, evaluation databases of this research are opted within this range to comply with the aforementioned benchmarking requirements.

[5] http://wang.ist.psu.edu/
[6] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[7] https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval

3.2.6 Performance Evaluation

Performance evaluation of *CBIRS* comprises three criteria as in the following subsections. Additionally, the competing methods are considered as one of the important criteria to prove the effectivness of the proposed methods over the existing and recent related methods.

A. Ground Truth

There are three methods to formulate the ground truth for database: 1) using database of predefined image categories; 2) image grouping; and 3) user judgment (Müller, Müller, Squire, Marchand-Maillet & Pun, 2001). In the first method, all groups of relevant images are already categorized into separate classes. The second method is dedicated for unstructured databases. Handling unstructured database can be achieved by learning-based techniques through two methods, supervised methods such as image classification and unsupervised methods such as image clustering where both of them can organize an unstructured image database (Datta et al., 2008). The last method uses experts to categorize the classes of the database. The first method is adopted in this research due to many reasons. The first reason is it represents the easiest method because the relevant images are already defined and categorized in different classes whereas the other two methods need to explore the whole database either by machine learning algorithms or by the experts to categorize the classes of the database. Additionally, these methods are time consuming particularly in the large database.

Moreover in all the three methods, image categories may contain not only visually similar images but also semantically similar images. In other words, images may have similar object (‘flowers’ for example), this means that these images are semantically similar. But the colour of flowers is different (some images have flowers of red colour whereas some images have flowers of yellow or white colours), this means that these images are visually not similar although they are semantically similar. This will decrease the accuracy of *CBIRS* that depend only on the visual similarity due to the limited capabilities of the current methods (Zhang, 2011). Thus, database must be cleaned by experts to avoid this situation (Zhang, 2011). In this research two cleaning methods are used. First one is forming a sub-database from the original dataset which contains classes of certain objects. The property of these objects is they have same colours in all image of this class such as tiger and zebra classes. This type of cleaning is called as *database cleaning*; it was the cleaning method that applied on the Caltech-101 database as mentioned in the Section 3.2.5. The second type of cleaning is performed when the whole database is considered in evaluation process, there is no database cleaning. It represents selecting samples from the suitable classes only to be queried in the retrieval system instead of selecting from all database classes; it can be called as *query cleaning*. This type of cleaning is performed in Corel database in the experiments of this research.

B. Evaluation Metrics

In this work, four quantitative metrics are selected to verify performance of the proposed methods. These metrics are cut-off precision value, P(10), average retrieval rank (ARR), average normalized modified retrieval rank (ANMRR) and the mean of average precision (MAP). Details and equations of computing these metrics as well as the previous works that used this metric are explained in Section 2.1.6 (A).

The reason of selecting the above four metrics as evaluation measures is that they measure the performance of retrieval system in different ranges, as depicted in the Figure 3.2, where P(10) measures the retrieval performance in the top 10 retrieved images, ARR measures the performance in the top N ranks images, N represents number of ground truth images of specific query. ANMRR measures the system performance through the top 2*N ranks images and finally MAP refers to the performance of the system through measure the precision of all relevant images (N_G) to the query over all database images. These metrics highlight the thesis' contributions in different perspectives where the applications that prefer specific metric can be determined. In other words, a specific application can use the contributions of this research if the results of their preferred metrics are good. For example, web-based applications prefer P(10) metric because the Internet users look to the first results page and prefer to reformulate the query instead of checking the second results page. Additionally, when searching is on a specific person in the criminals' database, for example, MAP metric is preferred because the police officers must investigate all retrieved images to find the wanted person (Penatti, 2012).

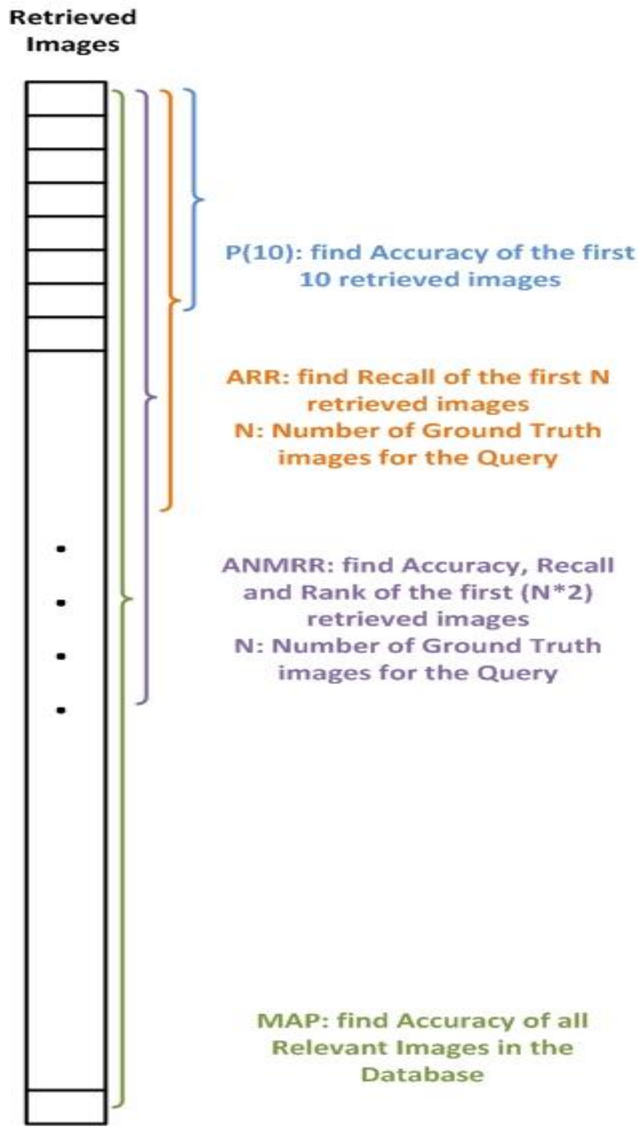


Figure 3.2: Accuracy metrics that are used in this research

C. Number and type of Queries

In Section 2.1.6 (B), there are many arguments presented to determine the numbers of queries to evaluate *CBIRS*. Therefore, number of queries in this thesis is selected to be more than 1% of database size. Moreover, diversity of queries is very important to ensure fair and honest results (Grubinger, 2007; Penatti et al., 2012). Therefore, the evaluation queries are selected from all classes of the database in this work except some cleaning is performed on the queries in Corel dataset as explained in Section 3.2.6 (A).

D. Competing methods

The competed descriptors that can be used to evaluate any proposed methods are standard methods of the proposed method, recent methods that have the best accuracy among all methods in the scientific fields, or both methods (Penatti et al., 2012). In this thesis, for first contribution, both standard and best recent methods are used for comparison with the proposed methods, which are MPEG-7 DCD (Yamada et al., 2001) and LBA DCD (Yang et al., 2008) repectively. In the second contribution, also both standard and recent correlogram descriptors are used for evaluation purpose. These methods are the original ColGrm (Huang et al., 1997; Kunttu et al., 2003), Autocorrelogram (Huang et al., 1997) and MPEG-7 Dominant Colour Descriptor (DCD) (Yamada et al., 2001) as standard methods and perceptual ColGrm (Kiranyaz et al., 2012) as recent method. In the last contribution, indexing methods, the method of evaluation is similar to the previous contributions. These competing methods are sequential search, K-means (Hartigan & Wong, 1979; Maimon & Rokach, 2005) as standard methods and recent K-means with B+-tree methods (KMB) as recent method (Yildizer et al., 2012).

3.2.7 CBIR Applications

CBIRS have been used in many types of applications. Depending on the application, *CBIRS* complexity will be determined and visual features will be selected. Some of potential applications that can be benefit from this research are illustrated in Section 7.3.

3.3 Summary

This chapter presents the methodology of this research (DRM) and discusses its phases to accomplish the objectives of this research. It discusses the settings of *CBIRS* components to meet the requirements of the application of this thesis. Query formulation method is identified. Selection of CBIR settings for core components, which are visual features, indexing methods and similarity measure functions, are justified. Evaluation metrics that are used to evaluate the proposed methods are also selected. Image collections that can be used to evaluate the work are determined. The design research methodology and settings of *CBIRS* represent the preamble to the details of research contributions that are presented in the next three chapters.

CHAPTER FOUR
A WEIGHTED DOMINANT COLOUR DESCRIPTOR

In this chapter, a DC-based semantic feature is proposed that determines a weight for each DC in the image. The resulted descriptor of this feature is called Weighted Dominant Colour Descriptor (WDCD). The proposed feature helps reduce the effect of image background on image matching decision where an object's colours receive much more focus. In addition, a modification to DC-based similarity measure is proposed. Such a modification includes adding a new term, the Mutual Colours Ratio (MCR), which improves the performance of the dominant colour descriptors because it alleviates background effects by adding MCR to image similarity measure. Section 4.1 presents the problem of current colour-based methods. Section 4.2 explains WDCD that introduces feature level-based solution to the background effect problem. Section 4.3 introduces a modification term, MCR, which is considered as a similarity measure level-based solution to the aforementioned problem. In Section 4.4, experimental results of the WDCD and MCR are reported. Finally, Section 4.4 summarizes this chapter.

4.1 Current Problem of DC-based CBIR Methods

Although extracting proper DCs can solve the problems of colour histogram (high dimensional and human perceptual problems), DCD (colours values and their percentages in the image) still lacks a good description about the object in the image (same histogram problem). Such a thing happens particularly when the background colour has large percentage among the dominant colours of the images. That is to say, it lacks the semantic information in its representation. In other words, most of the histogram-based and DC-based image retrieval approaches cannot apply to object recognition problem (Matas, Koubaroulis & Kittler, 2000). This is because the object occupies a portion (small or large) in an image (Das et al., 1997). The process of retrieving the images depends basically on the colour that occupies large area in the image (i.e.: the long bin in colour histogram or DC that has the large percentage of the image area). Actually this problem is also found in the most accurate colour-based methods such as colour *Correlogram* (Huang, Kumar, Mitra, Zhu & Zabih, 1997; Kunttu, Lepistö, Rauhamaa & Visa, 2003) and Border-Interior Classification (BIC) method (Renato et al., 2002). Even though these methods use advanced colour features like spatial correlations among colours (as in *Correlogram*) and somehow shape information (as in BIC), they still have the same large background effect problem as demonstrated in Chapter 5. Two modest solutions have been proposed in the literature to solve this issue. The first one was introduced by Krishnan and Christiyana (2007) and it was feature level-based solution. It is assumed that lighter colour in the image represents the object colour and the darker colour is the background but this assumption is uncertain for various image contents. Moreover, it depends only on the largest colour percentage of the object (only one colour) whereas the object may contain many small percentage colours. The second solution was proposed by Renato et al. (2002). It used similarity measure level-based solution, logarithm distance (*dLog*), to solve this problem but this method still suffers from the large percentage dominance problem, as discussed in the next chapter. In this chapter, two robust solutions are proposed to solve large background dominance problem.

4.2 Weighted Dominant Colour Descriptor

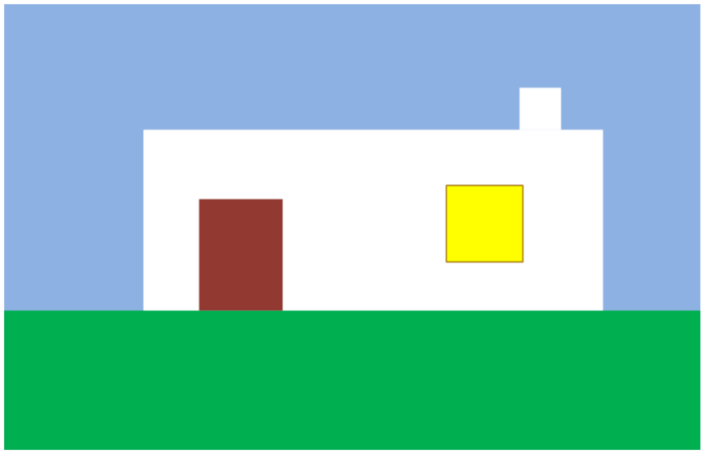
In the previous section it has been established that determining object and background colours are the main issues of this chapter. To obtain object DCs, two methods that depend on two assumptions need to be applied. Firstly, some studies show that the object is located at the centre of an image and that its size is approximately 25% from the image size only (Kim et al., 2003; Rodhetbhai, 2009). This is supported by the claim that photographers tend to locate the object in the middle of the picture when they snaps a photo (Kim et al., 2003). They also showed that the background colour will be distributed in the corners and borders of the images (Kim et al., 2003; Rodhetbhai, 2009). Therefore, this hypothesis is taken into account to extract object DCs where the DC that appears on the image border mostly does not belong to the object. An explanation of this method is detailed in Section 4.2.1 and its outcome is called the Border Weight of DC (BW-DC).

In the previous assumption, it is hard to semantically identify an object based on the extracted feature. Instead, it can be integrated with the second assumption, which is Salient Object Detection (SOD) method to be used as complementary part. There are two salient object detection methods used in this research (Talib, Mahmuddin, Husni & George, 2013b). The first detection method is used for natural images, known as the Global Contrast based Salient Region Detection (GC-SRD) that is proposed by Cheng et al. (2011) because of its effectiveness compared with other saliency methods. The second salient object detection method is used for cartoon images, due to its different characteristics, which is the Laplacian of Gaussian (LoG) filter with flood fill algorithm (Yu & Seah, 2011; Yu, Cheng & Tao, 2012).

An integration of BW-DC and SOD is necessary to detect the object in the image because the first assumption that the object is normally located at the centre and that the object does not touch the border is not true for all images. Although the effectiveness of the SOD method motivates its use in this research, some limitations do exist it is effective for single object only (as in GC-SRD) and it is not working with very complicated background (as in LoG). Therefore, importance (weight) of each DC in the image is extracted (according to its belonging to the object) and added to DCD. This help produces the semantic DCD, which is called the Weighted Dominant Colour Descriptor (WDCD), for content-based image retrieval.

4.2.1 Border Weight of Dominant Colours

The previous section illustrates the assumption that an object is located at the centre of the image while the background colour is distributed to corners and boundaries. Such a step includes extracting the dominant colours with their percentages from the image via using MP7DCD (Yamada et al., 2001) or fast LBA (Yang et al., 2008). Then, the weight of each DC (resulted from MP7DCD or LBA) is computed depending on its existence in the image border, where the weight of DC equal to frequency of it on the border, hence it is called as border weight (BW). The colour that has high frequency at the image borders will obtain a higher weight. This means that it is considered as the background colour. In addition, each colour that has low frequency in the border will obtain a lower weight (considered as object colour), as shows in Figure 4.1.



** Border Weight (blue) =0.65, BW (green) =0.35, BW (white) =0, BW (brown) = 0 and BW (yellow) =0.*

Figure 4.1: The way an image gives weight to each DC depending on whether it is an object or background.

One can notice from Figure 4.1, the way that the weight is given to each DC depends on the percentage of each colour in an image border. Each DC colour appears with a high frequency in the border; where it represents the background colour and obtains a high BW (BW=1 if the image has one background DC where its frequency is equal to border length). Moreover, the DC that does not appear in the border (frequency=0) gets a low weight (BW=0). It represents an object DC in the image (as in colours White, Brown and Yellow in Figure 4.1).

The calculation of BW of all DCs in Figure 4.1 can be illustrated as follows (image width=150 and image height=100). The border of an image comprises two horizontal and two vertical edges (i.e.: border length= (width + height)*2= (150+100)*2= 500). The frequency of the blue colour in the border is equal to 325 ($Freq_{blue}$=325), the frequency of the green colour in the border is 175 ($Freq_{green}$=175) whereas the frequencies of all the other colours (white, brown and yellow) are equal to zero. This is because they do not appear in the border ($Freq_{white}$=$Freq_{brown}$=$Freq_{yellow}$=0). Intuitively, one can notice that the blue and green colours represent the background whereas white, brown and yellow represent an object of an image. Therefore, the border weight of each DC can be computed by taking into account the frequencies on the image border, as shown in the following formula:

$$BorderWeight_{DC} = \frac{Freq_{DC}\,(Border)}{BorderLength} \tag{4.1}$$

From Eq. 4.1, one can compute the weight of all DCs in Figure 4.1, as stated below:

$$BorderWeight_{blue} = \frac{Freq_{blue}}{BoarderLenght} = \left(\frac{325}{500}\right) = 0.65$$

$$BorderWeight_{green} = \left(\frac{175}{500}\right) = 0.35$$

$$BorderWeight_{white} = Weight_{brown} = Weight_{yellow} = \left(\frac{0}{500}\right) = 0$$

Based on the above formula, one can notice that the DCs, which represent the background, obtain a higher BW than the DCs that represent the object. This first step state helps reduce the effect of background in similarity decision and gives some semantic information to the DCD by giving more importance to the object.

Nevertheless, there are some cases that conflict and refute this assumption. In the first case, some images have a large object that may touch the border of an image (as presented in Figure 4.2a). This case considers the object's DC as background's DC and thus the importance of object's DCs will be reduced. In the second case, the background colour has the same object colour (as presentd in Figure 4.2b); this will remove the object from consideration as well as the background. In the third case, there is a thin line surrounding the image (as present in Figure 4.2c); this will consider false background colour and the original background colour will consider as object colour. Therefore, salient object detection method can be used to determine and solve the conflicted cases and complement the proposed BW method.

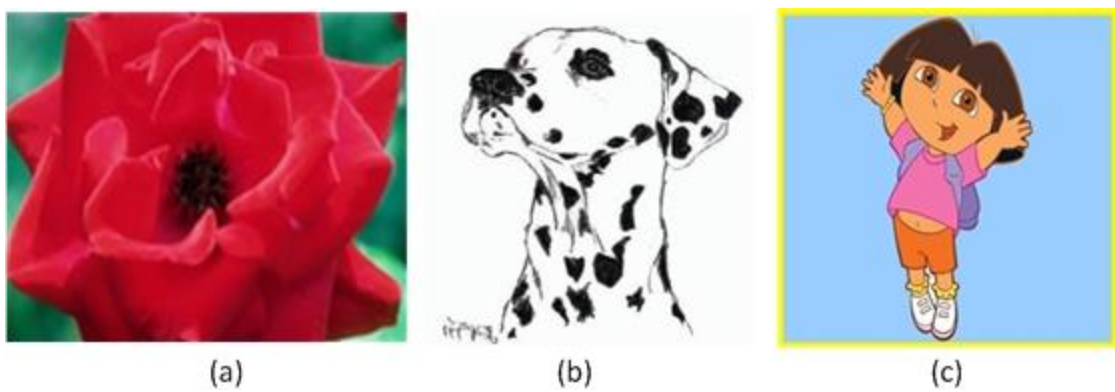


Figure 4.2: Three images that explain the conflict cases with border weight method.

Computing the border weight of DCs of an image can be achieved by using the following proposed algorithm:

Algorithm 4.1: BorderWeights-DC

Input

Wid, *Hig*: width and height of an input image

DC: Array of dominant colours values of an input image

N: number of dominant colours in an input image

Output

BorderWeight: Array of border weights for all DCs in the input image

BEGIN

1. Set frequency of each DC on the border to zero

$$Freq_k \leftarrow 0 \quad k = 0..N-1$$

2. Find frequency of each DC on the image border
   - { for each DC in the Image}
     $\forall\, DC_k \in DC \quad k = 0..N-1$
     - $\{find\ frequency\ of\ DC\ in\ horizontal\ border\ edges\}$

       $$if\ \left(DC_k == Pixel_{i,0}\ OR\ DC_k == Pixel_{i,hig-1}\right)$$

       $$Freq_k ++; \qquad i = 0..wid-1$$

     - $\{find\ frequency\ of\ DC\ in\ vertical\ border\ edges\}$

       $$if\ \left(DC_k == Pixel_{0,j}\ OR\ DC_k == Pixel_{wid-1,j}\right)$$

       $$Freq_k ++; \qquad j = 0..hig-1$$

3. Compute weight of each DC from their frequencies

$$BorderWeight_k \leftarrow \frac{Freq_k}{(wid+hig)*2} \qquad k = 0..N-1$$

4. Return $(BorderWeight_k)$

END

4.2.2 Salient Object Detection

The complement and second part of WDCD is detecting salient object in the image and giving weights to its DCs where these weights are called Salient Object Weights (SOW). The extraction of image salient object is illustrated in Figure 4.3.

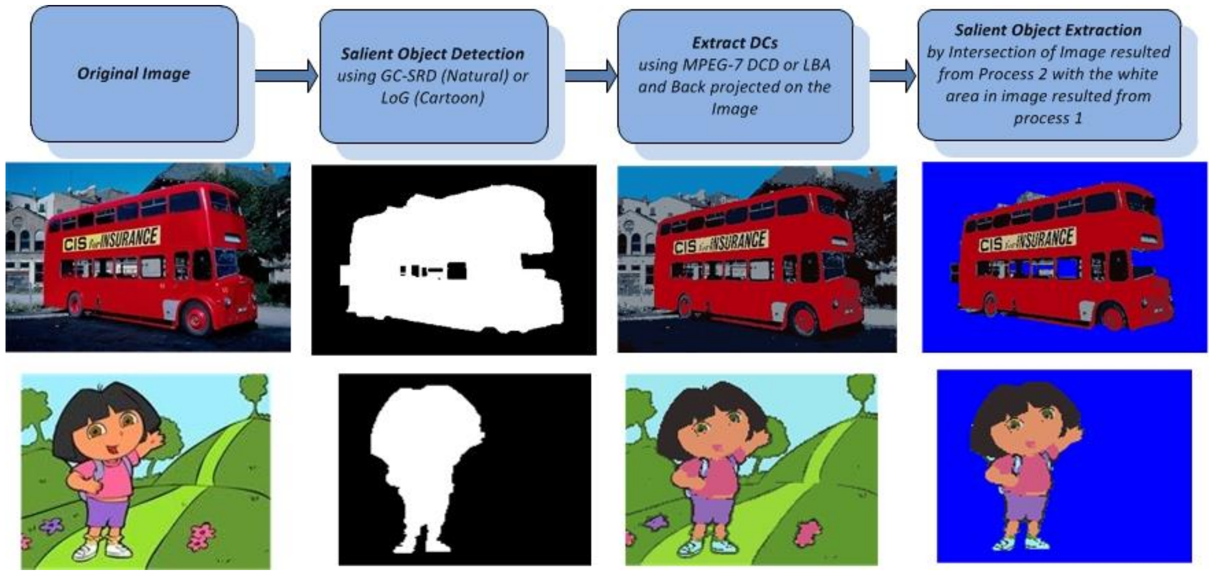


Figure 4.3: Two images examples on salient object extraction steps.

SOW can be computed by finding the ratio of frequencies of DCs in the extracted object to the total pixels of the object, as presented in Eq. 4.2, and its steps are depicted in Algorithm 4.2.

$$SalientObjectWeight_{DC} = \frac{Freq_{DC}\,(Object)}{No.of\ Pixels(Object)} \qquad (4.2)$$

Algorithm 4.2: Salient Object Weight

Input

wid, hig: width and height of an input image

DC: Array of dominant colours values of an input image

N: number of dominant colours in an input image

Output

SOW: Array of salient object weights of all DCs in the input image

BEGIN

1. Set frequency of each DC in the object to zero

$$Freq_k \leftarrow 0 \quad k = 0..N-1$$

2. Find frequency of each DC in the extracted object
   - { for each DC in the Image}

     $\forall\, DC_k \in DC \qquad k = 0..N-1$

     - $\{find\ frequency\ of\ DC\ in\ the\ object\}$

       $if\ (DC_k == Object'sPixel_{i,j})$

       $Freq_k++; \qquad i = 0..wid-1, j = 0..hig-1$

3. Find size of object in the image
   - {for each pixel in the image}

     $\forall\, i \in 0..wid-1$

     $\forall\, j \in 0..hig-1$

     - $\{Compute\ number\ of\ pixels\ in\ object\ (which\ is\ not\ blue)\}$

       $if\ (Pixel_{i,j} \neq Blue\ Color)$

       $ObjectSize++;$

4. Compute weight of each DC from their frequencies

$$SOW_k \leftarrow \frac{Freq_k}{ObjectSize} \qquad k = 0..N-1$$

5. Return (SOW_k)

END

It is worth mentioning that the input image of salient object extraction process is the original image instead of the image that resulted from DCD, as presented in Figure 4.3. This is because the saliency regions and edges faded away in the DCD's resulted images, hence triggering an important question - Is it possible to use SOD methods alone to determine the object and its colours? If so there is no need to compute border weight. The answer was mentioned previously in Section 4.2, that the current SOD methods are inaccurate for detecting the object, or objects, in complicated or high-detailed images as shown in Figure 4.4.

Figure 4.4: Examples to illustrate the inaccuracy of the SOD methods, (a) GC-SRD and (b) LoG.

4.2.3 Extracting Final Weight of Dominant Colours

Lastly, to extract final weights of DCs, the three weights are taken into account. These weights are BW, SOW and DC's percentages (or weights) in the DCD's resulted image (DCW). For each of the three input weights, there are two symbols, either "Large" (L) or "Small" (S) that is used to describe them depending on specific thresholds. The thresholds, which were extracted experimentally to be considered in this work, are 0.05, 0.10 and 0.10 for SOW, BW and DCW respectively, as depicted in Eq. 4.3. Additionally, description (either "L" or "S") of all these weights can be denoted as DSOW, DBW and DDCW respectively as presented in Eq. 4.3 and Table 4.1. Since there are three weights, therefore there are eight possible cases of their combinations. To describe these cases, it is necessary to mention the indication of using each weight where SOW of certain DC with symbol "L" indicates that DC belongs to the object with high percentage otherwise it belongs to the object with low percentage. BW with symbol "L" also indicates that DC belongs to border (background) with high percentage otherwise it belongs to the background with low percentage. Finally, DCW with symbol "L" indicates that this DC has large percentage in the image; otherwise it refers that this DC has small percentage in the image. Final weights of DCs can be computed as in Table 4.1.

$$DSOW = \begin{Bmatrix} L & if\ SOW > 0.05 \\ S & if\ SOW \leq 0.05 \end{Bmatrix},\ DBW = \begin{Bmatrix} L & if\ SOW > 0.10 \\ S & if\ SOW \leq 0.10 \end{Bmatrix},$$

$$DDCW = \begin{Bmatrix} L & if\ DCW > 0.10 \\ S & if\ DCW \leq 0.10 \end{Bmatrix} \qquad (4.3)$$

Table 4.1: *Final DC's Weights extraction from original image using three inputs – SOW, BW and DCW.*

Case No.	DSOW	DBW	DDCW	Final DC Weight	Case Description
1	L	L	L	**Max(SOW, 1-BW, DCW)**	Confused DC, it belongs to Object and Background with high percentage (e.g. Figure 4.2 a, b).
2	L	L	S	**Max (SOW,1-BW)**	Same as above but has small percentage in the image.
3	L	S	L	**1**	Confirmed DC, it represents big Object.
4	L	S	S	**1**	Confirmed DC, it represents small Object.
5	S	L	L	**1- BW**	It represents a background DC of large percentage, it must obtain low weight.

6	S	L	S	**1- BW**	It may be a thin line around the image, it must be ignored (e.g. Figure 4.2 c).
7	S	S	L	**DCW**	Confused DC, hence we consider its percentage in the image.
8	S	S	S	**DCW**	Same as above but it is a small object.

The final weight of DC represents an importance of this DC in the image. That means, if it is high (≈1) then the DC belongs to the object and must be considered. If it is low (≈0) then it belongs to the background and must be removed from consideration when computing similarity measure. Any other values, when the weight value is high, indicate the importance of this DC, which means that the DC is important to be considered. From Table 4.1, one can notice that there is no zero value set to the DC weight in all cases. This is because zero value will remove the DC completely from consideration whereas this colour may have some importance, but there might be a mistake when its weights (BW or SOW) are computed. Hence, zero weight value is avoided and it may be resulted implicitly through the cases of Table 4.1 (for example: image in Figure 4.2c, yellow colour recieves weight equal to 0 because it matches case 6).

Case 1 (“L” for three input weights “LLL”) indicates that the DC belongs to the object and background with high percentage and it has large percentage in the image. To determine if the DC belongs to the object or background is a confusing task. In this case, the maximum among these weights of DC is selected to represent its importance in the image because it may be a large object that covers large area in the image (such as the image I in Figure 4.2a), thus its weight must be selected carefully. BW represents background weight of certain DC, so it cannot be compared with SOW, thus its reverse (1-BW) is considered to represent its object weight regarding to border. Case 2 (“LLS”) is similar to case 1 but the DC has small percentage in the image, hence the maximum between two object weights (SOW, 1-BW) is considered as final weight. Case 3 and 4 (“LSL” and “LSS”) refer to the cases where the DC appears obviously in the extracted salient object and disappear from the border. Hence it is confirmed as object colour and must obtain full importance by giving it the value of “1” as final weight. Case 5 and 6 (“SLL and SLS”) refer to the case where DC appears with large percentage in the background and does not appear in the object. Hence it is considered as background colour and its effect from similarity measure is removed by giving it a low weight (1-BW). DC in case 5 represents background colour that has large percentage in the image while DC in case 6 represents just a thin line surrounding the image and it should be ignored as shown in the Figure 2c. Case 7 (“SSL”) represents the case where DC does not appear in the object area nor in the background area but it has large percentage in the image. Removing it from consideration may lead to a big mistake because it may be a missing object; thus its percentage in the image is considered as final weight. Case 8 (“SSS”) refers to the same previous case but DC is a small percentage; hence, a low weight is given.

4.2.4 Similarity Measure of the Proposed WDCD

MPEG-7 DCD (Yamada et al., 2001) used quadratic distance to compute the dissimilarity measures between the two images, as shown in Eq. 4.4.

$$D_Q(I_1, I_2) = \sum_{i=0}^{N-1} p_i^2 + \sum_{j=0}^{M-1} p_j^2 - \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} 2a_{i,j}\, p_i p_j \tag{4.4}$$

where I_1 and I_2 represent the two images whose similarities are required to be measured. After the process of extracting the DCs of the images, the latter can represent the features, as the following: F1= {(C_i, P_i), i=0,…,N-1} represents features of I_1 that has N dominant colours; F2= {(C_j, P_j), j=0,…,M-1} represents features of I_2 that has M dominants colours, C and P represent colour value and the percentage of each DC in the image, respectively. Finally, $a_{i,j}$ represents the similarity coefficient between the colours C_i and C_j. It can be computed using Eq. 4.5.

$$a_{i,j} = \begin{cases} 1 - \dfrac{d_{i,j}}{d_{max}} & if\ d_{i,j} \leq Th_d \\ 0 & if\ d_{i,j} > Th_d \end{cases} \tag{4.5}$$

where $d_{i,j}$ represents Euclidean distance between C_i and C_j, the abbreviation C represents the 3-D colour values (in CIE-Luv colour space), which can be computed in Eq. 4.6.

$$d_{i,j} = |C_i^L - C_j^L| + |C_i^U - C_j^U| + |C_i^V - C_j^V| \tag{4.6}$$

The threshold Th_d represents the maximum distance whereby the two colours are considered similar, and d_{max}= α Th_d, α=1.2. The latter state assumes that the maximum distance between the two colours is slightly greater than colour threshold. As it is stated previously, this distance has serious drawbacks. Accordingly, it does not satisfy human perception (Po & Wong, 2004; Yang et al., 2008). Therefore, Yang et al. (2008) proposes a new efficient similarity measure for DC as shown in the following equations:

$$S_{i,j} = [\, 1 - |p_i(i) - p_j(j)|\,] \times \min\left(p_i(i), p_j(j)\right) \tag{4.7}$$

$$SIM^{\,yang}(I_1, I_2) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} a_{i,j}\, S_{i,j} \tag{4.8}$$

$$D_{yang}(I_1, I_2) = 1 - SIM^{yang}(I_1, I_2) \tag{4.9}$$

In Eq. 4.7, *p* represents the percentage of DC in the image, $S_{i,j}$ refers to the similarity between colour percentages. On the other hand, $a_{i,j}$, in Eq. 4.8, represents colour similarity between the two colours C_i and C_j, as indicated in Eq. 4.5. In Eq. 4.8, $SIM^{yang}(I_1,I_2)$ represents the similarity ratio of the two images. Finally, to measure the dissimilarity between the two images, one can use Eq. 4.9. Yang et al. (2001) pinpointed that such a measure resembles the mechanism of human perception of colours. Besides, it helps to overcome problems of quadratic distance and proved its efficiency over the two improvements of quadratic distance as proposed by (Ma et al., 1997) and (Mojsilovic et al., 2000).

Therefore, the present similarity measure conducted in this research depends on D_{yang} dissimilarity distance. However, it is modified to be able to obtain the proposed semantic feature, which is the weight of DCs that is based on either belonging to the object or background. The adapted similarity measure can be formulated using the Eq. 4.10 and 4.11.

$$W_{i,j} = \min(W_i, W_j) \tag{4.10}$$

$$SIM^{W}\,(I_1, I_2) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} a_{i,j}\, S_{i,j}\, W_{i,j} \tag{4.11}$$

where $W_{i,j}$ in Eq. 4.10 represents the intersection of weights of the two DCs, *i* and *j*, which in return represent the smaller weights. That is, it reduces the importance of the colour if it represents the background colour in either the first or second image. As for the DC's weights (W_i and W_j), they can be computed by the Table 4.1.

The $SIM^{yang}(I_1,I_2)$ in Eq. 4.8 helps compute the similarity between the two images. It is further increased whenever the images are more similar (in terms of colour value $a_{i,j}$ and percentages $S_{i,j}$). In Eq. 4.11, the DC's weight is multiplied with the formula to decrease the consideration of the background DC or keeps them as object. This is done by multiplying them with a low weight or 0, or multiplying them with high weight or 1, respectively. Finally, the dissimilarity measure that depends on the DC's weight (D_W) can be computed as follows:

$$D_W(I_1, I_2) = 1 - SIM^W(I_1, I_2) \qquad (4.12)$$

To illustrate the effectiveness of the proposed method in object-based image retrieval compared with that of Yang et al. (2008) and his dissimilarity distance (D_{yang}), a cartoon image example is presented, as shown in Figure 4.5.

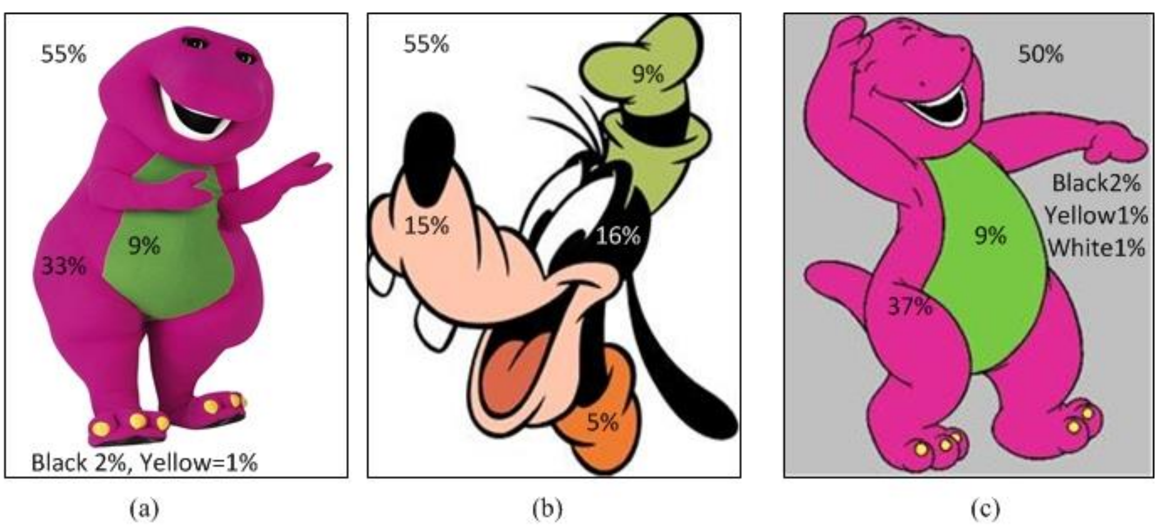


Figure 4.5: Examples to show the effectiveness of the proposed WDCD in object-based image retrieval.

As it is shown previously, SIM^{yang} (I_1,I_2) or $SIM^W(I_1,I_2)$ are used to measure the similarity between the two images. On the other hand, D_{yang} or D_W is dissimilarity measure that is used for the retrieval purpose. Such a step is performed by inversing the result of *SIM*, by $D(I_1,I_2)$=1-$SIM(I_1,I_2)$ to make the distance small for all the similar images and large for all the dissimilar images. Therefore, SIM can only be computed, as presented in Table 4.2 and Table 4.3, to show the similarity among the cartoon images that are presented in Figure 4.5.

Table 4.2: *Values of SOW, BW, DCW and Final Weights of DCs in the three cartoon images presented in Figure 4.5.*

		White	Green	Pink/ Orange	Black	Yellow/ PeachPuff	Gray
Image1 (a)	SOW	0.02	0.20	0.73	0.04	0.02	N/A
	BW	1	0	0	0	0	N/A
	DCW	0.55	0.09	0.33	0.02	0.01	N/A
	FinalW	**0**	**1**	**1**	**0.02**	**0.01**	
Image2 (b)	SOW	0.03	0.20	0.11	0.36	0.33	N/A
	BW	0.96	0	0	0.04	0	N/A
	DCW	0.55	0.09	0.05	0.16	0.15	N/A
	FinalW	**0.04**	**1**	**1**	**1**	**1**	
Image3 (c)	SOW	0.02	0.20	0.74	0.04	0.02	0
	BW	0	0	0	0	0	1
	DCW	0.01	0.09	0.37	0.02	0.01	0.50
	FinalW	**0.01**	**1**	**1**	**0.02**	**0.01**	**0**

Table 4.3: SIM^{yang} *(Eq. 4.8) and* SIM^{W} *(Eq. 4.11) similarity measure for Figure 4.5*

	White	Green	Pink	Black	Yellow	PeachPuff, Orange, Gray	Overall Similarity
$SIM^{Yang}(I_1,I_2)$	[1-0]* 0.55	[1-0]* 0.09	0	[1-0.14]* 0.02	0	0	**0.65**
$SIM^{Yang}(I_1,I_3)$	[1-0.54]* 0.01	[1-0]* 0.09	[1-.04] * 0.33	[1-0]* 0.02	[1-0]* 0.01	0	**0.44**
$SIM^{W}(I_1,I_2)$	[1-0]* 0.55* 0	[1-0]* 0.09* 1	0	[1-0.14] * 0.02 * 0.02	0	0	**0.090**
$SIM^{W}(I_1,I_3)$	[1-0.54]* 0.01* 0	[1-0]* 0.09* 1	[1-.04] *0.33* 1	[1-0]* 0.02* 0.02	[1-0]* 0.01* 0.01	0	**0.41**

Referring to Table 4.3, one can notice that SIM^{yang} depends basically on the colour that has the largest percentage regardless whether it represents the background colour or the object colour. Therefore, the similarity value of image 1 (Barney with White background) and image 2 (Goofy with White background) is larger than that of image 1 (Barney with White background) with image 3 (Barney with Gray background). Using the adapted method, one can notice that its SIM^{W} helps remove the effect of background on similarity value. The reduction is based on the object colour and is obtained by multiplying the background colour by 0, as shown in the column “white” in Table 4.3. Additionally, the proposed method depends on the largest colours of object and alleviates effect of the small percentage colours such as “Yellow” and “Black”.

4.3 Similarity Measure Level-based Solution

To improve the retrieved results, similarity measure level-based solution is proposed in this section. One can modify the similarity measure by adding the ratio of the mutual colours between the two compared images. This modification has a number of advantages: 1) it alleviates the major dependencies of the similarity measure on the largest percentage DC. This modification takes other colours into consideration even though these colours have small percentages; 2) it enhances the rank of the most similar relevant images by shifting the images that have similar colours up and shifting the others down. This modification enhances retrieving the images; especially, when the image classes that have more than one related colours, such as the images of beach that are frequently come with the sky and sand regardless the colours of objects that exist on the beach (people or trees). From another perspective, this modification will conflict with the proposed WDCD because the latter concerns with object only regardless of its background. As example, the category of elephant images in Corel dataset comes in two different backgrounds, sky and water and grass and tree. Hence, MCR will decrease the similarity between two images that have different backgrounds even though they have the same object. Therefore, little change is performed on this ratio to suit the proposed WDCD. This modification, enhances the result of MP7DCD (Yamada et al., 2001), Yang (Yang et al., 2008) and the proposed descriptor. The MCR of the two images DCs can be computed using Algorithm 4.3.

Algorithm 4.3: Mutual Colour Ratio

Input

DC1, DC2: Arrays of dominant colour values of the two compared images

W1, W2: Arrays of Final Weights of DCs for the two compared images

N, M: number of dominant colours in the two compared images

Output

MCR: Mutual Colour Ratio of the two images

BEGIN

1. Set Mutual Colours counter to zero

$$MC \leftarrow 0$$

2. Find mutual colours by pass on all DCs of two images
   - { for each DC in the Images}

     $\forall\, dc_i \in DC1 \quad i = 0..N-1$

     $\forall\, dc_j \in DC2 \quad j = 0..M-1$

     - $\{ count\ number\ of\ similar\ colors\ between\ two\ images \}$

       $if\ \left(distance\left(dc_i, dc_j\right) < Th_d\right)$

       $MC++;$

       or

       $MC = MC + \min\left(W1_i, W2_j\right);/*\ for\ \ WDCD\ \ */$

3. Compute ratio of mutual colours of two images

$$MCR \leftarrow \frac{MC}{Max(M,N)}$$

4. Return (MCR)

END

As shown in Algorithm 4.3, W1 and W2 represent the final weights of the two compared images. These weights are used to compute MCR for the proposed WDCD where the MCR for WDCD considers weights of colours instead of “1” during calculation of its value to overcome the aforementioned conflict with WDCD. The similarity measure of the proposed approach (Eq. 4.11) is modified by adding MCR as shown in Eq. 4.13.

$$SIM^{W}\,(I_1, I_2) = \left(\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} a_{i,j}\,S_{i,j}\,W_{i,j}\right) * MCR \tag{4.13}$$

The similarity measure of MP7DCD (Eq. 4.4) and Yang (Eq. 4.8) can also be modified as illustrated below:

$$D_Q(I_1, I_2) = \sum_{i=0}^{N-1} p_i^2 + \sum_{j=0}^{M-1} p_j^2 - \left(\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} 2a_{i,j}\,p_i p_j\right) * (1 + MCR) \tag{4.14}$$

$$SIM^{yang}\,(I_1, I_2) = \left(\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} a_{i,j}\,S_{i,j}\right) * MCR \tag{4.15}$$

4.4 Experimental Evaluation

In this section, a number of experiments are conducted to evaluate the proposed WDCD and MCR modification term.

4.4.1 Experimental Setup

This section is dedicated to identify some setup parameters that are used in the experiments of this chapter. These parameters are image datasets that are used for the purpose of verifying and comparing the performance of the proposed descriptor with the competing descriptors. The latter is the second parameter, which are used for comparison with the proposed one. Besides, to measure the performance of the competing descriptors, quantitative metrics are used as the third parameter for measuring the performance of competing descriptors.

A. Image Datasets

Evaluation of the proposed WDCD was carried out on three datasets: 1) Corel-1K dataset; 2) Caltech-101 dataset that contains 101 classes. In this dataset, 26 classes are selected. This is because, each class is characterized by its images' object that have the same colour to show effectiveness of the proposed WDCD in colour-based object image retrieval. Accordingly, this dataset will be called as Caltech-26 henceforth in this chapter; 3) Cartoon-5K dataset. Selection of cartoon dataset is intended to show effectiveness of WDCD. This is because; cartoon characters normally appear in the same colours in most cartoon images (Jiebo & Crandall, 2006; Khan et al., 2012) (that fits the objective of this contribution). In addition, this type of image often does not suffer from illumination variation. Some samples from cartoon dataset are presented in Figure 4.6. For more details about these datasets, refer to Section 3.2.5.

Figure 4.6: Samples from Cartoon-5K Dataset.

B. Competing Descriptors

The descriptors that are selected to be compared with the proposed WDCD are MPEG-7 DCD (Yamada et al., 2001) and LBA DCD (Yang et al., 2008). This is because the former represents the original DC descriptor whereas the latter is the best DC descriptor so far. In this context, Yang et al. (2001) shows that their descriptor surpasses the other DC descriptors such as (Ma et al., 1997; Mojsilovic et al., 2000; Po & Wong, 2004; Yamada et al., 2001) in both accuracy and time. Hence, there is no need to compare with them.

On the other hand, the comparison with other Object-Based Image Retrieval (OBIR) methods is unfair because these methods combine other features with colour, such as texture and shape, to obtain the results. The proposed WDCD descriptor is just a step forward in object-based image retrieval. It needs to be combined with other features, such as spatial colour relations such as (Kiranyaz et al., 2012; Wong et al., 2007) or shape such as Khan et al. (2012), to be suitable for OBIR methods. Moreover, comparison with some colour-based object recognition techniques such as Geusebroek (2006) that depends on colour invariant feature is also out of the scope of this research.

C. Performance Measure Metrics

Four quantitative performance measure metrics are utilized to measure the performance of all competing descriptors as mentioned in Section 3.2.6 (B).

4.4.2 Retrieval Performance

Retrieval performance of the competing descriptors in the three specified datasets are measured using the four aforementioned metrics (ARR, ANMRR, MAP and P(10)). Diversity of queries is very important to ensure fair and honest results (Grubinger, 2007), thus the evaluation queries are selected from all classes of the database.

A. Retrieval performance of Corel-1K Dataset

Figure 4.7 illustrates more about the visual comparison of the competing DCDs on Corel-1K dataset. Additionally, the four evaluation metrics are computed in accordance with 33 queries on this dataset (3.3% from total dataset size) as presented in Table 4.4.

Table 4.4: *Evaluation Metrics for Corel-1K dataset.*

Descriptors	ARR	ANMRR	MAP	P(10)
MP7DCD	0.225	0.722	0.239	0.41
MP7DCD+ MCR	0.241	0.708	0.250	0.44
LBADCD	0.330	0.598	0.328	0.56
LBADCD+ MCR	0.342	0.581	0.345	0.62
Proposed WDCD	0.374	0.536	0.384	0.62
WDCD+ MCR	**0.380**	**0.535**	**0.387**	**0.64**

Figure 4.7: Visual Comparison of the proposed WDCD with MPEG-7 and LBA DCD in Corel-1K Dataset.

As shown in Table 4.4, the proposed WDCD helps improve the performance of the image retrieval process. The percentages of improvement of WDCD (without MCR) over original LBA and original MPEG-7 are presented in Table 4.5 (left) in terms of ARR, ANMRR, MAP and P(10). The average improvement percentages of the proposed descriptor are 11.8% and 36.5% over LBA and MPEG-7 descriptors respectively. Moreover, the newly proposed similarity measure modification (MCR) also enhances the retrieving performance of all descriptors (MPEG-7, LBA and WDCD) by 4.9%, 5.2%

and 1.35%, respectively in terms of average percentages of the four used metrics, as presented in Table 4.5 (on right).

Table 4.5: *Improvement Percentages of the proposed descriptor WDCD over original LBA and MPEG-7 descriptors in Corel-1K database.*

Improvement Ratio	WDCD over LBA	WDCD over MPEG-7	LBA+MCR over LBA	MPEG7+MCR over MPEG-7	WDCD+MCR over WDCD
ARR	11.7	39.8	3.5	6.6	1.5
ANMRR	11.5	34.7	2.8	1.9	0.1
MAP	14.5	37.7	4.9	4.4	0.7
P(10)	9.6	33.8	9.6	6.8	3.1
Average	**11.8%**	**36.5%**	**5.2%**	**4.9%**	**1.35%**

** Improvement Percentages (Gain) of the proposed descriptor WDCD (without MCR) over original LBA and MPEG-7 descriptors (Left) and improvement Percentage of adding MCR to all descriptors (Right) in Corel-1K database with 33 queries that their results are depicted in Table 4.4.*

From the presented results, the proposed descriptor outperforms the other descriptors and all their enhanced versions (that contain modification term MCR) in all four evaluation metrics. In Corel dataset, there are many classes that have objects of different colours within the same class such as bus, flower and others, as presented in Figure 4.10. This certainly will effect on the accuracy of the WDCD, which mainly depends on colours in its retrieval. Therefore, only 26 categories from 101 categories of Caltech dataset, which have the same coloured-objects, are selected to show effectiveness of the proposed descriptor, evaluation result of Caltech-26 is presented in the next section.

B. Retrieval performance of Caltech-26 Dataset

Figure 4.8 shows the visual comparisons among the competing descriptors in the Caltech-26 dataset. Table 4.6 showcases the quantitative comparisons that are computed using the four evaluation metrics ARR, ANMRR, MAP and P(10).

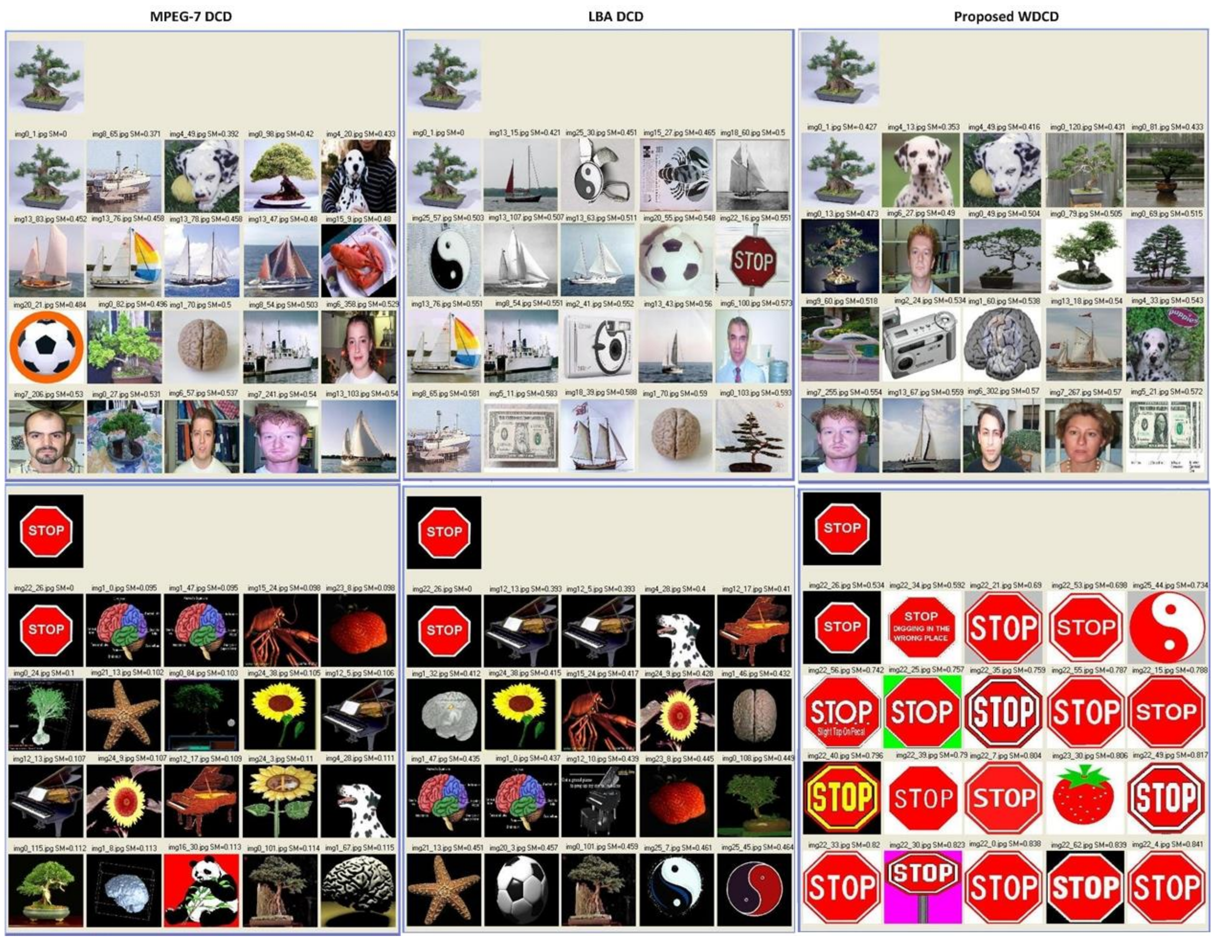


Figure 4.8: Visual Comparison of the proposed WDCD with MPEG-7 and LBA DCD in Caltech-26 Dataset.

Table 4.6: *Evaluation metrics for competing descriptors on Caltech-26 dataset.*

Descriptors	ARR	ANMRR	MAP	P(10)
MP7DCD	0.098	0.871	0.094	0.25
MP7DCD+ MCR	0.111	0.855	0.106	0.30
LBADCD	0.186	0.763	0.182	0.40
LBADCD+ MCR	0.218	0.725	0.205	0.45
Proposed WDCD	0.293	0.642	0.274	0.54
WDCD+ MCR	**0.298**	**0.635**	**0.280**	**0.56**

** Four Evaluation metrics values for all competing descriptors according to 47 queries (1.8 % from the dataset size) in Caltech-26 dataset.*

From Table 4.6, one can notice that the proposed descriptor WDCD performs better than all the other competing descriptors. It increases the performance average by 55.3% and 28.6% over MPEG-7 and LBA respectively in terms of average of the four evaluation metrics, as presented in Table 4.7 (left). In addition, MCR modification term enhances the average of performance for LBA, MPEG7 and WDCD by 10.5%, 10.3% and 2.1% respectively in terms of average of the four evaluation metrics, as shown in Table 4.7 (right).

Table 4.7: *Improvement Percentages of the proposed descriptor WDCD over existing descriptors on Caltech-26 database.*

Improvement Ratio	WDCD over LBA	WDCD over MPEG-7	LBA+MCR over LBA	MPEG7+MCR over MPEG-7	WDCD+MCR over WDCD
ARR	36.5	66.5	14.6	11.7	1.6
ANMRR	18.8	35.6	5.2	1.8	1.1
MAP	33.5	65.6	11.2	11.3	2.1
P(10)	25.9	53.7	11.1	16.6	3.5
Average	**28.6%**	**55.3%**	**10.5%**	**10.3%**	**2.1%**

** Improvement Percentages of the proposed descriptor WDCD (without MCR) over original LBA and MPEG-7 descriptors (Left) and improvement Percentage of adding MCR to all descriptors (Right) in Caltech-26 database with 47 queries that their results are depicted in Table 4.6.*

C. Retrieval performance of Cartoon-5K Dataset

Colour feature plays an essential role in cartoon images (Jiebo & Crandall, 2006; Khan et al., 2012). Therefore, the researchers used this type of images to test their colour-based methods such as the work that is proposed by Khan et al. (2012), which uses cartoon images in colour-based object detection. Jiebo and Crandall (2006) uses flag database to test its object detection method but they refer to specific databases suitable for colour-based object detection such as flag, logos and cartoon databases. Khan et al. (2012), introduced new cartoon image dataset of 18 classes and 586 images as total. This motivates us to introduce large cartoon dataset (Cartoon-5K) to test the proposed descriptor WDCD. Visual comparisons among all the competing descriptors in Cartoon-5K dataset is depicted in Figure 4.9. Additionally, Table 4.8 presents the quantitative comparisons among descriptors using aforementioned four metrics for 106 queries (2.1% from database size).

Table 4.8: *Four evaluation metrics for competing descriptors applied on Cartoon-5K dataset.*

Descriptors	ARR	ANMRR	MAP	P(10)
MP7DCD	0.049	0.935	0.047	0.17
MP7DCD+ MCR	0.057	0.927	0.050	0.19
LBADCD	0.075	0.905	0.068	0.21
LBADCD+ MCR	0.090	0.886	0.080	0.26
Proposed WDCD	0.156	0.806	0.134	0.34
LWDCD+ MCR	**0.166**	**0.794**	**0.140**	**0.37**

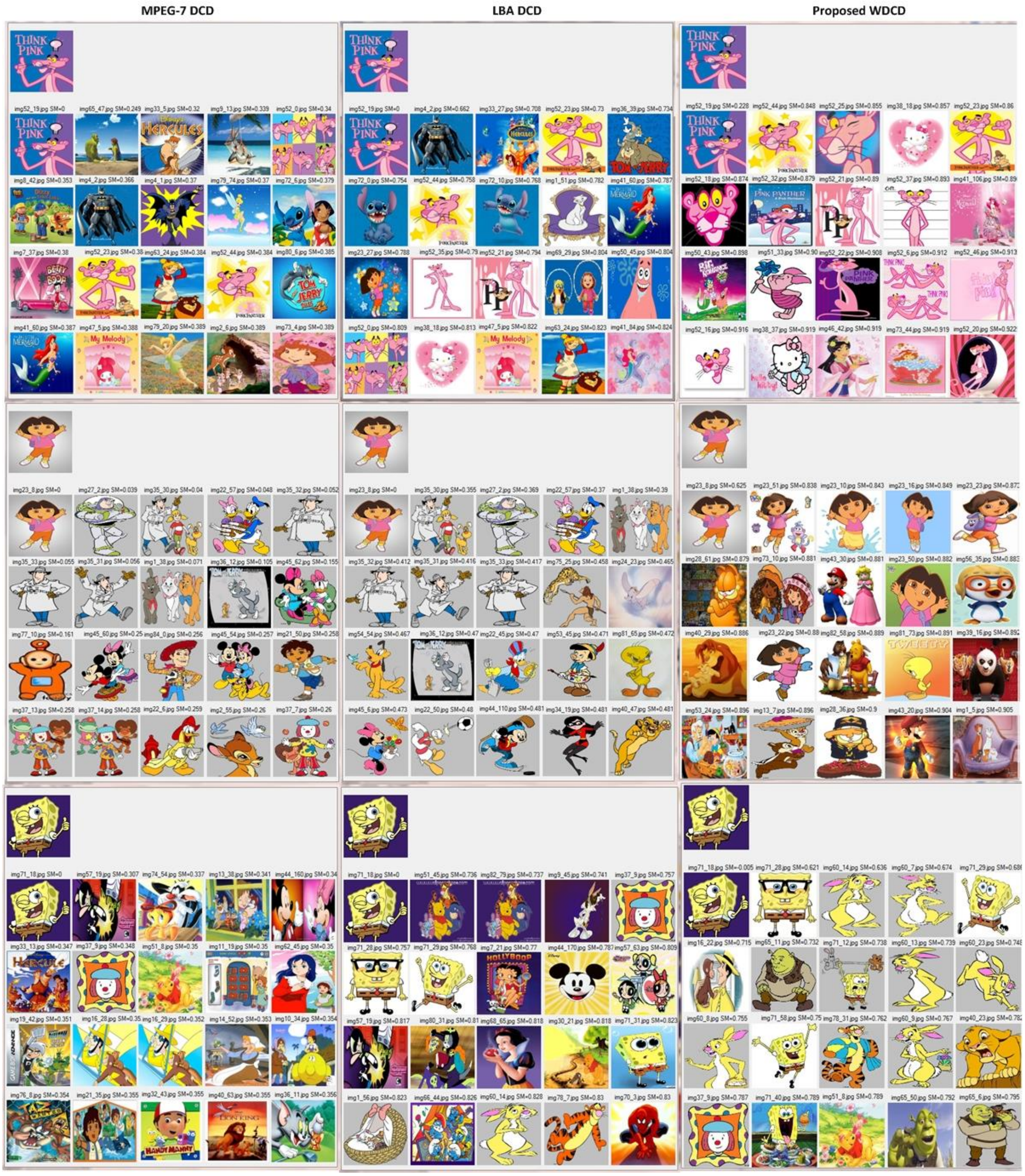


Figure 4.9: Visual Comparison of the proposed WDCD with MPEG-7 and LBA DCD in Cartoon-5K Dataset.

The visual and quantitative comparison of Cartoon dataset shows the discrimination power of the proposed semantic descriptor. With this method, the same cartoon character with different backgrounds can be retrieved. On the other hand, the other competing descriptors retrieve different characters with similar background. Moreover, as shown in Table 4.8, one can observe that combining the proposed descriptor and proposed similarity measure modification can successfully lead to the best retrieval performance. The high performance of the proposed descriptors in the cartoon images back to the cartoon character often has the same colours. Despite of the large number of images and diversity of Cartoon-5K dataset, the improving ratio in this dataset exceeds those of the two natural images datasets. The proposed descriptor results in improving the rates by 49.8% and 37.8% over MPEG-7 and LBA, respectively, in terms of the average of the four used metrics, as shown in Table 4.9 (left part). MCR further improves the rates by 13.2%, 7.8% and 4.9% of LBA, MPEG-7 and WDCDs respectively, as presented in Table 4.9 (right part).

Table 4.9: *Improvement Ratio of the proposed descriptor WDCD over existing descriptors in Cartoon-5K database.*

Improvement Ratio	WDCD over LBA	WDCD over MPEG-7	LBA+MCR over LBA	MPEG7+MCR over MPEG-7	WDCD+MCR over WDCD
ARR	51.9	68.5	16.6	14.0	6.0
ANMRR	12.2	16.0	2.1	0.8	1.5
MAP	49.2	64.9	15.0	6.0	4.2
P(10)	38.2	50.0	19.2	10.5	8.1
Average	**37.8%**	**49.8%**	**13.2%**	**7.8%**	**4.9%**

** Improvement Ratio of the proposed descriptor WDCD (without MCR) over original LBA and MPEG-7 descriptors (Left) and improvement ratio of adding MCR to all descriptors (Right) in Cartoon-5K database with 106 queries that their results are depicted in Table 4.8.*

As mentioned before, Corel dataset has some classes in which the object (of the same class) has different colour; this will degrade the proposed descriptor accuracy. Figure 4.10 shows examples of how the proposed descriptor retrieved results worse than the other competing descriptors.

The reason behind the failure of the proposed descriptor in dealing with such cases is that the object of the query image has different colours from its ground truth images. Besides, the query has a similar background colour to these ground truth images. Hence, the MPEG-7 and LBA DCDs that depend on the colour of the large percentage (background in this case) will outperform the proposed WDCD. However, the latter provides more semantic information (the colour of the object) than the previous methods. For more illustration consider the retrieval results of the yellow flower image in Figure 4.10. The image shows that the previous methods managed in retrieving images of flowers with different colours. The proposed method of this research, on the other hand, managed to retrieve images of yellow flowers and yellow objects only. That is because the proposed descriptor is mainly an object's colour-based descriptor.

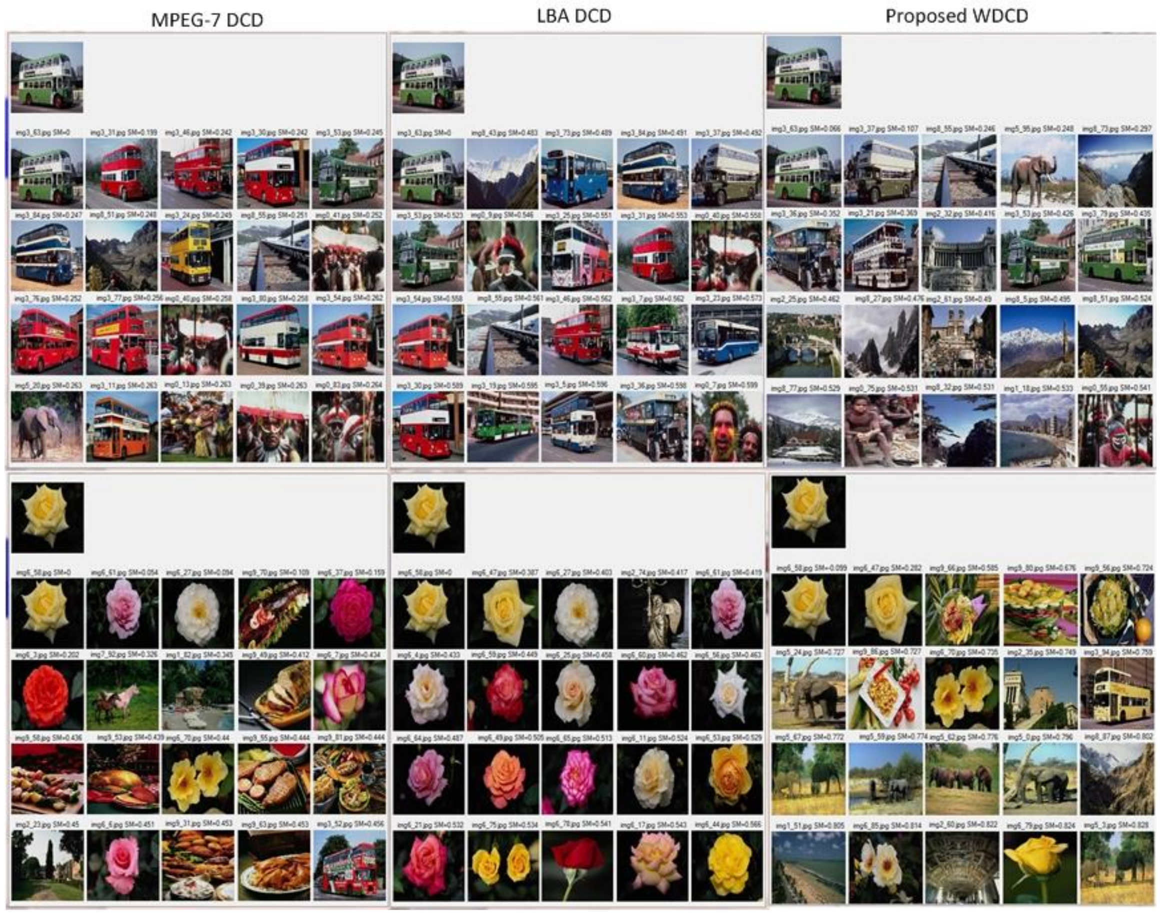


Figure 4.10: An example to show outperformance of MPEG-7 and LBA DCDs over the proposed WDCD in some queries.

Moreover, in all evaluation datasets, there are many classes that have the same object colour (e.g. yellow flower and yellow bus in Corel dataset, SpongeBob and yellow Rabbit in cartoon dataset as shown in Figure 4.9 and many others). This allows retrieving different object of the same colour that in turn will degrade performance of the proposed descriptor. Therefore, additional features need to be integrated with colour (such as spatial colours relations as proposed by Kiranyaz et al. (2012) and shape such as the one proposed by Khan et al. (2012)) to complement the proposed descriptor to be semantic and suitable for object-based image retrieval.

From another perspective, time is one of the important issues that need be considered in retrieval systems, especially in web-based image retrieval systems. To extract dominant colours from single image, GLA that is used in MPEG-7 DCD requires 2.5 seconds while LBA requires 0.37 seconds. To extract salient object of an image, GC-SRD requires 1.5 seconds while LoG with flood fill algorithm requires 1.2 seconds. All experiments were conducted using Dual Core 2.0 GHz CPU with 3 GB RAM; the time is averaged from tens of experiments on different image resolutions. Therefore, the time required for the proposed method compared with the MPEG-7 and LBA DCDs is presented in Table 4.10. The accuracy of the proposed descriptor tends to be equal in using any of DCs extraction method (GLA or LBA). Therefore, the restriction only in the time required for each one; for faster retrieval, LBA is preferred.

Table 4.10: *Average Feature Extraction Time required for the proposed descriptor and the two compared descriptors.*

Descriptor	MPEG-7 DCD	LBA DCD	Proposed WDCD
Time *(second)*	2.5	0.37	In Natural Images (LBA) 1.87 (MPEG7) 4.0 In Cartoon Images (LBA) 1.57 (MPEG7) 3.7

** The Experiments are conducted in Dual Core 2.0 GHz machine with 3 GB RAM.*

4.5 Summary

This chapter introduces two possible solutions to the problem that most colour-based methods suffer from. This problem is concerned with the dominant of large percentage colours of the query image (regardless of whether they are background or object colours) on retrieving images. One of these solutions is feature level-based solution, an adapted weighted dominant colour descriptor, which can be used as a step forward in object-based image retrieval. The mechanism of the proposed descriptor is based on weight assignment to each DC in the image in accordance to whether it belongs to the object or to the background. The background colours, which are in contact with the image borders and out of salient object area, receive a lower weight whereas the object colours, which are located at the salient object area and do not touch the border, receive a higher weight. Such a method helps alleviate the background effect. Additionally, the second solution is similarity measure level-based solution, MCR, which is a modification term that is added to similarity measure for enhancing retrieval accuracy. Moreover, new Cartoon-5K dataset is introduced to test the proposed descriptors. The experimental results further show that the proposed semantic feature with the newly introduced similarity measure modification outperform the existing descriptors. In the next chapter, an advanced colour-based method is proposed, which is enhancement to a colour *Correlogram*.

CHAPTER FIVE

A COMPACT-GENERALIZED DOMINANT-COLOUR CORRELOGRAM DESCRIPTORS

One of the promising colour techniques in image retrieval is colour *Correlogram* but it also suffers from high computations and storage drawback. In this chapter, two compact representations of colour *Correlogram* are presented; the first representation is proposed while the second is adapted from an existing one. The first representation is Compact-Generalized *Correlogram* (CGC), which compacts the colours and generalizes the distances of the original *Correlogram* descriptor. The second representation is Dominant Colours-Based *Correlogram* (DCBC), which is also a compact and conceptual *Correlogram* descriptor. It computes spatial correlations of the few image's DCs instead of the large number of quantized colours that are used by the original descriptor. The integration of the two representations is also introduced. Weighting DCs, which have been discussed in Chapter 4, is applied on two complicated DC-based descriptors than DCD to show its efficiency and generality. Section 5.1 presents the problems of colour *Correlogram* and its limitations in large databases. In Section 5.2, *Correlogram* feature extraction process is analyzed and a proposed compact-generalized *Correlogram* is introduced. Limitations of existing similarity measure of current DC-based *Correlogram* are addressed and new similarity measure is proposed in Section 5.3. Moreover, weighting DCBC is also presented in the latter section. Experimental results of all competing methods are reported in Section 5.4. Conversion from large number of colours into few dominant colours (that is used in colour *Correlogram* in Section 5.3.1) and weighting DCs (that is applied in Chapter 4 and Section 5.3.2) methods will be applied on another colour-based method, Border-Interior Classification (BIC), to ensure their generality; thus their generic frameworks will be detailed in Section 5.5. Finally, summary of this chapter is presented in Section 5.6.

5.1 The Problem of Colour Correlogram

Colour *Correlogram* is one of the most promising colour-based approaches. It preserves spatial correlations of colour information for accurate image retrieval than a colour histogram. Colour *Correlogram* approach demonstrates its effectiveness compared with colour histogram (Swain & Ballard, 1991) and earlier spatial-colour approach that called Colour Coherence Vector (Pass et al., 1997). Additionally, *Correlogram* is considered as one of the best colour descriptors in large database (Pedronette & Torres, 2012; Penatti et al., 2012). Therefore, it is selected by this research to solve its drawbacks.

Colour *Correlogram* is a table indexed by colour pairs (C_i, C_j) where k^{th} entry specifies the probability of finding a colour C_i at a distance k from a colour C_j in the image; i, j are indexes to colours within range of m quantized colours and k is a distance within range of maximum distance d. The problem of *Correlogram* lies in its expensively cost in terms of memory space and computation time where it required $O(m^2d)$ complexity. This imposes infeasibility problem, especially in the memory space where it requires several gigabytes for large database. This space may not be available in the normal modern computers or even in limited-sources devices such as mobile devices. Therefore, *Autocorrelogram* (Huang et al., 1997) is proposed to reduce the time and space complexity into $O(md)$ by finding spatial correlation of each colour with itself only. The accuracy of *Autocorrelogram* is certainly lower than original *Correlogram* because it ignored correlations of a particular colour with the other colours; it keeps correlation with the same colour only. In this chapter, a compact-generalized representation of colour *Correlogram* is proposed (Talib, Mahmuddin, Husni & George, 2013a). It reduces complexity of the *Correlogram* from $O(m^2d)$ into $O(\frac{m^2}{2} + \frac{m}{2})$. It is a little bit more complex than *Autocorrelogram* (or less than *Autocorrelogram* in some cases when d is large).

On the other hand, colour descriptors, including colour *Correlogram*, have a weakness in image recognition or discrimination because the naive rules that they are based on are not simulated to human visual system (Kiranyaz et al., 2010, 2012). Therefore, many improvements were achieved in this field. A wealth of research has been conducted on human colour perception such as Broek et al. (2004), Kiranyaz et al. (2012) and Mojsilovic et al. (2000). They showed that human uses few prominent or dominant colours of the image to judge the similarity. They report there are two rules to model human visual and colour perception, which are as follows: The first rule states that the two images are considered similar if they have the same DCs. The second rule indicates that the two images perceive similar if they have same the DCs' distribution irrespective of their content (Kiranyaz et al., 2012). In other words, human considers images' DCs and their spatial distributions in judging the colour similarity of the compared images. Therefore, dominant colour descriptors have been introduced by many research (Babu et al., 1995; Deng et al., 1999; Fauqueur & Boujemaa, 2002; Manjunath et al., 2001; Mojsilovic et al., 2002; Wong et al., 2007; Yamada et al., 2001; Yang et al., 2008) instead of descriptors that use a large number of colours such as colour histogram, colour moments and others (Gong et al., 1996; Swain & Ballard, 1991). According to previous discussion, colour *Correlogram* has perceptually and infeasibility problems.

5.2 Compact-Generalized Correlogram Descriptor

The colour *Correlogram* offers the best performance among the GCDs and SCDs, as mentioned in Section 5.1. However, it has a serious drawback due to its massive consumption for time and memory space. Through critical analysis to its process, some reduction can be achieved to its time and feature vector space as follows.

ColGrm ($\gamma_{Ci,Cj}^{(k)}$) is a table of probabilities for finding a spatial correlation of certain colour with the other colours within an image in a specific distance. This table is indexed by the triple (C_i,C_j,k), C_i and C_j represent the colours that their neighbouring probabilities in a distance k need to be known. Indexes values i, j are within m quantized colours, k value is within maximum distance d. *ColGrm* keeps spatial correlation among colours in the image, *ColGrm* table can be depicted in Figure 5.1.

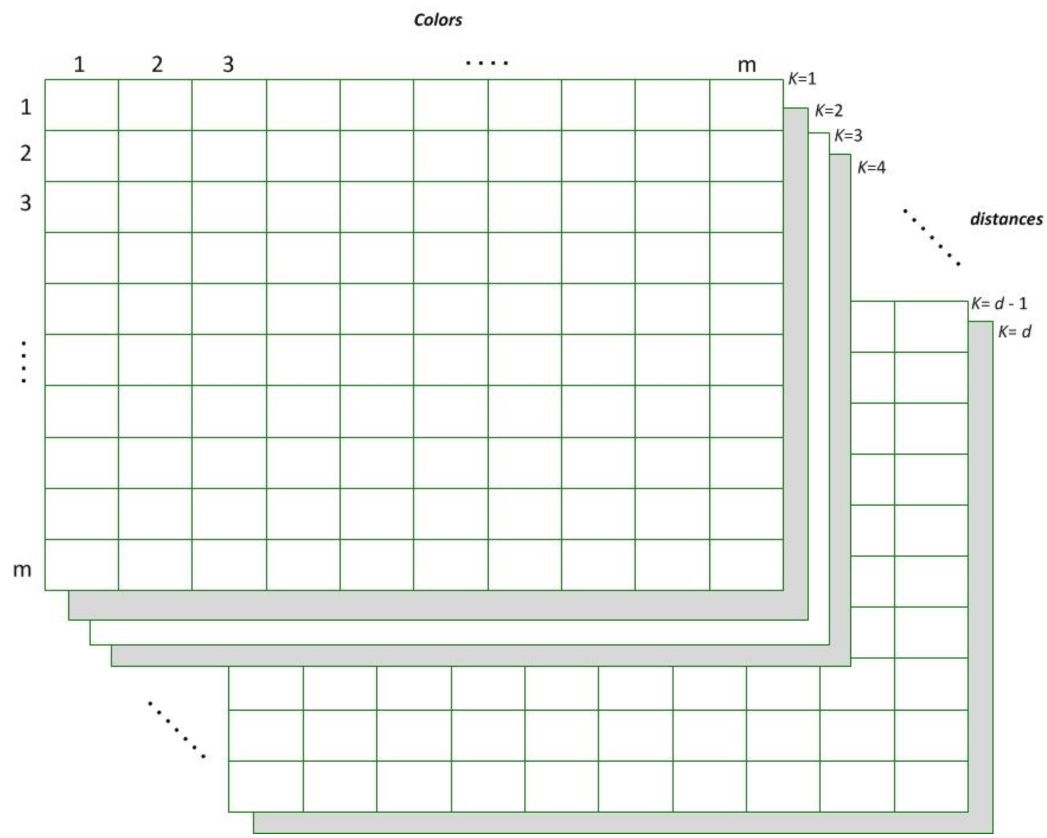


Figure 5.1: Original colour Correlogram feature vector representation, of complexity $O(m^2d)$.

From Figure 5.1, one can notice that the massive storage space of this representation lies in existence of colours and distances. Therefore, the proposed method is focused on reduction to these two factors without significant degradation to performance of the original *ColGrm*.

5.2.1 Colour Reduction

In the first factor (i.e. colours), the square matrix of colours, which can be noticed in Figure 5.1, contains probabilities of finding colour *i* at the distance *k* from colour *j*. In a proper logical analysis for this colour representation, there is a repetition of information. The probability of finding colour *i* with specific distance from colour *j* is located in two positions in the matrix, in locations *(i, j)* and *(j, i)*. Intuitively, the existence of white colour besides black colour, for example, represents the same meaning of existence the black colour besides white colour. This occurs when the black colour is on the right of white colour; and of course the white colour is on the left of the black colour. Therefore, the co-occurrence matrix of colours in the original representation is increased in the two locations *Co-Occurrence(black, white)* and *Co-Occurrence(white, black)*, as shown in Figure 5.2.

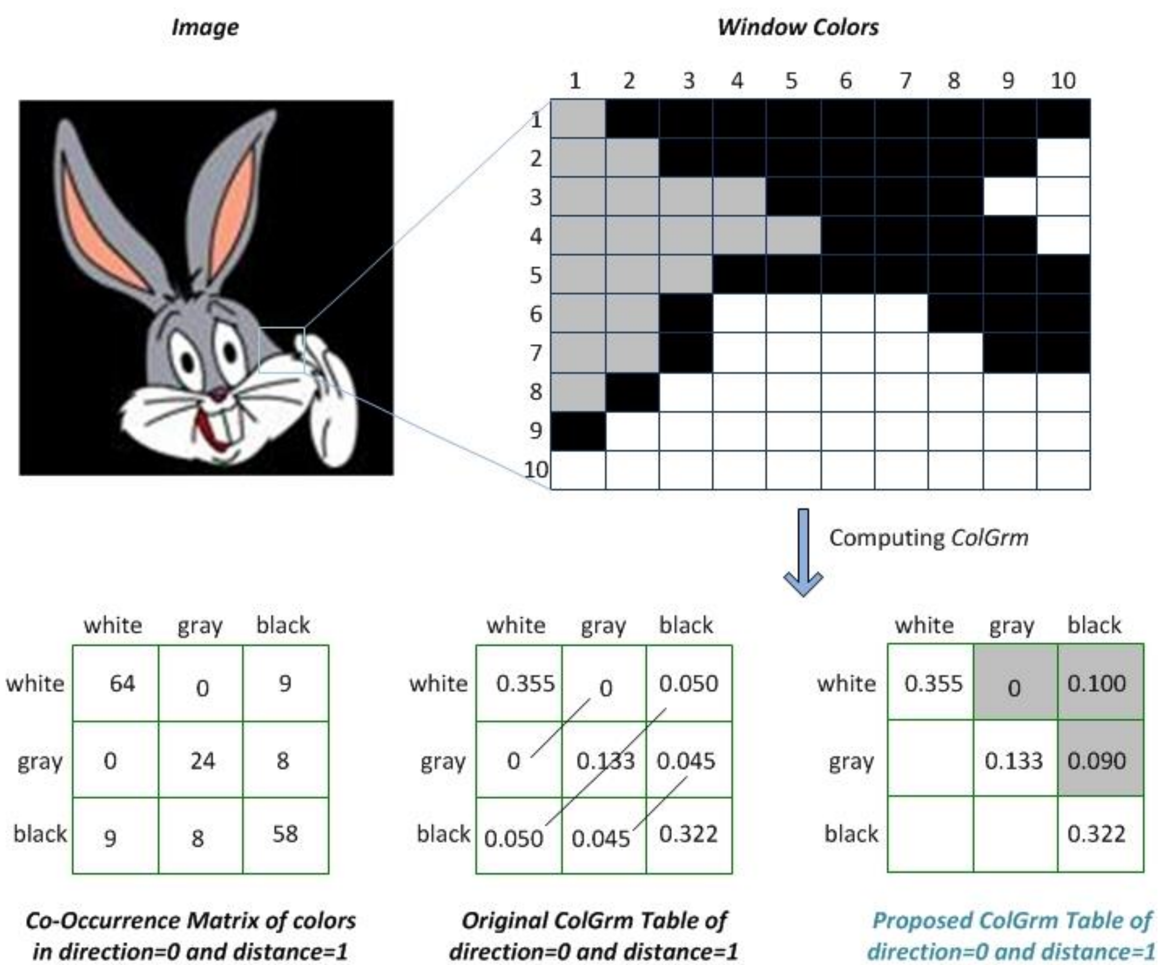


Figure 5.2: An example of computing ColGrm Table to Image's Window.

Figure 5.2 is an example of image's window that has three colours white (w), gray (g) and black (b). A simple setting (direction=0 and distance=1) is considered for ease of explanation. 'Direction equal to zero' and 'distance equal to one', mean that only direct horizontal (left and right) neighbours of pixels are considered during extracting process the *ColGrm* table. To simplify the explanation, co-occurrence matrix is shown in Figure 5.2, which one of its elements is *Co-occurrence(white, black)* = 9, which means there are nine horizontal black neighbours to the white colour. *ColGrm* table holds the probability instead of number of colours occurrence, so by dividing the co-occurrence matrix by the number of all neighbours in this 10*10 window, which is 180. From *ColGrm* table, one can notice that *ColGrm(w, g)=ColGrm(g, w)* and all other elements in the lower triangular matrix are similar to the elements of upper one. Therefore, repeating these elements are useless, where one element is sufficient for each pair of colours instead of two elements. Keeping the upper triangular matrix with the main diagonal is enough to keep the whole matrix. The upper triangular matrix in the new proposed representation is duplicated to substitute the absence of the lower matrix. From this finding, the *ColGrm* complexity can be reduced approximately to half $O(\frac{m^2}{2}+\frac{m}{2})$ instead of $O(m^2)$, as shown in Figure 5.3 in shaded cells. Therefore, only upper triangular matrix and main diagonal are needed to be computed and saved.

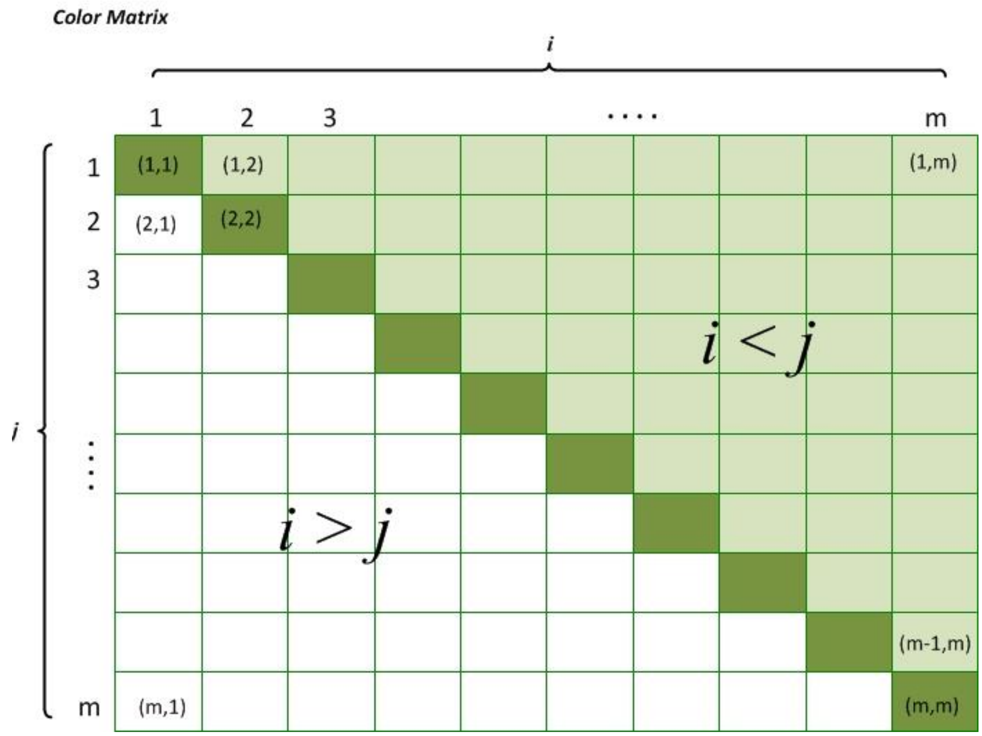


Figure 5.3: New ColGrm matrix representations.

In the proposed method, when passing on all images' colour values to produce *ColGrm* matrix; the following algorithm is added to increase co-occurrence matrix, instead of normal increment of the matrix.

Algorithm 5.1: Increment_CoOccurrence_Matrix

Input

CoOccMatrix: Co-Occurrence Matrix of Colour Correlogram

C_i, C_j: indexes of Colours i and j in Co-Occurrence Matrix.

d: distance between Colours Indexes C_i and C_j

Output

CoOccMatrix: Co-Occurrence Matrix of Colour Correlogram after increment

BEGIN

1. Check colours indexes if they are in upper triangular matrix then increase their occurrence, otherwise exchange indexes and increase the colours occurrence

If (C_i <= C_j)

```
            CoOccMatrix(Ci, Cj, d) ++;
      // increment CoOccMatrix element if it is in upper triangular matrix
        else
           CoOccMatrix(Cj, Ci, d) ++;
     // exchange of Ci and Cj locations if they represent lower triangular matrix
          End if
     2.   Return (CoOccMatrix)
```

END.

Algorithm 5.1 is used to increase the upper triangular matrix and the main diagonal of the *ColGrm* matrix. Besides, the dissimilarity measure equation remained the same as in original *Correlogram* as depicted in Eq. 5.1.

$$ColGrm\ dis-similarity\ (Q,I) = \sum_{i}^{m}\sum_{j}^{m}\sum_{k}^{d} \frac{|\,\gamma_{ci,cj}^{(k)}(Q) - \gamma_{ci,cj}^{(k)}(I)|}{1 +\ \gamma_{ci,cj}^{(k)}(Q) + \gamma_{ci,cj}^{(k)}(I)} \qquad (5.1)$$

5.2.2 Distance Reduction

Distance is the second specified factor to reduce the complexity of *ColGrm* feature vector. As a mentioned earlier in Section 2.3, the number of distances required for *ColGrm* to capture the true spatial correlations of the colours is from 10% to 50% of the smallest dimension in the image. This will consume CPU time and memory space, as depicted in Figure 5.1. Therefore, reduction of time and space is required to make the *ColGrm* applicable for the large database. The proposed solution for reducing these distances is a generalization. The proposed generalization scheme that can be applied for distances is averaging of all distances. Distances are very important to measure how many pixels the certain colour is far from the others. For example, image with three colours as in Figure 5.2, when d=1 and *ColGrm(white, gray, d)* = 0.15 that means probability of finding white colour far from gray colour by one pixel is 0.15. When d=2 and *ColGrm(white, gray, d)* = 0.149, it means that probability of finding white colour far from gray colour by two pixels is 0.149 and so on. Unfortunately this pixel-based structure is one of the main drawbacks of ColGrm (Kiranyaz et al., 2010, 2012). It characterizes the colour vicinity at a pixel-level, which is not just infeasible to achieve (in terms of time and space) in high resolution images but also it is meaningless with regard to the human visual system. This is because individual pixels are not perceived by the human eye. To eliminate this effect from *ColGrm* and generalize the distance, average of all distances can be computed. Therefore, the layers of distance, as shown in Figure 5.1, can be abbreviated to one layer that contains probabilities of finding each colour with other colours in the image in general; regardless of specific distance as shown in Figure 5.4. In Figure 5.1, the size of image is 10 by 10, so distance=5 is selected as 50% of the smallest dimension in the image. Now, when distance=5 and generalization of *ColGrm* is applied, one layer is produced. For example *ColGrm*(white, gray) = 0.145 means that probability of finding white colour far from gray colour is 0.145 in general. This will ensure the generality of descriptor, eliminate pixel-level dependency (especially in high resolution images) and describe the spatial correlations among colours in general for all images in the database. This will draw a general vision of image contents (colours) instead of depending on individual distances of colours that lack of feasibility and human perception. Hence, the complexity of distance will become 1 general distance (instead of d distances in the original *ColGrm*). The total complexity of the proposed *ColGrm* after colour and distance reduction is $O\left(\frac{m^2}{2}+\frac{m}{2}\right)$ instead of $O(m^2d)$, as shown in Figure 5.4.

The complexity of the proposed compact *ColGrm* is $O\left(\frac{m^2}{2}+\frac{m}{2}\right)$ during the image retrieval process as well as storage space; but it is $O(m^2d)$ during the feature extraction process because all distances must be computed first prior to find their average. Feature extraction is an offline process, which means that it is performed once and then the feature vectors are saved in a database to be ready for online retrieval process (refer to Figure 5.5).

In Figure 5.5, the online process in the proposed descriptor consists of two steps, the first one is feature extraction for the query image and the second step is similarity measure that is computed between query and all database images. Feature extraction process (that is computed for one image only) requires $O(m^2d)$ complexity whereas the similarity measure process (that computed for all database images) requires $O\left(\frac{m^2}{2}+\frac{m}{2}\right)$ complexity. Therefore, this will not significantly affect the speed of the interactive process with the user and the proposed method can be used in real time applications. The steps of *ColGrm* distances' generalization process is summarized in Algorithm 5.2.

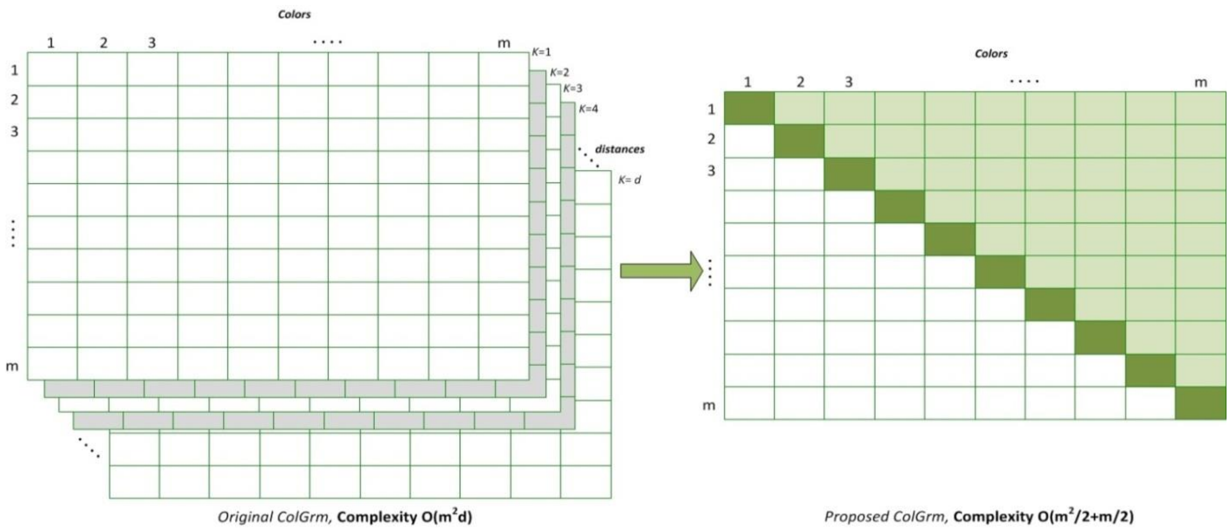


Figure 5.4: Complexity of the proposed Compact ColGrm descriptor (shaded cells only) compared with the Original One.

Algorithm 5.2: Correlogram Distances' Generalization

Input

ColGrmTable: Colour Correlogram Table of size m*m*dis (m^2dis)

m: number of colours in the ColGrm Table

dis: number of distances in ColGrm Table

Output

Generalized_ColGrmTable: Generalized Colour Correlogram Table of size m*m (m^2)

BEGIN

1. Find Generalization of all distances in ColGrm Table
   - { pass on all Colours in the ColGrm Table}

     $\forall\, i \;\in 0..m-1$

     $\forall\, j \;\in 0..m-1$

     - { find Summation of all distances in the ColGrm Table }

       $Sum \leftarrow 0$

       $\forall\, d \;\in 1..dis$

       $Sum \leftarrow Sum + \; ColGrmTable(i, j, d);$

     - { Generalize all distances by Average of them }

       $Generalized\_ColGrmTable(i, j) \leftarrow Sum \,/\, DistanceNo$
2. Return ($Generalized\_ColGrmTable$)

END

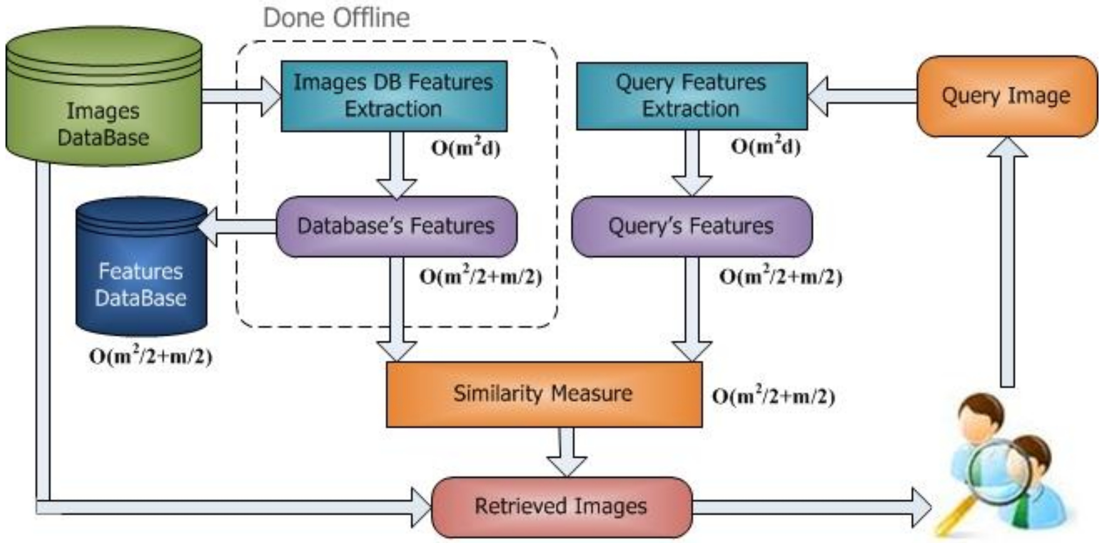


Figure 5.5: Complexity of the proposed Compact-Generalized Correlogram in all stages of CBIR System.

5.3 The Adapted DC-based Correlogram Descriptor

As mentioned in Section 5.1, DC-based approaches are introduced to solve the perceptual problem of conventional colour-based approaches, where they simulate the human colour perception. One of the most promising DC-based approaches is the method that has been proposed by Kiranyaz et al. (2012). Kiranyaz and colleagues integrate DCs with *ColGrm* to solve problems of both methods. These problems are lacking of spatial colours information of DC descriptors as well as infeasibility problem of original *ColGrm* descriptor, especially in large databases. This method is called “Perceptual Correlogram”. DCs are extracted from image by method similar to that of Deng et al. (1999), which is simulating human colour perception. Then these DCs are back projected on the image to extract colour *Correlogram* depending on DCs. An overview of this method is depicted in Figure 5.6.

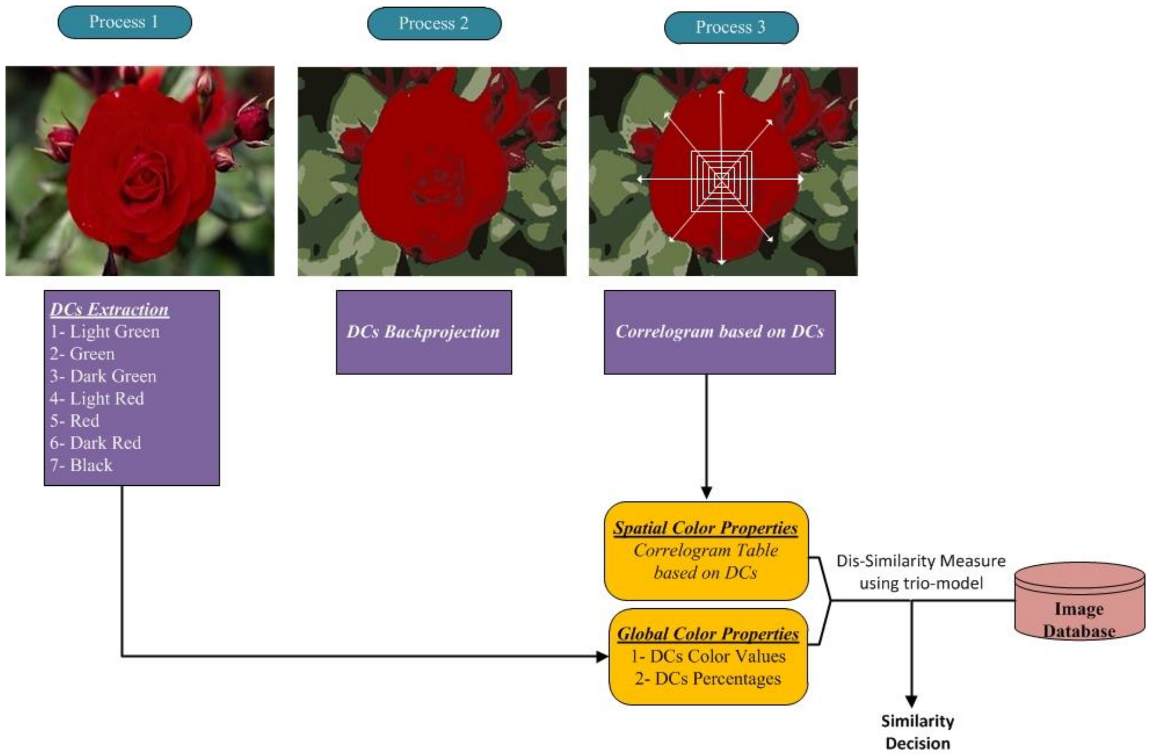


Figure 5.6: An overview of Perceptual Correlogram (source: Kiranyaz et al., 2012).

This method proposes trio-model to measure the dissimilarity of the two images as follows (Kiranyaz et al., 2012):

$$P_{trio}(Q,I) = P_{\emptyset}(Q,I) + \left(\alpha P_G(Q,I) + (1-\alpha)P_{Corr}(Q,I)\right) \qquad (5.2)$$

Trio-model has three measuring metrics which are $P_{\emptyset}$, P_G and P_{Corr}. The first metric, $P_{\emptyset}$, measures how much is the mismatching colours and their percentages in the two compared images, as depicted in Eq. 5.3, W_i and C_i represent percentages and colour values in mismatching colour list ($S^{\emptyset}$) (Kiranyaz et al., 2012). The other two metrics (P_G and P_{Corr}) measure the difference between the matched colours of the two images. P_G measures the global difference between the two images, as expressed in Eq. 5.4 where N_m represents number of matching colours of two images, Ts represents colour similarity threshold and β is the value between 0 and 1, it represents adjustment between the two terms of Eq. 5.4 (Kiranyaz et al., 2012). P_{Corr} measures the spatial (or *ColGrm*) difference between the two images, as shown in Eq. 5.5 where *MC* represents list of similar (matched) colours between the two images Q and I (Kiranyaz et al., 2012). $\gamma_{Ci,Cj}^{(k)}$ is the probability of finding DC Ci at distance k from DC Cj.

$$P_{\emptyset}(Q,I) = \frac{\sum(w_i | c_i \in S^{\emptyset})}{2} \quad \leq 1 \tag{5.3}$$

$$P_G(Q,I) = \beta \sum_{i=1}^{Nm} |W_i^Q - W_i^I| + (1-\beta)\frac{\sqrt{\sum_{i=1}^{Nm}\left(c_i^Q - c_i^I\right)^2}}{T_s N_m} \quad \leq 1 \tag{5.4}$$

$$P_{Corr}(Q,I) = \sum_{i,j \in MC} \sum_{k=1}^{d} \left\{ \begin{array}{cc} 0 & if\ \gamma_{Ci,Cj}^{(k)}(Q) = \gamma_{Ci,Cj}^{(k)}(I) = 0 \\ \frac{|\gamma_{Ci,Cj}^{(k)}(Q) - \gamma_{Ci,Cj}^{(k)}(I)|}{\gamma_{Ci,Cj}^{(k)}(Q) + \gamma_{Ci,Cj}^{(k)}(I)} & else \end{array} \right\} \tag{5.5}$$

From other perspective, $P_{\emptyset}$ and P_G measure the global differences whereas P_{Corr} measures the spatial differences between the compared images. In proper critical analysis to the trio-model, serious drawbacks have been identified. The first drawback occurs in computing the *ColGrm* dissimilarity metric (P_{Corr}), whereas the second drawback lies in existence of P_G and $P_{\emptyset}$ with P_{Corr}, where (P_G and $P_{\emptyset}$) compute general dissimilarity and represent different perspective than *ColGrm* dissimilarity (P_{Corr}). Limitation of P_{Corr} is identified by comparing it (Eq. 5.5) with dissimilarity measure of original *ColGrm* that is shown in Eq. 5.1. The results of both dissimilarity measures can be compared in Table 5.1.

Table 5.1: *Comparison between Original and Perceptual ColGrm dissimilarity measures.*

Colour Probability of First Image x	Colour Probability of Second Image y	Dissimilar Value of Original *ColGrm* $\frac{x-y}{1+x+y}$	Dissimilar Value of Perceptual *ColGrm* $\frac{x-y}{x+y}$	Percentage of x, y in image and difference amount
0.5	**0.5**	0	0	Large (zero diff)
0.5	**0**	0.333	1	Large (large diff)
0.005	**0**	0.005	1	Small (large diff)
0.5	**0.1**	0.25	0.66	Large (large diff)
0.005	**0.001**	0.004	0.66	Small (large diff)
0.5	**0.4**	0.05	0.11	Large (small diff)
0.005	**0.004**	0.001	0.11	Small (small diff)

Dissimilar values that resultant from both methods can be analyzed as follows. The similarity measure of the method that proposed by Kiranyaz et al. (2012) have serious problem. The problem is its dissimilarity measure does not discriminate between large and small percentages of the probabilities values according to image, whereas the dissimilarity value of large percentages *ColGrm* probabilities (colours) in the image are equal to those of small probabilities. This matter is contrary to the human visual perception, because the human eyes cannot recognize colours of small percentages while it can for the large colour percentages. While the original *ColGrm* dissimilarity keeps these percentages as it is; if the percentages is large the dissimilar value will be large and if it is small the result will also be small, as illustrated in Figure 5.7. In this example, there is object of colours white and black and its percentage in the top images (of blue background colours) are large (50% and 10%). The *ColGrm* probability values that correspond to these large percentages colours are also large, especially in the main diagonal of *Correlogram* matrix, which measures the probabilities of finding each colour with itself, as shown in Figure 5.9. The *ColGrm* dissimilarity measure of perceptual descriptor (Kiranyaz et al., 2012) is equal to 0.66 whereas in original *ColGrm* descriptor it is equal to 0.25. In the bottom images (of green background colours), the percentages of objects (and also *ColGrm* probability values) are small (5 percent and 1 percent). The dissimilar value of perceptual descriptor is also 0.66 (same value when the object has large percentage) while the original *ColGrm* dissimilar value is 0.037 (compatible with the small percentage of the object). Perceptual *ColGrm* considers only the different amount, as can be noted in the "large diff" and "small diff" in Table 5.1. If the ratio between the two *ColGrm* probability values (x and y) is small then the dissimilar value is small and vice versa. For example, if the ratio of difference between x and y is 5:1 (irrespective of *ColGrm* probability (colour) percentages in the image is 0.5:0.1 or 0.005:0.001) the dissimilar value will be 0.66. But from the human perspective, it is noticeable if the percentage of colour in the image is 0.5 (50 percent of the image area) whereas it is not noticeable if the colour percentage was 0.005 (0.5 percent of an image area). While in dissimilarity measure of original *ColGrm*, both *ColGrm* probability (colour) percentages and differences are considered.

The dissimilar value of perceptual descriptor is illogical because the image also has other colours (details). Moreover, the other metrics ($P_{\emptyset}$ and P_G) in dissimilarity measure have values harmonious with colours' percentages (from 0 to 1); this will conflict with P_{Corr}, which its value is fixed in both large and small percentages of colour. In other words, if there are five matched colours when comparing two images and each one of these colours has probability equals to the objects in the bottom images in Figure 5.7. This mean the P_{Corr} will become 0.66*5= 3.3; this value is not compatible with $P_{\emptyset}$ and P_G, which their values range between 0 and 1. Therefore, the dissimilarity measure of original *ColGrm* is better than that in perceptual *ColGrm* (Kiranyaz et al., 2012), but how can it be applied in this new environment, in which the number of DCs and the DCs values themselves of the two images are not equals as in *ColGrm*. The solution is proposed in the new sub-section.

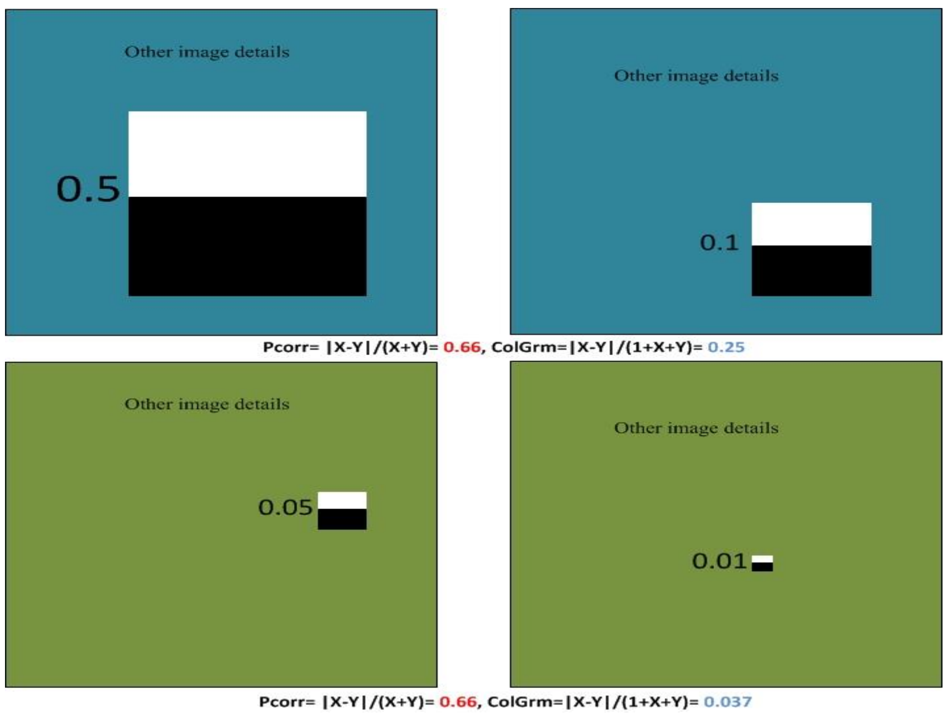


Figure 5.7: The drawback of ColGrm dissimilarity measure (P_{Corr}) of Perceptual ColGrm compared with that of Original ColGrm.

5.3.1 A Proposed Duo-Model for DC-based *ColGrm* Dissimilarity Measure

In the perceptual *ColGrm* dissimilarity measure, combining $P_{\emptyset}$ and P_G (global difference) with P_{Corr} (spatial difference) in the same metric is unsuitable because their values and perspective are different. In perceptual *ColGrm*, the authors are forced to use them with P_{Corr} because the latter metric compute the dissimilarity of matched colours only and leave the mismatched colours. Whereas in the original *ColGrm*, the dissimilarity measure equation computes the dissimilar values for matched and mismatched colours together. Therefore, adaptation to the perceptual *Correlogram* is proposed and is referred to as *DC-Based ColGrm (DCBC)* henceforth. The concept of original *ColGrm* can be applied, which is computing matched and mismatched colours in the same metric. In the adapted method, the probability values of the matched colours between the two images are compared directly whereas the mismatched colours for each of the two images are compared with zero, as shown in Figure 5.8, using original *ColGrm* dissimilarity measure (Eq. 5.1).

***ColGrm* of Image 1** that have Colors: 1,2,3,6,8

	1	2	3	4	5	6	7	8
1	0.14	0	0.02			0.005		0.04
2	0	0.16	0.01			0.004		0.006
3	0.02	0.01	0.04			0.001		0.02
4								
5								
6	0.005	0.004	0.001			0.21		0.04
7								
8	0.04	0.006	0.02			0.04		0.08

***ColGrm* of Image 2** that have Colors: 1,2,3,5,7

	1	2	3	4	5	6	7	8
1	0.10	0	0.02		0.002		0.04	
2	0	0.22	0.01		0.004		0.006	
3	0.02	0.01	0.04		0.001		0.02	
4								
5	0.002	0.004	0.001		0.11		0.04	
6								
7	0.04	0.006	0.02		0.04		0.08	
8								

**Note: An example of Original ColGrm Matrix of Two Images, each with 8 colours, the shaded cells is considered as 0's during computation of dissimilarity measure.*

Figure 5.8: An example of original ColGrm matrix of two images with not matched colours.

For example, Figure 5.8 shows two images' *ColGrms*. The first one has the colours of numbers 1, 2, 3, 6 and 8; whereas, the second images has the colours 1, 2, 3, 5 and 7. Colours 1, 2 and 3 are matched colours and they will be compared directly, whereas the mismatched colours 6 and 8 of image 1 and colours 5 and 7 of image 2 will be compared with 0's. Hence, similar to the original *ColGrm*, the corresponding probability values of the mismatching colours in *DCBC* method can be considered as zeros. This exactly will simulate the original *ColGrm* and it can consider the second term in addition to P_{Corr} in a new proposed dual-model dissimilarity measure. The proposed duo-model of the adapted DCBC can be expressed as follows.

$$P_{duo}(Q,I) = P_{match}(Q,I) + P_{mismatch}(Q,I) \tag{5.6}$$

$$P_{match}(Q,I) = \left\{ \begin{array}{ll} \sum_{i,j \,\epsilon\, MC}^{m} \sum_{k=1}^{d} \left(\frac{|\gamma_{ci,cj}^{(k)}(Q) - \gamma_{ci,cj}^{(k)}(I)|}{1 + \gamma_{ci,cj}^{(k)}(Q) + \gamma_{ci,cj}^{(k)}(I)} * a_{i,j} \right) & if\ i = j\ \&\ d \le 5 \\ \sum_{i,j \,\epsilon\, MC}^{m} \sum_{k=1}^{d} \frac{|\gamma_{ci,cj}^{(k)}(Q) - \gamma_{ci,cj}^{(k)}(I)|}{1 + \gamma_{ci,cj}^{(k)}(Q) + \gamma_{ci,cj}^{(k)}(I)} & otherwise \end{array} \right\} \tag{5.7}$$

$$\begin{aligned} P_{mismatch}(Q,I) \\ = \quad & \sum_{i,j \,\epsilon\, MMC_Q} \sum_{k=1}^{d} \frac{|\gamma_{ci,cj}^{(k)}(Q) - 0|}{1 + \gamma_{ci,cj}^{(k)}(Q) + 0} \\ + \quad & \sum_{i,j \,\epsilon\, MMC_I} \sum_{k=1}^{d} \frac{|\gamma_{ci,cj}^{(k)}(I) - 0|}{1 + \gamma_{ci,cj}^{(k)}(I) + 0} \end{aligned} \tag{5.8}$$

MC represents list of the matched colours between the two images *Q* and *I*. MMC_Q and MMC_I represent lists of the mismatched colours of images *Q* and *I* respectively. In addition, $a_{i,j}$ represents the similarity ratio between the colours C_i and C_j. It can be computed using the following equation:

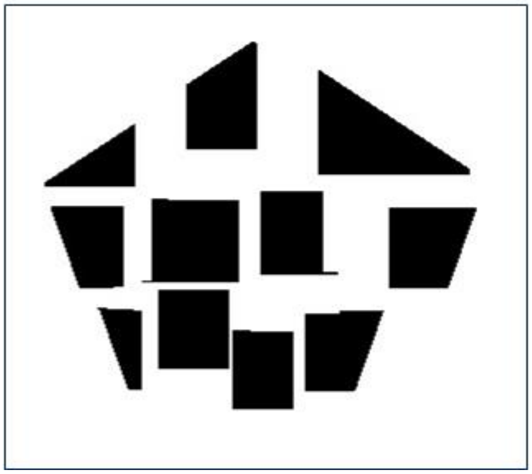
$$a_{i,j} = \begin{cases} 1 - \frac{d_{i,j}}{d_{max}} & if\ d_{i,j} \le T_c \\ 0 & if\ d_{i,j} > T_c \end{cases} \tag{5.9}$$

where $d_{i,j}$ represents L1 distance between *Ci* and *Cj*, the abbreviation C represents the 3-D colour values in CIE-LUV colour space, which can be computed in Eq. 5.10.

$$d_{i,j} = |C_i^L - C_j^L| + |C_i^U - C_j^U| + |C_i^V - C_j^V| \tag{5.10}$$

The colour threshold *Tc* represents the maximum distance whereby the two colours are considered similar and it sets to 10 and $d_{max}= \alpha$ Tc, α=1 or 1.2. In Eq. 5.7, $a_{i,j}$ will be multiplied to *ColGrm* dissimilarity values when d ≤ 5. The reasons behind multiplying only the main diagonal of *ColGrm* array by the colour similarity ratio ($a_{i,j}$) is that the main diagonal values often represent the percentages of colours in the image (especially when *d* is small) because the diagonal contains the probability of finding each colour with itself, as depicted in Figure 5.9, except the colours that are too scattered in the image. This is rarely used in images that are converted into eight DCs images. Whereas the other values in the *ColGrm* matrix represent the probabilities of finding a certain colour with other colours (spatial correlations). Hence, multiplying colour similarity ratio with DCs' percentages simulate DC-based approaches to alleviate problem of the matched colours are not identical, they are just similar.

Image



Color-Percentage Table

Black	0.241
White	0.759

CorreloGram Table

	Black	White
Black	0.23633	0.01072
White	0.01072	0.74222

**Note: This example to explain that percentages of colours in the image are approximately equal to the main diagonal of ColGrm when d is small.*

Figure 5.9: An example of colours percentages match with main diagonal of ColGrm.

Briefly, the differences between the adapted DC-based *ColGrm* descriptor and perceptual *ColGrm* descriptor lies in two positions. The first difference is the perceptual descriptor depends on metrics from different perspectives where it has three metrics to measure the dissimilarity between two images. $P_{\o}$ and P_G are used to measure global differences of colours; these metrics are produced from DC's approaches perspective. P_{Corr} measures the spatial correlations of matched colours only between the two images; this metric represents *ColGrm* perspective. Combining different perspective metrics may lead to inconsistency of these metric that in turn will produce inaccurate dissimilarity values. On the other hand, the adapted DC-based *ColGrm* depends on *ColGrm* perspective only, which measures global and spatial colour differences together efficiently. Therefore its accuracy is better than perceptual descriptor (as shown in experimental results in Section 5.4). Second difference lies in the dissimilarity measure of the matched colours in the perceptual *ColGrm* descriptor (P_{Corr}), it differs from the original metric. The new metric has serious limitation as explained early in Table 5.1; it cannot differentiate between large and small probabilities of colours correlation in the image, whereas there is no such case in the adapted DCBC method.

5.3.2 A Weighted DC-based Correlogram

In the latter section, dominant colours are used to solve infeasibility and perceptual problems of original *Correlogram* to produce DCBC. Thus, the two solutions of WDCD, which are mentioned in Chapter 4, Weight of DC (WDC) and Mutual Colour Ratio (MCR), can be applied also on DCBC to reduce the effect of dominant large percentage colour(s) on image retrieval process, especially when this large percentage colour(s) is the background colour.

Applying the two solutions of WDCD on advance descriptors such as *Correlogram* is not a straightforward task. Analyzing and understanding the way *ColGrm* works is required for adding these solutions (WDC and MCR).

In *ColGrm*, there are spatial correlations among colours in the image. Each element in *ColGrm* table relates to two colours and not one (spatial correlation of two colours at specific distance). If one of these two colours is a background colour (it has low weight) then the correlation importance of these colours must be decreased or totally removed (when the WDC equal zero). In other words, when removing a background colour, the correlations of this background colours with other image colours will become not useful and they will be considered as not existing. It is worth mentioning that each *ColGrm* table (of each image) is divided into two parts; first for matched colours with the compared image and second part for mismatched colours. Hence, WDC is applied to both of them to each image, as detailed in Algorithm 5.3, Algorithm 5.4 and Algorithm 5.5. Then it computes the dissimilarity measure of the two images normally using Eq. 5.7 and Eq. 5.8.

Algorithm 5.3: Weighted DC-based Correlogram

Input

CGT1, *CGT2*: Colour Correlogram Tables of the two compared images each of them of size (8^2dis) as maximum.

DCs1, *DCs2*: Dominant Colours Values of the two Compared images.

Warr1, *Warr2*: Final Weight Arrays of the two compared images computed by Table 4.1.

dis: number of distances in Correlogram Tables.

Output

MutualCGT1, *MutualCGT2*: Correlogram Tables of the two images that produced from of the matched colours between the two compared images.

RemCGT1, *RemCGT2*: Correlogram Tables of the two images that resulted from the mismatched (remaining) colours between two images.

BEGIN

1. Find Matched and Mismatched DCs between the two compared images

   Call *Find_matched_DCs* (DCs1, DCs2, **out** MatchedColourList, **out** MismatchedColoursList1, **out** MismatchedColoursList2);

2. Find Correlograms for two images of Matched Colours and Mismatched Colours

   Call *Find_Mutual_ColGrms* (CGT1, DCs1, CGT2, DCs2, **out** MutualCGT1, **out** MutualCGT2, **out** RemCGT1, **out** RemCGT2);

3. Find Weighted ColGrms for Mutual Correlograms of the Two images
   - { Find Weighted ColGrms for Mutual ColGrms for two images }

$\forall\, i \in 0..Length(MatchedColorList) - 1$

$\forall\, j \in 0..Length(MatchedColorList) - 1$

$\forall\, d \in 1..dis$

{

$W1 = min\,(Warr1[i], Warr1[j]);$

$MutualCGT1\,[i,j,d] = MutualCGT1\,[i,j,d] * W1;$

$W2 = min\,(Warr2[i], Warr2[j]);$

$MutualCGT2\,[i,j,d] = MutualCGT2\,[i,j,d] * W2;$

}

4. Find Weighted ColGrms for Remaining Correlograms
   - { for the First Remaining ColGrm Table }

$\forall\, i \in 0..Length(RemCGT1) - 1$

$\forall\, j \in 0..Length(RemCGT1) - 1$

$\forall\, d \in 1..dis$

{

$W1 = min\,(Warr1[i], Warr1[j]);$

$RemCGT1[i,j,d] = RemCGT1[i,j,d] * W1;$

}

   - { for the Second Remaining ColGrm Table }

$\forall\, i \in 0..Length(RemCGT2) - 1$

$\forall\, j \in 0..Length(RemCGT2) - 1$

$\forall\, d \in 1..dis$

{

$W2 = min\,(Warr2[i], Warr2[j]);$

$RemCGT2[i,j,d] = RemCGT2[i,j,d] * W2;$

}

5. Return $(MutualCGT1, MutualCGT2, RemCGT1, RemCGT2)$

END.

Algorithm 5.4: Find_matched_DCs

Input

DCs1, *DCs2:* Dominant Colours Values of the two Compared images

N1, *N2:* number of Dominant Colours in the two compared images

Output

MatchedColourList: Array of Matched Colours between two images

MismatchedColoursList1: Array of Colours in Image1 that mismatched with Image2

MismatchedColoursList2: Array of Colours in Image2 that mismatched with Image1

BEGIN

1. Find matched DCs between DCs1 and DCs2
   - { pass on all DCs in DCs1}

     $\forall\, dc1_i \in DCs1 \qquad i = 0..N1$

     - { pass on all colours in DCs2 to check their matching with *dc1*}

       mindis←1000 ; index← -1

       $\forall\, dc2_j \in DCs2 \qquad j = 0..N2$

       - {Compute distance of *dc1* with all colours in *dc2* to find nearest colour of DCs2}

         d← LUV_Distance ($dc1_i$, $dc2_j$) ;

         if (d < mindis) {mindis ←d; index← j;}

     - {put *dc1* and the nearest colour of *dc2* (if any) in the Matched Colour List, otherwise put *dc1* in Mismatched List 1 }

       If (d < Colour_Threshold)

       Add ($dc1_i$ and $dc2_{index}$ into MatchedColourList)

       Else

       Add ($dc1_i$ into MismatchedColoursList1)

2. Put the mismatched colours of DCs2 in Mismatched List 2
   - {pass on all colours in DCs2 to check their affiliation to Matched Colour List }

$\forall\, dc2_j \;\in DCs2 \qquad\qquad j = 0..N2$

○ {check *dc2* if it is not found in MatchedColourList }

If (*dc2*$_j$ not Exist in MatchedColourList)

Add (*dc2*$_j$ into MismatchedColoursList2)

3. Return (MatchedColourList ,MismatchedColoursList1,MismatchedColoursList2)

END.

Algorithm 5.5: Find_Mutual_ColGrms

Input

CGT1, CGT2: ColGrm Tables of the two compared images.

DCs1, DCs2: Dominant Colours Values of the two Compared images.

Output

MutualCGT1, MutualCGT2: ColGrm of Image1 and 2 respectively of matched DCs only.

RemCGT1, RemCGT2: ColGrm of Image1 and 2 respectively of mismatched DCs.

BEGIN

1. Find Matched DCs between the two compared images

   Call *Find_matched_DCs* (DCs1, DCs2, **out** MatchedColourList, **out** MismatchedColoursList1, **out** MismatchedColoursList2);

2. Find ColGrms of the Matched Colours for the first image
   - { pass on all colours in Matched Colour List }

   $\forall\, i \in indexes\ of\ DCs1\ in\ MatchedColorList$ [0.. Length(MatchedColourList)-1]

   $\forall\, j \in indexes\ of\ DCs1\ in\ MatchedColorList$

   ○ { move ColGrm Values to New ColGrm Tables}

   MutualCGT1[i, j]=CGT1[index(colour(i)) in DCs1, index(colour(j)) in DCs1];

3. Find ColGrms of the Matched Colours for the second image
   - { pass on all colours in Matched Colour List }

   $\forall\, i \in indexes\ of\ DCs2\ in\ MatchedColorList$ [0.. Length(MatchedColourList)-1]

   $\forall\, j \in indexes\ of\ DCs2\ in\ MatchedColorList$

○ { move ColGrm Values to New ColGrm Tables}

MutualCGT2[i, j]=CGT2[index(colour(i)) in DCs2, index(colour(j)) in DCs2];

4. Find ColGrms of the Mismatched Colours for the first image
   - { pass on all colours in Mismatched Colour List1 }

     $\forall\, i \in indexes\ of\ DCs1\ in\ MismatchedColorList1$ [0 .. Length(MismatchedColourList1)-1]

     $\forall\, j \in indexes\ of\ DCs1\ in\ MismatchedColorList1$

     ○ { move ColGrm Values to New ColGrm Tables}

     RemCGT1[i, j]=CGT1[index(colour(i)) in DCs1, index(colour(j)) in DCs1];

5. Find ColGrms of the Mismatched Colours for the second image
   - { pass on all colours in Mismatched Colour List2 }

     $\forall\, i \in indexes\ of\ DCs2\ in\ MismatchedColorList2$ [0 .. Length(MismatchedColourList2)-1]

     $\forall\, j \in indexes\ of\ DCs2\ in\ MismatchedColorList2$

     ○ { move ColGrm Values to New ColGrm Tables}

     RemCGT2[i, j]=CGT2[index(colour(i)) in DCs2, index(colour(j)) in DCs2];

6. Return ($MutualCGT1, MutualCGT2, RemCGT1, RemCGT2$)

END.

Based on MCR, it is related to images' DCs and far from *ColGrm* table, thus it can be computed directly from DCs respective to the importance (weights) of DCs, as depicted in Algorithm 5.6. Then it can be added to the dissimilarity duo-model (Eq. 5.6) to yield a result of an adapted dissimilarity measure for Weighted DCBC (WDCBC) as in Eq. 5.11.

$$DisSimilarity\ Measure_{WDCBC}(Q,I) = (\,P_{match}(Q,I) + P_{mismatch}(Q,I)) * (1 - MCR) \quad (5.11)$$

Algorithm 5.6: Mutual Colour Ratio for Weighted DC-based methods

Input

DCs1, DCs2: Dominant Colours Values of the two Compared images

Warr1, Warr2: Final Weight Arrays of the two compared images computed by Table 4.1

N1, N2: number of Dominant Colours in the two compared images

Output

MCR: Mutual Colour Ratio between two images

BEGIN

1. Find Matched DCs between the two compared images

   Call *Find_matched_DCs* (DCs1, DCs2, **out** MatchedColourList, **out** MismatchedColoursList1, **out** MismatchedColoursList2);

2. Clear the counter of mutual colours

   MC_Counter ← 0

3. Find Mutual Colours between two images according to importance of colours

   $\forall\, i \in 0..Length(MatchedColorList) - 1$

   MC_Counter ← MC_Counter + Min(Warr1[i],Warr2[i])

4. Compute Mutual Colour Ratio

   MCR← MC_Counter / Max(N1, N2)

5. Return (MCR)

END.

5.4 Experimental Evaluation of Correlogram

In this section, a number of experiments have been conducted to evaluate the proposed and adapted *Correlogram* with original *ColGrm* and *Autocorrelogram* descriptors.

5.4.1 Experimental Setup

This section is dedicated to identify some setup parameters that will be used in the experiments of the current work. These parameters are image datasets that are used for the purpose of verifying and comparing the performance of the proposed and adapted *ColGrm* descriptors together with the original *ColGrm* descriptors. The competing descriptors are the second parameter, which are used for comparison with the proposed descriptors. Besides, to measure the performance of the competing descriptors, quantitative metrics will be used as the third parameter for measuring the performance of the candidate and proposed descriptors. Moreover, all experiments are carried out on a 32-bit machine of 1.8 GHz processor with 3 GB memory.

A. Image Datasets

Evaluating the proposed Compact-Generalized *ColGrm* descriptor, adapted DC-based *ColGrm* descriptor and their integration are conducted on two datasets: 1) Corel-10K dataset; 2) Cartoon-11K dataset. This latter database is an expansion to the Cartoon-5K dataset that is introduced in Chapter 4. The two datasets are used to show superiority of the proposed descriptors in the large databases. These datasets are explained in Section 3.2.5.

B. Competing Descriptors

The descriptors that are selected to be compared with the proposed CGC, adapted DCBC, integration of them and Weighted DCBC are the original *ColGrm* (Huang et al., 1997; Kunttu et al., 2003) (whenever it can be applied), *Autocorrelogram* (Huang et al., 1997), MPEG-7 Dominant Colour Descriptor (DCD) (Yamada et al., 2001) and perceptual *ColGrm* (Kiranyaz et al., 2012). This is because the first two descriptors represent the original *ColGrm* descriptor, which are considered as the base of the proposed descriptors. The third descriptor, DCD, is the base of any DC-based approaches, which is used in adapted DCBC. Lastly, perceptual *ColGrm* represents the original descriptor that is adapted to produce DCBC descriptor.

C. Performance Measure Metrics

Quantitative performance measure metrics are utilized to measure the accuracy of the proposed descriptors with the other *ColGrm* descriptors that are candidate for the purpose of comparison. These metrics are ARR, ANMRR, P(10) and MAP; details of these metrics are explained in Section 2.1.6.1 and Section 3.2.6.2.

Additionally, the complexity of the proposed descriptors, in terms of time and memory space, is urgently computed, as fifth metric, to prove their applicability in the large databases. Applicability of the proposed descriptors in the large databases is the main purpose of this chapter; the accuracy metrics are used to prove that the compactness of the proposed descriptors does not significantly degrade the performance (if any).

5.4.2 Retrieval Performance

Retrieval performance of the competing descriptors in previous specified datasets can be measured using the accuracy and complexity metrics; complexity metrics represent computing time and memory space that is needed for each descriptor. Time is divided into feature extraction time (offline) and image retrieval time (online). Memory space is referred to as main memory or disk space required for the descriptors. Diversity of queries is also very important to ensure fair and honest results (Grubinger, 2007), thus the evaluation queries are selected from all classes of the databases.

A. Retrieval performance of Corel-10K Dataset

Some visual retrieval results are shown in Figure 5.10 of all candidate *ColGrm* descriptors that are performed on Corel-10K dataset with 117 queries. Additionally, the four evaluation metrics and complexity of memory space are also computed in Tables 5.2 and 5.3, respectively, to show the accuracy and efficiency of the proposed methods comparing with other descriptors.

Table 5.2: *Four Evaluation Metrics for all competing Correlogram descriptors on Corel-10K database.*

Descriptor Metric	Original Correlogram	*AutoCorrelogram*	Proposed CGC	MPG-7 DCD	Perceptual CG	Adapted DCBC	Integration CGC+DCBC	Weighted DCBD
ANMRR	0.825/827/NA	0.875/879/885	0.826/830/836	0.897	0.887/921/971	0.813/814/817	0.815/818/821	**0.793/795/805**
RR	0.138/135/NA	0.098/095/090	0.137/135/128	0.082	0.090/070/022	0.150/148/146	0.139/137/135	**0.168/165/156**
P(10)	0.37/0.36/NA	0.27/0.27/0.26	0.37/0.36/0.34	0.24	0.25/0.19/0.13	0.39/0.38/0.36	0.36/0.35/0.34	**0.44/0.44/0.43**
MAP	0.124/122/NA	0.086/080/076	0.123/120/114	0.067	0.069/051/023	0.128/125/122	0.119/117/116	**0.139/137/131**

**ANMRR, ARR, P(10) and MAP values for all competing descriptors on Corel-10K database with 117 queries (with no. of colours equals 3*3*3 = 27 colours and distance=5,10 and 40). NA means Not Applicable.*

Table 5.3: *Improvement Percentages of Proposed CGC over AutoCorrelogram Adapted DCBC over Original Correlogram and Weighted DCBC over DCBC in setting 27 colours and distance=5.*

	Proposed CGC over *AutoCorrelogram*	Adapted DCBC over Original *ColGrm*	Weighted DCBC over DCBC
ANMRR	28.1%	6.4%	9.6%
RR	28.4%	8.0%	10.7%
P(10)	27.0%	5.1%	11.3%
MAP	30.0%	3.1%	7.9%
Average	28.38%	5.65%	9.88%

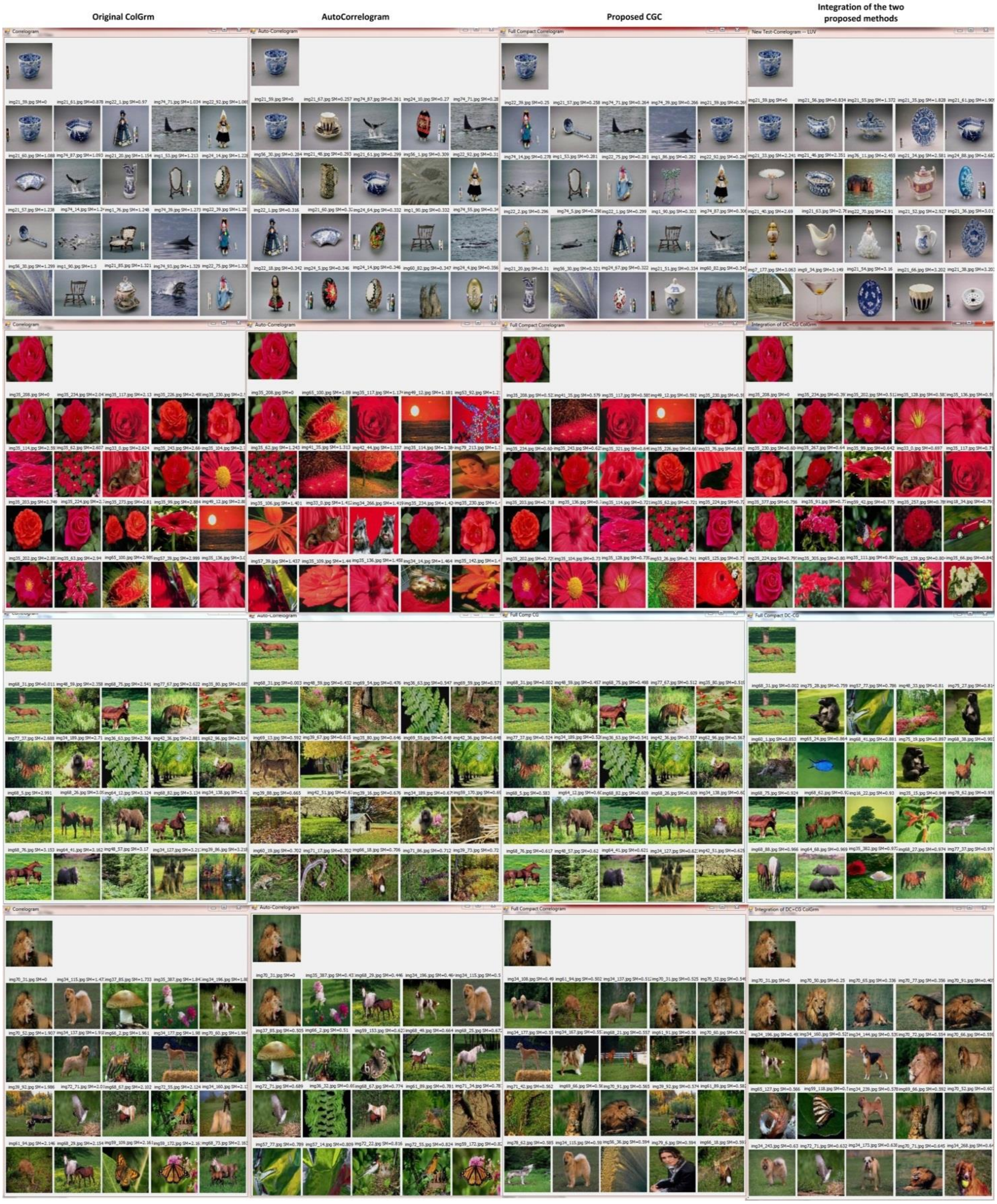


Figure 5.10: Visual Results of eight competing descriptors on Corel-10K database

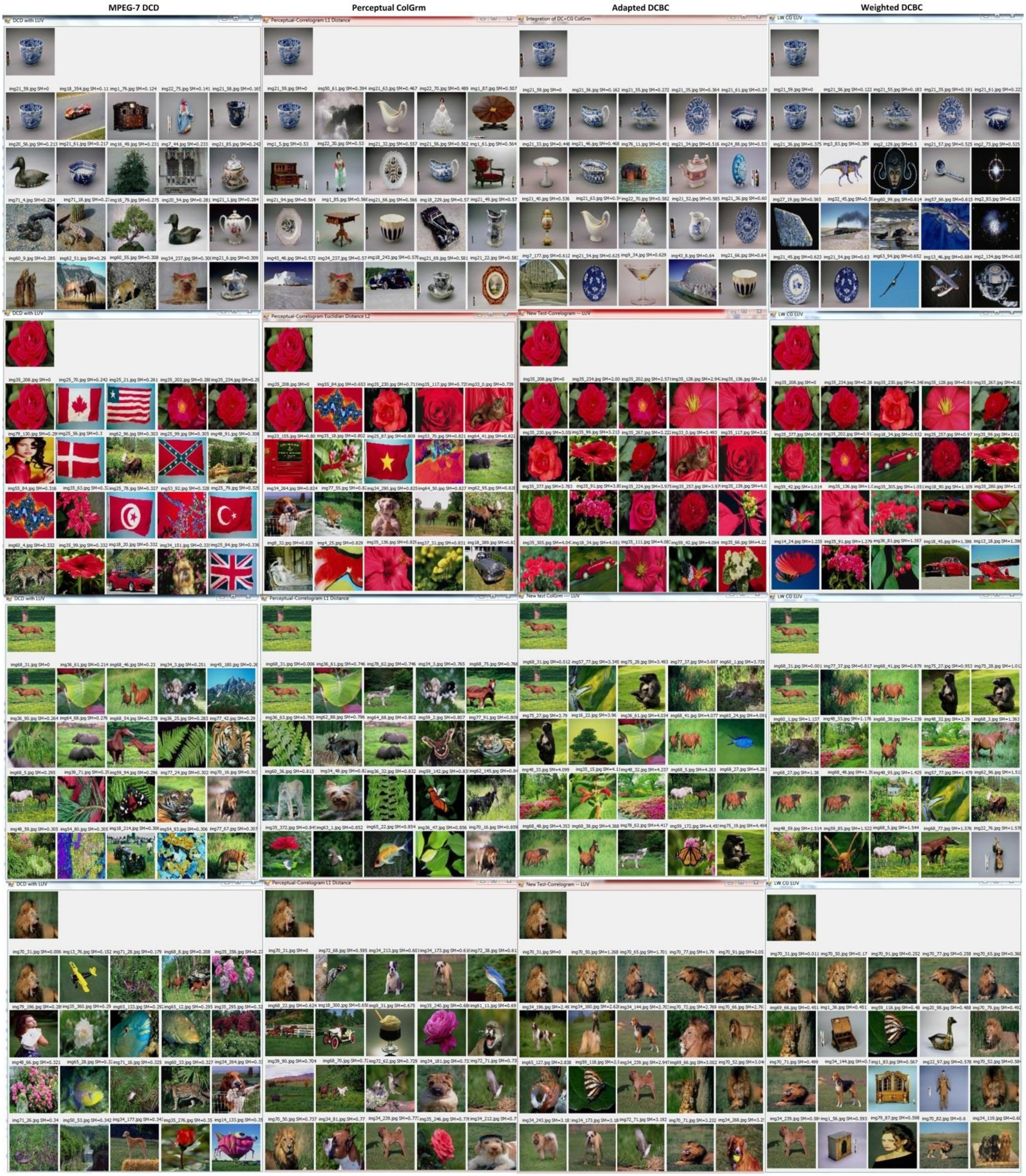


Figure 5.10: Visual Results of eight competing descriptors on Corel-10K database (Cont.)

Unlike other descriptors, single value of in the ‘MPEG7 DCD’ column, in Table 5.2, means that this descriptor does not have different setting of distances to compute. It depends on percentages of colours rather than distances among colours, which are used in spatial *ColGrm* methods.

In the left part of Table 5.2 (first three columns), one can notice that the best accuracy values are these of original *ColGrm*, better than proposed CGC, but it is applied in minimum settings (3*3*3 colours of each band and distances are 5 and 10) only, as shown in Table 5.2 and 5.4. This slight degradation of accuracy of the proposed descriptor is due to the generalization of distances that is losing values of the accurate distances. But when a comparison is made, increasing the setting such as 4*4*4 colours and 5, 10 and 40 distances, the *Autocorrelogram* and the proposed CGC can be applied only and the proposed descriptor outperforms on the *Autocorrelogram*. This is due to its preserve spatial correlation of each colour with other colours in the image whereas the *Autocorrelogram* has spatial correlation of each colour with itself and ignore the others. In *ColGrm* descriptors, the accuracy is decreased when the number of distances is increased because the unsuitable distances will effect on the suitable distances. In other words, there is certain distance, or distances, that represent the actual distance between specific colour and other colours in the image. This distance is called suitable distance and thus others are unsuitable. Although increasing the distances of the proposed descriptor, the memory space and image retrieval time is still $O\left(\frac{m^2}{2}+\frac{m}{2}\right)$, which are online processes where they are performed when comparing query image *ColGrm* with all database images’ *ColGrms*. The increasing of distances affects on the feature extraction process for all database images, which is offline process (performed once only when creating the database away from interaction with users). Additionally, it effects on online *ColGrm* extraction process of query image only; where the complexity of its computation and memory space complexity is $O(m^2d)$ that is equaled to original *ColGrm*. The complexity of proposed descriptor in all CBIR stages is depicted in Figure 5.5.

In the middle part of Table 5.2 (Second three columns), the adapted DCBC outperforms the three original descriptors (DCD, *ColGrm* and perceptual *ColGrm*; Perceptual *ColGrm* is the descriptor that integrates the DCD and *ColGrm*). The adapted descriptor shows more accuracy than its original version (Kiranyaz et al., 2012) because the latter has many drawbacks as mentioned in Section 5.3. The complexity of the perceptual and proposed DCBC descriptors are $O(8^2d)$ as maximum, where 8 represents the maximum DCs that can be extracted from the image. Additionally, one can notice the significant degradation accuracy of the perceptual descriptor when the distance is increased. This is because the incompatibility between spatial dissimilarity (P_{Corr}) and global dissimilarity ($P_{\emptyset}$ and P_G,). When the distance is increased, P_{Corr} is changed significantly while the global dissimilarity values ($P_{\emptyset}$ and P_G,) remain unchanged. This emphasize that the dissimilarity measure of the perceptual *ColGrm* descriptor has serious limitation.

Integration of both proposed methods is achieved by applying compactness and generalization concepts of the CGC (first proposed descriptor) on DC-based *ColGrm* (second adapted descriptor). It also outperforms all three original descriptors (MPEG-7 DCD, *ColGrm* and Perceptual *ColGrm*) with complexity $O\left(\frac{8^2}{2}+\frac{8}{2}\right) = O(36)$ as maximum. The single value within an entire row in Table 5.4 means that either this descriptor does not have different distances in its computations (such as MPEG7 DCD) or this descriptor produces the same memory space for all distances (such as proposed CGC and Integration of CGC and DCBC). From Table 5.3 and 5.4, one can notice that the integration of CGC and DCBC is indeed a promising approach in its little consumption not just in memory space but also in images retrieval time and accuracy.

Table 5.4: *Memory Space for Features' Database of all competing descriptors for Corel-10K dataset and 27 Colours.*

***ColGrm* Method**	**Distance=5**	**Distance=10**	**Distance=40**
Original *ColGrm*	278.1 M	556.2 M	2.17 G
Autocorrelogram	10.3 M	20.6 M	82.4 M
Proposed CGC		28.8 M	
MPEG7 DCD		0.85 M	
Conceptual *ColGrm*	25.2 M	49.7 M	196.1 M
Proposed DCBC	25.2 M	49.7 M	196.1 M
Integration of CGC+DCBC		3.6 M	

When increasing setting of *ColGrm* into 4 colours for each band (4*4*4=64 colours), the proposed CGC outperformed on all other DC-based competing descriptors. This is because the variety of colours (64 in CGC) is higher significantly than DC-based *ColGrm* approach (which has 8 colours as a maximum)).

Table 5.5: *Evaluation metrics for all Correlogram descriptors on Corel-10K database.*

Descriptor / Metric	Original Correlogram	*AutoCorrelogram*	Proposed CGC	MPEG-7 DCD	Perceptual CG	Adapted DCBC	Integration CGC+DCBC	Weighted DCBC
ANMRR	NA	0.840/845/852	**0.781/785/789**	0.897	0.887/921/971	0.813/814/817	0.815/818/821	0.793/795/805
RR	NA	0.127/120/112	**0.176/172/169**	0.082	0.090/070/022	0.150/148/146	0.139/137/135	0.168/165/156
P(10)	NA	0.33/0.33/0.31	**0.49/0.49/0.48**	0.24	0.25/0.19/0.13	0.39/0.38/0.36	0.36/0.35/0.34	0.44/0.44/0.43
MAP	NA	0.114/110/103	**0.165/0.161/155**	0.067	0.069/051/023	0.128/125/122	0.119/117/116	0.139/137/131

** ANMRR, ARR, P(10) and MAP values for all competing descriptors on Corel-10K database with 117 queries (with no. of colours equals 4*4*4 = 64 colours and distance=5,10 and 40). NA means Not Applicable.*

Table 5.6: *Accuracy Improvement Percentages of Proposed CGC over AutoCorrelogram and Weighted DCBC in setting 64 colours and distance=5.*

	Proposed CGC over *AutoCorrelogram*	Proposed CGC over Weighted DCBC
ANMRR	26.9%	5.4%
RR	27.8%	4.5%
P(10)	32.6%	10.2%
MAP	30.9%	15.7%
Average	29.55%	8.95%

As shown in Table 5.5, the original colour *Correlogram* is inapplicable in the setting of 64 (4*4*4) colours. This is because it has serious limitations, which are high computational complexity and memory storage (see Table 5.7); only *Autocorrelogram* and all compact descriptors can be applied on the machine that performed the experiments. Additionally, both proposed descriptors and their integration outperformed *Autocorrelogram* and perceptual *ColGrm*. The key contribution of this work is solving the feasibility problems, computations and memory space, of original *ColGrm*. In other words, any increment in the setting (such as distance or number of colours) of *ColGrm* will cause that the only proposed descriptors can be applied. Results of increasing the setting to more than four colours in each band is not displayed in this research because it showed similar results to four colours' setting.

Table 5.7: *Size of Features for all competing descriptors of 64 colours in Corel-10K Database.*

ColGrm Method	Distance=5	Distance=10	Distance=40
Original *ColGrm*	1.52 G	3.1 G	12.2 G
Autocorrelogram	24.4 M	48.8 M	195.3 M
Proposed CGC		158.7 M	
MPEG7 DCD		0.85 M	
Conceptual *ColGrm*	25.2 M	49.7 M	196.1 M
Proposed DCBC	25.2 M	49.7 M	196.1 M
Integration of CGC+DCBC		3.6 M	

In the experiments, there are some classes that show worse accuracy in DC-based descriptors than original *ColGrm* because their images are varied in colours and this will impose effect on any limited colour-based descriptors such as DCD and in turn this will effect on the DC-based *ColGrm* descriptors. The original *ColGrm* and proposed CGC can alleviate this problem because they use abundance of colours.

B. Retrieval performance of Cartoon-11K Dataset

Visual image retrieval results of all candidates *ColGrm* descriptors that is performed on Cartoon-11K dataset with 158 queries are shown in Figure 5.11. Additionally, the four evaluation metrics are also computed in Tables 5.8, 5.9, 5.10 and 5.11, respectively, to show the accuracy of the proposed methods compared to other descriptors.

From Table 5.9, one can notice that the adapted DCBC descriptor outperforms all competing descriptors including perceptual descriptor. Whereas the proposed descriptor CGC shows same accuracy to the original *ColGrm* but with significant reduction in the complexity where it is reduced from $O(m^2d)$ to $O(\frac{m^2}{2}+\frac{m}{2})$.

In Table 5.10, the setting is set to four colours and the proposed CGC outperforms the adapted DCBC due to the abundance of its colours that can be expressed on the image content more efficiently than DCs. Storage space requirement for cartoon database is approximately equal to this in Corel database that is depicted in Tables 5.4 and 5.7. These Tables show that the compactness of the proposed, adapted and their integration descriptors that in turn will speed up image retrieval process.

Figure 5.11: Visual Results of Seven Competing Descriptors on Cartoon-11K database.

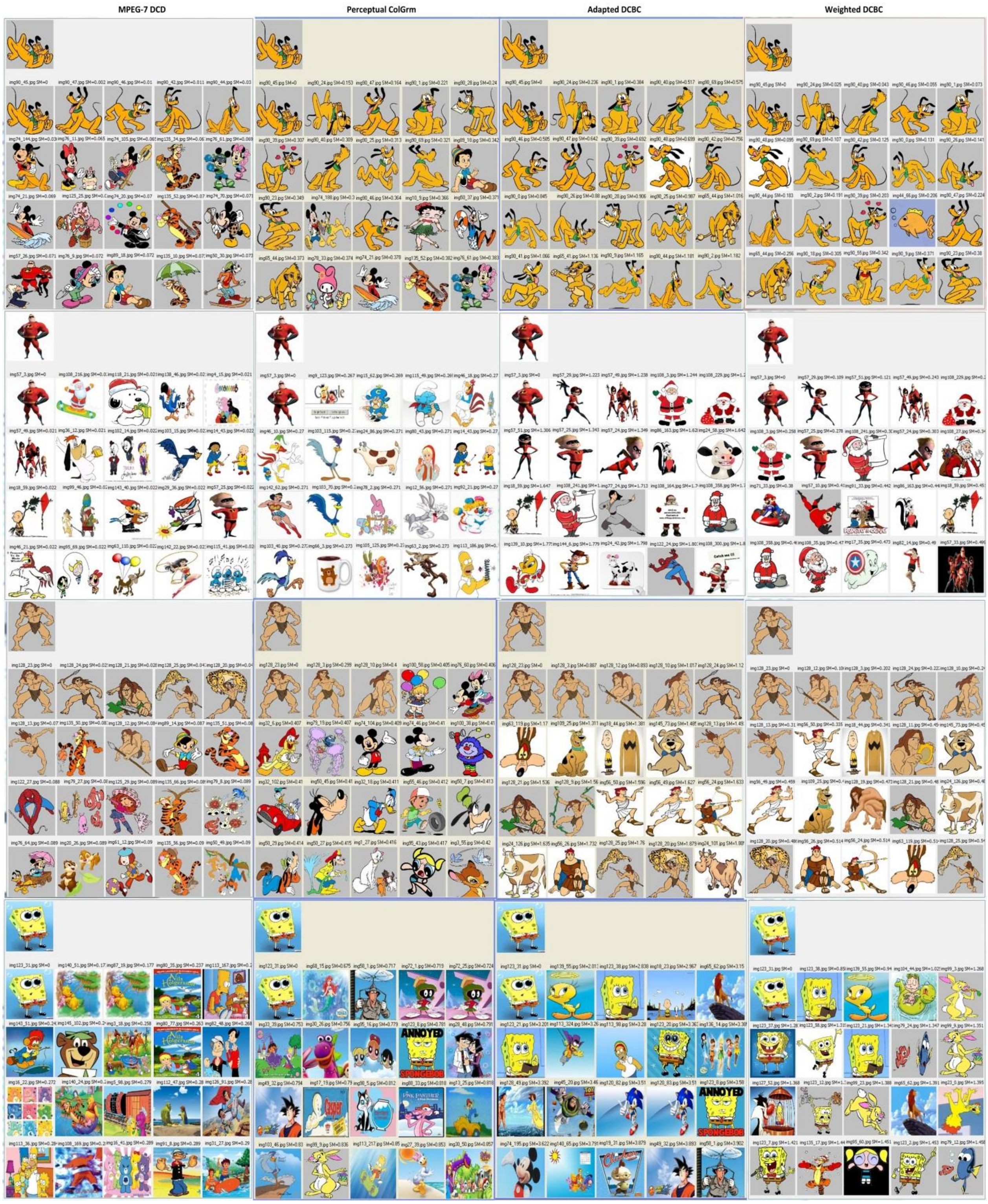


Figure 5.11: Visual Results of Seven Competing Descriptors on Cartoon-11K database (cont.)

Table 5.8: *Evaluation metrics for all Correlogram descriptors on Cartoon-11K database with 27 colours.*

Descriptor / Metric	Original Correlogram	*AutoCorrelogram*	Proposed CGC	MPEG-7 DCD	Perceptual CG	Adapted DCBC	Integration CGC+DCBC	Weighted DCBC
ANMRR	0.853/0.853/NA	0.880/890/902	0.853/854/855	0.945	0.927/944/969	0.838/838/839	0.841/844/851	**0.810/813/815**
ARR	0.118/117/NA	0.094/088/077	0.117/117/116	0.041	0.057/041/023	0.130/130/130	0.126/123/117	**0.152/147/141**
P(10)	0.35/.35/NA	0.29/.26/.24	0.35/.35/.35	0.08	0.20/.17/.10	0.39/.38/.39	0.37/.37/.36	**0.41/0.41/0.40**
MAP	0.098/097/NA	0.075/069/060	0.097/097/097	0.029	0.045/038/023	0.105/105/104	0.102/098/094	**0.124/121/118**

** ANMRR, ARR, P(10) and MAP values for all competing descriptors on Cartoon-11K database with 158 queries (with no. of colours equals 3*3*3 = 27 colours and distance=5,10 and 40). NA means Not Applicable.*

Table 5.9: *Improvement Percentages of Adapted DCBC over Correlogram and Weighted DCBC over DCBC in setting 27 colours and distance=5 in Cartoon-11K dataset.*

	Proposed CGC over *AutoCorrelogram*	Adapted DCBC over Original *ColGrm*	Weighted DCBC over DCBC
ANMRR	18.4%	9.2%	14.7%
RR	19.7%	9.2%	14.5%
P(10)	17.1%	10.2%	4.9%
MAP	22.7%	6.6%	15.3%
Average	19.48%	8.8%	12.35%

Table 5.10: *Evaluation metrics for all Correlogram descriptors on Cartoon-11K database with 64 colours.*

Descriptor / Metric	Original Correlogram	*AutoCorrelogram*	Proposed CGC	MPEG-7 DCD	Perceptual CG	Adapted DCBC	Integration CGC+DCBC	Weighted DCBC
ANMRR	N/A	0.867/870/892	0.830/833/835	0.945	0.927/944/969	0.838/838/839	0.841/844/851	**0.810/813/815**
ARR	N/A	0.107/100/089	0.136/135/133	0.041	0.057/041/023	0.130/130/130	0.126/123/117	**0.152/147/141**
P(10)	N/A	0.32/.28/.25	0.41/.40/.38	0.08	0.20/.17/.10	0.39/38/39	0.37/.37/.36	**0.41/0.41/0.40**
MAP	N/A	0.083/079/070	0.114/110/108	0.029	0.045/038/023	0.105/105/104	0.102/098/094	**0.124/121/118**

** ANMRR, ARR, P(10) and MAP values for all competing descriptors on Cartoon-11K database with 158 queries (with no. of colours equals 4*4*4 = 64 colours and distance=5,10 and 40). NA means Not Applicable.*

Table 5.11: *Accuracy Improvement Percentages of Proposed CGC over AutoCorrelogram and Weighted DCBC in setting 64 colours and distance=5 in Cartoon-11K dataset.*

	Proposed CGC over *AutoCorrelogram*	Weighted DCBC over Proposed CGC
ANMRR	21.8%	10.5%
RR	21.3%	10.5%
P(10)	22.0%	0.0%
MAP	27.2%	8.1%
Average	23.10%	7.28%

In the Cartoon-11K dataset, weighted DCBC outperforms all competing descriptors in both colour settings 27 and 64. This is because the nature of cartoon dataset that fits the purpose in which the weighted method is designed for. It contains cartoon characters that share almost similar colours in image database.

5.5 Generalizing both of Large Quantized Colours into Few Dominant Colours Conversion and Weighting of Dominant Colours Methods

A conversion from large number of colours-based *ColGrm* into few dominant colours-based *ColGrm* is proposed by (Kiranyaz et al., 2012) to solve infeasibility and perceptual problem of original *ColGrm*. In Section 5.3, the problem of trio-model similarity measure is addressed and solved by the new duo-model similarity measure. Moreover, assigning weights to DCs, which is applied to DCD in Chapter 4, is also applied to DC-Based *ColGrm* in Section 5.3.2. This is performed to reduce the dominance problem of large percentage colours on image retrieving results. To generalize the proposed methods, two generic frameworks are proposed. The first one is converting a large number of colour-based methods into few dominant colours-based methods. The second framework is weighting the dominant colours of DC-based descriptors. This generalization can be achieved by applying these methods (concepts) to other descriptors. Therefore, in this section, Border Interior pixel Classification method is selected to apply these methods to proof their generality on colour-based methods. This selection is justified by: 1) BIC is considered as one of the best colour-based descriptors with *ColGrm* (Pedronette & Torres, 2012; Penatti et al., 2012); 2) It has different format compared to simple methods (like colour histogram) and complicated methods (like colour *Correlogram*); 3) It is proposed for broad image domains (Renato et al., 2002) and can be applied to a wide range of applications, hence the proposed method will be useful for many upcoming research.

5.5.1 Border-Interior pixel Classification method Overview

Border-Interior pixel Classification (BIC) is a compact and efficient image retrieval method and is designed for wide image domains. It classifies image's pixels into either border or interior where if the pixel has one of its 4-neighbours (horizontal and vertical neighbours only) of different colour than its colour then this pixel will be classified as border pixel otherwise it will classify as interior pixel (all its neighbours have same its colour). This method is considered a powerful method because it has somewhat semantic information about object's shape in the image (Penatti et al., 2012; Renato et al., 2002). Furthermore, it is practically tested to be at the forefront of colour-based descriptors in its accuracy beside *Autocorrelogram* (as replacement of *Correlogram* because its infeasibility problem in the large image databases) (Pedronette & Torres, 2012; Penatti et al., 2012). Moreover, it used new logarithmic distance (*dLog* distance) to reduce the effect of large percentage colours on images similarity measure. This *dLog* distance offers more compact representation to BIC features representation. This is because the result of *dLog* distance ranges from 0 to 9, which allows using 4-bits only to save these values instead of 8-bits to save values ranging from 0-255 resultant from L_1 or L_2 distance.

BIC used 64 (4*4*4) colours in RGB space for building its features because this number of colours is widely used in the literature and it is effective (Renato et al., 2002). Therefore, BIC has 128-bins histogram (64 bins represent how many pixels of each colour is classified as border pixels and 64 bins represent how many pixels for each of 64 colours classified as interior pixels). Each value of BIC histogram is first normalized to be between 0- 255 (8-bits required). Then it is converted by *dLog* distance into values between 0-9 (4-bits required) to compact histogram size from 128 bytes into 64 bytes. To find similarity of two images, *dLog* distance is applied to their BIC histograms as shown in the following equations:

$$dLog(BICh1, BICh2) = \sum_{i=0}^{M-1} |f(BICh1[i]) - f(BICh2[i])| \qquad (5.12)$$

$$f(x) = \begin{Bmatrix} 0 & if\ x = 0 \\ 1 & if\ 0 < x \leq 1 \\ |log_2(x)| & otherwise \end{Bmatrix} \qquad (5.13)$$

BICh1[i] and *BICh2[i]* represent BIC histograms bins of the two compared images; M represents number of bins for BIC histogram, which equals to 128 in this case. *f(x)* function is used to take logarithm of histogram bins. Examples of BIC results are depicted in Figure 5.12, where black pixels represent border pixels and white pixels represent interior pixels.

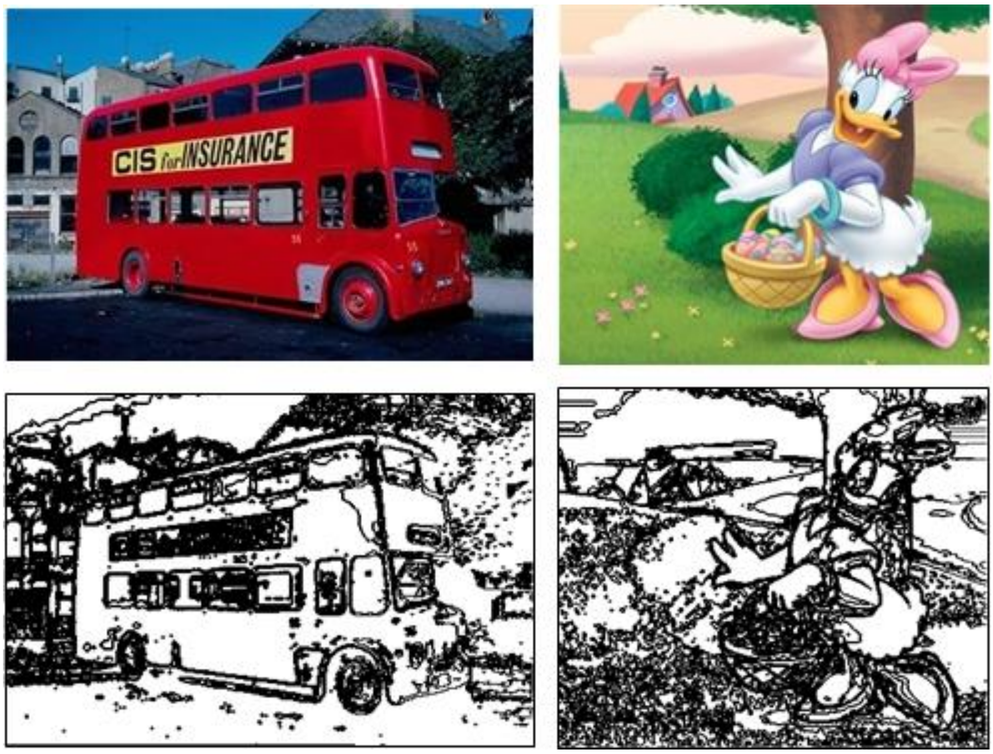


Figure 5.12: Results of BIC to show Border and Interior Pixels classification.

5.5.2 Dominant Colours-based BIC Approach

First step to convert any large number of colours into a few DCs method is find the DCs of an image. MPEG-7 DCD (Yamada et al., 2001) is used in this work to extract 8 DCs (as maximum) to be a representative of the image. Therefore, 16 bins BIC histogram will be used to describe border and interior pixels (8-bins represent border pixels for each one of the 8 DCs and another 8-bin for representing interior pixels for same DCs). Moreover, normalization is made on the histogram to obtain a range of 0-255 values; then applying *f(.)* function (Eq. 5.13) in second step to convert histogram bins values into a range between 0 and 9 that required half byte (4-bits) only to store the bin value. *dLog* distance (Eq. 5.12) is also used to compare similarities between two images. Since there is no static colour space (such as 64 colours) in DC-based methods; thus a reference to each of the eight DCs is used and all DCs values is saved separately. The similarity measure of DC-based BIC will be different than that in original BIC.

An approach that is similar to DC-based *Correlogram* will be used (duo-model dissimilarity measure). Firstly, a list of matched colours (using L_1 distance in LUV colour space with colour threshold=25) as well as lists of mismatched colour for both images are determined. Secondly, BIC histogram bins of matched list are compared directly whereas the bins of mismatched lists are compared with zero (exactly the same concept that is applied in DC-based *Correlogram* in Chapter 4), as expressed in Eq. 5.14, Eq. 5.15 and Eq. 5.16. This approach is simulated to the original BIC method where the colour that exists in one image and does not exist in the second one; the histogram bin of the first image will have a certain value while that bin of the second image will be zero. Overall similarity process of original BIC and DC-based BIC are depicted in Figures 5.13 and 5.14 respectively.

$$Dissimlarity\ Measure_{DC_{BIC}}(Q,I) = P_{Match}(Q,I) + P_{Mismatch}(Q,I) \qquad (5.14)$$

$$P_{Match}(Q,I) = \sum_{i=0}^{ML-1} \left| f\left(BICh_Q[i]\right) - f\left(BICh_I[i]\right) \right| \qquad (5.15)$$

$$P_{Mismatch}(Q,I) = \sum_{i=0}^{MML_Q-1} \left| f\left(BICh_Q[i]\right) - 0 \right| + \sum_{i=0}^{MML_I-1} \left| f\left(BICh_I[i]\right) - 0 \right| \qquad (5.16)$$

ML represents list of matched colours, MML_Q and MML_I represent mismatched lists for query image (Q) and database image (I). $BICh_Q$ and $BICh_I$ represent BIC histogram for query and database images.

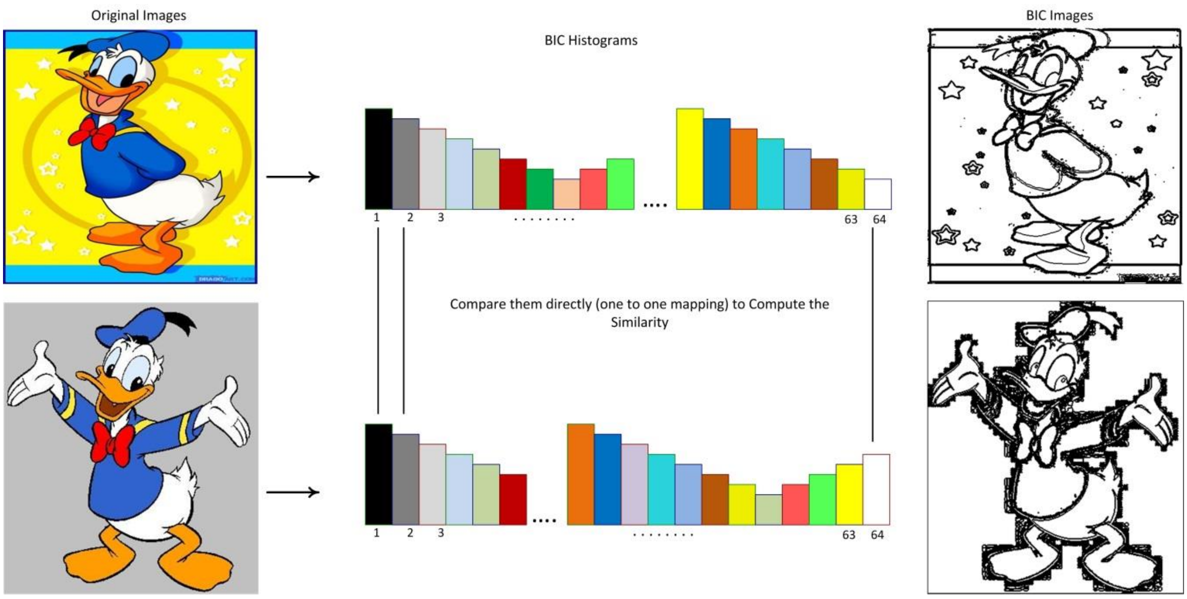


Figure 5.13: Similarity Measure for two images in original BIC method.

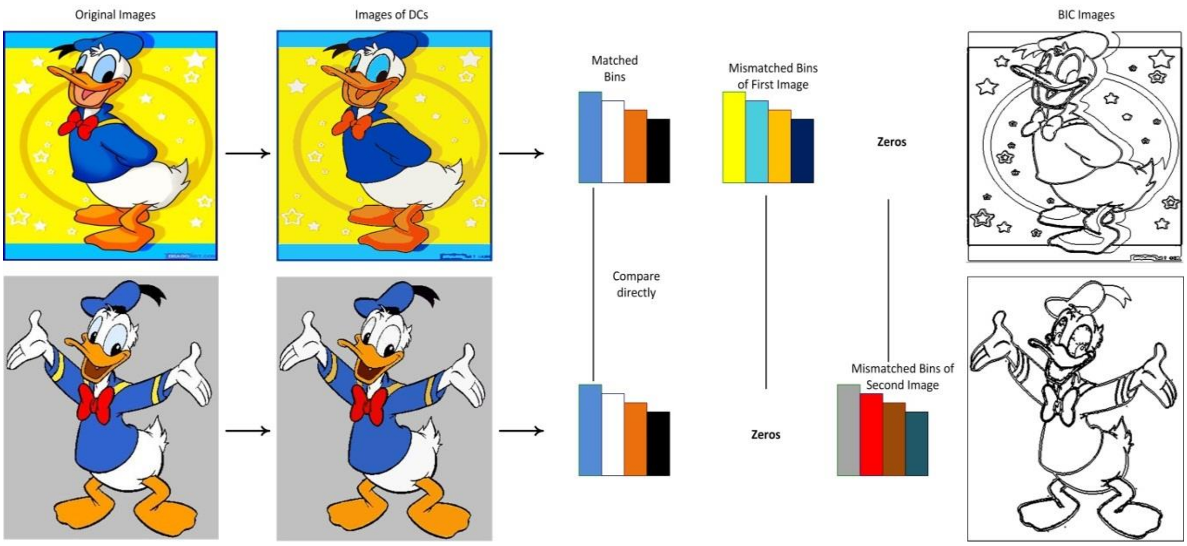


Figure 5.14: Similarity Measure for two images in the proposed DC-based BIC.

To compare between the original and proposed methods, three criteria can be considered. First criterion of comparison is the required storage space for each method. As mentioned before, original BIC requires 64-bytes for its 128-bins (half byte (4-bits) for each bin) while the proposed DC-based BIC requires 8-bytes only to store its 16-bins; in addition to 16-bytes for 8 RGB DCs values (each DC has 2-bytes because each band has 5-bits). The total bytes required for the proposed method is 24 (8+16) bytes, which mean less than the half size of original BIC. The second criterion of comparison is the required time for both methods. The average time needed for extracting 128-bin histogram is 1.5 seconds while the time for DC-based BIC is 3.5 seconds because it contains the time of extracting DCs. This extracted time is computed for tens of images with different resolutions to ensure generality and accuracy of measurement. The first two criteria are used to compare the efficiency of both methods. Third comparison criterion is an accuracy of proposed method compared to the original method. Accuracy of BIC and all its adapted versions is illustrated in Section 5.4.4. An extraction process of DC-based BIC histogram is depicted in Algorithm 5.7.

Algorithm 5.7: DC-based BIC

Input

DCImg: DC Image of the original image

DCs: Dominant Colours Values of the image

N: number of DCs in the image

Output

DC-BICh: DC-based BIC Histogram of the image

BEGIN

1. Clear *DC-BIC* histogram values

   $\forall\, i \in 0..N * 2$ // N = 8 as maximum

   DC-BICh[i] ← 0

2. Find DC-BIC Histogram from DC Image
   - { pass on all colours in the DC Image }

     $\forall\, i \in 0..width(DC\ Image)$-1

     $\forall\, j \in 0..height(DC\ Image) - 1$

     - { determine the current pixel, it is Border or Interior pixel}

       if (pixel(i , j)≠ any one of its 4-neighbour pixels) *// border pixel*

       DC-BICh[index(pixel(i,j) in DCs)]← DC-BICh[index(pixel(i,j) in DCs)]+1;

       Else *// Interior pixel*

       DC-BICh[index(pixel(i,j) in DCs)+N]← DC-BICh[index(pixel(i,j) in DCs)+N]+1;

3. Return *(DC-BICh)*

END

The best colour-based methods have dominance problem of large percentage colours on image similarity. Therefore, to solve this problem, weighting of DCs is applied on DCD in Chapter 4 and then on DC-based *Correlogram* in Section 5.3.2. In the next section, the weighting of DCs method is applied on DC-based BIC to demonstrate two issues. The first one is proving that the large percentage colour dominance problem and second issue is generalizing the DCs weighting method.

5.5.3 Weighted DC-based BIC

DC-based BIC results can be enhanced by adding both weights to its DCs and MCR. The enhanced version is called as a Weighted DC-based BIC method (WDCBIC). In DC-based BIC, the DC-BIC histogram of an image is divided into two parts. The first one is for matched colours with second image and the second one is for mismatched colours. Weights of DCs are added to histogram before dividing the histogram as illustrated in Algorithm 5.8. Then a dividing process can be performed to compute the dissimilarity measure of the two compared images using duo-model as depicted in Eq. 5.14.

Algorithm 5.8: Weighted DC-based BIC

Input

DCBICh1,*DCBICh2*: DC-based BIC histograms of the two compared images

Warr1, *Warr2*: Final Weight Arrays of the two compared images computed by Table 4.1

N1,*N2*: number of DCs in the two Compared images

Output

WDCBICh1, WDCBICh2: Weighted DC-BIC histograms of the two compared images.

BEGIN

1. Find Weighted DC-BIC histograms for Two images
    - { Find Weighted DC-BIC histograms for first image }
      $\forall\, i \in 0..N1-1$
      {
      $WDCBICh1[i] = DCBICh1[i] * Warr1[i];$
      $WDCBICh1[i+N1] = DCBICh1[i+N1] * Warr1[i];$
      }
    - { Find Weighted BIC histograms for second image }

$\forall\, i \in 0..N2-1$

{

$WDCBICh2[i] = DCBICh1[i] * Warr2[i];$

$WDCBICh2[i+N2] = DCBICh1[i+N2] * Warr2[i];$

}

2. Return ($WDCBICh1,\ WDCBICh2$)

END.

As noted from Algorithm 5.7, the weight of certain DC is multiplied by two values, border and interior pixels values (*DC-BICh1[i]*, *DCBICh1[i+N]*), where the DC-BIC histogram contains two parts first *N* values for border pixels and second *N* values for interior pixels. *N* represents number of DCs in the image; *N*=8 as maximum in DC-based BIC whereas in original BIC is *N*=64.

According to Mutual Colour Ratio, it can be computed using Algorithm 5.6, then added to the dissimilarity measure (Eq. 5.14) to result Eq. 5.17 that is exactly similar to that of weighted DC-based *ColGrm* (Eq. 5.11).

$$DisSimilarity\ Measure_{WDC_{BIC}}(Q,I) = (\,P_{match}(Q,I) + P_{mismatch}(Q,I)) * (1 - MCR) \qquad (5.17)$$

5.5.4 Experimental Evaluation of BIC

Evaluation of the proposed DC-based BIC and its weighted version with the original BIC is conducted on Cartoon-11K dataset only, as it satisfies the scope of this research (colour object-based image retrieval), to show efficiency and accuracy of this DC-based descriptor on large databases.

Four quantitative metrics are utilized to measure the accuracy of the proposed descriptors with the original BIC descriptor. These metrics are ARR, ANMRR, P(10) and MAP that have been explained in Section 4.4.1.3.

Table 5.12: *ANMRR, ARR, P(10) and MAP values for all BIC versions on Cartoon-11K database with 158 queries.*

Accuracy Metrics	Original BIC	DC-based BIC	Weighted DC-based BIC
ANMRR	0.832	0.822	**0.808**
ARR	0.134	0.143	**0.154**
P(10)	0.41	0.42	**0.43**
MAP	0.112	0.115	**0.126**

Table 5.13: *Storage Space, Time and No. of histogram bins performance metrics for all BIC versions on Cartoon-11K database with 158 queries.*

Efficiency Metrics	Original BIC	DC-based BIC	Weighted DC-based BIC
Storage (Byte)	64	**24**	56
Time (s)	**1.5**	3.5	3.6
BIC Histogram Bins	**128**	16	16

In the results presented in Table 5.12 and 5.13, the outperformance of weighted DC-based BIC approach in notable in terms of accuracy of all four metrics. From other side, DC-based BIC is surpassed in number of bytes required to store each method. As mentioned previously, DC-based BIC requires saving DCs values in addition to BIC histogram whereas the weighted version of BIC requires saving weights of DCs in addition to DCs values and histogram. In the last metric, original BIC outperforms all of its enhanced versions. Improvement ratio of the proposed methods over original one is presented in Table 5.14.

Table 5.14: *Accuracy Improvement Ratio of proposed BIC methods over original BIC method.*

	DC-based BIC over BIC	Weighted DC-based BIC over DC-based BIC	Weighted DC-based BIC over original BIC
ANMRR	1.2%	1.7%	2.9%
ARR	6.3%	7.1%	13.4%
P(10)	2.4%	2.3%	4.7%
MAP	2.6%	8.7%	11.3%
Average	**3.13%**	**4.95%**	**8.08%**

Percentage of improvement of weighted DC-based BIC is not very significant. This is because BIC (like *Correlogram*), is an advanced colour descriptor and its accuracy is high compared with other simple colour descriptor (such as colour histogram or DCD) (Pedronette & Torres, 2012; Penatti et al., 2012). Therefore, BIC accuracy needs little efforts to be ideal, in terms of colour values, percentages and colour spatial relations. These little efforts represent weights of DCs (Weighted DC-based BIC). Although the additional time of extracting the dominant colours, there is significantly reduction in the time of the overall retrieval time because the similarity measure process, which perform on all database images (thousands of images), depends on the BIC histogram bins that is reduced significantly from 128 into 16 bins only, as shown in Table 5.13.

5.5.5 Generic Frameworks for Converting Large number of Colours into Few DCs and Weighting DCs

To generalize the methods of converting large number of colours into few dominant colours and weighting DC-based descriptors, designing generic framework is required to be a guide for future research. The first framework of "Converting large number of colours into few DCs" is depicted in the Figure 5.15. Weighting DC-based methods Framework is the second framework as shown in Figure 5.16.

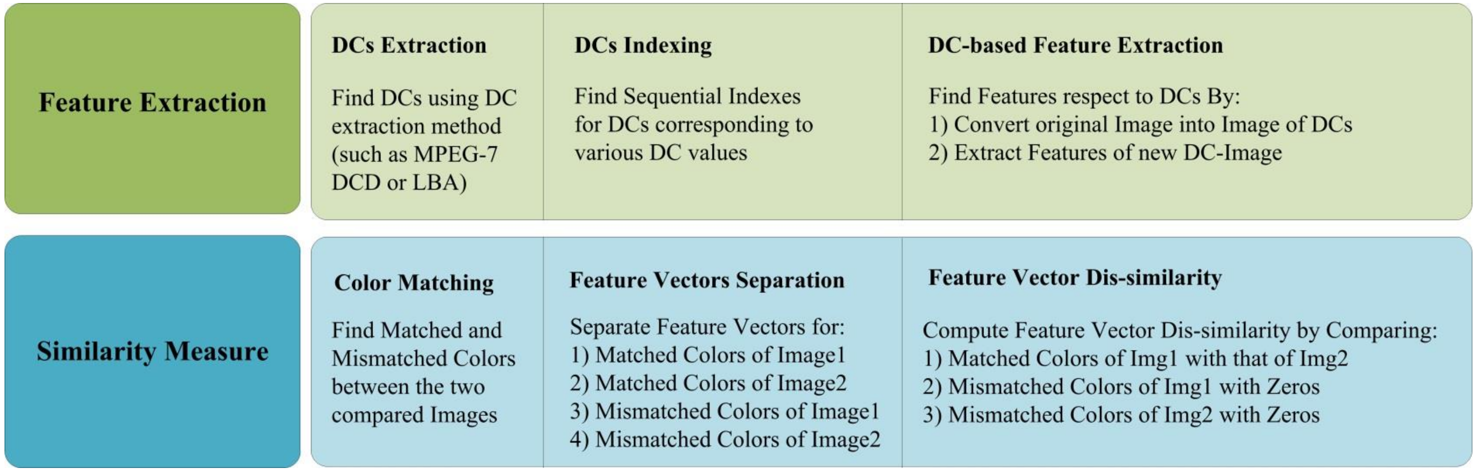


Figure 5.15: Large Number of Colours into Few DCs Conversion Generic Framework.

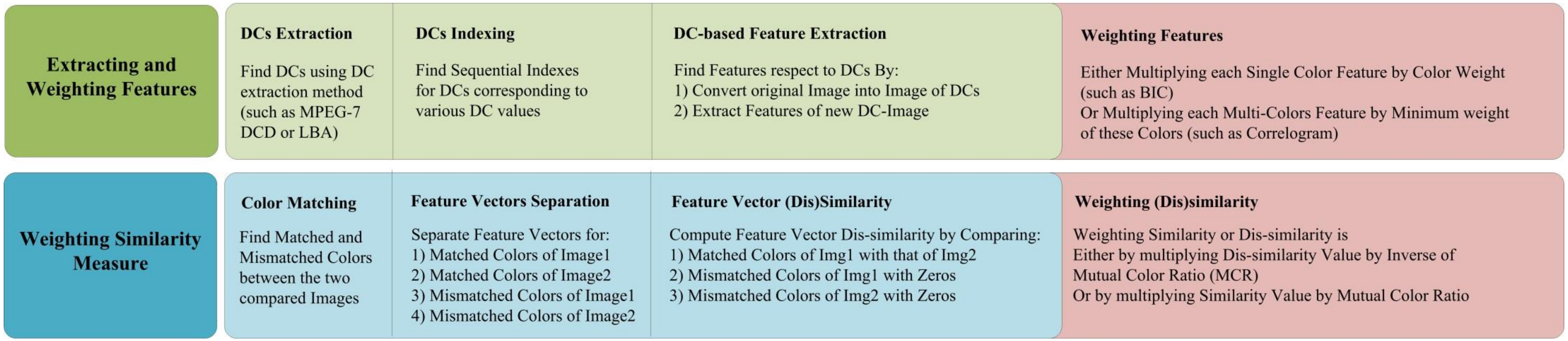


Figure 5.16: Weighting DC-based Descriptors Generic Framework.

There are shared components between the two frameworks. These components are:

1) DCs Extraction Method: MPEG-7 DCD, LBA DCD, or any other DC extraction method (such as colour naming (van de Weijer, Schmid, Verbeek & Larlus, 2009)) can be used to extract dominant colours from the image.

2) DCs indexing: this step aims to give sequence indexes (references) to arbitrary DCs that resulted from DCs extraction method. This process to organize DCs and prepare them for matching process. For example, the new indexes of DCs are 0, 1, 2, 3, …, 7 (if there are 8 DCs). These indexes refer to the original RGB DCs such as (12, 50, 125), (234, 76, 200) … (124, 67, 189).

3) DC-based Feature Extraction: this process uses DCs to extract features from an image instead of large number of quantized colours. This process can be performed after obtaining image of DCs by substituting each original pixel's colour with corresponding DC. All previous processes are included inside Feature extraction stage. Next processes will be located within similarity measure stage.

4) Colour Matching: this process contains matching between DCs of the two compared images. The results of colour matching are lists of matched and mismatched colours of the two images.

5) Feature Vectors Separation: this important process includes separation of feature vector (of each one of the two compared images) into two vectors;

first one is feature vector that contain features related to the matched colours and the other is for features that are related to the mismatched colours. This is because the dissimilarity measure formula that will be applied on each one is different.

6) Feature Vector Dissimilarity: This process computes dissimilarity value between two images by comparing feature vectors of matched colours directly (one-to-one) whereas comparing the feature vectors of mismatched colours with zeros. Actually, comparing features with zeros is to simulate the similarity measure of original descriptors.

All the previous processes represent framework of converting from large number of colours into few DCs descriptors. Additionally, all these processes also belong to weighting DC-based descriptors framework with additional two processes as follows:

1) Weighting Features: this process relates to feature extraction process where the feature can be weighted after extracted from the image. Weighting the feature depending on the weight of colour or colours that this feature extracted from. For example, in BIC histogram, each bin is corresponded to one DC in the image. Thus, the bin will be multiplied by weight of this DC. In more complicated descriptor, such as *Correlogram*, each feature of *Correlogram* table is associated with two DCs because it measures the probability of finding one of them beside the other in certain distance. In this

case, weighting this feature will be achieved by multiplying the feature by smallest weight of the two DCs that corresponds to this feature.

2) Weighting Dissimilarity: as mentioned in the weighting DCs original method that was described in Chapter 4, there are two solutions to address the problem related to large percentage dominance colour. The latter include feature level and similarity measure level-based solutions. Weighting features, which explained in the previous point, represents the feature level. The current point represents the similarity measure level. In this level, Mutual Colour Ratio (MCR) is proposed and it will be multiplied to the original value of dissimilarity. It is worth mentioning, if the similarity measure process computes similarity value between two images, it can be multiplied by MCR normally. While if the similarity measure process computes dissimilarity value, it will be multiplied by (1-MCR) to match effect of similarity measure process.

One of the important issues that must be mentioned here is the weighting process can work effectively only in specific image domain, which is same coloured object images database. This image domain contains many types of images including cartoon images, flag images, logo images and many natural images that have same coloured object such as images of tiger, zebra and others.

5.6 Summary

This chapter introduces four contributions of this research. Two of them are dedicated to solve the problem of this chapter which is the high complexity (in terms of memory space and computations time) of *Correlogram* descriptor, which is considered as one of the best colour descriptors. The first contribution is proposing a compact representation of colour *Correlogram*. Two steps of reduction are considered, first one is colour-based reduction and the second is distance-based reduction algorithm. The second contribution is proposing a new duo-model of similarity measure of an existing DC-based *Correlogram*. This model overcomes the drawbacks of the existing trio-model. The new proposed model simulates the original similarity measure of *Correlogram* descriptor. The other two contributions in this chapter are two generic frameworks. The first is to generalize the weighting DCs concept that is proposed in Chapter 4. The generalization of this concept is achieved by applying it on two further colour descriptors, which are *Correlogram* and BIC. The second framework is the conversion from large number of colours into few DCs, which is proposed in the second contribution of this chapter, DC-based *Correlogram*. Conversion into DCs concept is applied on BIC descriptor, in addition to *Correlogram*, to generalize it. Moreover, new Cartoon-11K dataset is introduced to test the proposed descriptors. The experimental results further show that the proposed descriptors offer significant reduction to the complexity, in terms of memory space and time, with higher accuracy.

CHAPTER SIX
DOMINANT COLOUR-BASED INDEXING METHODS

In this chapter, dominant colours of an image are indexed to avoid sequential search in a large database. This will speed up an image retrieval process in addition to improve the accuracy of colour-based descriptor because it narrows the search space. Images are indexed using 3-D RGB and perceptual LUV colour spaces. In the searching process, dominant colours in the query image are used independently to find images that contain similar colours in a reduced search space instead of the whole database search space. Therefore, query image is matched with images in this reduced space only to result the final retrievals. Two indexing methods are proposed in this chapter; first one is RGB colour space-based index structure while second one uses RGB space to build perceptual LUV colour space-based index structure.

6.1 Problem of Colour-based Indexing Methods

The first problem of colour-based indexing methods is high dimensional problem where it called "*curse of dimensionality*" problem. This problem occurred due to attempts to index large number of histogram colours where many dimensions reduction approaches are used. Decreasing number of colours in the image has been proposed to solve the aforementioned problem and many colour quantization techniques are used. Dominant colours were proposed as the most effective solution in this context where few colours are extracted to represent the image. Vector quantization methods are used to index images database using DCs but "*colour approximation"* problem emerged in these indexing methods. Details of colour-based problems are explained in Section 2.4.

Therefore, two colour indexing methods are proposed. The first one belongs to SP methods, where RGB colour space is divided into small partitions using uniform *Octree* colour quantization method (Gervautz & Purgathofer, 1990) combined with B+-tree method that will be used for representing colour percentages to filter irrelevant images in early stage. The second proposed method uses first uniform RGB colour space partitioning method to construct efficient indexing method for perceptual LUV colour space, which needs time consuming non-uniform vector quantization for partitioning this colour space. Characteristics of the two proposed methods are detailed in the next sections.

6.2 RGB Indexing Method

In large image databases, indexing is an urgent matter to reduce the search space for the retrieval process and in turn to speed up the process (Liu & Yang, 2013; Arslan, Yazıcı, Saçan, Toroslu & Acar, 2013). Most colour-based methods perform sequential search in their retrieval process; this will impose delay in the time of image retrieval process. In this section, DC-based indexing method is proposed to reduce search space for all colour-based methods (not only DC-based methods) to speed up retrieval process as well as preserve retrieval accuracy.

The proposed colour indexing technique is similar to multiple keywords searching, where if you have a statement of several words, the search of each word is performed separately to find matches of each word then the final result represents joining of these matches. In the proposed techniques, the image represents the statement and DCs of the image represent the keywords.

The proposed method is motivated by the question, which is, “where do we need to search exactly, to reduce search space rather than doing whole database search space?” The key answer is searching depends on a fixed range queries. In other words, searching on images that only have colours of distance less than or equal the maximum distance (that consider the two colours are similar) to the query image colours. As mentioned before, using tree-like indexing in the fixed range queries is ineffective (Deng et al., 2001; Samet, 1990). Therefore, fixed space partitioning method is used in this research. Before building database index structure, similarity between two colours must be considered and maximum distance between these colours also needs to be determined because the index structure will depend upon them. This will be detailed in the next section.

6.2.1 Maximum Distance between Similar Colours

As mentioned previously, the key of the proposed method is a similarity among colours within fixed range. That means, the searching can be done only in a specific range within distance, which is the maximum distance between two colours to consider them as similar colours. Therefore, below is an explanation of what is the maximum difference (distance) value (MxDV) that can be considered in the proposed approach to assume the two colours are similar.

In Eq. 6.1, Euclidian distance between two 3-D colours i and j is depicted to determine the maximum distance between two colours by comparing it with certain threshold value Th_d. This threshold value was assumed to be 10, 20 or 25 by (Yamada et al., 2001) and (Yang et al., 2008). From the maximum threshold value

(25), maximum difference value (minimum difference is zero) between the two colours (each colour has three channels Red, Green and Blue in RGB colour space) can be computed to consider it in the proposed colour indexing method.

An example to show how to find the difference between two colours values is depicted as follows:

$$d_{i,j} = \sqrt{(C_i^R - C_j^R)^2 + (C_i^G - C_j^G)^2 + (C_i^B - C_j^B)^2} \qquad (6.1)$$

by substitution $d_{i,j}$ with Th_d (maximum = 25)

$$25 = \sqrt{\left(C_i^R - C_j^R\right)^2 + \left(C_i^G - C_j^G\right)^2 + \left(C_i^B - C_j^B\right)^2}$$

by squaring both sides

$$25^2 = (C_i^R - C_j^R)^2 + (C_i^G - C_j^G)^2 + (C_i^B - C_j^B)^2$$

$$625 = (C_i^R - C_j^R)^2 + (C_i^G - C_j^G)^2 + (C_i^B - C_j^B)^2$$

the difference of each channel can be written as one variable as follows:

$$Diff\_R = \left(C_i^R - C_j^R\right), \qquad Diff\_G = \left(C_i^G - C_j^G\right), \qquad Diff\_B = \left(C_i^B - C_j^B\right)$$

The formula will be as below:

$$625 = {Diff_R}^2 + {Diff_G}^2 + {Diff_B}^2$$

Here, two assumptions can be assumed: ***first one*** is assuming that all differences are equal

$$Diff\_R = \quad Diff\_G = Diff\_B = Diff$$

$$625 = Diff^2 + Diff^2 + Diff^2$$

$$625 = 3Diff^2$$

$$Diff = \sqrt{(625/3)} = \sqrt{208} \approx \mathbf{14}$$

From the first assumption, maximum difference value of each channel is obtained; it is 14 to consider colours are similar.

The second assumption is assuming the differences of two channels are zeros and one channel only has a difference, as follows:

$$625 = Diff^2 + 0^2 + 0^2$$

$$Diff = \sqrt{625} = \mathbf{25}$$

From this assumption, a maximum difference value of any RGB colour channel is 25. Therefore, the MxDV of any colour channel in the two assumptions are 14 and 25 to produce Euclidian distance equal to 25. Actually, these difference values are virtual, experiments on certain database must be conducted to know, what is the best MxDv that can be used to build an efficient indexing structure in colour-based CBIR for certain database? These experiments are conducted on the new Cartoon-11K database and Corel-10K to extract the best settings to these database.

6.2.2 Indexing Structure

In DC-based methods (MPEG-7 DCD for example), dominant colours are extracted using dynamic quantization method (GLA clustering as example) and most likely the image is quantized to 5-bits colours. To find actual colours of an image, the conversions of these quantized colours to 8-bits colours (their origin) can be done by multiplying them by 8 (or shift the value by 3-bits to the left). An example that illustrates this process is shown in Figure 6.1 and Table 6.1.

Figure 6.1: An example of colour quantization of DCD.

From Figure 6.1, the two images have two different object colours (white and grey). The difference of original colours is 255-210= 45 in each band. After colour quantization process, the difference of resulted colours becomes 31-26= 5 in each band (31 in binary numbering system equal to 11111 (5-bits)). Clearly, this difference cannot represent the actual difference value between the White and Grey colours. The actual difference can be obtained after multiplying the colour value by 8. This results in a difference which equals to 248-208= 40. This figure represents the actual difference between the colours with slight changes due to quantization process.

Table 6.1: *An illustration of colour quantization method that applied in Figure 6.4.*

Image	Original Colours Values	Quantized Colours Values (5-bits)	Quantized Colours*8
Image 1	**White** (255,255,255)	(31,31,31)	(248,248,248)
	Black (0, 0, 0)	(0, 0, 0)	(0, 0, 0)
Image 2	**Gray** (210,210,210)	(26,26,26)	(208,208,208)
	Black (0, 0, 0)	(0, 0, 0)	(0, 0, 0)

From Figure 6.2, one can note that the values of the first three bits (bit0, bit1 and bit2) of each colour channel after quantization and multiplying them by 8 are zeros. That means, the similarity of colours will depend on the remaining 5 bits (bit3, bit4, bit5, bit6 and bit7). Since the maximum difference between two channels is 25 (as mentioned previously), then the changing of the two bits (bit3 and bit4) is within this range. This is because the weights of these bits are 8 and 16 respectively; their summation is 24 and that approximately equals to MxDV (25) of two colours. In this regard, bit7, bit6 and bit5 are out of tolerance range of colours to be similar.

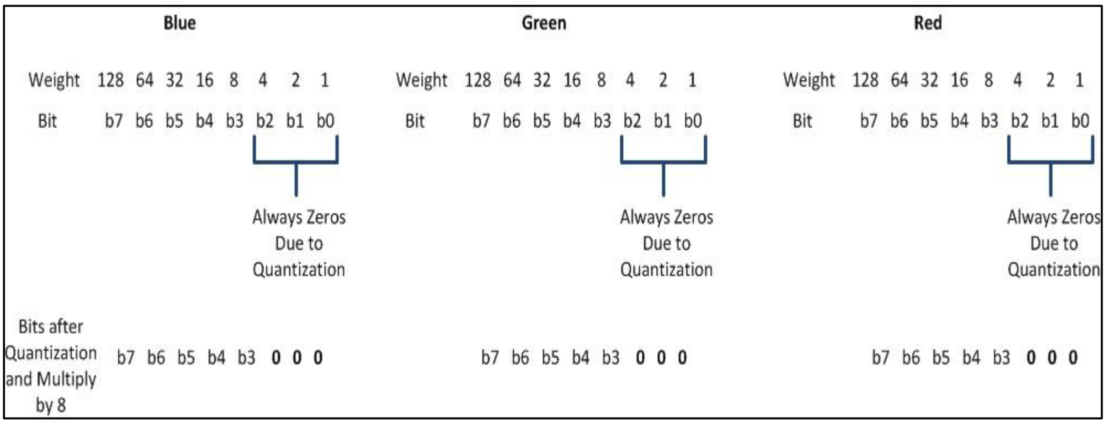


Figure 6.2: Weights and values of bits for three colour channels (R, G, B) after quantization.

Thus, all these three bits will be the first and main level of colour similarity; they will be used to differentiate among dissimilar colours. In other words, if the compared colours are different in these 3 bits, then these colours definitely will be not similar. The other two bits (bit3 and bit4 of each channel) can be used separately to be the second and third level respectively of colour similarity as shown in Figure 6.3. Hence, first indexing dimension contains 512 cell (3-bits from each channel=9-bits, number of cells=2^9=512). The second indexing dimension contains 8 cells (1-bit from each channel=3-bits, 2^3=8). The third dimension has 8 cells that represent the remaining one bit (bit3). An example of how to put certain colour into indexing structure is depicted in Figure 6.4. Overall index structure is presented in Figure 6.5.

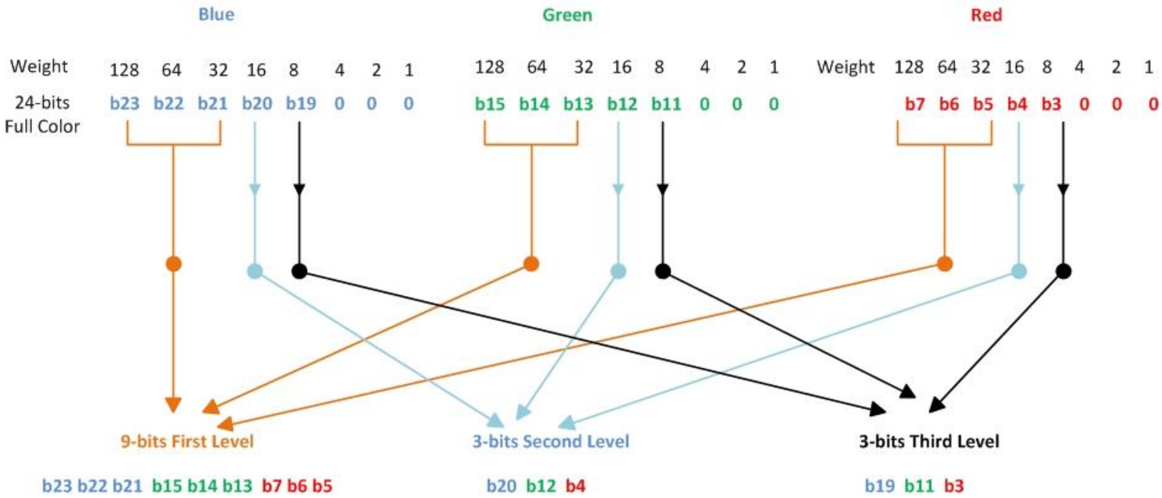


Figure 6.3: Building index structure from colour channels.

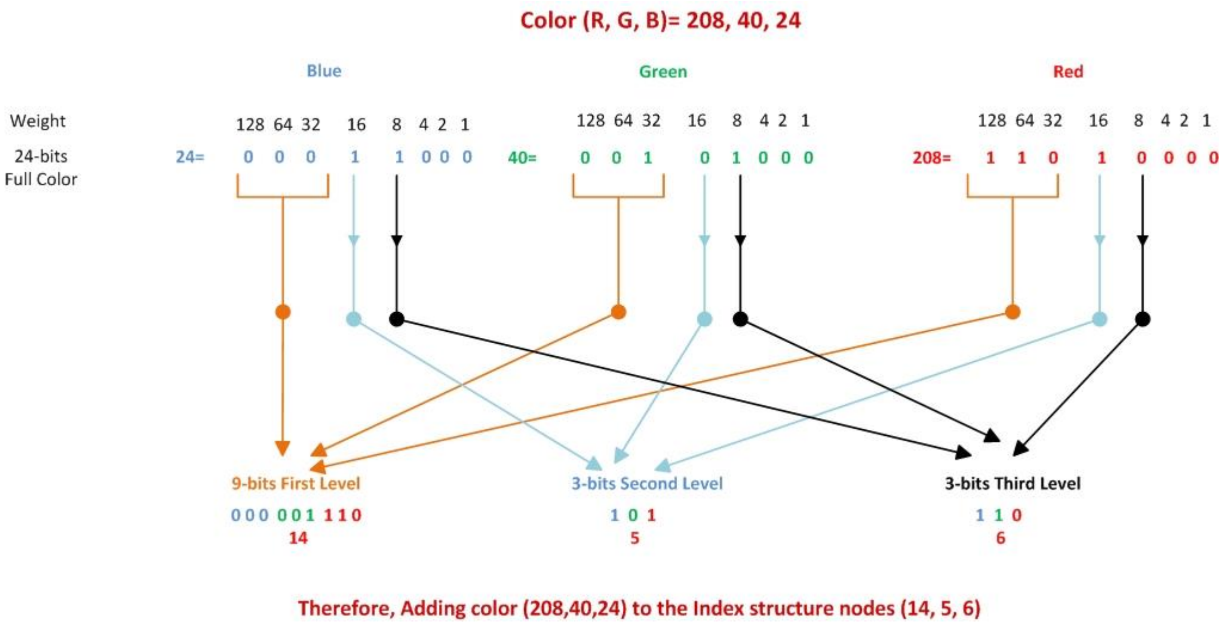


Figure 6.4: Obtaining indexing values from colour channels values.

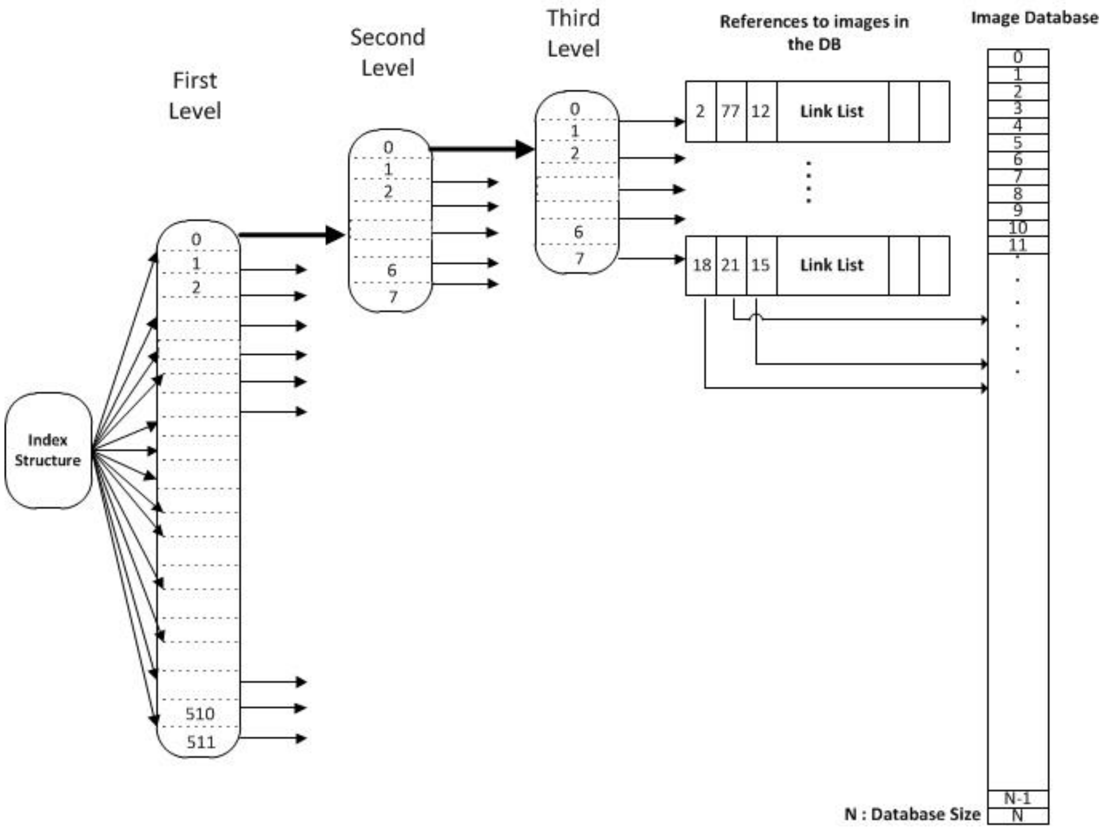


Figure 6.5: Structure of proposed RGB Indexing Method

Building the database index structure can be expressed using the following algorithms:

Algorithm 6.1: Build Index Structure for Database

Input

DB Images: List of all Images in the database

Output

Index Structure: Fixed size Index structure, its definition is Array [512, 8, 8] of dynamic list.

BEGIN

1. Set all Index Structure Cells to Empty list

$$\forall\, i = 0..511, \qquad j = 0..7, \qquad k = 0..7$$

$$Index\ Structure(i, j, k) \leftarrow empty\ list\ (null)$$

2. Build Index Structure for all Images in database

- $\{Pass\ by\ all\ images\ in\ Database\}$

  $$\forall\, Image\ \in DBImages$$

  - $\{\ Pass\ by\ all\ domiant\ colors\ in\ the\ image\}$

    $$\forall\, Color\ \in Image's\ DCs$$

    - { Extract three levels values (location) of Index Structure from colour }

      $$Call\ Extract\_Index\_Dimensions(Color, \boldsymbol{out}\ Dim1, \boldsymbol{out}\ Dim2, \boldsymbol{out}\ Dim3)$$

    - { Add Image Reference to the List of Index Cell that identified by extracted dimensions }

      $$IndexStructure(Dim1, Dim2, Dim3).Add\ (Image\ Ref)$$

3. Return $(Index\ Structure)$

END.

Algorithm 6.2: Extract_Index_Dimensions

Input

Colour: Colour Value to extract its index location (dimensions)

Output

Dim1, *Dim2*, *Dim3*: Location in the Index Structure to put the input Colour

Meaning of Symbols

(&) means logical AND, ($\gg$) is Shift logical Right, ($\ll$) means Shift logical Left, *(H)* means hexadecimal value, *(mod)* means modulus or remainder.

BEGIN

1. Extract three channels (R,G,B) from Colour Value

$$R \leftarrow Color\ mod\ 256 \quad \boldsymbol{or}\ \ R \leftarrow Color\ \&\ 0000FFH$$

$$G \leftarrow \left(\frac{Color}{256}\right) mod\ 256\ \ \boldsymbol{or}\ \ G \leftarrow Color\ \&\ 00FF00H$$

$$B \leftarrow \left(\frac{Color}{256^2}\right) mod\ 256\ \ \boldsymbol{or}\ \ B \leftarrow Color\ \&\ FF0000H$$

2. Find First Index Level value from RGB values

$$Dim1 \leftarrow (R\ \&\ E0H) \gg 5 + (G\ \&\ E0H) \gg 2 + (B\ \&\ E0H) \ll 1$$

3. Find Second Index Level value from RGB values

$$Dim2 \leftarrow (R\ \&\ 10H) \gg 4 + (G\ \&\ 10H) \gg 3 + (B\ \&\ 10H) \ll 2$$

4. Find Third Index Level value from RGB values

$$Dim3 \leftarrow (R\ \&\ 08H) \gg 3 + (G\ \&\ 08H) \gg 2 + (B\ \&\ 08H) \ll 1$$

5. Return $(Dim1, Dim2, Dim3)$

END.

From algorithms 6.1 and 6.2, one can notice that the index structure contains three levels. The first one represents the three most significant bits (bit7, bit6 and bit5) of the colour bands (channels). The weights of these bits are too high whereas their weights are 128, 64 and 32. They are out of the tolerance value of colour similarity difference which is 25. Thus, separating these bits from the other two bits is necessary to differentiate between similar and not similar colours because if these bits are different in the two compared colours, then there is no chance of these colours to be similar. The other 2-bits (bit4 and bit3) however, have weights to allow them to be within tolerance range of similar colours; their weights are 16 and 8 respectively. To see the effect of each one of these two bits on colour similarity and in turn on retrieval accuracy, separating these two bits into two index levels is performed. Obviously, their effect depends on the variety of colours the database. Structure of indexing method is presented in Figure 6.5.

In the proposed indexing method, all or some DCs of the database images are taken in consideration to see what is the effect of excluding some DCs from indexing structure?. This effect will be shown when the experiments on different number of colours (8, 5, 3 and 1 DC) are conducted. Building an index structure can be summarized in some steps as follows: First, compute three index levels values of each DC to find the cell or list that contains the references to all images in the database that have same colour. Then add reference of the image into this list. For example, if there are 8 DCs in an image, then reference of this image will be added 8 times to 8 lists; each list represents one of image DCs. Later in the searching process, this will allow matching with images that have same or similar colours only instead of all database images. Moreover, the proposed indexing method is dynamic, which means the insertion and deletion operations are straightforward; i.e.: without needing to reconstruct the index structure again when adding new images or removing undesirable images from database. Adding a new image to the database can be achieved by computing 3 levels values and adding reference of the new image to the 8 lists (or less) depending on the DCs of this image. Whereas removing an image from index structure is performed by only removing references of this image from 8 lists that represent DCs of the image. Therefore, index structure of these cases is untouched.

Colour percentage plays an important role in similarity measure of certain colour with its corresponding colour in other images where the similar colours are considered as dissimilar if their percentages have large difference such as similarity measure of different colour descriptors such as MPEG-7 DCD, LBA, BIC and *Correlogram*. Therefore, filtering images (that have large difference in percentage) out in the early stage helps in reducing search space and in turn speeds up the retrieval process. In Deng (2001), filtering process is performed online during query processing. Filtering out the dissimilar images in terms of colours is achieved firstly; then percentages of colours are matched through pass by all images sequentially to perform second level of filtering. This online filtering is impractical because it is time consuming even though Deng (2001) claims that this process is less expensive than image similarity measure process in its computations. Actually, this online process is time consuming process in large database; thus, achieving it offline is mandatory in large sized database. Therefore, the proposed RGB indexing structure is extended to include partitions of colour percentages. Single level B+-tree is used to represent colour percentages. The unique node of B+-tree contains three entries (four pointers) as depicted in Fig 6.6. B+-tree can be added to all leaf nodes (in the third index level), hence the index structure can be viewed as shown in Figure 6.7.

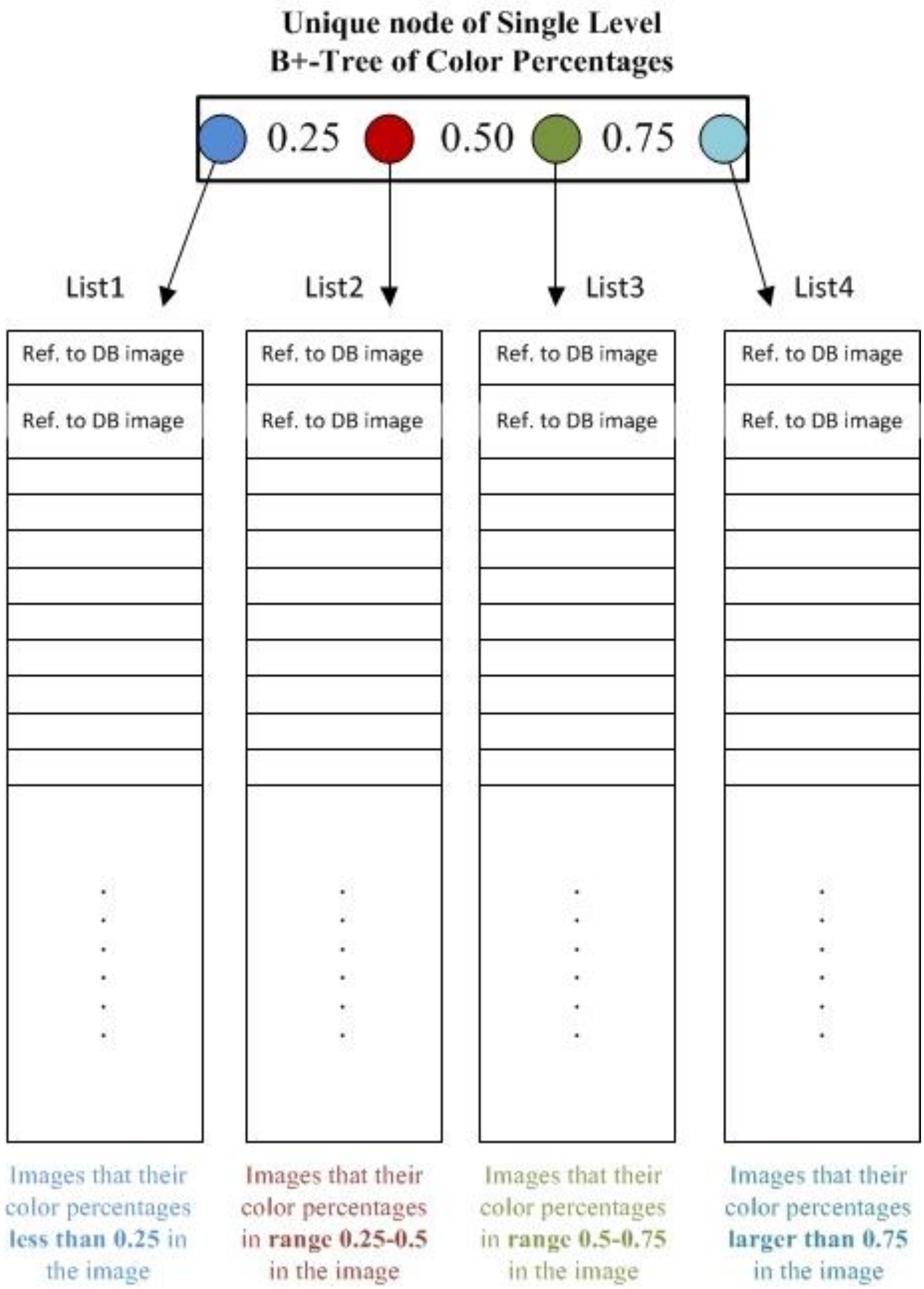


Figure 6.6: Single Level B+-Tree for representing Colour Percentages

From Figure 6.6, images of colour percentage less than 0.25 will be inserted in List1 whereas the images of colour percentages within range 0.25 to 0.5 will be inserted in List2 and so on. This representation gives flexibility of representing colour percentages where the whole range (of colour percentages) is divided into four parts; this maintains on reasonable increase of index structure size as well as enough difference between colour percentages to differentiate between various contents. The index structure now is four times size larger than before but with significant reduction of search space, as will be explained in experiments results.

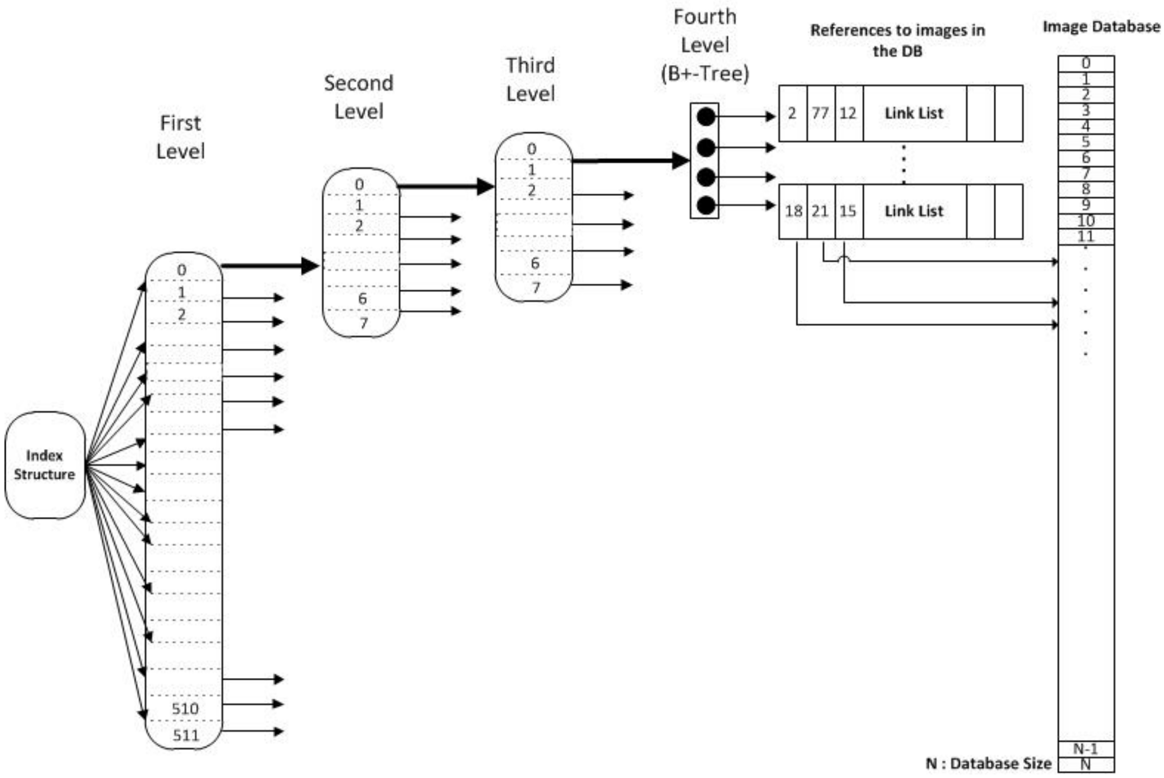


Figure 6.7: Indexing Structure with B+-Tree to represent Colour Percentages

6.2.3 Searching Process

In this process, query is required to find its similar images in the database. Searching process includes the following steps:

1. For each DC in the query image, find database images that are similar in both colours values and colours percentages; this is by reaching to suitable node(s) in index structure and in turn to the database images that are associated with this node(s).

a. Reaching to the nodes in the third level of index structure, which are similar to the query colour, is considered as the first level of similarity (it is called colour-based similarity). In this level, all false matched images that do not contain colours similar to the query will be eliminated.

b. Second level of similarity is a percentage-based similarity. Each reached node in point (a) has B+-tree structure of colour percentages. Two paths of B+-tree that are nearest to the query colour percentage will be selected to obtain the candidate images for comparison. For example, if query colour percentage is 0.22, this mean it belong to the List 1 (of range from 0 to 0.25) of B+-tree. In this work, List1 and nearest list to it (List2) are selected to obtain candidate images for comparison. In Algorithm 6.3, B+-tree searching method is detailed.

2. Merging images references that resulted from each DC of the query image to produce search space of the query, which is called as reduced search space (RSS).

3. Calculate dissimilarity distance between query and all images in the RSS and then rank them accordingly.

In step 1, most false match images will be removed according to different colour tolerance value and colour percentages. Three colour tolerance values are used according to the maximum distance value that is extracted from Section 6.2.1. These colour tolerance values (CTV) are 0, 8 and 24 regarding to 2-bits (bit3 and bit4) of colour channels. If CTV=0 (no tolerance in colour difference), that means the images of same colour only are allowed to compare with the query image. Only one node in the third index level for each DC will be nominated to the step 1 (b) as shown in Figure 6.8.

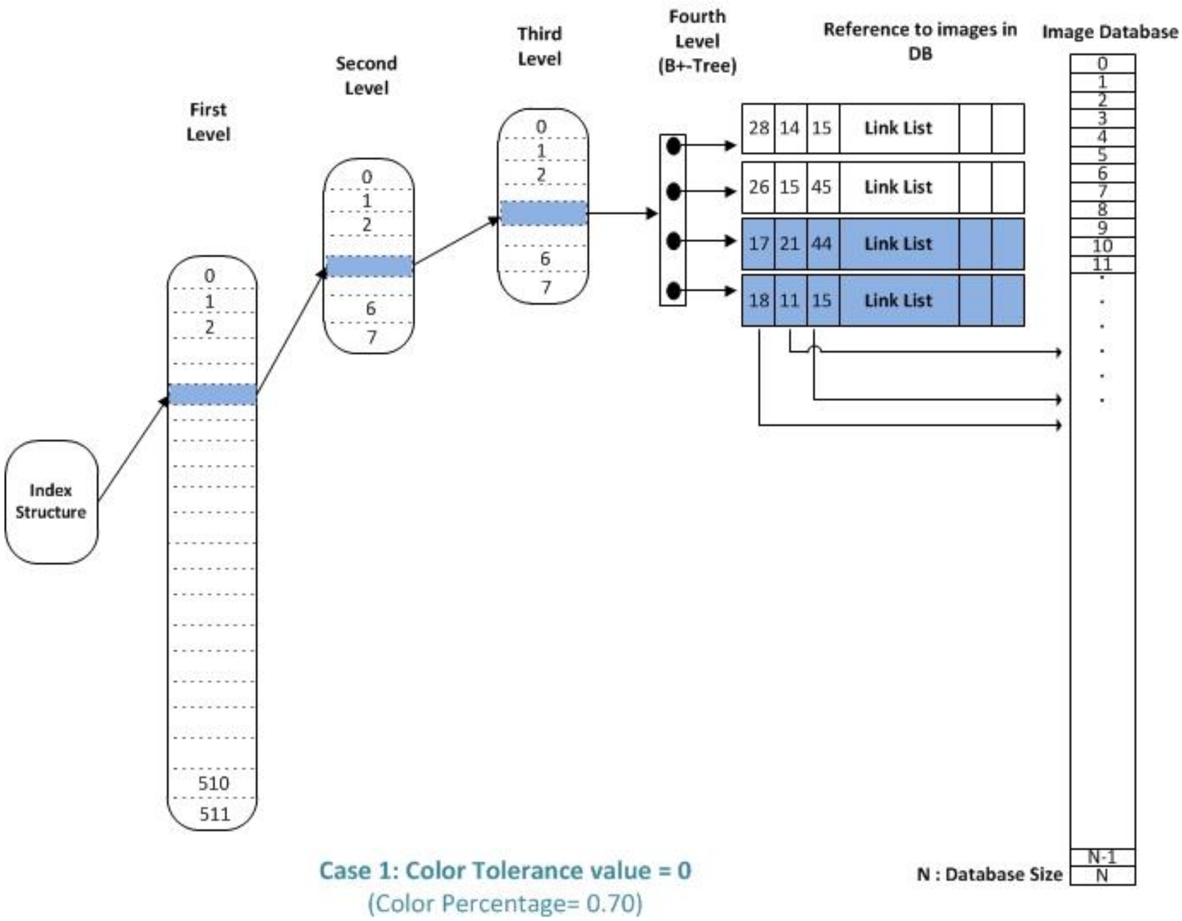


Figure 6.8: Image retrieval process when colour tolerance value = 0.

In step 1 (b), the colour percentage will be checked as second level of similarity. The three threshold values of the B+-tree are fixed in this research, which are 0.25, 0.50 and 0.75 (as depicted in Figure 6.6). Four lists are used to keep the images of different colour percentages. In retrieval process, two of these lists will be visited instead of one that is similar to query colour percentages. For example as in Figure 6.11, if the query colour percentage is 0.70; hence, the appropriate list for this percentage value is in List 3 because it contains all colours that have percentage within range from 0.5 to 0.75. In order to give more flexibility to the proposed method, List 3 as well as another neighbouring list will be selected. The selection of this another list will depend on the colour percentage value whether it is near to List 2 or List 4. In the case of 0.70 percentage value, it is near to the List 4 than List 2; hence, List 3 and List 4 are selected to obtain images that will be matched with the query in steps 2 and 3. Selection of two lists instead of one gives the retrieval process some flexibility to match images that have some scaling operations (enlargement or reduction in size) with its original size. Additionally, this flexibility is necessary for non-natural images (including cartoon images) because the cartoon producer is free to display cartoon characters in different sizes whereas each character has at most the same colours and shape. Algorithm of selecting the lists of B+-tree is explained in Algorithm 6.3.

Algorithm 6.3: Find Percentages Lists

Input

Percentage: Colour Percentage Value to find its suitable percentages lists.

Output

Lists: Lists of names for suitable percentages lists that corresponding to given percentage.

BEGIN

1. Set Lists of percentage lists names to empty list

   $$Lists \leftarrow Empty\ List\ (null)$$

2. Find First suitable list for Colour Percentage

   Select Case (*Percentage*)

   Case $Percentage \leq 0.25$: $Lists.add$ (List1)

   Case $0.25 < Percentage \leq 0.5$: $Lists.add$ (List2)

   Case $0.5 < Percentage \leq 0.75$: $Lists.add$ (List3)

   Case $Percentage > 0.75$: $Lists.add$ (List4)

3. Find Second suitable list for Colour Percentage

   Select Case (*Percentage*)

   Case $Percentage \leq 0.25$: $Lists.add$ (List2)

   Case $0.25 < Percentage \leq 0.5$:

   If (*Percentage* <= 0.375) $Lists.add$ (List1)

   Else $Lists.add$ (List3)

   Case $0.5 < Percentage \leq 0.75$:

   If (*Percentage* <= 0.625) $Lists.add$ (List2)

   Else $Lists.add$ (List4)

   Case $Percentage > 0.75$: $Lists.add$ (List3)

4. Return ($Lists$)

END.

In CTV=8, 8 nodes in the third indexing structure will be nominated to step 1 (b) as depicted in Figure 6.9. Tolerance value 8 represents the weight of bit3 in each colour channel. In other words, bit3 of the retrieved images is free to be equaled or not (tolerance) to the query while bit4 must be equal like the other 3-bits (bit5, bit6 and bit7) that must be equaled in all cases; this will allow tolerance in colour value by 8.

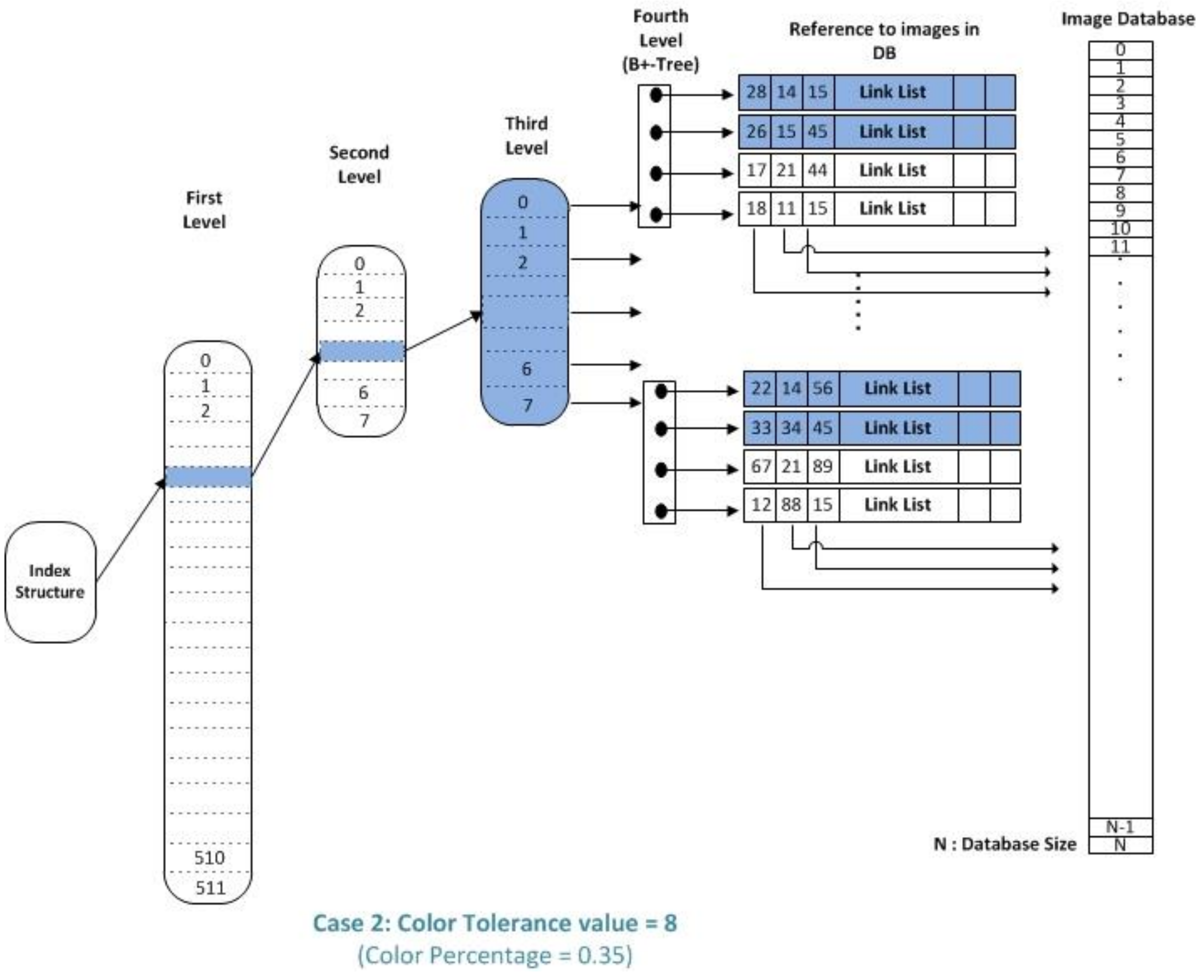


Figure 6.9: Image retrieval process when colour tolerance value = 8.

Last case is when CTV=24, 64 nodes in the third level of indexing structure (8 nodes in the second level) will participate in step 1 (b). This is because; the two bits (bit3 and bit4) are allowed to be not matched (tolerance). Two bits for 3 channels (R, G, B) produce 6-bits and 2^6=64 possibilities as shown in Figure 6.10.

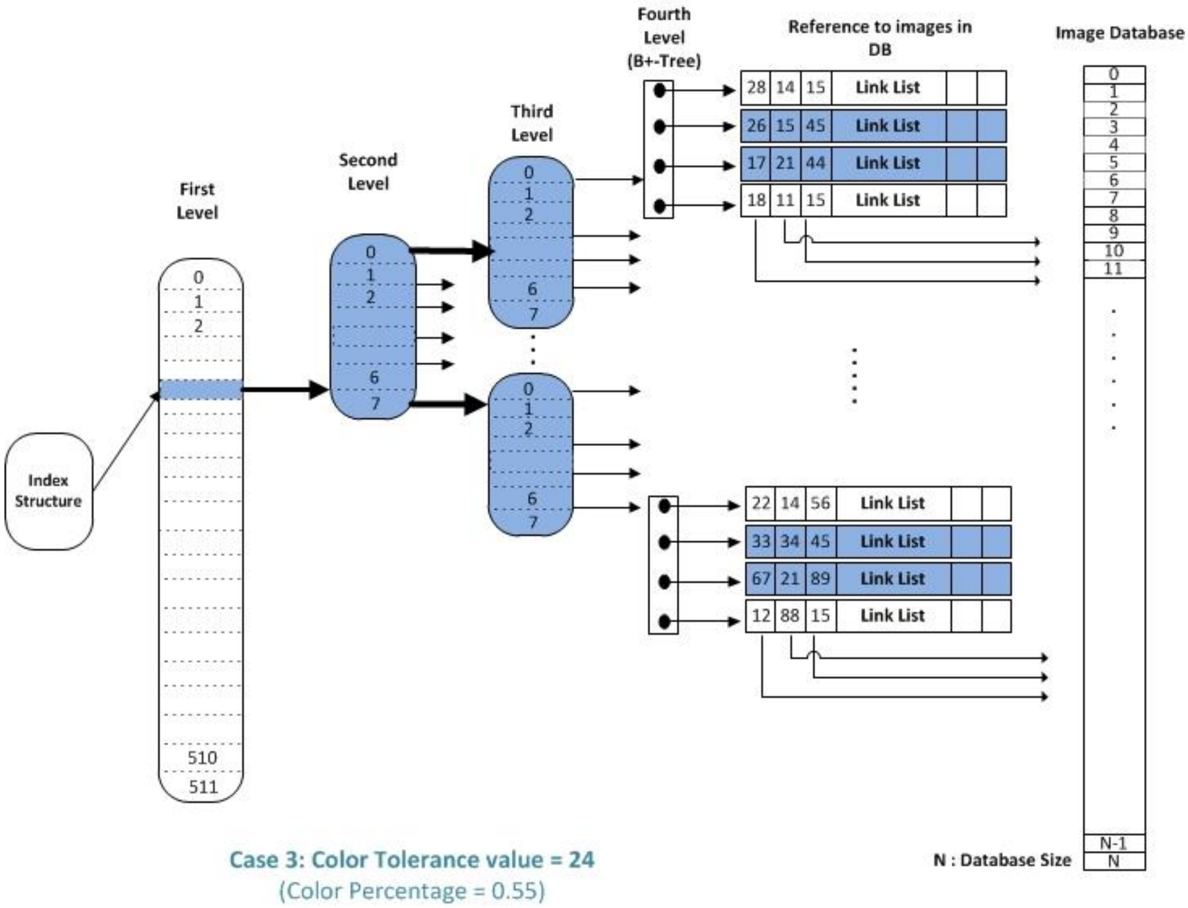


Figure 6.10: Image retrieval process when colour tolerance value = 24.

The difference of the proposed indexing scheme over space partitioning methods such as *kd-tree* structure lies in different aspects:

1) The fixed-size representation (array) is faster than dynamic-size representation (tree) of *kd-tree* in spite of some array locations could be left empty due to unavailability of some colours in the space. However, the memory space of fixed-size array is still smaller than dynamic representation that uses many memory references (pointers) to keep track the next nodes in the tree.

2) The new design of the proposed RGB method exploits the range query in building the index structure (3-levels structure), instead of changing query parameters to perform range query as in *kd-tree* that have 8-levels structure to represent RGB colour space. This new structure speeds up the search mechanism.

3) Embedding B+-tree representation in the last level (leaves nodes) of the proposed representation makes the search result more accurate and fast. This is because the colour percentage is used as fourth level of filtering to exclude the images that have different colour percentage than the query's colours percentages.

6.2.4 Experimental Evaluation

In the experiments evaluation, some parameters need to be set; this will be presented in first sub section. In the second sub section, indexing-based retrieval results will be detailed.

A. Experimental Setup

Parameters that need to be set in indexing method are number of indexed colours, performance metrics that can be used to measure the accuracy and efficiency of the proposed indexing method and databases that can be used for experiments. Lastly, the current indexing methods will be used for comparison to the proposed methods. Illustration of these parameters is presented in the next sub sections.

ii. Number of Indexed Colours

Indexing of images database can be performed with 8, 5, 3 or 1 different colours, to measure effect of each one on the retrieval performance. The database is indexed firstly according to maximum DCs in the image that equal 8; then reduce the number of indexed DCs into 5, 3 and 1. It is noteworthy to mention here that DCs will be sorted in descending order according to their colour percentage before the indexing process. The reason behind this is indexing the largest and important colours and ignores the others. For example, when using 5 colours only in indexing, the largest 5 percentage colours are indexed and the remaining 3 smallest percentage colours are neglected. The idea of not selecting all colours (8 colours) for indexing (after sorting them in descending order depending on their percentages) is the query and the retrieved images need not to be exactly have the same colours as long as the largest dominant colours of the two images are matched.

iii. Performance Metrics

Two types of metrics are used in this chapter. First is the efficiency (speed) of the image retrieval using proposed indexed structure compared with sequential searching retrieval for the whole database. Additionally, comparison will be made also with famous methods of vector quantization scheme. The second is the accuracy of image retrieval after reducing search space.

a. Efficiency Metrics

The main goal of indexing is to reduce search time compared to sequential search by reducing database images that will be matched with the query image. It eliminates unlikely (irrelevant) images from the matching process. It keeps desired (relevant) images only to be compared with the query. This produces what is known as Reduced Search Space (RSS) while the term WSS is denoted to the Whole Search Space hereinafter.

The reduced time needed for searching in the RSS can be computed by the percentage $\frac{RSS}{WSS}$%. The percentage $\frac{RSS}{WSS}$% can be called as Search Space Ratio (SSR) that represents ratio of images that are actually searched to the all images in the database (WSS) (Alexandrov et al., 1995). Whenever this ratio is small, the search is fast. For example, if SSR=25% then the search will be quarter the whole search space (i.e.: the indexing search is four times faster than sequential search). Therefore, SSR can be used to measure speed of the indexed search compared with sequential search. Nevertheless, SSR alone is not enough to determine speed of the retrieval process because the indexing process also introduces some overhead. Increasing this overhead will degrade performance of indexing method. Overhead of the proposed method is not significantly noticed during the search process. This is because it is fixed and only 1 node (in the third index structure level) will be checked when the colour tolerance value is equal 0. Likewise, 8 nodes will be checked when tolerance value is 8. Maximum number of nodes that will be checked is 64 nodes when CTV equals 24.

For more explanation, each node in the third level of index structure represents one colour in the quantized colour space (5-bits only, 3 RGB channels, no. of colours $=2^{15}$ = 32768). In the fourth level of indexing structure (B+-tree structure), there are four lists of images, which represent images that contain certain colour (corresponding to colour in the node of third level). The number of images in these lists depends on the existence of this colour in the database. Two lists of the four will be selected to be compared with the query image in the retrieval process. The number of images in these lists is different depending on the percentage of this colour in the images. Therefore, an overhead of indexing structure is computed for the two datasets with various queries and it is averaged in Table 6.2.

Table 6.2: *An Average of Overhead Ratio of the proposed indexing method in two different datasets.*

Colour Tolerance Value	Corel-10K	Cartoon-11K
24	0.8800%	0.9000%
8	0.0800%	0.0750%
0	0.0029%	0.0035%

**An Average of Overhead Ratio of the proposed indexing method in two different datasets with maximum indexing colours (8-colours) with different Colour Tolerance values.*

As shown from Table 6.2, experiments on two large datasets, in terms of colours, are conducted to find an average overhead of the proposed indexing method. To add overhead to SSR $(\frac{RSS}{WSS})$, the percentage of this overhead based on the sequential search time of whole dataset (WSS) must be computed. Therefore, an Overhead Ratio (OHR) can be computed as (Alexandrov et al., 1995):

$$OHR = \frac{Overhead\ Time}{WSS\ Time}$$

where this ratio can be added to SSR to find total ratio (reduced time) of the proposed indexing method compared to the sequential time. The overhead resulting from the maximum number of indexed colour (8 colours) is tabulated in Table 6.2. In addition, different colour tolerance values are taken into consideration when computing this ratio. Maximum computed ratio was 0.9 % which consider as low overhead compared with other indexing scheme.

b. Accuracy Metrics

In the second type of metrics, three quantitative performance metrics will be utilized to measure the accuracy of different colour descriptors when are used in the proposed indexing method as well as other competing indexing methods. These metrics are ARR, ANMRR and P(10) that were illustrated previously in Section 4.4.1.3. MAP is one of the important metrics that is used in this research because its effectiveness and popularity in CBIR field. MAP accuracy metric is excluded in this contribution because it measures the accuracy of retrieval system depending on all relevant images (in the database) to the query instead of depending of only relevant retrieved images. In indexing methods, not all relevant images can be retrieved because the reduced space may neglect some images that are not very similar to the query. Thus, computing MAP in this situation is unfair because inequality of number of relevant images will result different accuracy values even if the ranks of the most relevant retrieved images are identical. Whereas other metrics depend on relevant retrieved images instead of all relevant images such as P(10), ARR and ANMRR. Figure 6.11 shows the difference between MAP and other metrics in sequential and indexed methods.

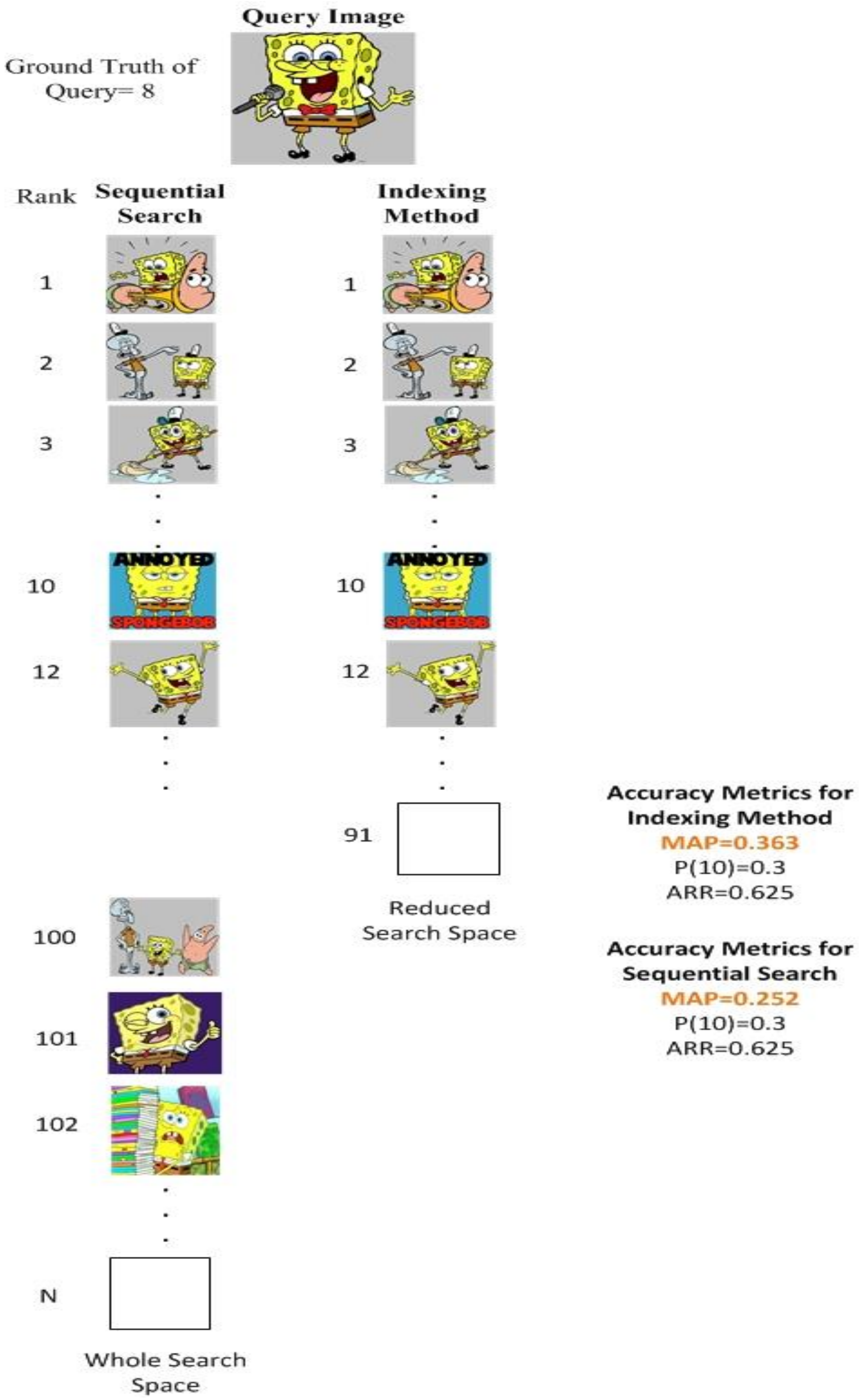


Figure 6.11: An example of instability of MAP Accuracy Metric in Indexing Method because its dependence on all relevant images instead of relevant retrieved images.

iv. Evaluation Datasets and Competing Indexing Methods

Evaluating the proposed indexing techniques will be conducted on two datasets: Cartoon-11K and Corel-10K. These databases are already illustrated in Section 5.4.1 (A). Selection of these datasets is necessary to show effectiveness of the proposed index structure on these large databases.

Indexing methods that have been selected to compete with the proposed methods are sequential search, K-means (KM) (Hartigan & Wong, 1979; Maimon & Rokach, 2005) and recent K-means with B+-tree methods (KMB) (Yildizer et al., 2012). Sequential search is a conventional method in CBIR for searching in the database. The accuracy resulted from sequential search will be considered as optimal accuracy because searching in this method include whole database (WSS). Therefore, all competing indexing methods accuracies will be compared with it to check the degradation that can be obtained from these methods due to the reduction of search space. The best method will be the method that has less degradation for accuracy. K-means method is selected among many existing colour indexing methods. This is because; the tree-like structure methods (such as R-tree, SS-tree) are not efficient in fixed range queries (Deng et al., 2001; Samet, 1990). Lattice structure (Deng et al., 2001) is complicated method in obtaining its parameter such as lattice cell radius, desired and actual search radius in hexagonal lattice structure; hence there is no research which had performed a comparison on it. Additionally, most colour indexing methods that used R-tree indexing and its modifications such as (Babu et al., 1995) and (Sudhamani & Venugopal, 2007) used clustering method to obtain the representative of the colours in each tree node. Moreover, K-means is attractive for recent studies such as (Yildizer et al., 2012) to enhance it and obtaining better results. Therefore, K-means (Hartigan & Wong, 1979; Maimon & Rokach, 2005) and its recent adapted version KMB (Yildizer et al., 2012) are selected to compare with the proposed indexing methods. Proposed methods that will be used in comparison are *Octree*-like RGB indexing method and *Octree*-like RGB with Colour Percentage Filter (CPF) that used B+-tree for representing images of different colour percentages.

B. RGB Indexing-based Retrieval Performance

Experiments will be conducted on two datasets which are Corel-10K and Cartoon-11K databases to measure performance of the proposed indexing method. These datasets are different in terms of image content (colour and variety) as well as their sizes are somewhat large to fit the objective of designing indexing methods. The performance can be measured by time and accuracy as mentioned before. The comparison between the time of the indexing methods (the proposed, KM and KMB) and the time of sequential search can be achieved by computing SSR+OHR to represent the ratio of the time of indexing methods to sequential method. Accuracy of the indexing methods, using ARR, ANMRR and P(10) also must be compared with that of sequential search to measure the performance of the proposed indexing method in selecting relevant images from whole database. The proposed indexing method is applied on different types of descriptors. First type is large quantized number of colours-based descriptors which is colour *Correlogram* and *Autocorrelogram*. Second and third types are dominant colours-based descriptors. The second type is a pure DC-based descriptor, MPEG-7 DCD, which has the worst accuracy than other descriptors. The last type is a combination between first and second type of descriptors, which are DC-based *Correlogram* (DCBC) and weighted DCBC (WDCBC). The last descriptors showed their results better than others, as explained in Chapter 5.

i. Results of Cartoon-11K dataset

This section is dedicated to present the results of Cartoon dataset where Table 6.2 shows accuracy results of colour *Correlogram* descriptor when applying different indexing methods.

Table 6.3: *Accuracy and Efficiency metrics for Correlogram Descriptor using sequential search and all competing indexing methods applied on Cartoon-11K Dataset.*

Colour *ColGrm*	**Indexed colour=8**		**Indexed colour=5**		**Indexed colour=3**		**Indexed colour=1**	
	P(10)/ ARR/ ANMRR	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**
Sequential Search	0.350/ 0.118/ 0.852				100%			
K-Means Clustering	0.310/ 0.100/ 0.874	45.8%+ 0.70000%	0.320/ 0.102/ 0.872	40.8%+ 0.40000%	0.270/ 0.089/ 0.889	24.1%+ 0.10000%	0.220/ 0.076/ 0.905	14.5%+ 0.06000%
K-Means with B+Tree	**0.350**/ 0.115/ 0.856	76.6%+ 1.30000%	**0.350**/ 0.116/ 0.856	71%+ 1.10000%	0.310/ 0.104/ 0.870	39.8%+ 0.60000%	0.230/ 0.080/ 0.899	26.7%+ 0.40000%
Proposed Octree CTV =24	**0.360/ 0.122/ 0.848**	**57.3%+ 0.90000%**	**0.360/ 0.122/ 0.848**	**49.8%+ 0.08000%**	**0.360/ 0.120/ 0.850**	**40.5%+ 0.00700**	0.320/ 0.097/ 0.879	24.5%+ 0.00080%
Proposed Octree CTV=8	**0.350/** 0.115/ 0.857	**27%+ 0.07500%**	**0.350/** 0.113/ 0.859	**25%+ 0.02000%**	**0.350/** 0.108/ 0.866	**22.1%+ 0.00400%**	0.310/ 0.087/ 0.890	16.3%+ 0.00050%
Proposed Octree CTV=0	0.340/ 0.100/ 0.875	14%+ 0.00350%	0.340/ 0.099/ 0.877	13.6%+ 0.00050	0.320/ 0.093/ 0.884	12.8%+ 0.00020%	0.270/ 0.081/ 0.898	10.8%+ 0.00001%

Proposed Octree+CPF CTV =24	**0.360/ 0.122/ 0.848**	**47.2%+ 0.80000%**	**0.350/ 0.122/ 0.848**	**39.2%+ 0.07000%**	**0.350/ 0.121/ 0.850**	**29.7%+ 0.00500**	0.310/ 0.093/ 0.883	15.1%+ 0.00060%
Proposed Octree+CPF CTV=8	**0.360/** 0.115/ 0.857	**18.9%+ 0.06500%**	**0.350/** 0.113/ 0.859	**17.2%+ 0.01500%**	**0.350/** 0.108/ 0.865	**14.8%+ 0.00300%**	0.310/ 0.084/ 0.892	10.5%+ 0.00040%
Proposed Octree+CPF CTV=0	0.340/ 0.100/ 0.874	9.3%+ 0.00030%	0.340/ 0.099/ 0.876	9%+ 0.00400	0.320/ 0.092/ 0.884	8.5%+ 0.00015%	0.270/ 0.079/ 0.900	7.4%+ 0.00030%

** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for Colour Correlogram Descriptor (colours=27,distance=5) using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, proposed Octree and proposed Octree with Colour Percentage Filtering (CPF)) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*

Analysis the results of all competing indexing methods that shown in Table 6.2 can be summarized in the following points:

1. K-means clustering method reduces search space (that mean, the time) to half (and more) according to number of indexed colours; but the accuracy is degraded. This is because; the comparison of query DCs is performed with the cluster centroids instead of actual colours inside the clusters. Cluster centroid represents an approximation to all cluster members, thus comparison with it will produce some errors.

2. K-means with B+-tree (KMB) indexing method outperforms the original K-means in enhancing the retrieval accuracy but with increasing search space. This is because; some missing nearest images to the query that are located in the other clusters are reached in this method (this will increase accuracy and search space). Additionally, the colours in the suitable range inside one cluster are selected instead of all cluster members. Hence, this helps to avoid searching in the whole space. Therefore, this method succeeds in obtaining good accuracy (compared to K-means) with reasonable search space.

3. The Proposed *Octree* indexing and *Octree* with CPF methods have different settings involving four different number of indexed colours (8, 5, 3 and 1) and three colour tolerance values (24, 8 and 0). The accuracy value of the proposed indexing is increased in some settings (that presented in the bold font) than sequential search method. This is due to the following reasons; first, the query's DC is reached to the exact corresponding colour value in the index structure and some colours around it (according to tolerance value).

This in turn will lead to the query be compared with images that have similar colours only. In this case, there is no need for an approximation of the colour values (like in K-means method) that have some errors and leads to compare with some images of not similar colours. Second reason for increasing the accuracy is narrowing the search space to include images of similar colours only and in turn the rank of some relevant images will be enhanced. As depicted in Table 6.2, the search space is significantly reduced into 22% without degradation to the accuracy that measured by P(10), which represent the accuracy of the first page of retrieval results of *CBIRS*. P(10) is very important in web-based application where the user prefers to redo the search instead of going to next page of the search (Penatti et al., 2012). On the other hand, K-means method could not maintain the retrieval accuracy whereas KMB could, in the 8- and 5-indexed colours but with significantly increase to the search space. Moreover, the accuracy of the proposed *Octree* method (in most settings) is better than the accuracy of K-means and KMB with outperforming in reducing search space, the *Octree* has SSR lower than K-means and KMB.

4. Colour Percentage-based Filtering (CPF) method using B+-tree is proposed to speed up the retrieval process by considering only the images that have similar colours as well as similar colour percentage. The result showed that this filtering process succeed in reducing SSR by 10% (in average) without degradation to the accuracy.

The main disadvantage of the proposed *Octree* method is that it has same problem of all space partitioning-based indexing methods, which is locating the query image on the border of the partition. This will lead to the loss of some similar images that are located in the neighbour partitions. Although with this disadvantage, the proposed *Octree* method still outperforms the KMB method that solved this problem by considering some neighbour clusters. It is worth mentioning, the experiments in Table 6.2 is conducted on colour *Correlogram* descriptor of settings 27 (3*3*3) colours and distance =5. The results of other descriptors will be depicted in the following tables.

Table 6.4: *Accuracy and Efficiency metrics for AutoCorrelogram Descriptor using sequential search and all competing indexing methods applied on Cartoon-11K Dataset.*

Colour *AutoColGrm*	**Indexed colour=8**		**Indexed colour=5**		**Indexed colour=3**		**Indexed colour=1**	
	P(10)/ ARR/ ANMRR	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**
Sequential Search	0.290/ 0.094/ 0.880				100%			
K-Means Clustering	0.260/ 0.085/ 0.893	45.8%+ 0.70000%	0.270/ 0.084/ 0.894	40.8%+ 0.40000%	0.230/ 0.077/ 0.903	24.1%+ 0.10000%	0.180/ 0.068/ 0.915	14.5%+ 0.06000%
K-Means with B+Tree CG=0,Cs=2 2	0.280/ 0.093/ 0.883	76.6%+ 1.30000%	0.280/ 0.093/ 0.882	71%+ 1.10000%	0.260/ 0.089/ 0.889	39.8%+ 0.60000%	0.190/ 0.072/ 0.912	26.7%+ 0.40000%
Proposed Octree CTV =24	**0.290/ 0.101/ 0.874**	**57.3%+ 0.90000%**	**0.290/ 0.102/ 0.873**	**49.8%+ 0.08000%**	**0.300/ 0.101/ 0.874**	**40.5%+ .00700**	0.270/ 0.082/ 0.897	24.5%+ .00080%
Proposed Octree CTV=8	**0.300/ 0.099/ 0.877**	**27%+ .07500%**	**0.300/ 0.099/ 0.878**	**25%+ 0.02000%**	**0.290/ 0.094/ 0.883**	**22.1%+ .00400%**	0.270/ 0.077/ 0.904	16.3%+ .00050%
Proposed Octree CTV=0	0.280/ 0.090/ 0.888	14%+ 0.00350%	0.280/ 0.089/ 0.890	13.6%+ .00050	0.270/ 0.083/ 0.897	12.8%+ .00020%	0.230/ 0.072/ 0.909	10.8%+ .00001%

Proposed Octree+CPF CTV =24	**0.290/ 0.101/ 0.874**	**47.2%+ 0.80000%**	**0.300/ 0.103/ 0.873**	**39.2%+ .07000%**	**0.300/ 0.102/ 0.873**	**29.7%+ 0.00500**	0.270/ 0.081/ 0.897	15.1%+ .00060%
Proposed Octree+CPF CTV=8	**0.300/ 0.102/ 0.875**	**18.9%+ .06500%**	**0.300/ 0.101/ 0.876**	**17.2%+ .01500%**	**0.300/ 0.097/ 0.881**	**14.8%+ .00300%**	0.270/ 0.076/ 0.904	10.5%+ .00040%
Proposed Octree+CPF CTV=0	0.290/ 0.092/ 0.886	9.3%+ .00030%	0.290/ 0.091/ 0.887	9%+ 0.00400	0.280/ 0.084/ 0.895	8.5%+ .00015%	0.230/ 0.072/ 0.909	7.4%+ 0.00030%

** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for Colour AutoCorrelogram Descriptor (colour=27, distance=5) using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, proposed Octree and proposed Octree with Colour Percentage Filtering (CPF)) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*

Table 6.5: *Accuracy and Efficiency metrics for MPEG-7 Dominant Colour Descriptor using sequential search and all competing indexing methods applied on Cartoon-11K Dataset.*

MPEG-7 DCD	Indexed colour=8		Indexed colour=5		Indexed colour=3		Indexed colour=1	
	P(10)/ ARR/ ANMRR	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**
Sequential Search	0.230/ 0.060/ 0.922				100%			
K-Means Clustering	0.200/ 0.057/ 0.926	45.8%+ 0.70000%	0.210/ 0.059/ 0.926	40.8%+ 0.40000%	0.180/ 0.051/ 0.935	24.1%+ 0.10000%	0.140/ 0.041/ 0.947	14.5%+ 0.06000%
K-Means with B+Tree CG=0,Cs=2 2	**0.230/** 0.059/ **0.922**	76.6%+ 1.30000%	**0.230/** 0.059/ 0.923	71%+ 1.10000%	0.210/ 0.056/ 0.927	39.8%+ 0.60000%	0.150/ 0.042/ 0.945	26.7%+ 0.40000%
Proposed Octree CTV =24	**0.230/ 0.060/ 0.921**	**57.3%+ 0.90000%**	**0.240/ 0.061/ 0.921**	**49.8%+ 0.08000%**	**0.240/ 0.059/ 0.922**	**40.5%+ 0.00700**	0.230/ 0.057/ 0.926	24.5%+ 0.00080%
Proposed Octree CTV=8	**0.240/ 0.060/ 0.921**	**27%+ 0.07500%**	**0.240/ 0.061/ 0.921**	**25%+ 0.02000%**	**0.240/ 0.060/ 0.922**	**22.1%+ 0.00400%**	0.230/ 0.056/ 0.927	16.3%+ 0.00050%

Proposed Octree CTV=0	**0.230/ 0.060/ 0.922**	14%+ 0.00350%	**0.230/ 0.061/ 0.922**	13.6%+ 0.00050	0.230/ 0.058/ 0.925	12.8%+ 0.00020%	0.200/ 0.054/ 0.930	10.8%+ 0.00001%
Proposed Octree+CPF CTV =24	**0.230/ 0.060/ 0.921**	**47.2%+ 0.80000%**	**0.240/ 0.060/ 0.921**	**39.2%+ 0.07000%**	**0.240/ 0.059/ 0.923**	**29.7%+ 0.00500**	0.230/ 0.056/ 0.927	15.1%+ 0.00060%
Proposed Octree+CPF CTV=8	**0.240/ 0.060/ 0.922**	**18.9%+ 0.06500%**	**0.240/ 0.060/ 0.921**	**17.2%+ 0.01500%**	**0.240/ 0.060/ 0.922**	**14.8%+ 0.00300%**	0.230/ 0.055/ 0.928	10.5%+ 0.00040%
Proposed Octree+CPF CTV=0	**0.230/ 0.060/ 0.922**	9.3%+ 0.00030%	**0.230/ 0.061/ 0.922**	9%+ 0.00400	0.230/ 0.058/ 0.925	8.5%+ 0.00015%	0.200/ 0.053/ 0.931	7.4%+ 0.00030%

** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for MPEG-7 Dominant Colour Descriptor using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, proposed Octree and proposed Octree with Colour Percentage Filtering (CPF)) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*

Table 6.6: *Accuracy and Efficiency metrics for DC based-Correlogram Descriptor using sequential search and all competing indexing methods applied on Cartoon-11K Dataset.*

DCBC	Indexed colour=8		Indexed colour=5		Indexed colour=3		Indexed colour=1	
	P(10)/ ARR/ ANMRR	SSR+ OHR	ARR/ ANMRR/ P(10)	SSR+ OHR	ARR/ ANMRR/ P(10)	SSR+ OHR	ARR/ ANMRR/ P(10)	SSR+ OHR
Sequential Search	0.370/ 0.131/ 0.837				100%			
K-Means Clustering	0.340/ 0.111/ 0.862	45.8%+ 0.70000%	0.340/ 0.112/ 0.859	40.8%+ 0.40000%	0.300/ 0.096/ 0.880	24.1%+ 0.10000%	0.220/ 0.078/ 0.903	14.5%+ 0.06000%
K-Means with B+Tree	**0.370/** 0.124/ 0.843	76.6%+ 1.30000%	**0.370/** 0.126/ 0.841	71%+ 1.10000%	0.330/ 0.110/ 0.863	39.8%+ 0.60000%	0.250/ 0.084/ 0.896	26.7%+ 0.40000%
Proposed Octree CTV =24	**0.380/ 0134/ 0.834**	**57.3%+ 0.90000%**	**0.380/ 0.133/ 0.835**	**49.8%+ 0.08000%**	**0.380/** 0.126/ 0.842	**40.5%+ 0.00700**	0.330/ 0.095/ 0.880	24.5%+ 0.00080%
Proposed Octree CTV=8	**0.370/** 0.119/ 0.851	**27%+ 0.07500%**	**0.370/** 0.117/ 0.854	**25%+ 0.02000%**	0.360/ 0.110/ 0.861	**22.1%+ 0.00400%**	0.320/ 0.086/ 0.889	16.3%+ 0.00050%
Proposed Octree CTV=0	0.360/ 0.103/ 0.871	14%+ 0.00350%	0.350/ 0.101/ 0.873	13.6%+ 0.00050	0.340/ 0.094/ 0.880	12.8%+ 0.00020%	0.280/ 0.081/ 0.897	10.8%+ 0.00001%

Proposed Octree+CPF CTV =24	**0.380/ 0133/ 0.834**	**47.2%+ 0.80000%**	**0.380/ 0.132/ 0.835**	**39.2%+ 0.07000%**	**0.380/** 0.126/ 0.843	**29.7%+ 0.00500%**	0.320/ 0.092/ 0.884	15.1%+ 0.00060%
Proposed Octree+CPF CTV=8	**0.370/** 0.120/ 0.851	**18.9%+ 0.06500%**	**0.370/** 0.117/ 0.854	**17.2%+ 0.01500%**	0.360/ 0.111/ 0.862	**14.8%+ 0.00300%**	0.310/ 0.085/ 0.892	10.5%+ 0.00040%
Proposed Octree+CPF CTV=0	0.360/ 0.102/ 0.872	9.3%+ 0.00030%	0.350/ 0.101/ 0.874	9%+ 0.00400	0.330/ 0.093/ 0.882	8.5%+ 0.00015%	0.270/ 0.078/ 0.901	7.4%+ 0.00030%

** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for DC based-Correlogram Descriptor using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, proposed Octree and proposed Octree with Colour Percentage Filtering (CPF)) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*

Table 6.7: *Evaluation metrics for Weighted DC-based Correlogram using sequential search and all competing indexing methods applied on Cartoon-11K Dataset.*

WDCBC	Indexed colour=8		Indexed colour=5		Indexed colour=3		Indexed colour=1	
	P(10)/ ARR/ ANMRR	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**
Sequential Search	0.410/ 0.153/ 0.810				100%			
K-Means Clustering	0.360/ 0.127/ 0.842	45.8%+ 0.70000%	0.380/ 0.130/ 0.838	40.8%+ 0.40000%	0.340/ 0.114/ 0.859	24.1%+ 0.10000%	0.260/ 0.087/ 0.892	14.5%+ 0.06000%
K-Means with B+Tree CG=0,Cs=2 2	0.400/ 0.147/ 0.817	76.6%+ 1.30000%	**0.410/** 0.148/ 0.816	71%+ 1.10000%	0.370/ 0.129/ 0.840	39.8%+ 0.60000%	0.280/ 0.097 0.879	26.7%+ 0.40000%
Proposed Octree CTV =24	**0.410/** 0.151/ 0.813	**57.3%+ 0.90000%**	**0.410/** 0.150/ 0.814	**49.8%+ 0.08000%**	**0.410/** 0.143/ 0.822	**40.5%+ 0.00700**	0.350/ 0.103/ 0.870	24.5%+ 0.00080%
Proposed Octree CTV=8	0.390/ 0.129/ 0.837	**27%+ 0.07500%**	0.380/ 0.125/ 0.842	**25%+ 0.02000%**	0.380/ 0.119/ 0.850	**22.1%+ 0.00400%**	0.330/ 0.093/ 0.881	16.3%+ 0.00050%
Proposed Octree CTV=0	0.370/ 0.111/ 0.861	14%+ 0.00350%	0.360/ 0.109/ 0.864	13.6%+ 0.00050	0.350/ 0.102/ 0.872	12.8%+ 0.00020%	0.300/ 0.087/ 0.890	10.8%+ 0.00001%

Proposed Octree+CPF CTV =24	**0.410/** 0.151/ 0.813	**47.2%+ 0.80000%**	**0.410/** 0.150/ 0.814	**39.2%+ 0.07000%**	**0.410/** 0.142/ 0.823	**29.7%+ 0.00500%**	0.340/ 0.100/ 0.874	15.1%+ 0.00060%
Proposed Octree+CPF CTV=8	0.390/ 0.130/ 0.837	**18.9%+ 0.06500%**	0.390/ 0.126/ 0.842	**17.2%+ 0.01500%**	0.380/ 0.119/ 0.850	**14.8%+ 0.00300%**	0.330/ 0.092/ 0.884	10.5%+ 0.00040%
Proposed Octree+CPF CTV=0	0.370/ 0.109/ 0.862	9.3%+ 0.00030%	0.360/ 0.107/ 0.865	9%+ 0.00400	0.350/ 0.100/ 0.874	8.5%+ 0.00015%	0.290/ 0.085/ 0.893	7.4%+ 0.00030%

** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for Weighted DC-based Correlogram Descriptor using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, proposed Octree and proposed Octree with Colour Percentage Filtering (CPF)) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*

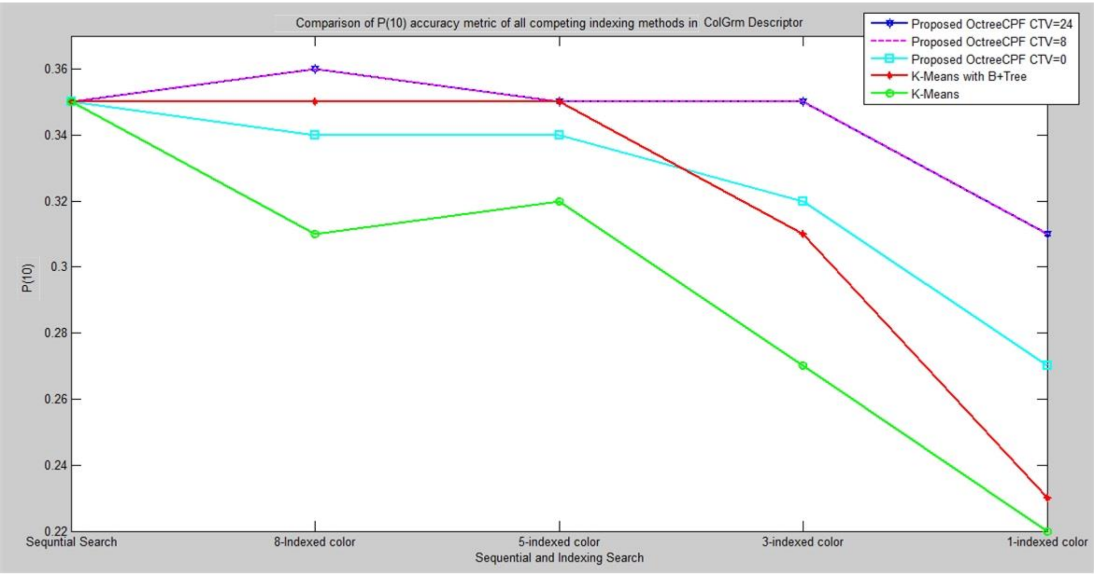
The above tables display the accuracy and efficiency metrics results of three different types of colour descriptors, as mentioned previously (*ColGrm* and *Autocorrelogram* from first type, MPEG-7 DCD is a second type and DCBC and WDCBC represent the third type of colour descriptors). The MPEG-7 DCD has the worst result while DC-based descriptors (DCBC and WDCBC) have best results. These descriptors results are affected by indexing methods as follows:

i. The proposed *Octree* indexing method narrowed the search space by collecting only the images of similar colours to the query. MPEG-7 DCD on the other hand, is not efficient for retrieving similar images (worst descriptor). Therefore, the proposed indexing method helps in enhancing its result noticeably as can be seen in Table 6.4, in which most indexing results (of different settings) are better than sequential search result. Except for the settings of 1 indexed colour and 0 tolerance value, these settings force coarse filtering on database images thus it produces poor results in all descriptors.

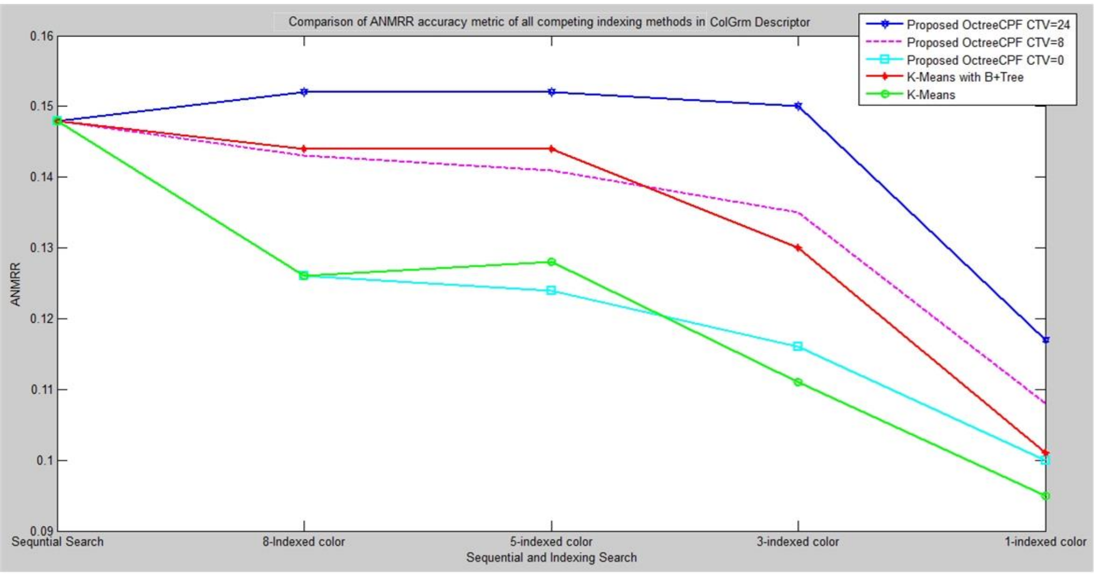
ii. A proposed indexing method also improves the accuracy of good-result descriptors (*Correlogram* and *Autocorrelogram*) but with improvement ratio smaller than that in MP7DCD, as can be seen in Tables 6.2 and 6.3. This is because the descriptors perform colours comparisons and their spatial relations, thus most similar images are already ranked well in the original searching process (searching whole space by sequential search). Therefore, narrowing search space of proposed indexing method is not highly effected on accuracy on these descriptors but it speeds up searching process by reducing the search space.

iii. Enhancement on accuracy of DC-based descriptors is not noticeable because these methods are already ranked images correctly in the sequential search but the search space is reduced. Additionally, indexing accuracy results tend to be little worse than sequential search results in most cases, this is because the problem of all space partitioning-based methods, which is loss some relevant images that lies in other partitions as well as the accuracy of these descriptors is already high.

To ease the comparison among all competing indexing methods, graphical results are depicted in Figures 6.12, 6.13, 6.14 and 6.15. In these figures, three descriptors are selected to represent the three previous mentioned types of descriptors that are used in quantitative comparison. *Correlogram* descriptor is used as representative on large quantized number of colours-based descriptors and is illustrated in Figure 6.12. The MPEG-7 DCD is used to represent DCD and is depicted in Figure 6.13, whereas DC-based *Correlogram* is used to represent DC-based descriptors. DC-based *Correlogram* is depicted in Figure 6.14. All figures are used to measure the accuracy of all competing indexing methods. P(10) and ANMRR are selected only as accuracy metrics because P(10) show the accuracy of the first page result in web-based application and ANMRR represents one value to measure three performance issues together, which are precision, recall and rank information of retrieved images. ANMRR in its nature has inverse value where the value 1 represents the worst case and value 0 represents best result. Thus, inversing its value is achieved before displaying it in graphical results. Moreover, Search space ratio is used in the latter figure (Figure 6.15) to point out reduction that has been obtained in each indexing method. Figure 6.15 (a) is used to measure efficiency (search space, which inversely proportional with the speed) of the competing indexing methods. Additionally, Figure 6.15 (b) is used to explain the effectiveness of colour percentage filtering (CPF) technique over proposed *Octree* indexing method, in reducing the search space.



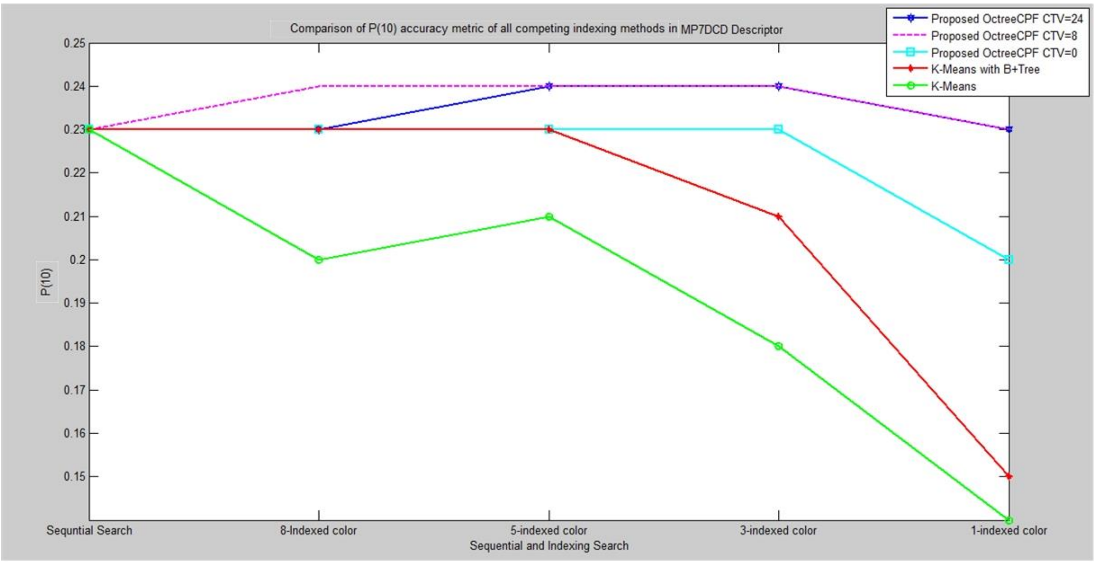
(a) **P(10) Accuracy Metric**



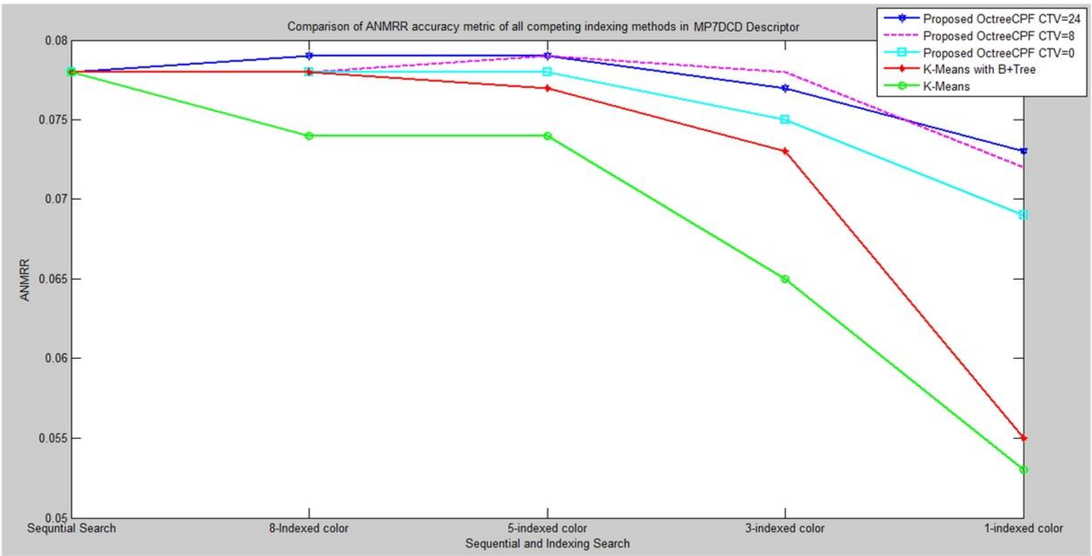
(b) **ANMRR Accuracy Metric**

Figure 6.12: Comparison the accuracy of indexing methods on ColGrm Descriptor using (a) P(10) and (b) ANMRR.

Graphical results on ColGrm descriptors show that the proposed method in 24 and 8 colour tolerance values outperforms the KM and KMB methods when P(10) is used. Also, ANMRR metric results show that Octree with 24 CTV outperforms other methods whereas the results of KMB and Octree with 8 CTV are similar. This result indicates that SSR can decide which one is better than another. From Figure 6.15, SSR of Octree with 8 CTV is smaller than KMB, therefore it is better. The problem that the proposed method (CTV=8) suffers from is locating the query on the border some times and there are many similar images in the neighbours partitions, thus achieving results similar to KMB. The settings of 1 indexed colour and 0 CTV refer to worst cases in all experiments, thus avoiding these setting is better to yield good retrieval accuracy.



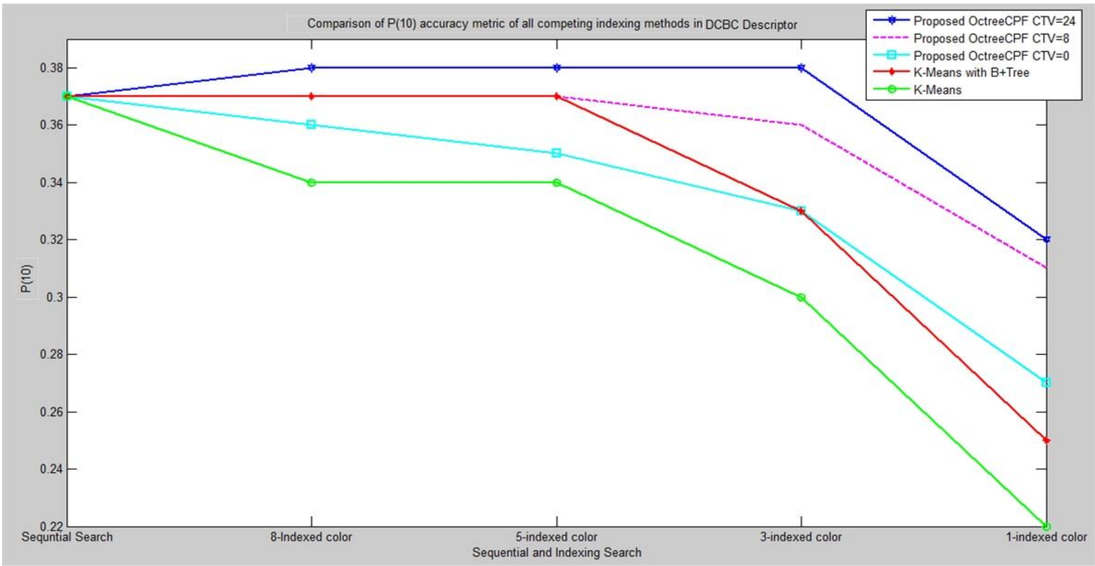
(a) P(10) Accuracy Metric



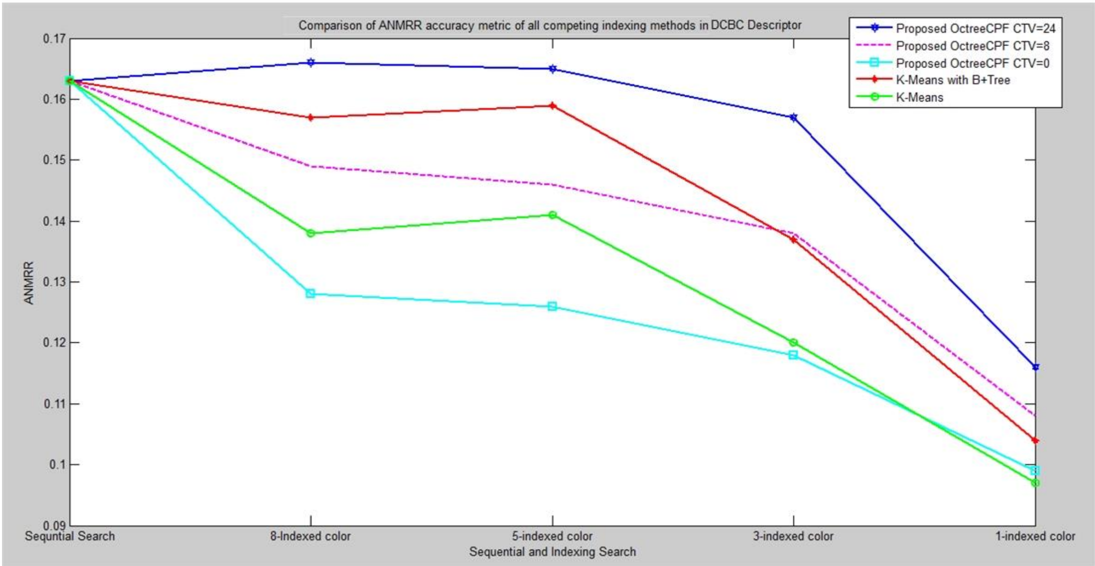
(b) ANMRR Accuracy Metric

Figure 6.13: Comparison the accuracy of indexing methods on MPEG-7 DCD using (a) P(10) and (b) ANMRR.

In MP7DCD, all settings of the proposed method showed good accuracy compared to KM and KMB. K-means clustering indexing method exhibited worst results (accuracy) among the competing descriptors, although it has low SSR but this will not intercede to it with the user. The user prefers to wait some time rather than obtaining bad results (Penatti et al., 2012). Nevertheless, the proposed method with settings 8 and 0 CTV has better accuracy and efficiency results than it.



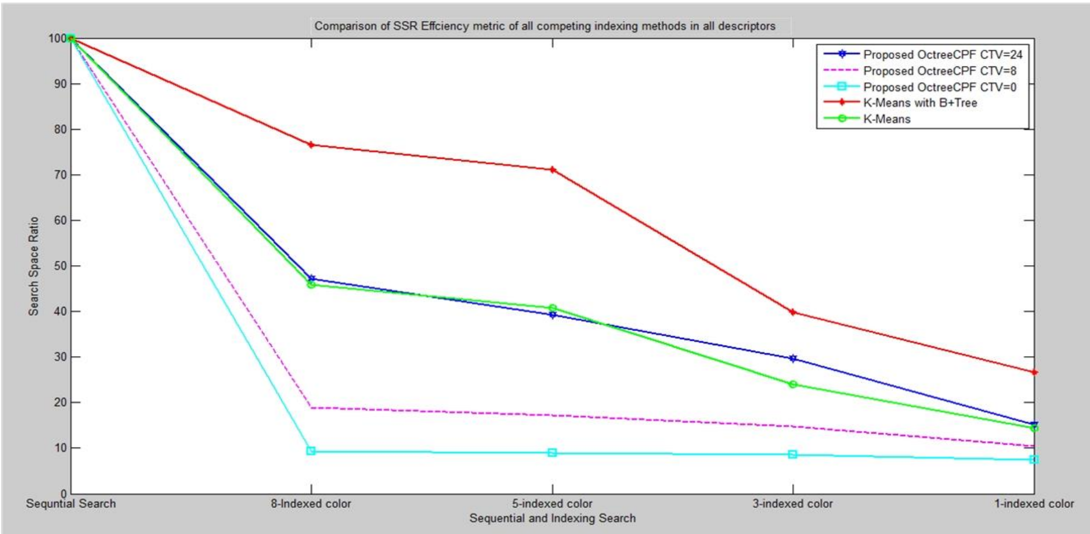
(a) P(10) Accuracy Metric



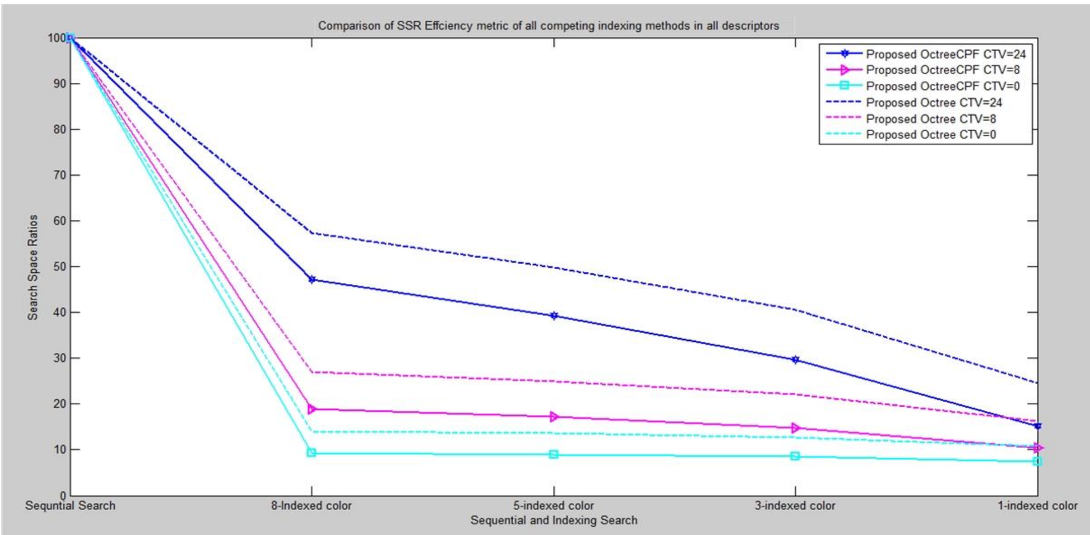
(b) ANMRR Accuracy Metric

Figure 6.14: Comparison the accuracy of indexing methods on DC-based ColGrm using (a) P(10) and (b) ANMRR.

The above results of DCBC descriptors show that KMB indexing method performs better than the proposed RGB indexing method in the setting of CTV=8. This is because; the proposed indexing depends on RGB colour space whereas the DCBC depends on LUV colour space in its similarity measure. The difference between these two spaces is the LUV is a perceptual colour space (it matches human visual system) and the two similar colours (they have small LUV distance) in this space may consider not similar colours in RGB space because they have large RGB distance. Therefore, the proposed RGB indexing will lose some perceptually similar colours that have large distance in RGB colour space. This reason motivates this research to propose perceptual LUV indexing instead of RGB indexing, as illustrated in next section.



(a) SSR Efficiency metric for all competing indexing methods



(b) SSR Efficiency metric for OcTree indexing method compared with OcTree with CPF method

Figure 6.15: Comparison the Search Space Ratio (SSR) for Cartoon-11K dataset on (a) different settings using Octree and (b) Octree vs. Octree+CPF.

In the above Figure 6.15 (a), it is shown that KMB has worst (large) SSR whereas the proposed method with 8 and 0 CTV has best (less) SSR. *Octree*-like method with 24 CTV and K-means represents the compromise between them. From these results, it can be concluded that the proposed *Octree*-like indexing method with 8 CTV (has low SSR) and 8 indexed colours (has high accuracy) has the best performance in general and especially in web-based applications.

ii. Results of Corel-10K dataset

In this section, graphical results are presented to validate the behaviour of the proposed indexing methods in natural image collections. Therefore, results of three different descriptors are depicted in the figures 6.16, 6.17, 6.18 and 6.19.

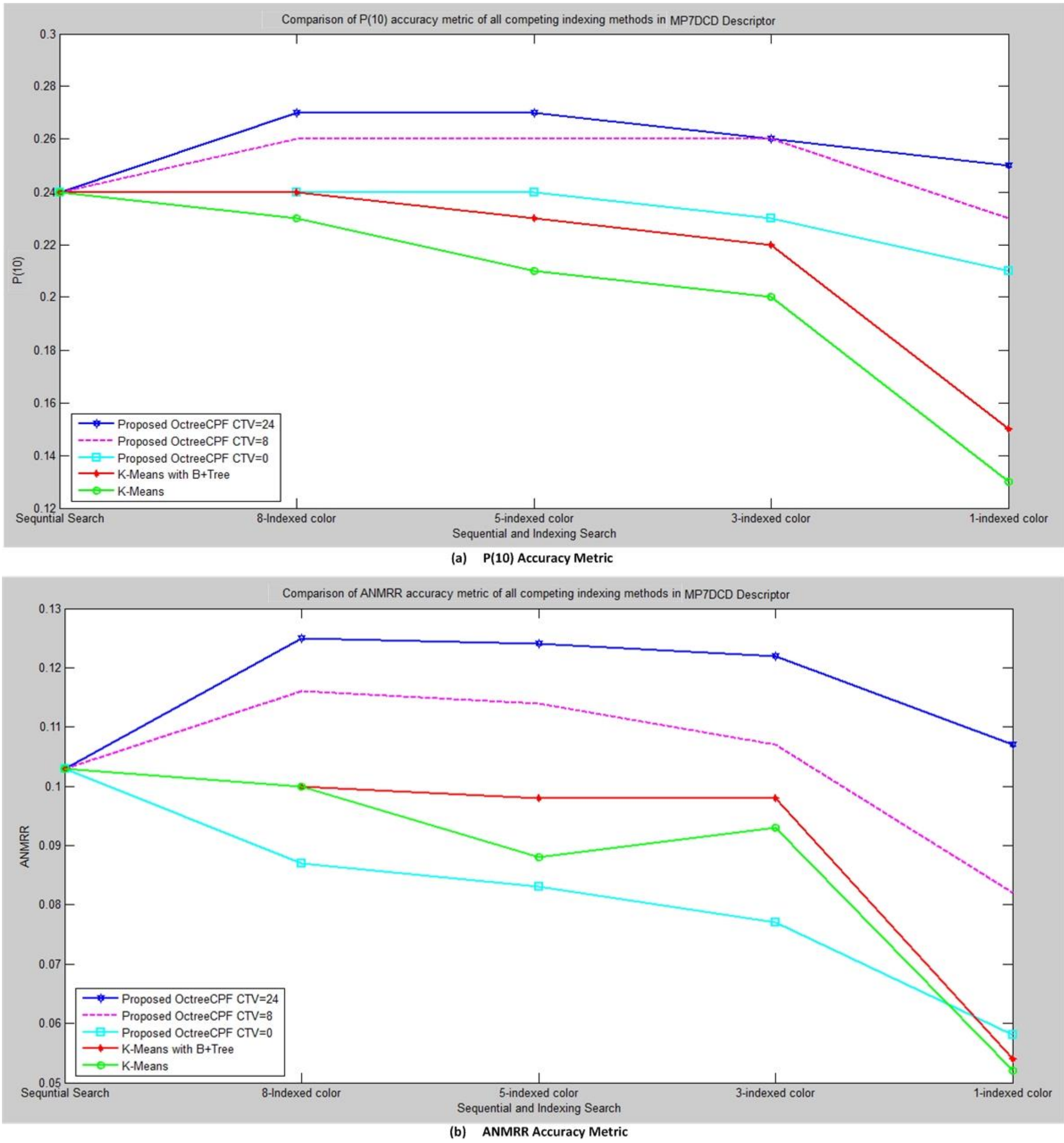
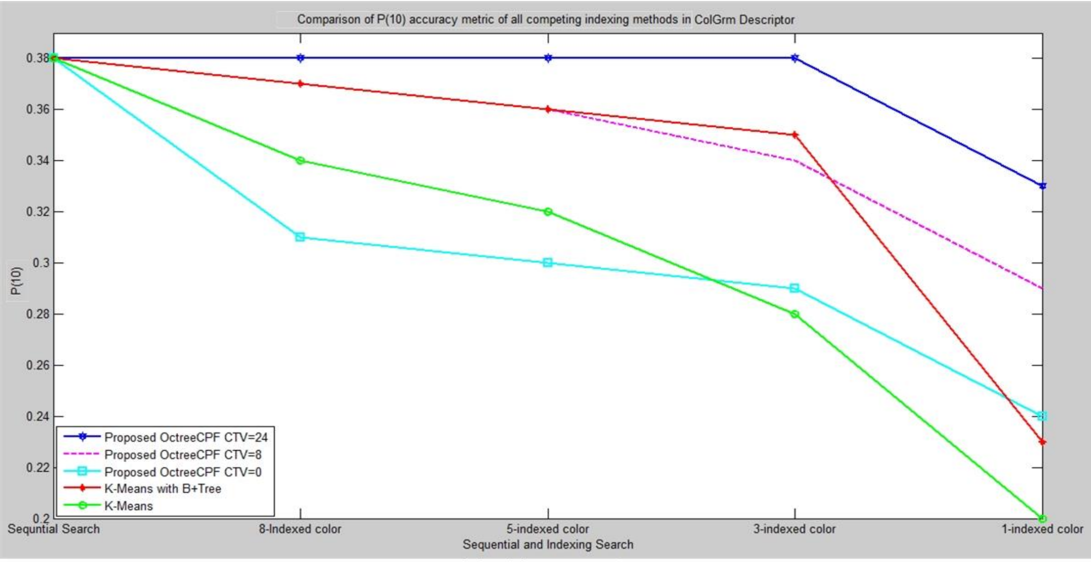
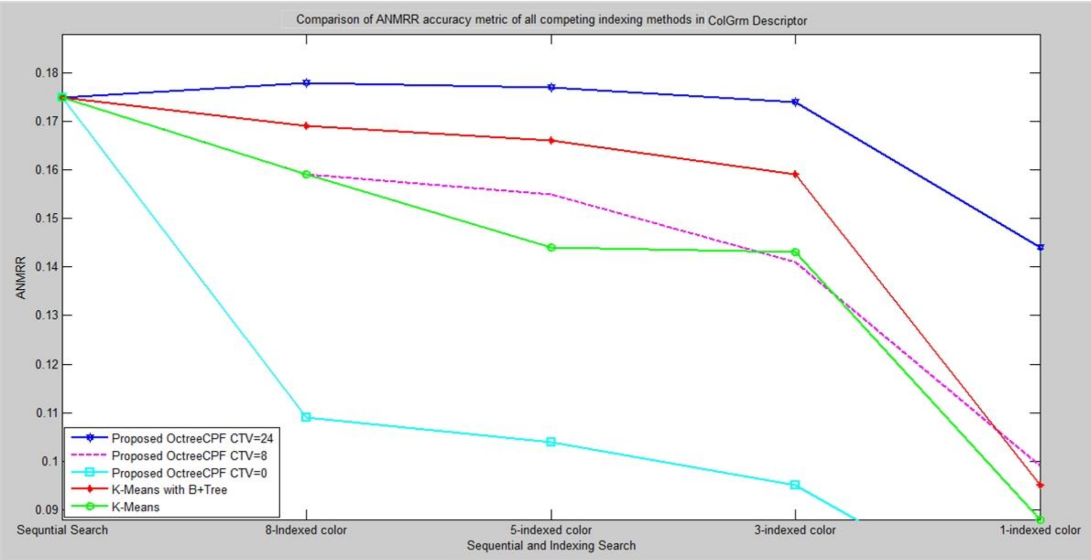


Figure 6.16: Comparison the accuracy of indexing methods on MP7DCD in Corel-10K database using (a) P(10) and (b) ANMRR.

Similar to the results of cartoon dataset, the proposed indexing methods in Corel-10K database, in MP7DCD and in CTV equals to 24 and 8, outperform the sequential search, KM and KMB in accuracy as well as in reduction the search space. The only 1 indexed colour is excluded from this outperforming due to coarse colour filtering; this can be noticed in all descriptors' experiments.



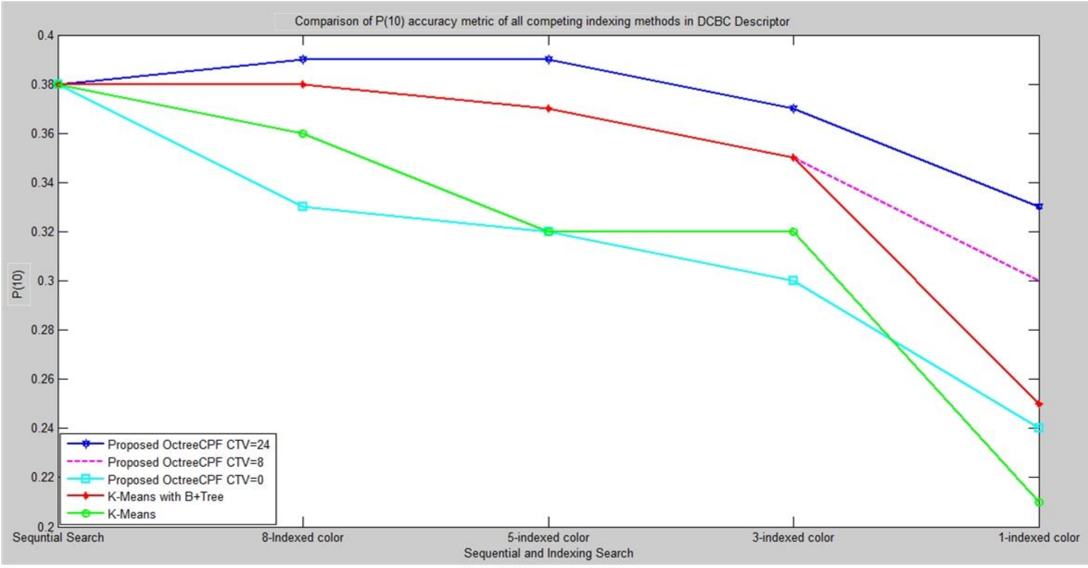
(a) P(10) Accuracy Metric



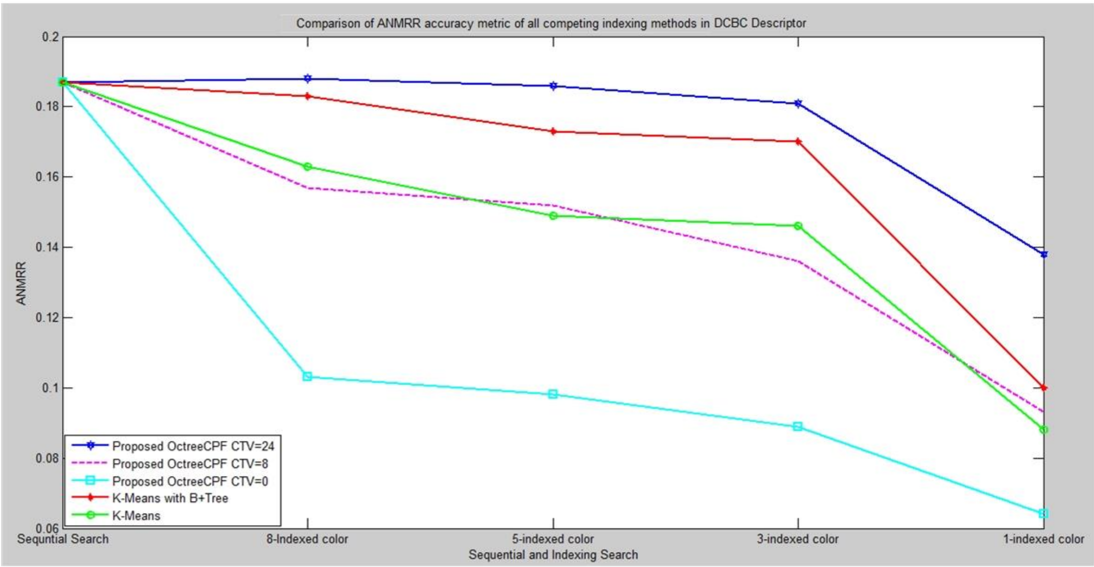
(b) ANMRR Accuracy Metric

Figure 6.17: Comparison the accuracy of indexing methods on ColGrm in Corel-10K database using (a) P(10) and (b) ANMRR.

Likewise the results of Cartoon-11K dataset, the proposed indexing method in *ColGrm* descript outperforms the sequential search, KM and KMB methods in terms of accuracy and reduction the search space when CTV=24 (except in 1 indexed colour).



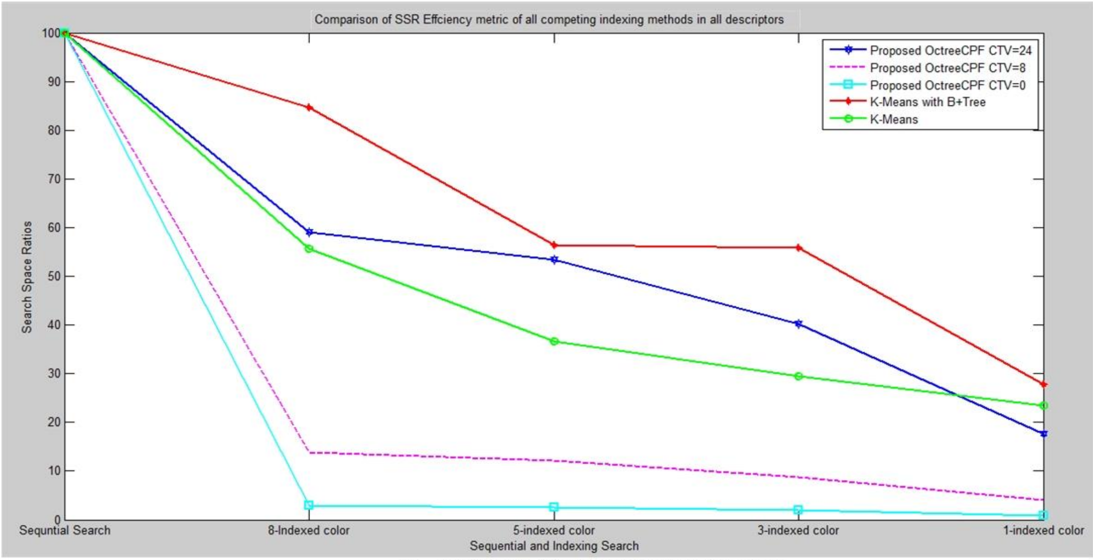
(a) P(10) Accuracy Metric



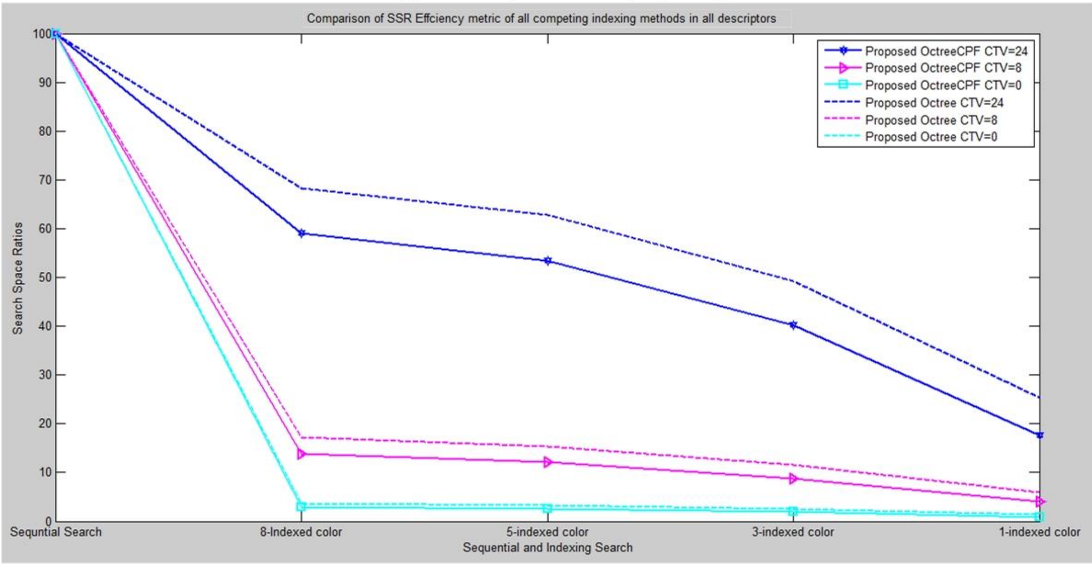
(b) ANMRR Accuracy Metric

Figure 6.18: Comparison the accuracy of indexing methods on DCBC in Corel-10K database using (a) P(10) and (b) ANMRR.

The outperforming of proposed indexing methods in DC-based ColGrm descriptor lies when CTV equals 24 and indexed colours are 8 and 5 colours only. The descriptors (such as DCBC and ColGrm) are good descriptors and only the high settings can enhance their results. The efficiency of proposed indexing methods in Corel-10K dataset is identical to that of Cartoon-11K where the KMB is the worst and the proposed indexing method of 8 and 0 CTV is the best, as depicted in Figure 6.19.



(a) SSR Efficiency metric for all competing indexing methods



(b) SSR Efficiency metric for OcTree indexing method compared with OcTree with CPF method

Figure 6.19: Comparison the Search Space Ratio of indexing methods in Corel-10K dataset using (a) different settings of Octree and (b) Octree vs.Octree+CPF.

It is worth mentioning, all the above results of different colour descriptors are obtained by similarity measure in perceptual LUV colour space whereas the proposed *Octree* indexing method is depended on RGB colour space. This certainly affects the accuracy of the proposed method. This is because the proposed method depends on maximum difference between colours is 24 in RGB space but there is a possibility that two similar colours may have large RGB distance (larger than 24) because this space is not consistent with human visual system (Xia & Kuo, 1998). The latter issue can be considered as a drawback of the proposed *Octree*-like indexing method where it depends on RGB and this will affect negatively on any descriptor that use another colour space. Little accuracy degradation of any indexing method is normal compared to significant reduction of search space (speed up searching process), as occurred in the proposed RGB indexing method. However, increasing or keeping the accuracy of sequential search process is one of characteristics the good indexing methods. Therefore, second promising indexing method is proposed to solve the above drawback of first proposed indexing method and its details is explained in the next section.

6.3 LUV Indexing Method

The characteristics of RGB colour distribution is uniform, thus it can be quantized using simple uniform quantization method such as *Octree* quantization method. Uniform quantization is a simple and straightforward method (Park, Park, Kim & Han, 1999; Wan & Kuo, 1998). The disadvantage of RGB colour space however is it is not a perceptual uniform colour space (Park et al., 1999).

On the other hand, perceptual colour spaces (that match human visual system) such as LUV, Lab and YUV have perceptual and non-uniform colour distribution. As a result of this, complicated and time consuming quantization methods must be used to quantize these spaces such as standard vector quantization (for example GLA method that used in MPEG-7 DCD) and product vector quantization methods (Wan & Kuo, 1998). For perceptual colour indexing, two methods can be used which are colour clustering methods and utilizing uniform colour spaces. Colour clustering methods were used in Babu et al. (1995) and Sudhamani and Venugopal (2007) while uniform colour spaces were used in the first proposed indexing method and Taycher (1997). The disadvantages of these two methods can be summarized as follows: (i) Colour clustering methods (such as K-means and KMB) have colours approximation problem that will lead to accuracy degradation of the indexing system as depicted in Section 6.2.4. (ii) When uniform quantization (such as *Octree*) for perceptual (non-uniform) colour spaces is used, it does not take into account the perceptual similarity between different bins. In other words, perceptual effect means that the same distance from two different points in the colour space makes the equal perceivable colour difference. For example, if two 3-D colours C1(X1, Y1, Z1), C2(X2, Y2, Z2) and distance D are given; then new colour C1+D (X1+D, Y1+D, Z1+D) has certain visual effect from colour C1 and it equals to the visual effect of new colour C2+D (X1+D, Y1+D, Z1+D) from colour C2. That means, similar difference will have the same effect on any colour in this colour space (Du-Sik et al., 1999). RGB and HSV colour spaces do not exhibit perceptual uniformity whereas CIE LUV, CIE Lab and YUV colour spaces have this perceptual effect (perceptual uniformity).

Therefore, the objective of this section is to propose a design of an indexing method for perceptual (non-uniform) colour spaces (such as LUV, Lab and YUV). The proposed perceptual indexing method uses uniform colour space (such as RGB in this research) with consideration on perceptual similarity of different colours.

6.3.1 Indexing Structure

The structure of the second proposed indexing structure also depends on uniform quantization of RGB colour space, similar to the *Octree* quantization method that used in the first proposed indexing method. The first difference of the LUV indexing from RGB indexing method is that each colour in the colour space will be reached individually at a time (in single query) instead of reaching many colours at a time through using colour tolerance value (CTV). CTV cannot use in this approach because perceptual concept cannot be guaranteed using RGB tolerance value.

To achieve the perceptual concept of LUV colour space (the space of this research), LUV colour distance (Manhattan or Euclidian distance) is used. Each colour in the 5-bits quantized space (that means, each node in the third level of the index structure) is compared with other colours in the space and stores the perceptually similar colours in a list belonging to this colour. This list will be denoted as Similarity List (SL) hereinafter, as depicted in Figure 6.20.

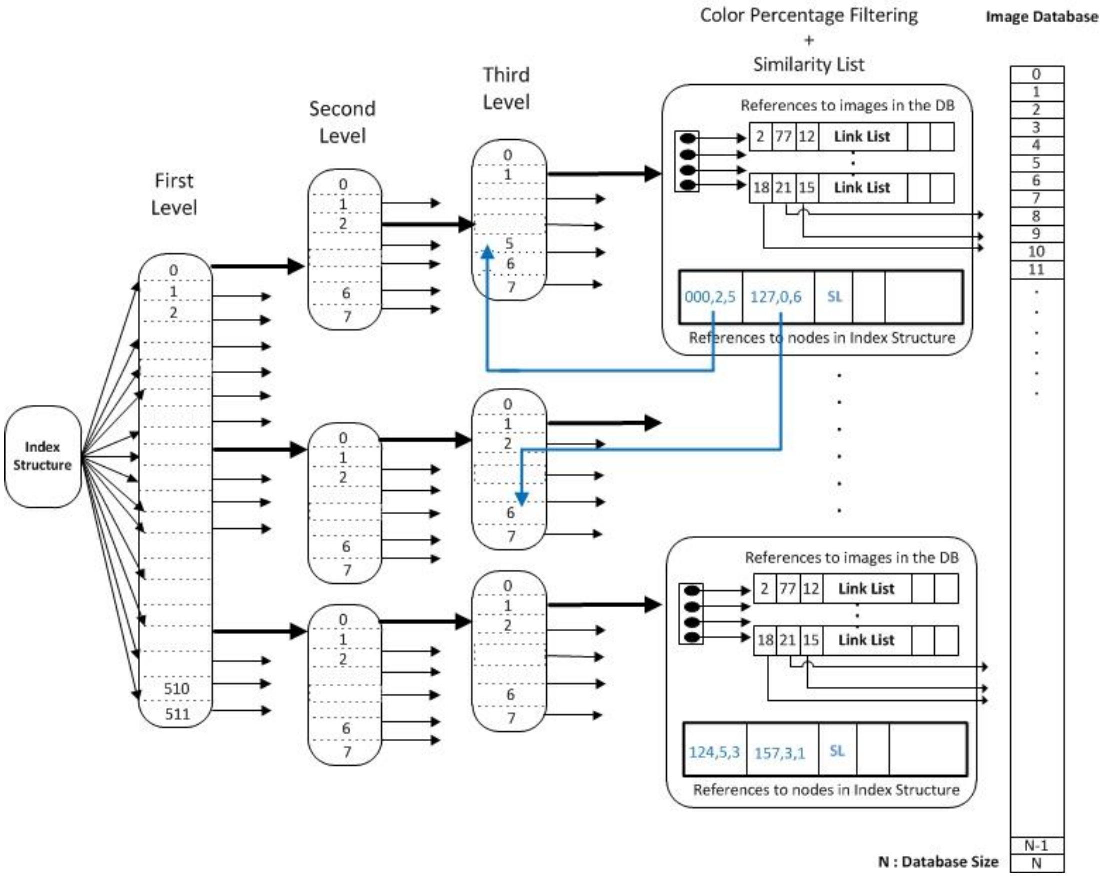


Figure 6.20: A Second Proposed Indexing Method for perceptual colour spaces.

In other words, each node of the third level in the index structure (that means, each colour in quantized space) has one SL that contains references to all colours in the space that are perceptually similar to this colour (node). The steps of building SL of all colours in the space is depicted in Algorithm 6.4.

Algorithm 6.4: Build Similarity Lists for Index Structure

Input

Index Structure: Octree-like Index Structure for Database (it has fixed structure as Array[512,8,8] of B+Tree[4] of link list of References to the Images in DB).

Output

SLs: Similarity Lists that associated with Index Structure of DB, its definition is Array [512,8,8] of linked list of Index Structure References (3 numbers represent the location of certain node in the Index structure)

BEGIN

1. Set all Similarity Lists to an Empty list

$$\forall\, i = 0..512, \qquad j = 0..7, \qquad k = 0..7$$

$$SLs(i, j, k) \leftarrow empty\ list\ (null)$$

2. Build Similarity List for each colour in Index Structure

- $\{Pass\ by\ all\ colors\ in\ Index\ Structure\}$

$$\forall\, i = 0..512, \qquad j = 0..7, \qquad k = 0..7$$

$$Call\ Find\_Color(i, j, k, \mathbf{out}\ Color1)$$

  - $\{Pass\ by\ all\ colors\ that\ located\ within\ certain\ range\ from\ (i, j, k)\ color\}$

$$\forall\ ii = i - R\,..\,i + R, \qquad jj = 0..7, \qquad kk = 0..7$$

$$(*\ if\ i - R < 0\ then\ i - R = 0, \qquad if\ i + R > 511\ then\ i + R = 511,$$
$$R\ depends\ on\ LUV\ Color\ distance\ (Threshold)\ *)$$

$$Call\ Find\_Color(ii, jj, kk, \mathbf{out}\ Color2)$$

    - { Check Similarity of all colours in the range with the index structure colour, then add colours to the SL that are less than certain threshold}

$$If\ Distance\_LUV\ (Color1, Color2) < Threshold\ then$$

$$SLs(i, j, k).Add(node\_ref(ii, jj, kk))$$

3. Return (SLs)

END.

One can note from Algorithm 6.4 that the searching for similar colours does not include all colour space; instead, it will search within specific range in the whole colour space, as shown in Figure 6.21. This range is from i-R to i+R in the first level of the index structure, where i represents the current index dimension in database index structure. R represents the maximum distance (from current index) that can be used to reach all perceptually similar colours in LUV colour space. R value can be extracted through experiments by checking all colours in the whole colour space. The similarity between each two colours must be less than or equal the certain colour threshold (distance). The R value is extracted for several colour thresholds as shown in Table 6.8.

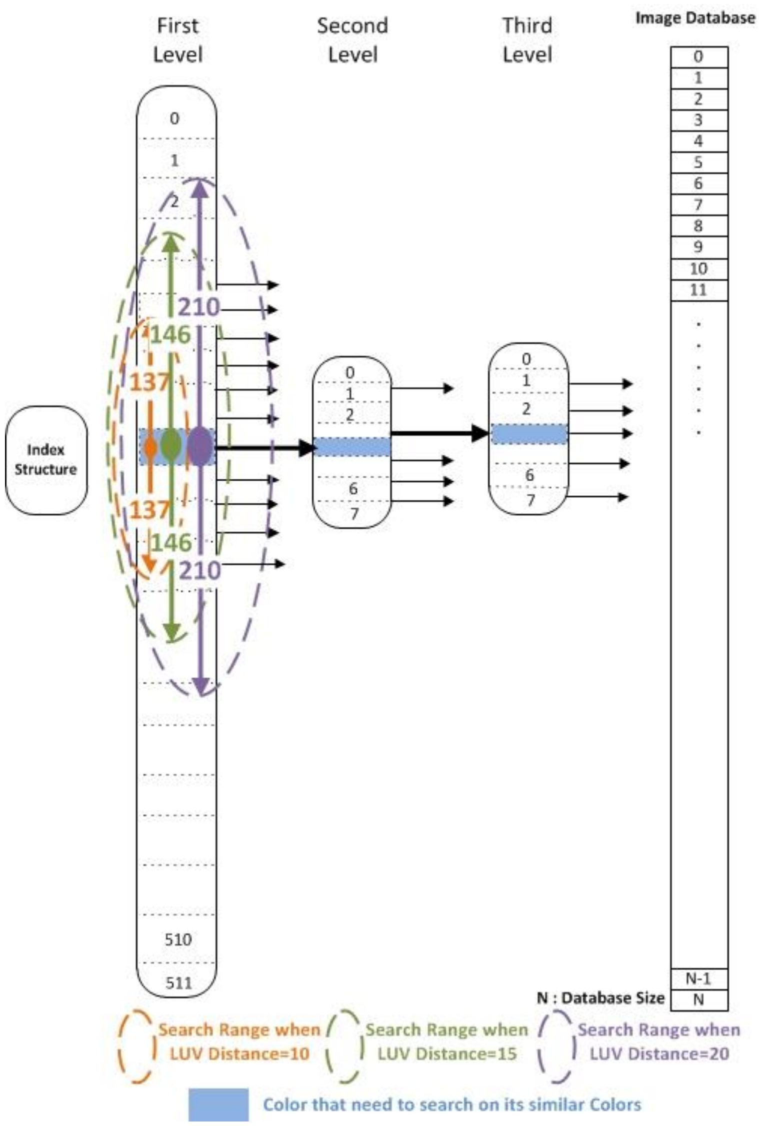


Figure 6.21: Illustration of Search Range when different LUV Colour Distances are applied.

Table 6.8: *R values of Different Colour Thresholds and the corresponding theoretical Percentage of the visited colours in RGB space*

Colour Threshold	R value	Percentage of visited colours in Colour Space
10	137	53.5%
15	146	57.0%
20	210	82.0%
25	219	85.5%

Theoretical percentage of visited colours according to the whole RGB colour space can be computed by the following equation:

$$Percentage\ of\ Visited\ Colors\ in\ the\ Space(R) = \frac{R * 2}{512} * 100\% \qquad (6.2)$$

The word “theoretical” is called to this percentage because of not all colours in this range will be actually visited depending on LUV colour distance; this percentage is a maximum percentage. This percentage is different from percentage of visited images in the database (Reduced Search Space (RSS)) because each colour in the space has different number of images in the database. This percentage is expensive compared to that of the first proposed RGB indexing methods that its search percentage is less than 1% in the all cases of CTV but this low percentage is achieved but with low accuracy of image retrieval process. Additionally, the visited colours percentage is performed in the offline phase (during building an index structure) in contrast to that of first proposed indexing method that is achieved in online phase.

6.3.2 Searching Process

In this process, a query is required to find its similar images in the database. Searching process includes the following steps:

1. For each DC in the query image, find database images that have similar in both colour and percentage; this is by reaching to the single node in the index structure and in turn to the database images that are associated with this node. In other words, it is the same retrieving process of first proposed indexing

method but the difference between them is the new method reaches to single node only where there is no colour tolerance value.

2. In each single node (of point 1) that corresponds to each of the DC in the query, there is a similarity list (SL) that will be used to reach to all similar colours in LUV colour space. It contains references to all nodes (colours) in the index structure and thus to all images that contain similar colours of the query image, as depicted in Figure 6.20.

3. Merging images that resulted from each DC of the query image to produce search space of the query, it represents reduced search space.

4. Calculate dissimilarity distance between query and all images in the RSS and then rank them accordingly.

In step 1, images of the same query DCs will be reached. In step 2, all images that have perceptually similar colours to the query DCs will be collected to produce RSS. It is worth mentioning, step 1 and 2 also filter the images that have different colour percentages using B+-tree that exist in each node in the index structure. This will help to reduce the search space that will emerge in step 3. Dissimilarity distances will be computed for specific descriptor to all images in the RSS to obtain the most similar images to the query. Summary of this process is illustrated in Figure 6.22.

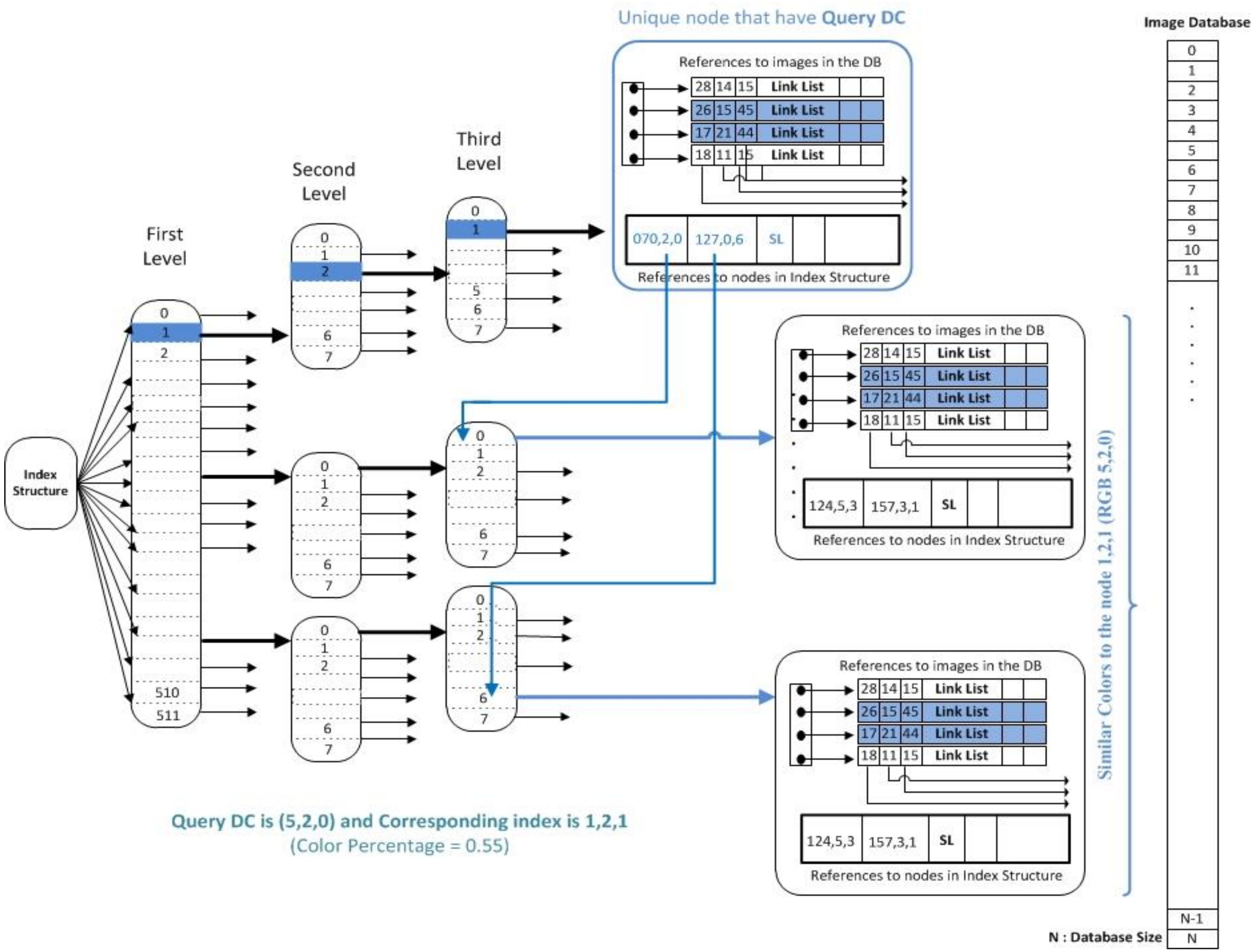


Figure 6.22: Image Retrieval Process using second proposed LUV indexing.

One of the important characteristics of the proposed index structure (that is used in both RGB and LUV indexing methods) is that it is dynamic to the database updating process (insertion and deletion). An insertion or deletion of an image from the database will effect on the link list inside B+-tree structure whereas the entire index structure is kept unchanged. In another side, K-means clustering and K-means with B+-tree need to construct the index structure when updating database because cluster centroids must be recomputed. The most important characteristic of the proposed LUV colour indexing method is that it does not have an approximation neither in matching the query DC with database colours (as in KM and KMB) nor in finding nearest similar colours to the query DC (as in the first proposed RGB indexing method). The proposed index structure includes all perceptually similar colours to the certain colour (node) in similarity list (SL) of this node. That means there is no approximation because only the similar colours will be included in the SL of this colour (node).

6.3.3 Experimental Evaluation

In the experiments evaluation, some parameters need to be set. This will be presented in first sub section. In the second sub section, indexing-based retrieval results will be detailed.

A. Experimental Setup

Parameters that are needed to be set in this proposed indexing method are: 1) number of indexed colours; 2) performance metrics that can be used to measure the accuracy and efficiency of the proposed indexing method; and 3) databases that can be used for experiments as well as the current indexing methods that can be compared with.

All parameters are similar to the first proposed RGB indexing method that is illustrated in Section 6.2.4 (A). Any changes to these parameters are mentioned in the text through explanation of the results.

B. LUV Indexing-based Retrieval Performance

Experiments are conducted on two datasets namely the Corel-10K and the Cartoon-11K databases to measure the performance of the proposed LUV indexing method. These datasets are different in terms of image content (colour and variety) as well as their sizes are large enough to fit the objective of designing the indexing methods. The performance can be measured by the time and accuracy. The comparison between the time of the indexing methods and that of sequential search can be achieved by computing SSR+OHR to represent the ratio of the time of indexing methods to sequential method. Accuracy of the indexing methods, using ARR, ANMRR and R(10) also must be compared with that of sequential search to measure the performance of the proposed indexing method in selecting relevant images.

i. Results of Cartoon-11K dataset

Some experiments are conducted on Cartoon-11K dataset to compare the results of different settings of the proposed LUV indexing method on three different colour descriptors. These results are depicted as quantitative and graphical results as below.

Table 6.9: *Evaluation metrics for ColGrm descriptor using sequential search, competing indexing methods and LUV indexing method applied on Cartoon-11K Dataset.*

Colour *ColGrm*	Indexed colour=8		Indexed colour=5		Indexed colour=3		Indexed colour=1	
	P(10)/ ARR/ ANMRR	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**
Sequential Search	0.350/ 0.118/ 0.852				100%			
K-Means Clustering	0.310/ 0.100/ 0.874	45.8%+ 0.70000%	0.320/ 0.102/ 0.872	40.8%+ 0.40000%	0.270/ 0.089/ 0.889	24.1%+ 0.10000%	0.220/ 0.076/ 0.905	14.5%+ 0.06000%
K-Means with B+Tree	**0.350**/ 0.115/ 0.856	76.6%+ 1.30000%	**0.350**/ 0.116/ 0.856	71%+ 1.10000%	0.310/ 0.104/ 0.870	39.8%+ 0.60000%	0.230/ 0.080/ 0.899	26.7%+ 0.40000%
Proposed LUV Indexing, Colour Distance=25	**0.350/ 0.118/ 0.852**	**90.5%+ 1.60000%**	**0.360/ 0.119/ 0.851**	**82.1%+ 1.40000%**	**0.360/ 0.120/ 0.851**	**69.6%+ 1.15000%**	0.360/ 0.117/ 0.855	45.6%+ 0.80000%
Proposed LUV Indexing+ CPF	**0.350/ 0.119/ 0.852**	**84.1%+ 1.40000%**	**0.360/ 0.119/ 0.851**	**72.5%+ 1.20000%**	**0.360/ 0.120/ 0.851**	**57.2%+ 0.80000%**	0.350/ 0.114/ 0.858	31.8%+ 0.50000%
Proposed LUV Indexing, Colour Distance=25,W	**Same as above**		**0.360/ 0.119/ 0.851**	**74.0%+ 1.10000%**	**0.360/ 0.120/ 0.850**	**56.7%+ 0.90000%**	0.350/ 0.116/ 0.854	23.6%+ 0.40000%

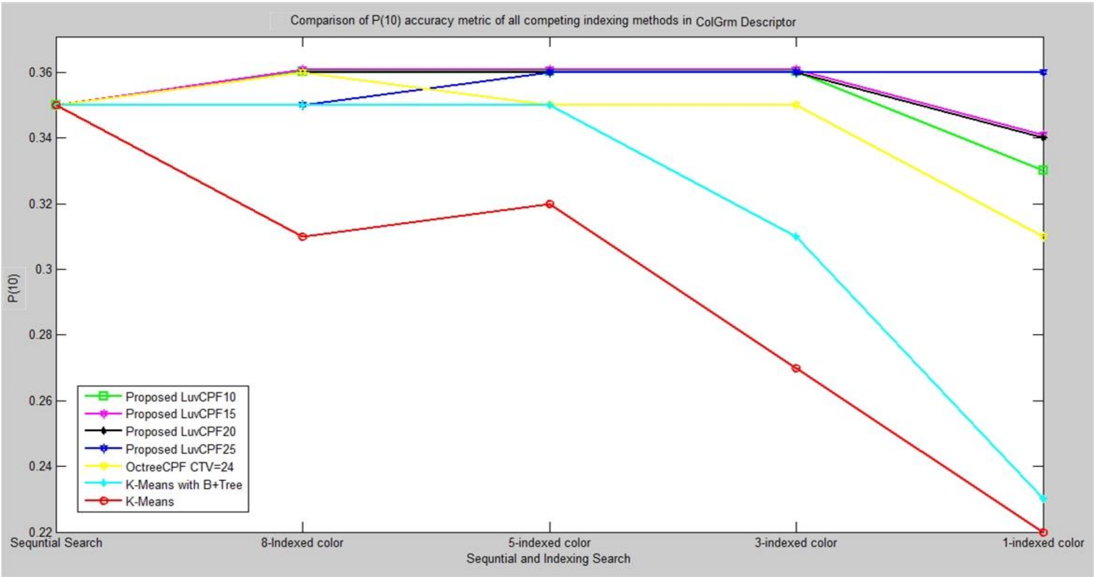
Proposed LUV Indexing+ CPF			**0.360/ 0.120/ 0.850**	**68.6%+ 1.00000%**	**0.360/ 0.120/ 0.850**	**41.0%+ 0.70000%**	**0.360/ 0.119/ 0.850**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20	**0.360/ 0.120/ 0.850**	**78.0%+ 1.30000%**	**0.360/ 0.121/ 0.850**	**68.0%+ 1.10000%**	**0.360/ 0.121/ 0.849**	**54.4%+ 0.80000%**	0.340/ 0.114/ 0.859	31.4%+ 0.50000%
Proposed LUV Indexing+ CPF	**0.360/ 0.121/ 0.849**	**67.9%+ 1.10000%**	**0.360/ 0.123/ 0.848**	**55.6%+ 0.88000%**	**0.360/ 0.123/ 0.847**	**40.7%+ 0.65000%**	0.340/ 0.112/ 0.860	19.1%+ 0.20000%
Proposed LUV Indexing, Colour Distance=20,W	**Same as above**		**0.360/ 0.122/ 0.848**	**57.2%+ 0.89000%**	**0.360/ 0.122/ 0.848**	**42.4%+ 0.75000%**	0.350/ 0.112/ 0.860	16.6%+ 0.16000%
Proposed LUV Indexing+ CPF			**0.360/ 0.123/ 0.847**	**51.0%+ 0.79000%**	**0.360/ 0.124/ 0.844**	**36.7%+ 0.58000%**	0.360/ 0.115/ 0.856	13.1%+ 0.11000%
Proposed LUV Indexing, Colour Distance=15	**0.360/ 0.122/ 0.848**	**59.4%+ 0.91000%**	**0.360/ 0.122/ 0.848**	**51.4%+ 0.78000%**	**0.360/ 0.122/ 0.847**	**41.6%+ 0.73000%**	0.340/ 0.114/ 0.860	25.8%+ 0.27000%
Proposed LUV Indexing+ CPF	**0.360/ 0.124/ 0.846**	**47.9%+ 0.7700%**	**0.360/ 0.124/ 0.845**	**39.3%+ 0.61000%**	**0.360/ 0.125/ 0.844**	**29.5%+ 0.48000%**	0.340/ 0.112/ 0.862	15.9%+ 0.13000%
Proposed LUV Indexing, Colour Distance=15,W	**Same as above**		**0.360/ 0.122/ 0.847**	**37.4%+ 0.61000%**	**0.360/ 0.121/ 0.849**	**26.7%+ 0.28000%**	0.350/ 0.103/ 0.870	10.9%+ 0.08000%

Proposed LUV Indexing+ CPF			0.360/ 0.125/ 0.843	31.8%+ 0.53000%	0.370/ 0.124/ 0.846	22.0%+ 0.22000%	0.360/ 0.106/ 0.866	8.2%+ 0.07000%
Proposed LUV Indexing, Colour Distance=10	0.360/ 0.120/ 0.849	37.1%+ 0.57000%	0.360/ 0.121/ 0.849	33.0%+ 0.55000%	0.360/ 0.117/ 0.854	28.2%+ 0.30000%	0.330/ 0.107/ 0.867	19.8%+ 0.23000%
Proposed LUV Indexing+ CPF	0.360/ 0.123/ 0.846	27.1%+ 0.29000%	0.360/ 0.123/ 0.846	23.2%+ 0.25000%	0.360/ 0.119/ 0.850	19.0%+ 0.21000%	0.330/ 0.103/ 0.870	12.7%+ 0.14000%
Proposed LUV Indexing, Colour Distance=10,W	Same as above		0.360/ 0.121/ 0.851	18.2%+ 0.21000%	0.360/ 0.114/ 0.858	13.1%+ 0.15000%	0.320/ 0.087/ 0.889	5.6%+ 0.02000%
Proposed LUV Indexing+ CPF			0.370/ 0.123/ 0.847	14.1%+ 0.16000%	0.370/ 0.117/ 0.853	9.9%+ 0.09000%	0.330/ 0.089/ 0.888	3.9%+ 0.00100%
Proposed RGB Octree Indexing CTV =24	0.360/ 0.122/ 0.848	57.3%+ 0.90000%	0.360/ 0.122/ 0.848	49.8%+ 0.08000%	0.360/ 0.120/ 0.850	40.5%+ 0.00700%	0.320/ 0.097/ 0.879	24.5%+ 0.00080%
Proposed Octree+CPF CTV =24	0.360/ 0.122/ 0.848	47.2%+ 0.80000%	0.350/ 0.122/ 0.848	39.2%+ 0.07000%	0.350/ 0.121/ 0.850	29.7%+ 0.00500%	0.310/ 0.093/ 0.883	15.1%+ 0.00060%

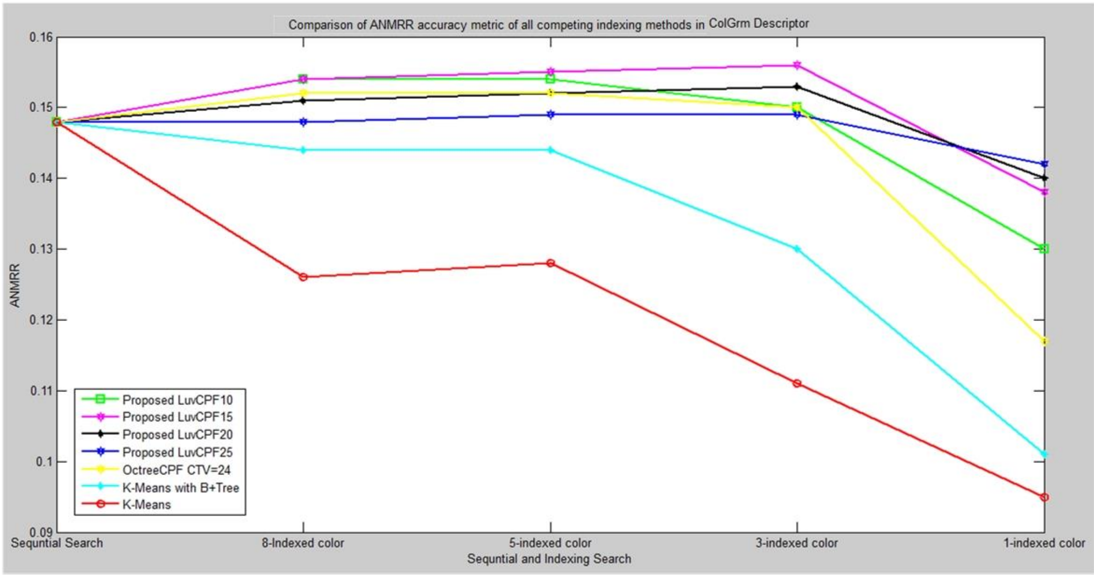
** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for Colour Correlogram Descriptor (colours=27,distance=5) using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, Octree, Octree with Colour Percentage Filtering (CPF) and Proposed LUV Indexing Method) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*

The results of all competing indexing methods that shown in Table 6.10 can be analysed as follows:

1- Accuracy of the results using LUV indexing method is better than those of sequential search in most settings (in four LUV colour distances 25, 20, 15 and 10 as well as in different indexed colours 8, 5, 3). This is because, the good coverage of this indexing to all similar colours in the space; this will lead to fetching of all images that have similar colours to the query DCs where there is no colour approximation at all. The unique case that has accuracy less than that of sequential search is when using 1 colour for indexing. The reason behind this is the single indexed colour may not actually be included in the image object whereas 3 or more colours will increase the chance of obtaining one or more colours of the object. Accuracy of all competing indexing methods is depicted in Figure 6.23. In this figure, one can notice that the proposed LUV indexing method outperformed other competing indexing method (KM and KMB) whereas the first proposed RGB indexing in its best setting can compete LUV accuracy when LUV colour distance equals to 25.



(a) P(10) Accuracy Metric



(b) ANMRR Accuracy Metric

Figure 6.23: Accuracy metrics of the proposed LUV indexing compared with other indexing methods in Correlogram using (a) P(10) and (b) ANMRR.

2- The first proposed RGB indexing method is used in its best setting of colour tolerance value (CTV=24) to compare with second proposed LUV indexing method. The accuracy of the new LUV indexing method (in most settings) is better than the accuracy that is produced from the RGB indexing method (in its best case), as depicted in Figure 6.23. Only one different case can be noticed that the RGB indexing is better in accuracy than LUV indexing which is in ANMRR metric and when the LUV distance is equal to 25. In this case, the large search space of this setting led to dispersion of the search and in turn decrease the accuracy of results whereas in the other cases of small LUV distance (20, 15 and 10) the accuracy of LUV indexing is better than RGB indexing (after excluding the case of 1-indexed colour because it is the worst among all settings of the proposed LUV indexing method).

3- Another type of experiment is applied in this proposed indexing method which is sorting the colours before indexing depending on semantic weights of colours (that proposed in Chapter 4) instead of depending on colour percentages. These colours weights represent the importance of colours in the image; high weight colour means that this is an object colour and low weight colour means that this is a background colour. The idea behind sorting colours depending on their weights is to make an image representative in the index structure more semantic, where the sorting depending on colour percentage may produce some incorrect information. Figure 6.24 presents an example about percentage- and weight-based sorting colours in the indexing process. This example shows the sorted colours of each sorting method. The colours order is very important in retrieval accuracy and efficiency. When using 8-colours indexing, there is no difference between the two sorting methods because of all images' DCs will be used regardless of sorting methods. The effect of colour weights-based sorting appears when not all DCs is used to represent the image (such as using 5, 3 or 1 colour).

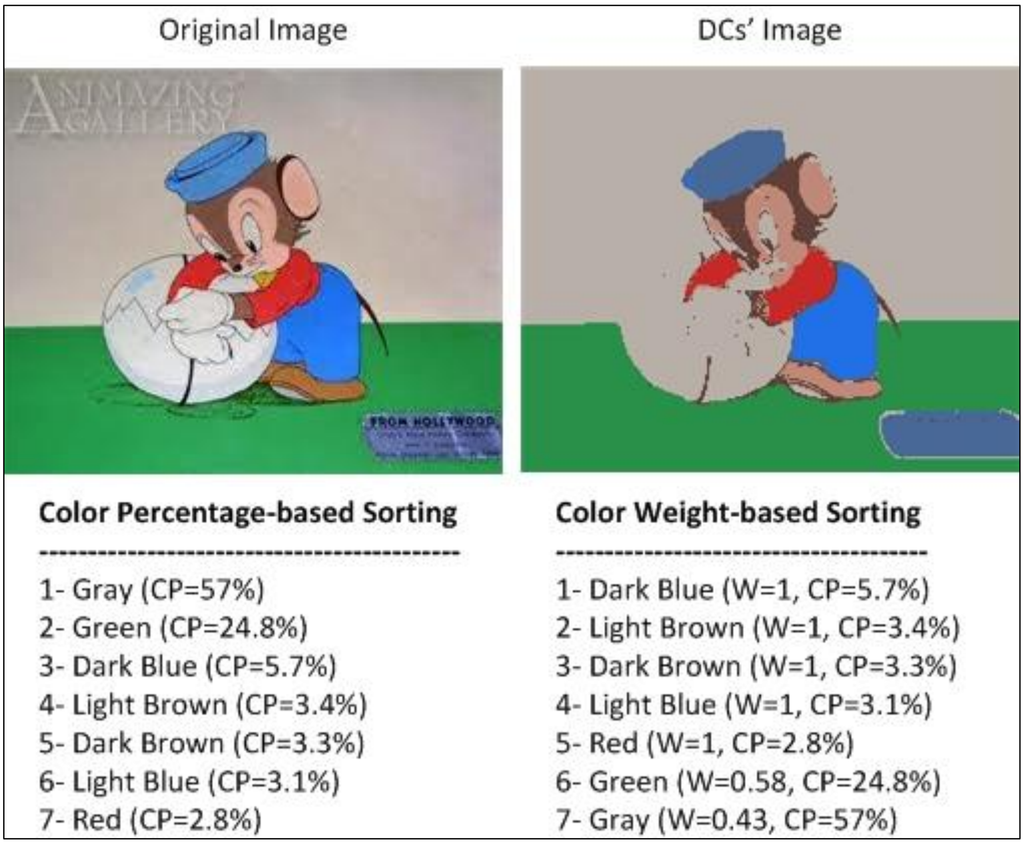


Figure 6.24: An example of Colour Percentage- and Colour Weight-based Sorting in the proposed LUV Indexing.

When using 5-colours indexing method as in Figure 6.24, the colour percentage-based sorting will use first 5-colours in the sorted list to represent the image (in other words, the image will be added in the lists of each of these 5-colours that are grey, green, dark blue, light brown, dark brown. That means that this image contains all these colours). First two colours (grey and green) in the list represent the background of the image where this information is incorrect because the colours grey and green did not reflect the object of the image and some important colours (colours of object) will be ignored such as red, dark brown and light blue. On the other flip, when 5 colours using colour weight-based sorting are indexed (as shown in Figure

6.24); first 5-colours that are sorted by their weights (which are computed according to their affiliation to the image object and border) will be considered in the proposed LUV indexing method for pointing out to the image. These 5-colours are indeed represented the object of the image and the other two background colours will be ignored. Therefore, the accuracy and efficiency (RSS) is increased using weight-based sorting method; as shown in Figure 6.25 and Figure 6.26 (b) respectively. One case only showed that percentage-based method is better than weight-based method; it is when LUV distance equals 10 and when the number of indexed colours is small (3 and 1 indexed colour). This is because when colour similarity range is small (10 and 15) this will lead to desertion of some perceptually similar colours as well as the number of colours is small to represent the object. According to this accuracy degradation, the value 10 for LUV distance is selected as last testing values in the experimental evaluation.

4- Search Space Ratio (SSR) of the proposed LUV indexing method is ranged from high search space (the worst) to very low search space (the best). The worst case occurs when LUV colour distance equals to 25 (it is similar to KMB) where both of them have large SSR. The medium case occurs when distance equals to 20 (it lower than KMB and higher than KM). The lower case occurs when LUV distance equals to 15 (it similar to K-means). The very low search space (the best) case (lower than K-means) occurs when the distance equals to 10; as depicted in Figure 6.26 (a) for SSR. According to these different search space ratios of different settings of LUV indexing method, the accuracy of it is higher than all other indexing methods.

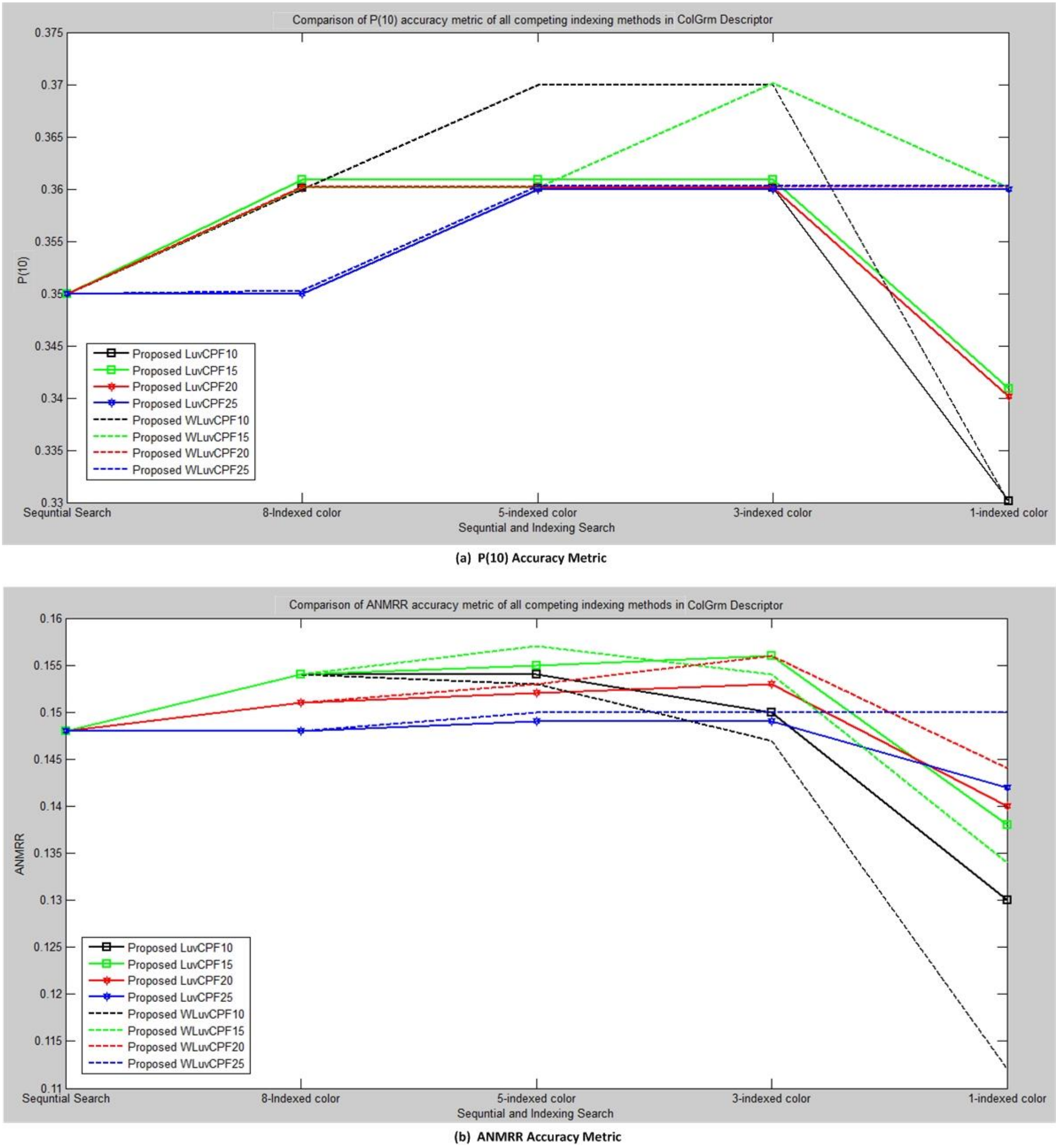
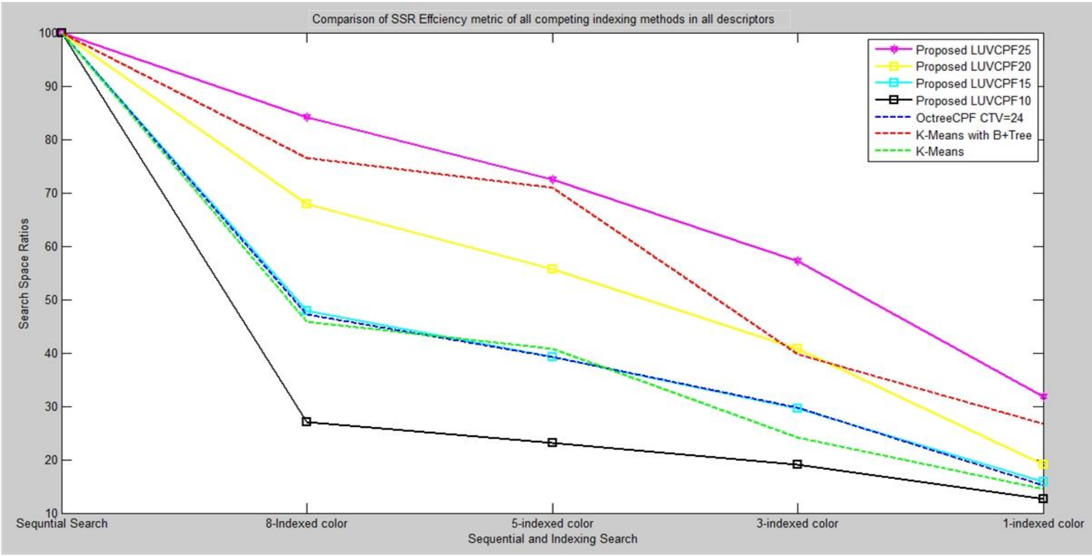
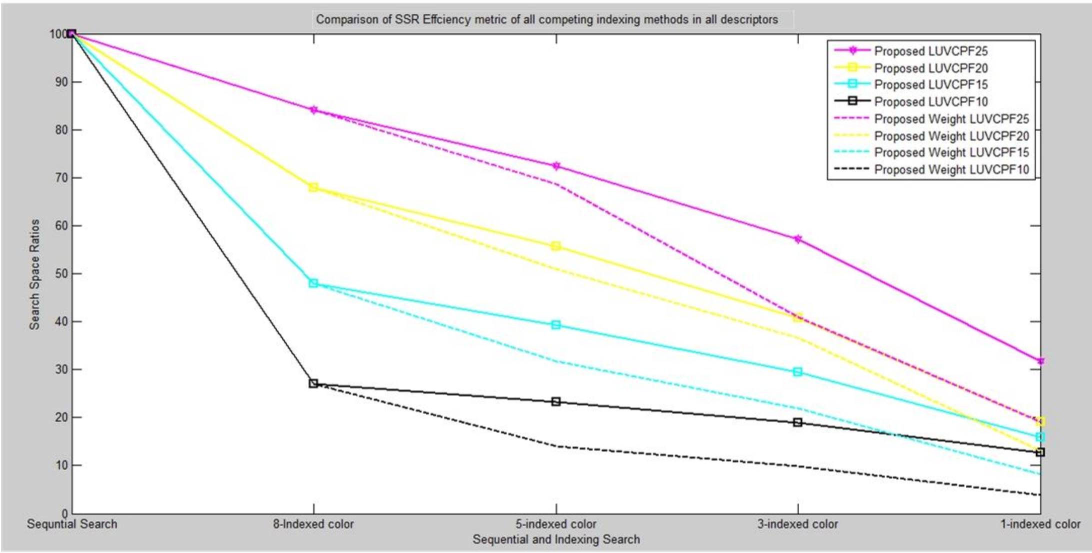


Figure 6.25: Comparison between colour percentage-based indexing method and Colour weight-based indexing method (WLuvCPF indexing).



(a) SSR Efficiency metric for all competing descriptors



(b) Comparison of SSR between Percentage and Weight-based indexing methods

Figure 6.26: SSR comparison for (a) LUV indexing in different settings and (b) CP- and CW-based sorting methods of LUV indexing.

Autocorrelogram descriptor behaviour is also checked in the proposed LUV indexing method (in different settings) with other competing indexing methods in Table 6.11. In this descriptor, the proposed indexing method outperforms the sequential search and all other indexing methods in accuracy.

Table 6.10: *Evaluation metrics for Autocorrelogram descriptor using sequential search, competing indexing methods and LUV indexing applied on Cartoon-11K Dataset.*

AutoColGrm	Indexed colour=8		Indexed colour=5		Indexed colour=3		Indexed colour=1	
	P(10)/ ARR/ ANMRR	SSR+ OHR	ARR/ ANMRR / P(10)	SSR+ OHR	ARR/ ANMRR/ P(10)	SSR+ OHR	ARR/ ANMRR/ P(10)	SSR+ OHR
Sequential Search	0.290/ 0.094/ 0.880				100%			
K-Means Clustering	0.260/ 0.085/ 0.893	45.8%+ 0.70000%	0.270/ 0.084/ 0.894	40.8%+ 0.40000%	0.230/ 0.077/ 0.903	24.1%+ 0.10000%	0.180/ 0.068/ 0.915	14.5%+ 0.06000%
K-Means with B+Tree	0.280/ 0.093/ 0.883	76.6%+ 1.30000%	0.280/ 0.093/ 0.882	71%+ 1.10000%	0.260/ 0.089/ 0.889	39.8%+ 0.60000%	0.190/ 0.072/ 0.912	26.7%+ 0.40000%
Proposed LUV Indexing, Colour Distance=25	**0.290/ 0.095/ 0.880**	**90.5%+ 1.60000%**	**0.290/ 0.096/ 0.879**	**82.1%+ 1.40000%**	**0.290/ 0.097/ 0.878**	**69.6%+ 1.15000%**	**0.290/ 0.096/ 0.880**	**45.6%+ 0.80000%**
Proposed LUV Indexing+ CPF	**0.290/ 0.096/ 0.879**	**84.1%+ 1.40000%**	**0.290/ 0.096/ 0.879**	**72.5%+ 1.20000%**	0.290/ 0.098/ 0.878	**57.2%+ 0.80000%**	**0.290/ 0.096/ 0.881**	**31.8%+ 0.50000%**
Proposed LUV Indexing, Colour Distance=25,W	**Same as above**		**0.290/ 0.097/ 0.877**	**74.0%+ 1.10000%**	**0.290/ 0.099/ 0.876**	**56.7%+ 0.90000%**	**0.290/ 0.099/ 0.876**	**23.6%+ 0.40000%**

Proposed LUV Indexing+ CPF			**0.290/ 0.098/ 0.876**	**68.6%+ 1.00000%**	0.290/ 0.099/ 0.877	**41.0%+ 0.70000%**	**0.300/ 0.103/ 0.872**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20	**0.290/ 0.097/ 0.878**	**78.0%+ 1.30000%**	**0.290/ 0.098/ 0.877**	**68.0%+ 1.10000%**	**0.290/ 0.099/ 0.876**	**54.4%+ 0.80000%**	**0.290/ 0.095/ 0.881**	**31.4%+ 0.50000%**
Proposed LUV Indexing+ CPF	**0.290/ 0.098/ 0.876**	**67.9%+ 1.10000%**	**0.290/ 0.099/ 0.875**	**55.6%+ 0.88000%**	**0.290/ 0.102/ 0.873**	**40.7%+ 0.65000%**	**0.290/ 0.097/ 0.878**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20,W	**Same as above**		**0.290/ 0.100/ 0.875**	**57.2%+ 0.89000%**	**0.300/ 0.101/ 0.873**	**42.4%+ 0.75000%**	**0.300/ 0.097/ 0.879**	**16.6%+ 0.16000%**
Proposed LUV Indexing+ CPF			**0.290/ 0.101/ 0.873**	**51.0%+ 0.79000%**	**0.300/ 0.103/ 0.871**	**36.7%+ 0.58000**	**0.310/ 0.101/ 0.874**	**13.1%+ 0.11000%**
Proposed LUV Indexing, Colour Distance=15	**0.290/ 0.099/ 0.875**	**59.4%+ 0.91000%**	**0.290/ 0.101/ 0.874**	**51.4%+ 0.78000%**	**0.300/ 0.102/ 0.873**	**41.6%+ 0.73000%**	0.290/ 0.095/ 0.881	25.8%+ 0.27000%
Proposed LUV Indexing+ CPF	**0.300/ 0.103/ 0.873**	**47.9%+ 0.77000%**	**0.300/ 0.104/ 0.871**	**39.3%+ 0.61000%**	**0.300/ 0.105/ 0.870**	**29.5%+ 0.48000%**	0.290/ 0.098/ 0.879	15.9%+ 0.13000%
Proposed LUV Indexing, Colour Distance=15,W	**Same as above**		**0.300/ 0.103/ 0.871**	**37.4%+ 0.61000%**	**0.300/ 0.103/ 0.872**	**26.7%+ 0.28000%**	0.300/ 0.092/ 0.884	10.9%+ 0.08000%

Proposed LUV Indexing+ CPF			**0.300/ 0.106/ 0.868**	**31.8%+ 0.53000%**	**0.300/ 0.106/ 0.868**	**22.0%+ 0.22000%**	**0.310/ 0.094/ 0.880**	**8.2%+ 0.07000%**
Proposed LUV Indexing, Colour Distance=10	**0.300/ 0.101/ 0.874**	**37.1%+ 0.57000%**	**0.300/ 0.101/ 0.873**	**33.0%+ 0.55000%**	**0.300/ 0.099/ 0.877**	**28.2%+ 0.30000%**	0.280/ 0.091/ 0.886	19.8%+ 0.23000
Proposed LUV Indexing+ CPF	**0.300/ 0.103/ 0.871**	**27.1%+ 0.29000%**	**0.300/ 0.104/ 0.870**	**23.2%+ 0.25000%**	**0.300/ 0.103/ 0.872**	**19.0%+ 0.21000%**	0.280/ 0.093/ 0.884	12.7%+ 0.14000%
Proposed LUV Indexing, Colour Distance=10,W	**Same as above**		**0.310/ 0.103/ 0.871**	**18.2%+ 0.21000%**	**0.300/ 0.101/ 0.875**	**13.1%+ 0.15000%**	0.270/ 0.080/ 0.899	5.6%+ 0.02000%
Proposed LUV Indexing+ CPF			**0.310/ 0.107/ 0.867**	**14.1%+ 0.16000%**	**0.320/ 0.105/ 0.869**	**9.9%+ 0.09000%**	0.290/ 0.083/ 0.896	3.9%+ 0.00100%
Proposed RGB *Octree* Indexing CTV =24	**0.290/ 0.101/ 0.874**	**57.3%+ 0.90000%**	**0.290/ 0.102/ 0.873**	**49.8%+ 0.08000%**	**0.300/ 0.101/ 0.874**	**40.5%+ 0.00700%**	0.270/ 0.082/ 0.897	24.5%+ 0.00080%
Proposed *Octree*+CPF CTV =24	**0.290/ 0.101/ 0.874**	**47.2%+ 0.80000%**	**0.300/ 0.103/ 0.873**	**39.2%+ 0.07000%**	**0.300/ 0.102/ 0.873**	**29.7%+ 0.00500%**	0.270/ 0.081/ 0.897	15.1%+ 0.00060%

** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for Colour Autocorrelogram Descriptor (colours=27,distance=5) using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, proposed Octree CTV=24 and proposed Octree with Colour Percentage Filtering (CPF) CTV=24) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*

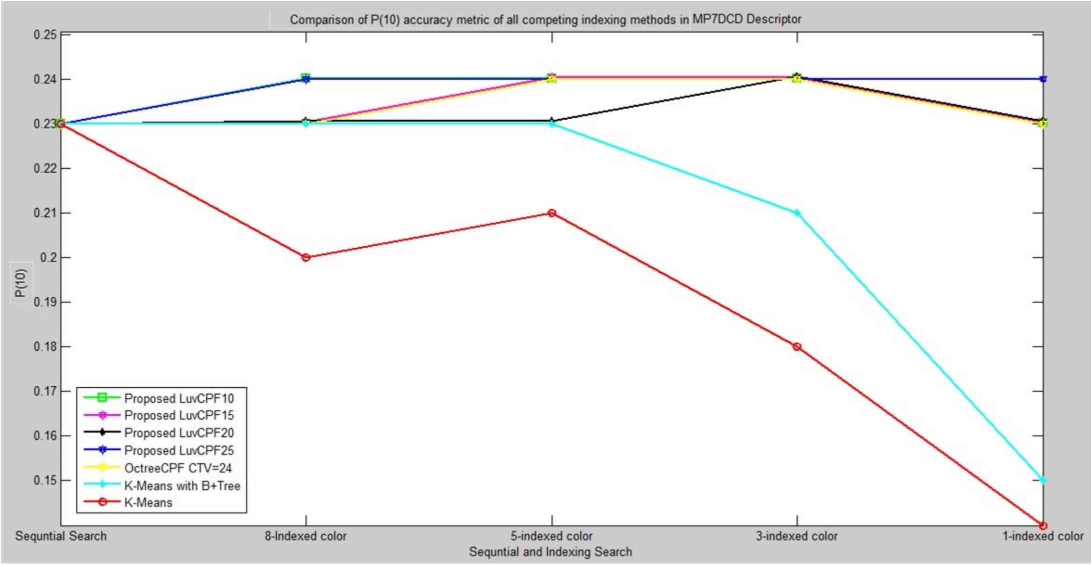
Results for MPEG-7 DCD using LUV indexing methods also are depicted in Table 6.12, Figure 6.27 and Figure 6.28.

Table 6.11: *Evaluation metrics for MPEG-7 DCD using sequential search and all competing indexing methods including LUV indexing applied on Cartoon-11K Dataset.*

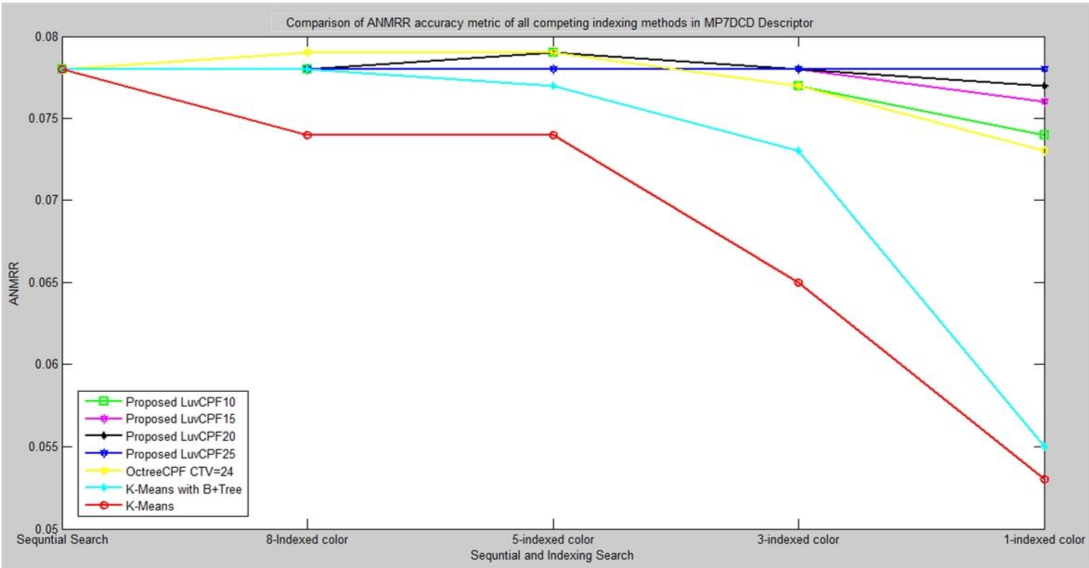
MPEG-7 DCD	**Indexed colour=8**		**Indexed colour=5**		**Indexed colour=3**		**Indexed colour=1**	
	P(10)/ ARR/ ANMRR	**SSR+ OHR**	**ARR/ ANMRR / P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**
Sequential Search	0.230/ 0.060/ 0.922				100%			
K-Means Clustering	0.200/ 0.057/ 0.926	45.8%+ 0.70000%	0.210/ 0.059/ 0.926	40.8%+ 0.40000%	0.180/ 0.051/ 0.935	24.1%+ 0.10000%	0.140/ 0.041/ 0.947	14.5%+ 0.06000%
K-Means with B+Tree	**0.230/** 0.059/ **0.922**	76.6%+ 1.30000%	**0.230/** 0.059/ 0.923	71%+ 1.10000%	0.210/ 0.056/ 0.927	39.8%+ 0.60000%	0.150/ 0.042/ 0.945	26.7%+ 0.40000%
Proposed LUV Indexing, Colour Distance=25	**0.230/ 0.060/ 0.922**	**90.5%+ 1.60000%**	**0.240/ 0.060/ 0.922**	**82.1%+ 1.40000%**	**0.240/ 0.061/ 0.922**	**69.6%+ 1.15000%**	**0.240/ 0.060/ 0.922**	**45.6%+ 0.80000%**
Proposed LUV Indexing+ CPF	**0.240/ 0.060/ 0.922**	**84.1%+ 1.40000%**	**0.240/ 0.061/ 0.922**	**72.5%+ 1.20000%**	**0.240/ 0.060/ 0.922**	**57.2%+ 0.80000%**	**0.240/ 0.059/ 0.922**	**31.8%+ 0.50000%**

Proposed LUV Indexing, Colour Distance=25,W	**Same as above**		**0.240/ 0.062/ 0.919**	**74.0%+ 1.10000%**	**0.240/ 0.064/ 0.917**	**56.7%+ 0.90000%**	**0.230/ 0.067/ 0.913**	23.6%+ 0.40000%
Proposed LUV Indexing+ CPF			**0.240/ 0.062/ 0.919**	**68.6%+ 1.0000%**	**0.240/ 0.064/ 0.917**	**41.0%+ 0.70000%**	**0.240/ 0.068/ 0.911**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20	**0.240/ 0.060/ 0.922**	**78.0%+ 1.30000%**	**0.240/ 0.060/ 0.922**	**68.0%+ 1.10000%**	**0.240/ 0.060/ 0.922**	**54.4%+ 0.80000%**	**0.230/ 0.060/ 0.923**	**31.4%+ 0.50000%**
Proposed LUV Indexing+ CPF	**0.230/ 0.060/ 0.922**	**67.9%+ 1.10000%**	**0.230/ 0.060/ 0.921**	**55.6%+ 0.88000%**	**0.240/ 0.060/ 0.922**	**40.7%+ 0.65000%**	**0.230/ 0.059/ 0.923**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20,W	**Same as above**		**0.240/ 0.064/ 0.916**	**57.2%+ 0.89000%**	**0.240/ 0.068/ 0.913**	**42.4%+ 0.75000%**	**0.240/ 0.068/ 0.912**	**16.6%+ 0.16000%**
Proposed LUV Indexing+ CPF			**0.240/ 0.064/ 0.916**	**51.0%+ 0.79000%**	**0.240/ 0.068/ 0.913**	**36.7%+ 0.58000%**	**0.240/ 0.068/ 0.911**	**13.1%+ 0.11000%**
Proposed LUV Indexing, Colour Distance=15	**0.240/ 0.060/ 0.922**	**59.4%+ 0.91000%**	**0.240/ 0.060/ 0.922**	**51.4%+ 0.78000%**	**0.240/ 0.060/ 0.922**	**41.6%+ 0.73000%**	0.230/ 0.059/ 0.924	25.8%+ 0.27000%
Proposed LUV Indexing+ CPF	**0.230/ 0.061/ 0.922**	**47.9%+ 0.77000%**	**0.240/ 0.061/ 0.921**	**39.3%+ 0.61000%**	**0.240/ 0.059/ 0.922**	**29.5%+ 0.48000%**	0.230/ 0.058/ 0.924	15.9%+ 0.13000%

Proposed LUV Indexing, Colour Distance=15,W	Same as above		0.240/ 0.067/ 0.913	37.4%+ 0.61000%	0.240/ 0.068/ 0.911	26.7%+ 0.28000%	0.230/ 0.066/ 0.915	10.9%+ 0.08000%
Proposed LUV Indexing+ CPF			0.240/ 0.067/ 0.913	31.8%+ 0.53000%	0.240/ 0.068/ 0.911	22.0%+ 0.22000%	0.230/ 0.067/ 0.913	8.2%+ 0.07000%
Proposed LUV Indexing, Colour Distance=10	0.240/ 0.061/ 0.922	37.1%+ 0.57000%	0.240/ 0.061/ 0.921	33.0%+ 0.55000%	0.240/ 0.059/ 0.923	28.2%+ 0.30000%	0.230/ 0.057/ 0.926	19.8%+ 0.23000%
Proposed LUV Indexing+ CPF	0.240/ 0.060/ 0.922	27.1%+ 0.29000%	0.240/ 0.060/ 0.921	23.2%+ 0.25000%	0.240/ 0.059/ 0.923	19.0%+ 0.21000%	0.230/ 0.056/ 0.926	12.7%+ 0.14000%
Proposed LUV Indexing, Colour Distance=10,W	Same as above		0.240/ 0.070/ 0.909	18.2%+ 0.21000%	0.240/ 0.072/ 0.908	13.1%+ 0.15000%	0.230/ 0.062/ 0.919	5.6%+ 0.02000%
Proposed LUV Indexing+ CPF			0.240/ 0.070/ 0.909	14.1%+ 0.16000%	0.240/ 0.073/ 0.906	9.9%+ 0.09000%	0.230/ 0.064/ 0.917	3.9%+ 0.00100%
Proposed RGB *Octree* Indexing CTV =24	0.230/ 0.060/ 0.921	57.3%+ 0.90000%	0.240/ 0.061/ 0.921	49.8%+ 0.08000%	0.240/ 0.059/ 0.922	40.5%+ 0.00700%	0.230/ 0.057/ 0.926	24.5%+ 0.00080%
Proposed *Octree*+CPF CTV =24	0.230/ 0.060/ 0.921	47.2%+ 0.80000%	0.240/ 0.060/ 0.921	39.2%+ 0.07000%	0.240/ 0.059/ 0.923	29.7%+ 0.00500%	0.230/ 0.056/ 0.927	15.1%+ 0.00060%



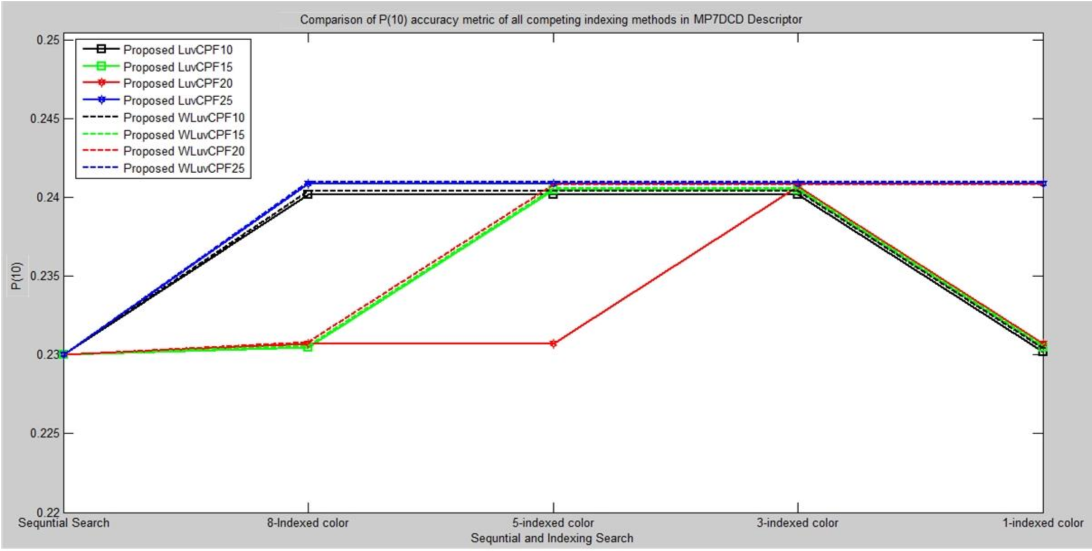
(a) P(10) Accuracy Metric



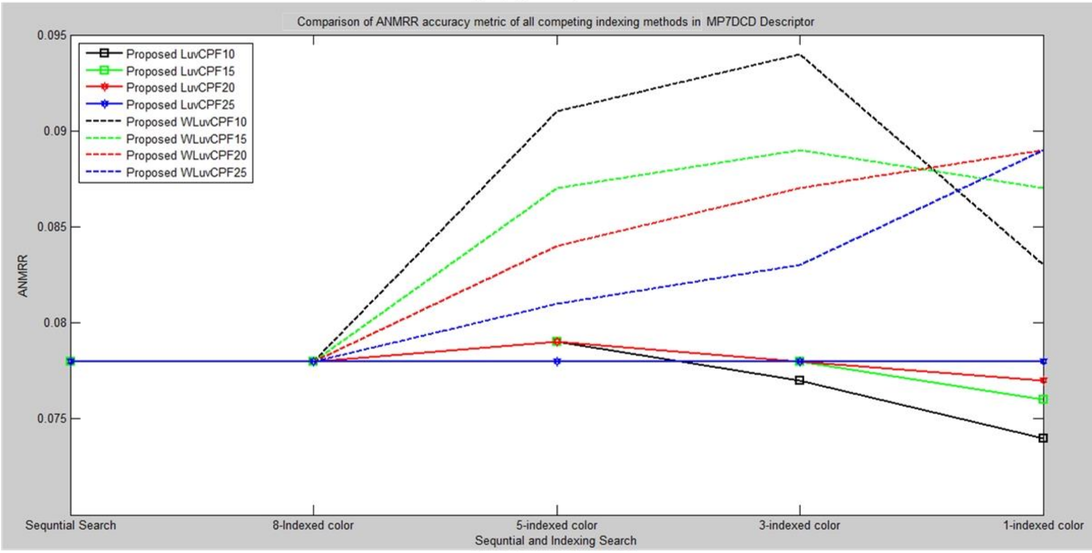
(b) ANMRR Accuracy Metric

Figure 6.27: Accuracy metrics of LUV indexing compared with other indexing methods in MPEG-7 DCD using (a) P(10) and (b) ANMRR.

The performance of the proposed LUV indexing is obvious in MP7DCD over other indexing methods as depicted in Figure 6.27. Additionally, colour weight-based sorting method also surpassed the performance of colour percentage-based sorting method in accuracy as shown in Figure 6.28 as well as in reducing search space as shown in Figure 6.26 (b).



(a) P(10) Accuracy Metric



(b) ANMRR Accuracy Metric

Figure 6.28: Comparison between Colour Percentage- and Colour Weight-based indexing method in MPEG-7 DCD using (a) P(10) and (b) ANMRR.

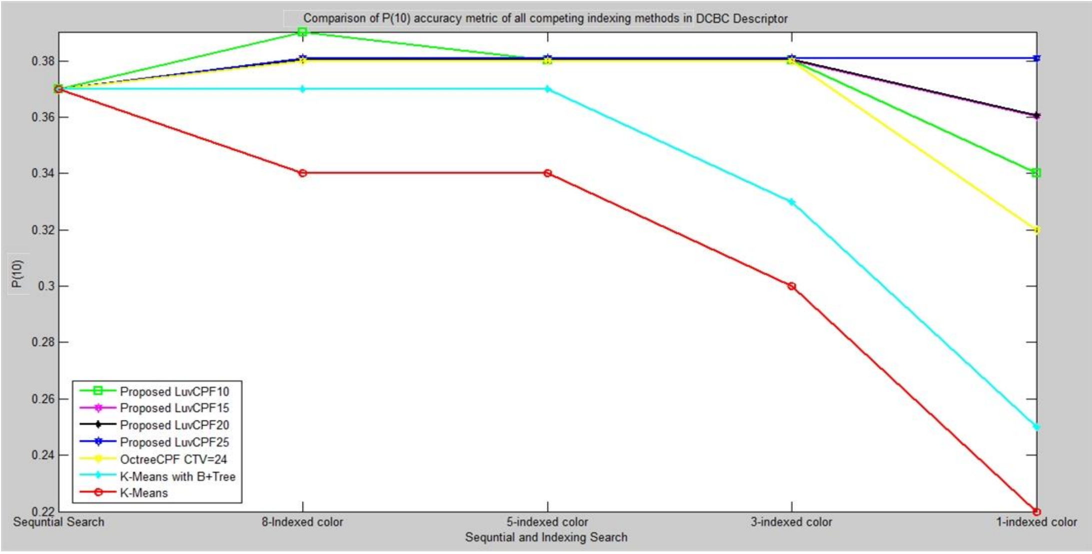
Table 6.12: *Evaluation metrics for DCBC descriptor using sequential search, competing indexing methods and LUV indexing applied on Cartoon-11K Dataset.*

DCBC	Indexed colour=8		Indexed colour=5		Indexed colour=3		Indexed colour=1	
	ARR/ ANMRR/ P(10)	SSR+ OHR	ARR/ ANMRR/ P(10)	SSR+ OHR	ARR/ ANMRR/ P(10)	SSR+ OHR	ARR/ ANMRR/ P(10)	SSR+ OHR
Sequential Search	0.370/ 0.131/ 0.837				100%			
K-Means Clustering	0.340/ 0.111/ 0.862	45.8%+ 0.70000%	0.340/ 0.112/ 0.859	40.8%+ 0.40000%	0.300/ 0.096/ 0.880	24.1%+ 0.10000%	0.220/ 0.078/ 0.903	14.5%+ 0.06000%
K-Means with B+Tree	**0.370/** 0.124/ 0.843	76.6%+ 1.30000%	**0.370/** 0.126/ 0.841	71%+ 1.10000%	0.330/ 0.110/ 0.863	39.8%+ 0.60000%	0.250/ 0.084/ 0.896	26.7%+ 0.40000%
Proposed LUV Indexing, Colour Distance=25	**0.380/ 0.131/ 0.837**	**90.5%+ 1.60000%**	**0.380/ 0.131/ 0.837**	**82.1%+ 1.40000%**	**0.380/ 0.131/ 0.837**	**69.6%+ 1.15000%**	**0.380/ 0.127/ 0.842**	**45.6%+ 0.80000%**
Proposed LUV Indexing+ CPF	**0.380/ 0.131/ 0.837**	**84.1%+ 1.40000%**	**0.380/ 0.131/ 0.837**	**72.5%+ 1.20000%**	**0.380/ 0.132/ 0.836**	**57.2%+ 0.80000%**	**0.380/ 0.124/ 0.846**	**31.8%+ 0.50000%**
Proposed LUV Indexing, Colour Distance=25,W	**Same as above**		**0.380/ 0.132/ 0.836**	**74.0%+ 1.10000%**	**0.380/ 0.134/ 0.834**	**56.7%+ 0.90000%**	**0.370/ 0.128/ 0.841**	**23.6%+ 0.40000%**

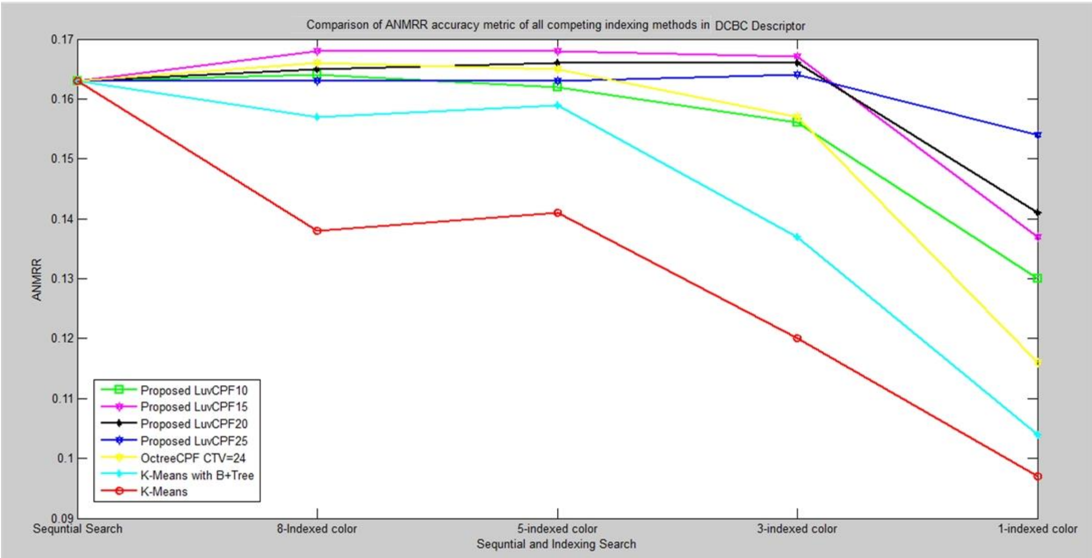
Proposed LUV Indexing+ CPF			**0.380/ 0.133/ 0.835**	**68.6%+ 1.00000%**	**0.380/ 0.134/ 0.833**	**41.0%+ 0.70000%**	**0.370/ 0.128/ 0.841**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20	**0.380/ 0.132/ 0.836**	**78.0%+ 1.30000%**	**0.380/ 0.132/ 0.836**	**68.0%+ 1.10000%**	**0.380/ 0.132/ 0.836**	**54.4%+ 0.80000%**	**0.360/ 0.118/ 0.854**	**31.4%+ 0.50000%**
Proposed LUV Indexing+ CPF	**0.380/ 0.132/ 0.835**	**67.9%+ 1.1000%**	**0.380/ 0.133/ 0.834**	**55.6%+ 0.88000%**	**0.380/ 0.134/ 0.834**	**40.7%+ 0.65000%**	**0.360/ 0.114/ 0.859**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20,W	**Same as above**		**0.380/ 0.133/ 0.833**	**57.2%+ 0.89000%**	**0.390/ 0.136/ 0.831**	**42.4%+ 0.75000%**	**0.370/ 0.122/ 0.849**	**16.6%+ 0.16000%**
Proposed LUV Indexing+ CPF			**0.380/ 0.134/ 0.833**	**51.0%+ 0.79000%**	**0.390/ 0.136/ 0.831**	**36.7%+ 0.58000%**	**0.370/ 0.123/ 0.848**	**13.1%+ 0.11000%**
Proposed LUV Indexing, Colour Distance=15	**0.380/ 0.134/ 0.833**	**59.4%+ 0.91000%**	**0.380/ 0.135/ 0.833**	**51.4%+ 0.78000%**	**0.380/ 0.134/ 0.834**	**41.6%+ 0.73000%**	**0.360/ 0.115/ 0.858**	**25.8%+ 0.27000%**
Proposed LUV Indexing+ CPF	**0.380/ 0.136/ 0.832**	**47.9%+ 0.77000%**	**0.380/ 0.136/ 0.832**	**39.3%+ 0.61000%**	**0.390/ 0.134/ 0.833**	**29.5%+ 0.48000%**	**0.350/ 0.111/ 0.863**	**15.9%+ 0.13000%**
Proposed LUV Indexing, Colour Distance=15,W	**Same as above**		**0.390/ 0.135/ 0.832**	**37.4%+ 0.61000%**	**0.380/ 0.133/ 0.835**	**26.7%+ 0.28000%**	**0.360/ 0.113/ 0.859**	**10.9%+ 0.08000%**

Proposed LUV Indexing+ CPF			**0.390/ 0.136/ 0.831**	**31.8%+ 0.53000%**	**0.380/ 0.134/ 0.834**	**22.0%+ 0.22000%**	**0.370/ 0.114/ 0.859**	**8.2%+ 0.07000%**
Proposed LUV Indexing, Colour Distance=10	**0.390/ 0.130/ 0.837**	**37.1%+ 0.57000%**	**0.380/ 0.129/ 0.839**	**33.0%+ 0.55000%**	**0.380/ 0.125/ 0.845**	**28.2%+ 0.30000%**	**0.350/ 0.108/ 0.866**	**19.8%+ 0.23000%**
Proposed LUV Indexing+ CPF	**0.390/ 0.131/ 0.836**	**27.1%+ 0.29000%**	**0.380/ 0.130/ 0.838**	**23.2%+ 0.25000%**	**0.380/ 0.127/ 0.844**	**19.0%+ 0.21000%**	**0.340/ 0.104/ 0.870**	**12.7%+ 0.14000%**
Proposed LUV Indexing, Colour Distance=10,W	**Same as above**		**0.390/ 0.128/ 0.842**	**18.2%+ 0.21000%**	**0.370/ 0.122/ 0.850**	**13.1%+ 0.15000%**	**0.330/ 0.091/ 0.886**	**5.6%+ 0.02000%**
Proposed LUV Indexing+ CPF			**0.390/ 0.128/ 0.841**	**14.1%+ 0.16000%**	**0.370/ 0.122/ 0.849**	**9.9%+ 0.09000%**	**0.330/ 0.092/ 0.885**	**3.9%+ 0.00100%**
Proposed RGB *Octree* Indexing CTV =24	**0.380/ 0134/ 0.834**	**57.3%+ 0.90000%**	**0.380/ 0.133/ 0.835**	**49.8%+ 0.08000%**	**0.380/** 0.126/ 0.842	**40.5%+ 0.00700%**	0.330/ 0.095/ 0.880	24.5%+ 0.00080%
Proposed *Octree*+CPF CTV =24	**0.380/ 0133/ 0.834**	**47.2%+ 0.80000%**	**0.380/ 0.132/ 0.835**	**39.2%+ 0.07000%**	**0.380/** 0.126/ 0.843	**29.7%+ 0.00500%**	0.320/ 0.092/ 0.884	15.1%+ 0.00060%

** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for Dominant Colour-based Correlogram (DCBC) Descriptor (distance=5) using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, proposed Octree CTV=24 and proposed Octree with Colour Percentage Filtering (CPF) CTV=24) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*



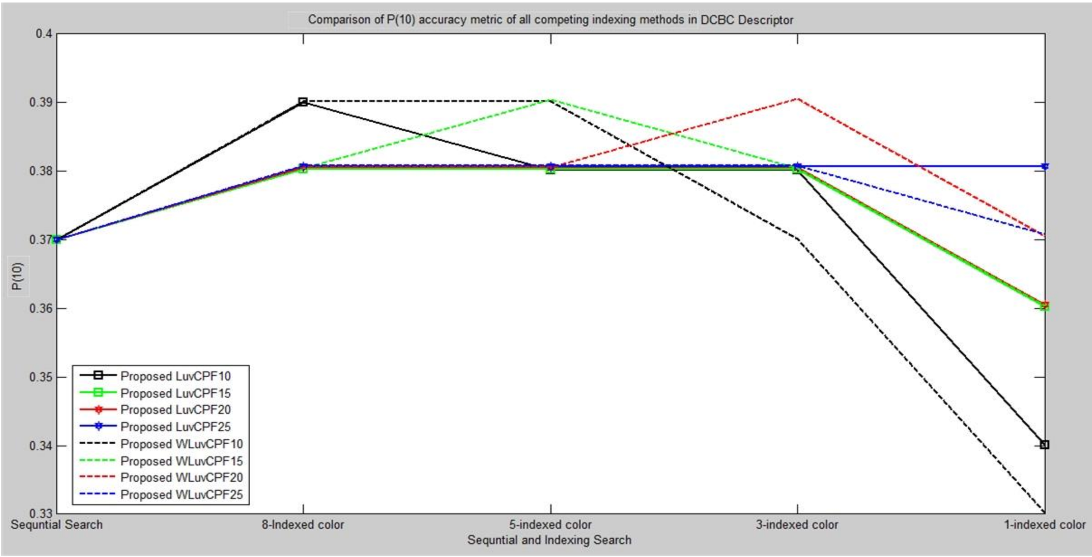
(a) P(10) Accuracy Metric



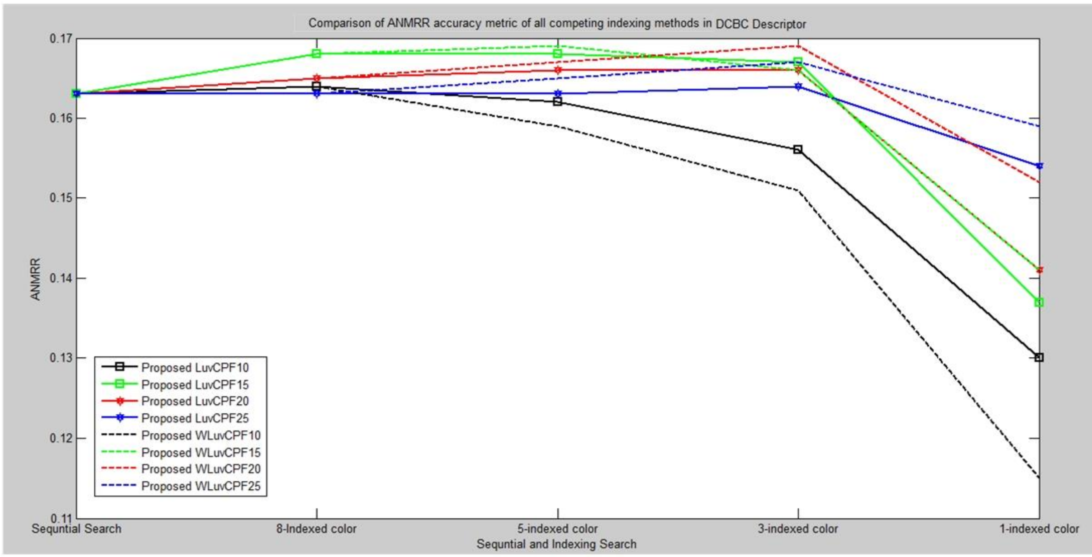
(b) ANMRR Accuracy Metric

Figure 6.29: Accuracy metrics of LUV indexing compared with other indexing methods in DC-based ColGrm using (a) P(10) and (b) ANMRR.

In the DCBC descriptor, the proposed LUV indexing method outperformed other indexing method except for the first proposed RGB indexing method. It surpasses the LUV indexing in some settings, as shown in Fig 6.29. Additionally, it outperforms the sequential search in all settings in P(10) metric and some settings in ANMRR metric. The worst results of proposed LUV indexing are obtained when LUV distance equals to 10 but it is still better than KM and KMB. As for the comparison between weight- and percentage-based sorting methods, weight-based method outperforms the percentage-based method except in the case when 1 colour only is indexed as well as when the LUV distance equals to 10 (the reasons are explained previously), as shown in Figure 6.30.



(a) P(10) Accuracy Metric



(b) ANMRR Accuracy Metric

Figure 6.30: Comparison between Colour Percentage- and Colour Weight-based indexing method in DC-based ColGrm using (a) P(10) and (b) ANMRR.

Table 6.13: *Evaluation metrics for Weighted DC-based ColGrm using sequential search, competing indexing methods and LUV indexing applied on Cartoon-11K Dataset.*

WDCBC	Indexed colour=8		Indexed colour=5		Indexed colour=3		Indexed colour=1	
	ARR/ ANMRR/ P(10)	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**	**ARR/ ANMRR/ P(10)**	**SSR+ OHR**
Sequential Search	0.410/ 0.153/ 0.810				100%			
K-Means Clustering	0.360/ 0.127/ 0.842	45.8%+ 0.70000%	0.380/ 0.130/ 0.838	40.8%+ 0.40000%	0.340/ 0.114/ 0.859	24.1%+ 0.10000%	0.260/ 0.087/ 0.892	14.5%+ 0.06000%
K-Means with B+Tree	0.400/ 0.147/ 0.817	76.6%+ 1.30000%	**0.410/** 0.148/ 0.816	71%+ 1.10000%	0.370/ 0.129/ 0.840	39.8%+ 0.60000%	0.280/ 0.097 0.879	26.7%+ 0.40000%
Proposed LUV Indexing, Colour Distance=25	**0.410/ 0.153/ 0.810**	**90.5%+ 1.60000%**	**0.410/ 0.153/ 0.810**	**82.1%+ 1.40000%**	**0.410/ 0.153/ 0.810**	**69.6%+ 1.15000%**	**0.410/ 0.140/ 0.824**	**45.6%+ 0.80000%**
Proposed LUV Indexing+ CPF	**0.410/ 0.153/ 0.810**	**84.1%+ 1.40000%**	**0.410/ 0.153/ 0.810**	**72.5%+ 1.20000%**	**0.410/ 0.153/ 0.809**	**57.2%+ 0.80000%**	**0.400/ 0.134/ 0.832**	**31.8%+ 0.50000%**
Proposed LUV Indexing, Colour Distance=25,W	**Same as above**		**0.410/ 0.153/ 0.811**	**74.0%+ 1.10000%**	**0.410/ 0.152/ 0.811**	**56.7%+ 0.90000%**	**0.410/ 0.142/ 0.823**	**23.6%+ 0.40000%**

Proposed LUV Indexing+ CPF			**0.410/ 0.153/ 0.811**	**68.6%+ 1.00000%**	**0.410/ 0.152/ 0.811**	**41.0%+ 0.70000%**	**0.410/ 0.142/ 0.823**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20	**0.410/ 0.152/ 0.810**	**78.0%+ 1.30000%**	**0.410/ 0.153/ 0.810**	**68.0%+ 1.10000%**	**0.420/ 0.153/ 0.810**	**54.4%+ 0.80000%**	**0.390/ 0.129/ 0.840**	**31.4%+ 0.50000%**
Proposed LUV Indexing+ CPF	**0.410/ 0.153/ 0.810**	**67.9%+ 1.10000%**	**0.410/ 0.154/ 0.809**	**55.6%+ 0.88000%**	**0.420/ 0.154/ 0.808**	**40.7%+ 0.65000%**	**0.370/ 0.121/ 0.847**	**19.1%+ 0.20000%**
Proposed LUV Indexing, Colour Distance=20,W	**Same as above**		**0.410/ 0.152/ 0.811**	**57.2%+ 0.89000%**	**0.420/ 0.153/ 0.810**	**42.4%+ 0.75000%**	**0.400/ 0.136/ 0.832**	**16.6%+ 0.16000%**
Proposed LUV Indexing+ CPF			**0.410/ 0.153/ 0.810**	**51.0%+ 0.79000%**	**0.420/ 0.153/ 0.809**	**36.7%+ 0.58000%**	**0.400/ 0.137/ 0.831**	**13.1%+ 0.11000%**
Proposed LUV Indexing, Colour Distance=15	**0.420/ 0.154/ 0.809**	**59.4%+ 0.91000%**	**0.420/ 0.153/ 0.809**	**51.4%+ 0.78000%**	**0.420/ 0.151/ 0.813**	**41.6%+ 0.73000%**	**0.380/ 0.124/ 0.845**	**25.8%+ 0.27000%**
Proposed LUV Indexing+ CPF	**0.420/ 0.154/ 0.808**	**47.9%+ 0.77000%**	**0.420/ 0.154/ 0.808**	**39.3%+ 0.61000%**	**0.420/ 0.153/ 0.811**	**29.5%+ 0.48000%**	**0.370/ 0.119/ 0.851**	**15.9%+ 0.13000%**
Proposed LUV Indexing, Colour Distance=15,W	**Same as above**		**0.420/ 0.152/ 0.811**	**37.4%+ 0.61000%**	**0.420/ 0.147/ 0.818**	**26.7%+ 0.28000%**	**0.390/ 0.123/ 0.847**	**10.9%+ 0.08000%**

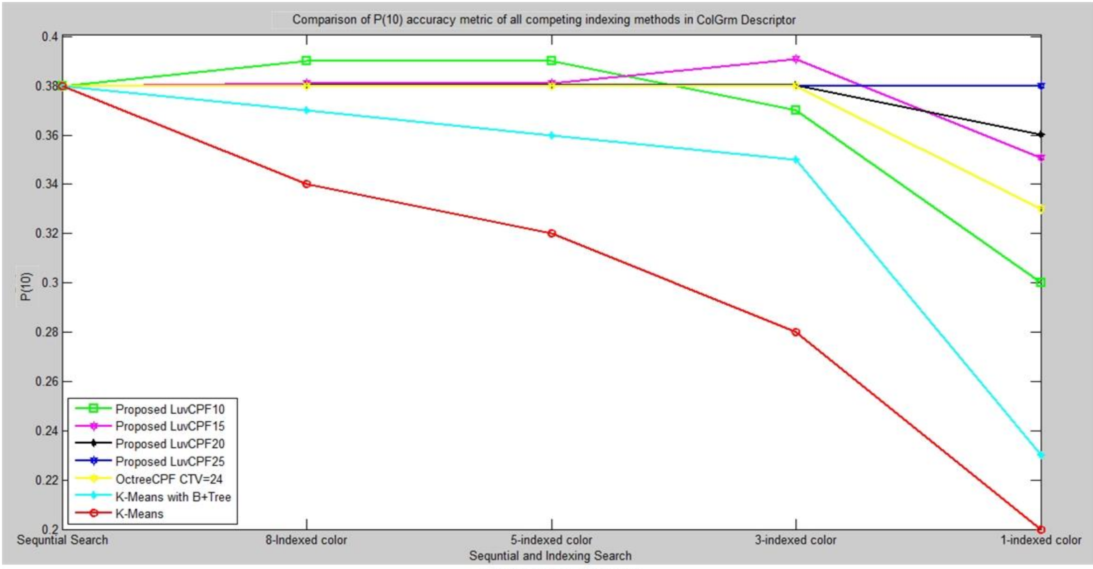
Proposed LUV Indexing+ CPF			**0.420/ 0.152/ 0.810**	**31.8%+ 0.53000%**	**0.420/ 0.148/ 0.817**	**22.0%+ 0.22000%**	**0.390/ 0.124/ 0.846**	**8.2%+ 0.07000%**
Proposed LUV Indexing, Colour Distance=10	**0.420/ 0.147/ 0.817**	**37.1%+ 0.57000%**	**0.420/ 0.145/ 0.819**	**33.0%+ 0.55000%**	**0.400/ 0.138/ 0.827**	**28.2%+ 0.30000%**	**0.360/ 0.116/ 0.855**	**19.8%+ 0.23000%**
Proposed LUV Indexing+ CPF	**0.420/ 0.149/ 0.815**	**27.1%+ 0.29000%**	**0.420/ 0.147/ 0.818**	**23.2%+ 0.25000%**	**0.410/ 0.140/ 0.826**	**19.0%+ 0.21000%**	**0.360/ 0.112/ 0.859**	**12.7%+ 0.14000%**
Proposed LUV Indexing, Colour Distance=10,W	**Same as above**		**0.420/ 0.142/ 0.825**	**18.2%+ 0.21000%**	**0.400/ 0.132/ 0.837**	**13.1%+ 0.15000%**	**0.360/ 0.098/ 0.876**	**5.6%+ 0.02000%**
Proposed LUV Indexing+ CPF			**0.420/ 0.142/ 0.823**	**14.1%+ 0.16000%**	**0.410/ 0.133/ 0.834**	**9.9%+ 0.09000%**	**0.360/ 0.100/ 0.875**	**3.9%+ 0.00100%**
Proposed RGB Octree Indexing CTV =24	**0.410/** 0.151/ 0.813	**57.3%+ 0.90000%**	**0.410/** 0.150/ 0.814	**49.8%+ 0.08000%**	**0.410/** 0.143/ 0.822	**40.5%+ 0.00700%**	0.350/ 0.103/ 0.870	**24.5%+ 0.00080%**
Proposed Octree+CPF CTV =24	**0.410/** 0.151/ 0.813	**47.2%+ 0.80000%**	**0.410/** 0.150/ 0.814	**39.2%+ 0.07000%**	**0.410/** 0.142/ 0.823	**29.7%+ 0.00500%**	0.340/ 0.100/ 0.874	**15.1%+ 0.00060%**

** Accuracy metrics (ARR, ANMRR, P(10)) and Efficiency metrics (SSR and OHR) for Weighted DC-based ColGrm Descriptor (distance=5) using sequential search compared with all competing indexing methods (K-means, K-Means with B+Tree, proposed Octree CTV=24 and proposed Octree with Colour Percentage Filtering (CPF) CTV=24) where different settings are applied. Cartoon-11K Dataset with 158 Queries is used in these experiments.*

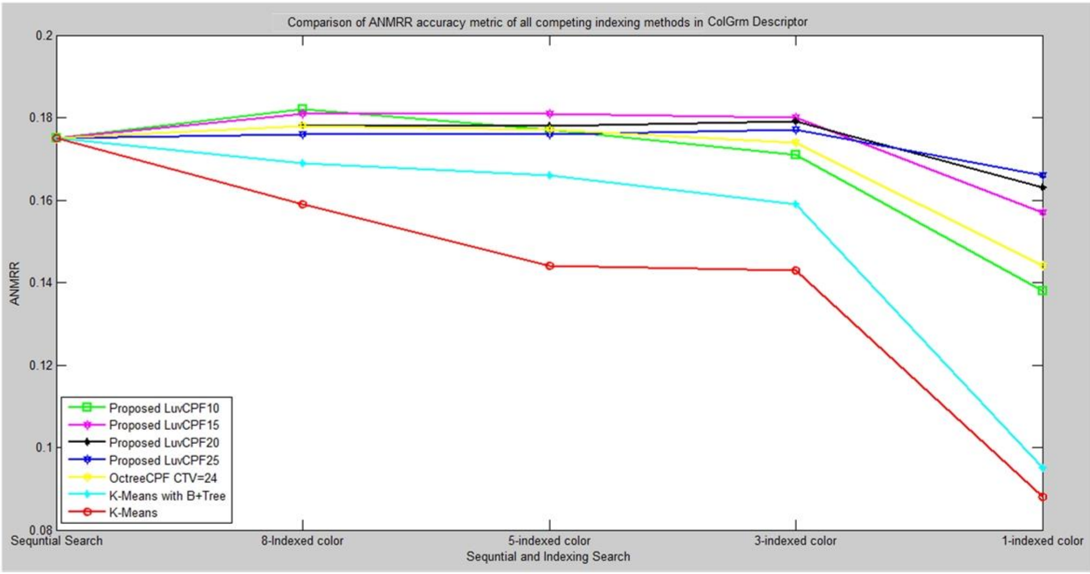
In this descriptor (WDCBC), the proposed indexing method outperforms the sequential search in metric P(10) almost in all settings. In ANMRR metric, the accuracy is degraded when LUV distance equals to 10 in all settings and when LUV distance equals to 15 in some settings. This is because, this descriptor is good descriptor and any restriction by indexing methods will lead to lose some relevant images.

ii. Results of Corel-10K dataset

In this section, graphical results for the Corel-10K dataset are presented. This is because, to assure that the proposed indexing method can be generalized to different databases.



(a) P(10) Accuracy Metric



(b) ANMRR Accuracy Metric

Figure 6.31: Accuracy metrics of LUV indexing compared with other indexing methods in ColGrm in Corel-10K dataset using (a) P(10) and (b) ANMRR.

Likewise to Cartoon-11K dataset, the proposed method when applied on *ColGrm* descriptor is outperformed other indexing methods and sequential search, as shown in Figure 6.31, except when the indexed colours equal 1. Moreover, weight-based sorting method is outperformed the percentage-based sorting methods in most settings, as depicted in Figure 6.32.

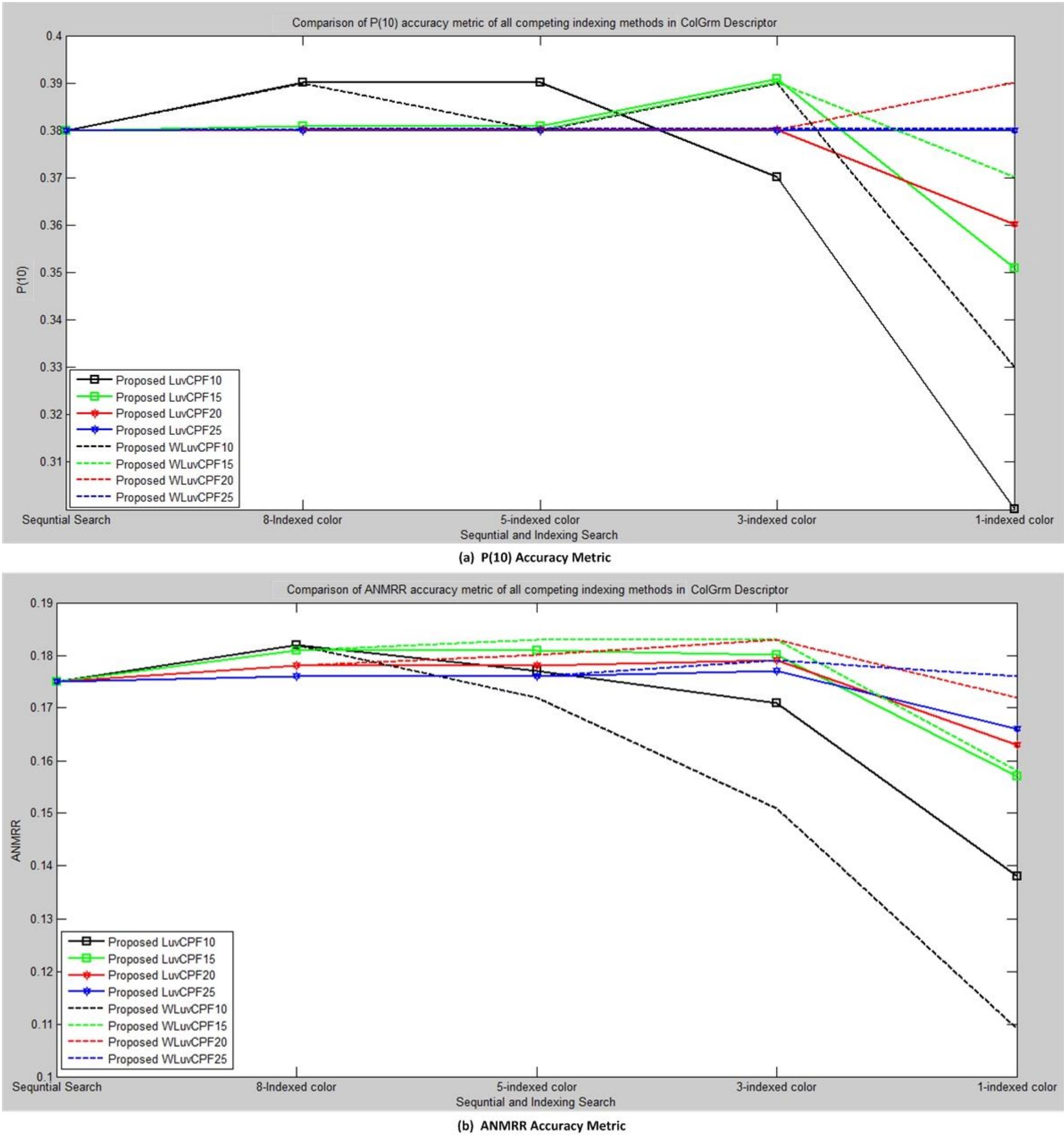
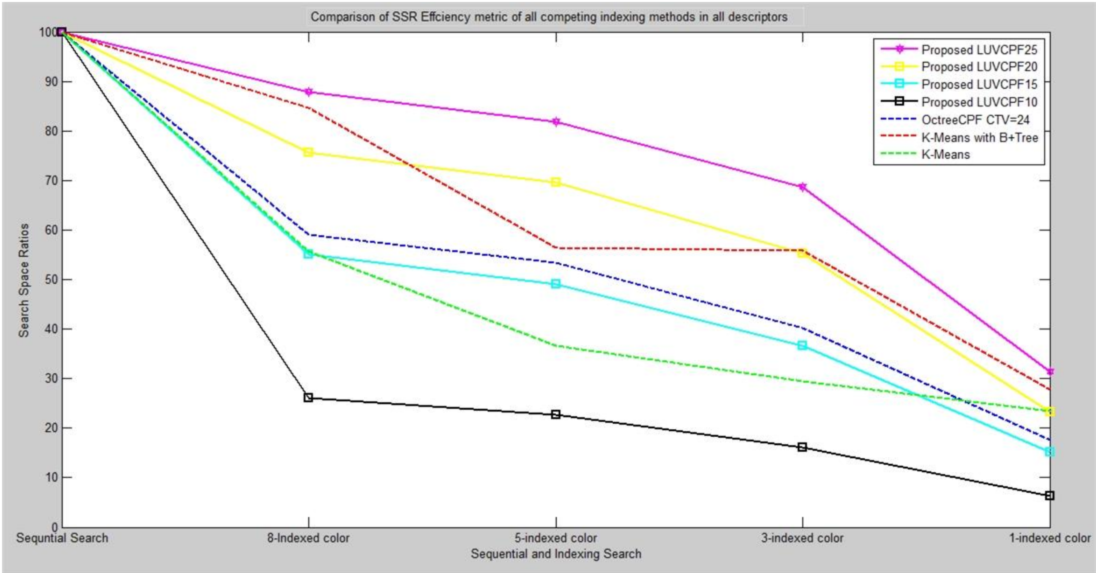
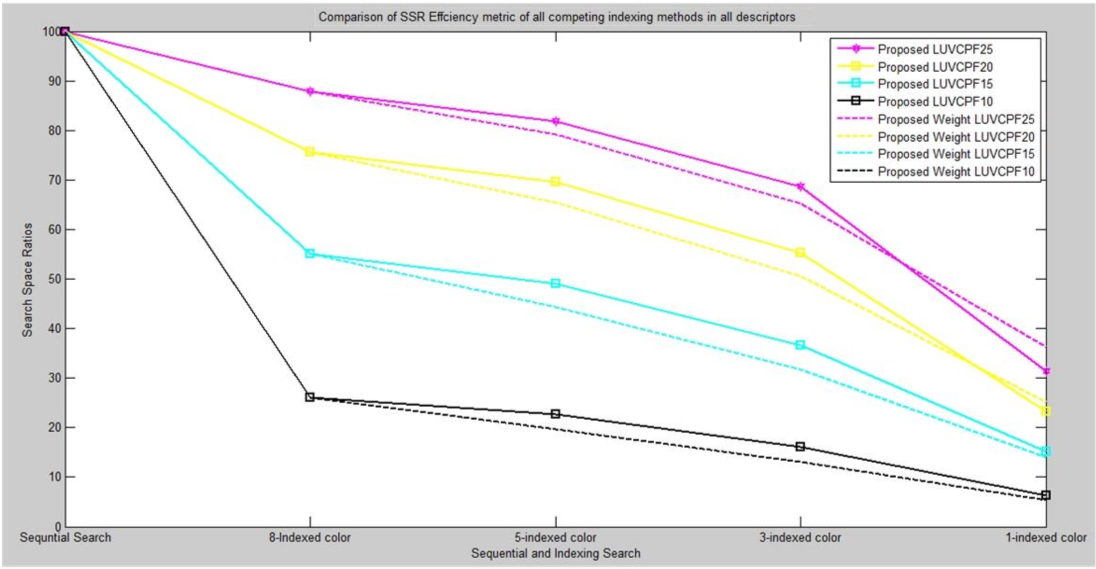


Figure 6.32: Comparison between Colour Percentage- and Colour Weight-based indexing methods in ColGrm descriptor using (a) P(10) and (b) ANMRR.



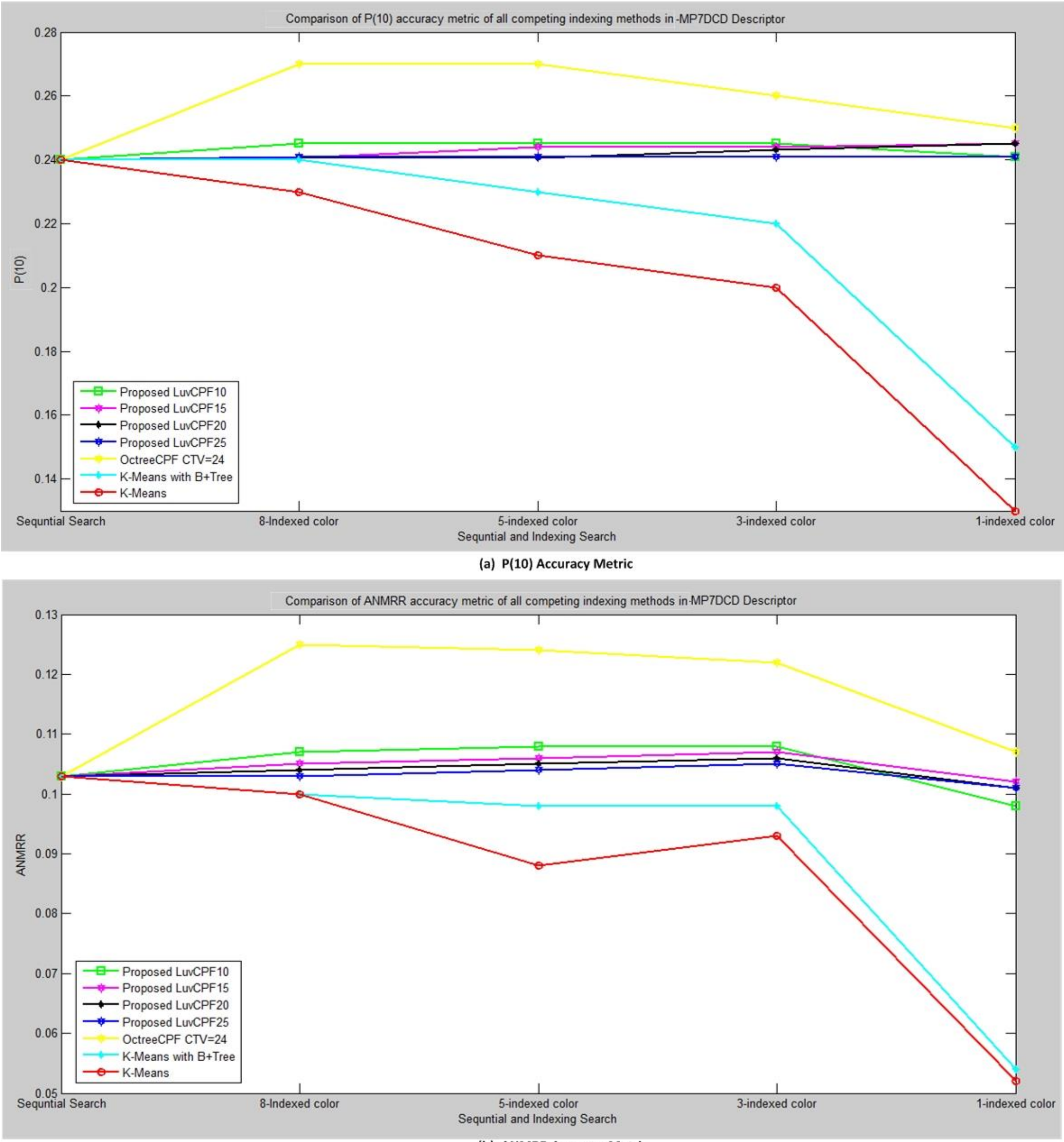
(a) SSR Efficiency metric for all competing descriptors



(b) Comparison of SSR between Percentage and Weight-based indexing methods

Figure 6.33: SSR comparison for (a) different settings in LUV indexing and (b) CP- and CW-based sorting methods in LUV indexing.

Similar to Cartoon-11K dataset, Figure 6.33 (a) shows that the proposed indexing method of LUV distance equals to 25 has largest search space whereas the distance of value 10 has smallest search space. KMB method has SSR similar to proposed method of distance 20 while KM and RGB of CTV=24 have SSR similar to proposed method when distance equals 15. In another side, weight-based sorting technique has SSR lower than percentage-based sorting method, as depicted in Figure 6.33 (b).

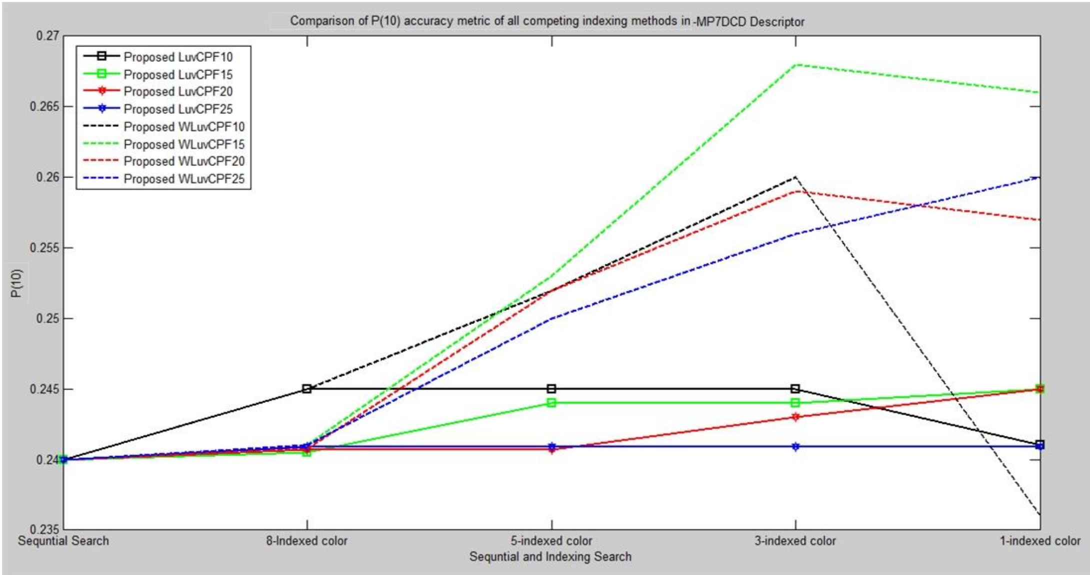


(a) P(10) Accuracy Metric

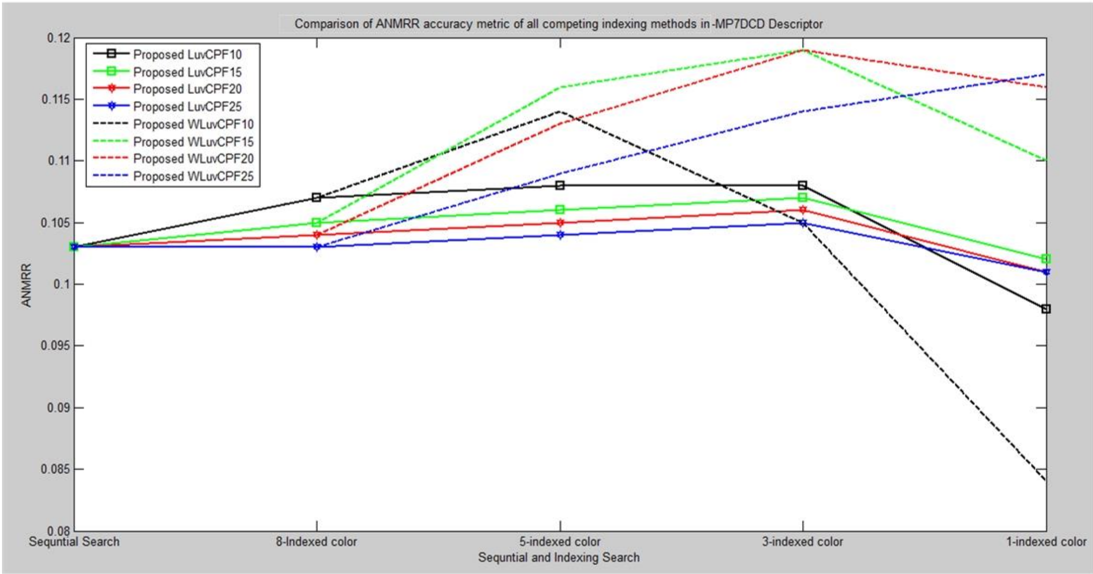
(b) ANMRR Accuracy Metric

Figure 6.34: Accuracy metrics of LUV indexing compared with other indexing methods in MPEG-7 DCD using (a) P(10) and (b) ANMRR.

In MP7DCD, all settings of the proposed LUV indexing method are outperformed the sequential search and other indexing methods, as shown in Figure 6.34.



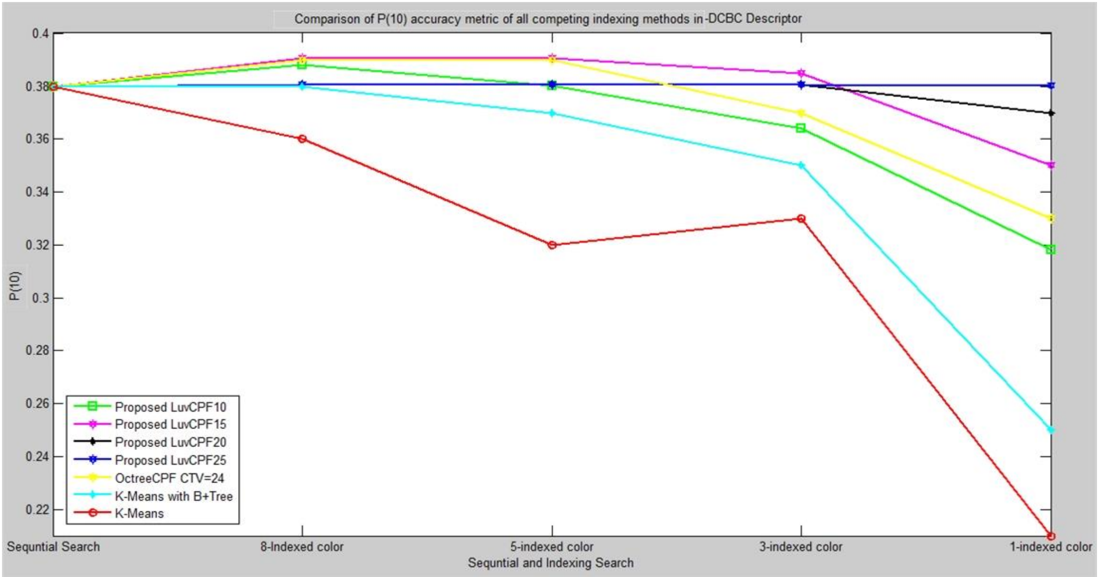
(a) P(10) Accuracy Metric



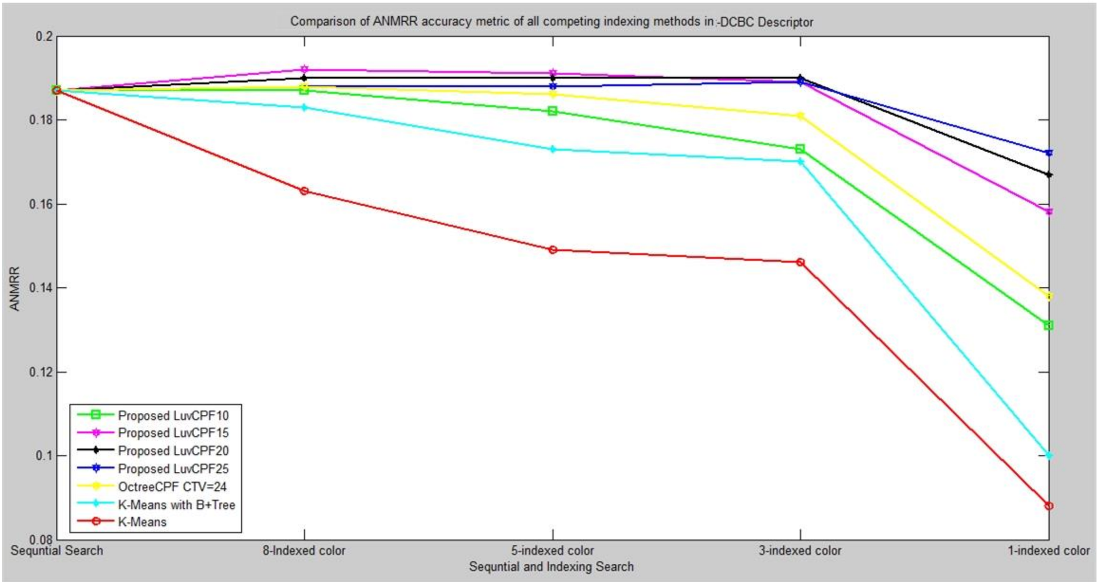
(b) ANMRR Accuracy Metric

Figure 6.35: Comparison between Colour Percentage- and Colour Weight-based indexing methods in MPEG-7 DCD using (a) P(10) and (b) ANMRR.

The Weight-based sorting method also surpasses the percentage-based sorting method in most settings, as depicted in Figure 6.35.



(a) P(10) Accuracy Metric



(b) ANMRR Accuracy Metric

Figure 6.36: Accuracy metrics of LUV indexing compared with other indexing methods in DCBC using (a) P(10) and (b) ANMRR.

In DCBC, LUV indexing method is also outperformed the other indexing methods and sequential search in most settings in both metrics, P(10) as shown in Figure 6.36 (a) and ANMRR as depicted in Figure 6.36 (b). Additionally, Figure 6.37 (a) explains that the weight-based method surpasses percentage-based method in most settings in terms of P(10) metric. In ANMRR metric, the two methods approximately have similar accuracy, as depicted in Figure 6.37 (b), except when the distance equals to 10 where the weight method is worse than percentage method.

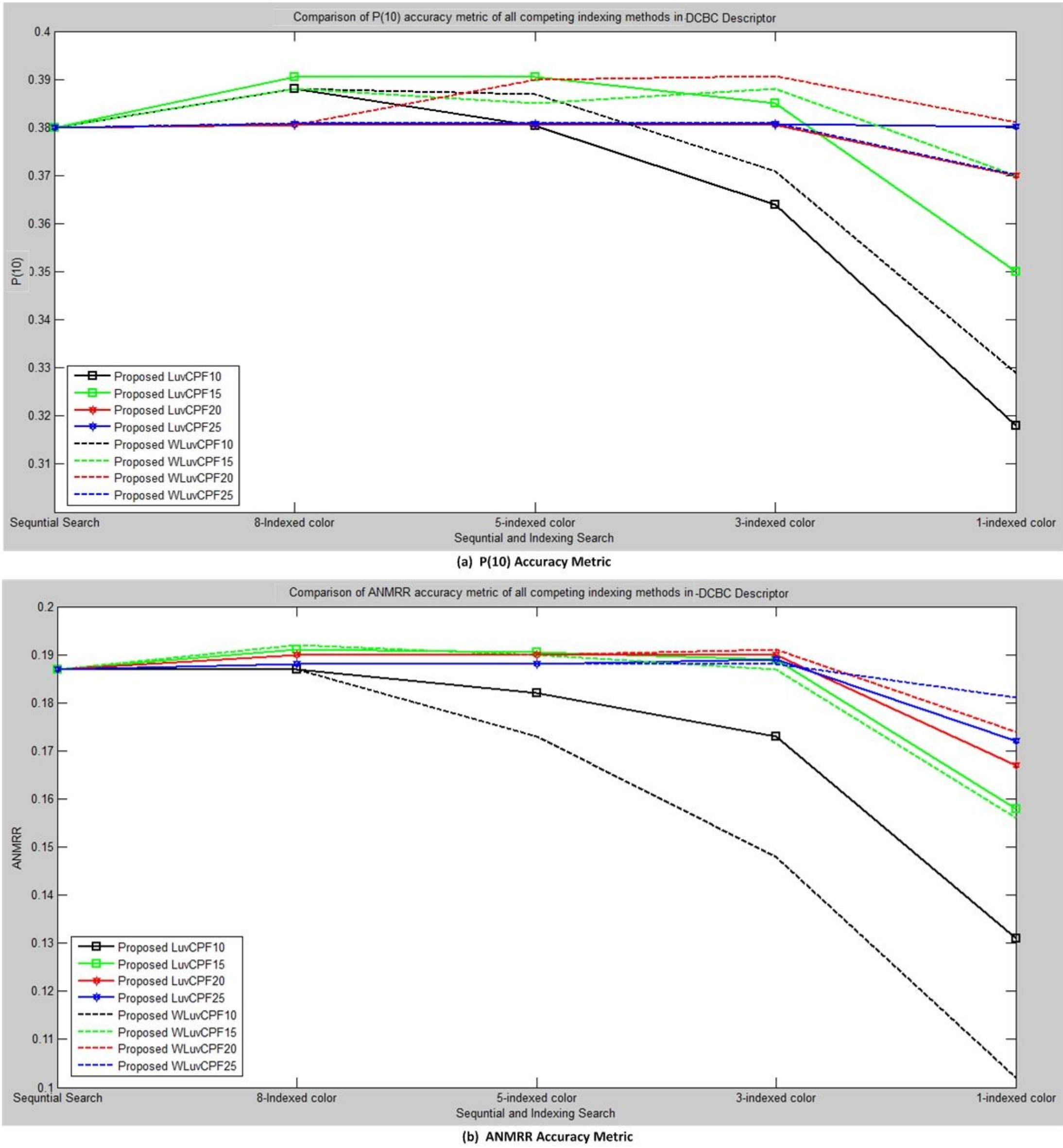


Figure 6.37: Comparison between Colour Percentage- and Colour Weight-based indexing methods in DCBC using (a) P(10) and (b) ANMRR.

6.4 Summary

In this chapter, indexing methods of CBIR with the advantages and disadvantages of each method are presented. Specifically, the problems of colour-based indexing methods such as high-dimensional problem for histogram-based methods and colour approximation problem of DC-based methods are addressed. Colour approximation problem is focused in this chapter. Accordingly, two DC-based indexing methods are proposed, the first for uniform RGB colour space whereas the second for non-uniform LUV colour space. These methods outperform the existing KM and KMB indexing methods in accuracy and efficiency metrics where several settings are applied. The search space ratio is reduced to less than 25% with preserving the same accuracy. The proposed RGB indexing method has the same problem of SP methods which is locating the border at the edge of partition, this lead to degradation of accuracy. Therefore, perceptual LUV indexing method is proposed to tackle this issue.

CHAPTER SEVEN
CONCLUSION AND FUTURE WORK

This chapter is dedicated to summarize the thesis achievements as well as to outline future directions in CBIR research field. A summary of the thesis achievements is presented in Section 7.1 where in this section the contributions of this research are explained. Section 7.2 offers some suggestions and future directions.

7.1 Contributions

The massive growth of digital image libraries imposes management and organization on these libraries to ease the retrieving and browsing operations. Thus, two methods are used for retrieving images from these libraries, annotation-based (text-based) image retrieval, which suffers from many drawbacks in large image database due to image annotation process that represents laborious and time consuming method and very much depending on user subjectivity. Hence, another method has been proposed, i.e. CBIR, which depends on image content in retrieving similar images from database. Low level features such as colour, texture and shape are widely used to analyse image content. These low level features suffer from semantic gap problem that corresponds to the difference in the representation of an image between computer representation and semantic meaning to a human. Therefore, these low level features with existing large database deteriorate the CBIR performance. Performance of image retrieval depends on two key factors, accuracy and speed. According to the importance of colour feature in most CBIRSs, this thesis focuses on improving the performance and reducing the complexity of colour-based CBIR through enhancing the methods that are related with the two aforementioned factors. The main contributions of this thesis as specified in Chapter 1 are as follows:

7.1.1 Weighted dominant colour descriptor

The first contribution of this thesis is related to the accuracy where enhancing the quality of features represents the solution for this issue. This enhancement is achieved by focusing on the object of the image. To enhance CBIR to be as object-based image retrieval, this requires addressing the problems that can impede this matter. Large background dominance problem is identified as the most challenging problem in most colour-based image retrieval methods (Krishnan et al., 2007; Renato et al., 2002). Two solutions are proposed, which formulate the first contribution in this research in order to overcome this problem. The first represents feature level-based solution where two algorithms are applied to differentiate between background colours and object colour. The first algorithm depends on the assumption that is the object tends to be near the centre of image and far from the border while the second algorithm uses an existing salient object detection method to determine the object colours. From the results of the two algorithms, weights are computed and assigned to all DCs of the image where high weights are assigned to the object's DCs and low weights are assigned to the background colours. To generalize this solution (weighting the DCs) to all colour-based methods, the weighting concept is applied on the best and most complicated colour descriptors, *Colour Correlogram* and *Border/Interior Classification* method. Therefore, a generic weighting DCs framework is proposed. The second represents similarity measure level-based solution. A new term, Mutual Colour Ratio (MCR), is proposed to alleviate the effect of large background. This term is added to the similarity measure functions of many descriptors to prove its effectiveness.

7.1.2 A compact representation of colour correlogram

The second contribution revolves around the second performance key factor, i.e. *time*. Two issues are related with the time factor namely computation complexity and memory space of the features. These two issues correlate with the size of extracted feature vector. In other words, the smaller the size, the smaller the computations and memory space is. Therefore, proposing a compact representation of the colour descriptor is a promising solution. One of the best colour descriptor in large database is *colour Correlogram* (Pedronette & Torres, 2012; Penatti et al., 2012). It is selected to be compacted so as to obtain good accuracy besides the compactness property. First solution is reducing number of colours through two approaches; first one was proposed by Kiranyaz et al. (2012) that used few DCs instead of large number of quantized colours where the colours are reduced to 8 DCs only. This solution suffers from imperfect similarity measure, hence, adapted similarity measure is proposed. Additionally, the method of conversion the quantized colours into few DCs with new adapted similarity measure are generalized by applying it on *Correlogram* and *BIC* methods thus a generic conversion to DCs framework is proposed. The second solution is related with the format of *Correlogram* descriptor. It contains square matrix of the colours, m, this will impose $O(m^2)$ complexity of colours. From proper analysis to this format, a new format is proposed to reduce the complexity approximately to the half, $O(m^2/2+m/2)$. Additionally, to reduce the *Correlogram* complexity, spatial distances among colours is focused. Instead of using distances equal to 50% of the smallest dimension of image, single distance is proposed to substitute the large number of distances. This is performed by averaging all distances of colours to generalize distance values. Therefore, second contribution includes the two algorithms for colour reductions and one for distance reduction algorithm as well as the generic conversion to DCs framework.

7.1.3 An enhanced colour indexing methods

The third contribution focuses on the indexing methods that can influence the both accuracy and time factors. The main effect of indexing method is to reduce the search space in the database and subsequently reducing the required time for image retrieval. The main orientation of this thesis is towards using few DCs instead of large quantized colours. In DCs concept, the problem of high-dimensional indexing that is faced the most colour-based methods such colour histogram that have large number of colour bins to be indexed is eliminated. The problem of most DC-based indexing methods is colour approximation, which exists in vector quantization (VQ) methods. This approximated value of cluster centroid causes a problem in finding similar colours to the cluster where many similar colours will be excluded as well as many non-similar colours will be considered as similar. Additionally, the process of producing DCs itself is also using colour approximation to extract few DCs from large number of quantized colours (such as GLA or LBA dynamic quantization methods). As a result from these two colour approximation methods, accuracy of DC-based methods is degraded. Therefore, VQ methods is avoided in this research, instead space partitioning method is used for colour indexing.

RGB colour space is partitioned using *Octree* colour quantization technique to construct first proposed indexing method. This method demonstrates its effectiveness over VQ-based indexing methods, especially when using colour percentage as a part of similarity process in addition to the colour value. An implication of colour percentage along with reduction the search space speeds up the image retrieval process. RGB colour space is characterized by its simple distribution and does not match to the human visual system. Thus, the trend towards indexing the perceptual colour spaces such as LUV is imposed. Nevertheless, the uniform quantization (such as *Octree*) is not preserving the perceptual similarity among the colours. Accordingly, new colour indexing method is proposed for perceptual colour spaces, such as LUV or Lab, using simple uniform quantization method, *Octree*. This method shows high accuracy, as expected, than non-perceptual (RGB) indexing method in spite of the search space is little more than RGB indexing method.

7.2 Future Work

This section concentrates on the future research recommendations based on this research. These recommendations can be outlined below:

i. In this research, dynamic quantization method (DQM) is used to extract the DCs. Two methods are used in this research, GLA and LBA. These methods need to be enhanced because in some cases two perceptual similar original colours are converted, using DQM, into not similar colours when measuring their distance. This will affect negatively the performance of DC-based image retrieval. Hence, an improved dynamic perceptual quantization method or an enhanced distance metric needs to be developed for this purpose.

ii. The main dependency of this research is on the colour feature that will produce some mismatched results to the query. This is due to either different objects have the same colour of query object or the same object has a different colour due to illumination variations (in case of natural images) or due to different colour selection by designer (in case of cartoon images). Accordingly, other features need to be added to the colour to overcome this limitation (the recommended feature is shape feature because it does not depend on the changes of colour).

iii. Text-based image retrieval is mandatory used in collaboration with CBIR to manage huge databases. In these methods, CBIR can be used to construct set of visual words; it is so-called *Bag of Visual Words (BOVW)*. These text words subsequently will be used in indexing and retrieving image efficiently. DCs recently are participated to construct these *BOVW* (Vidal et al., 2012). From these visual words, object can be described as words also and these object words can be used for building robust object-based image retrieval for large databases.

iv. Managing huge databases that contain millions of images is difficult using pure CBIR techniques. However, CBIR techniques are effective when is used on a medium sized database. This is because, it needs long time to extract features from query image as well as for searching inside the huge database. Thus, many techniques were used for this purpose but they are not suitable for this research. One of these techniques is classification-based similarity measure where it is used in many CBIR because of its accuracy and speed but it is avoided in this research for many reasons as illustrated in Section 2.1.4.2. Another technique is using text-based image retrieval with CBIR to ease and speed up the search but using text is conflicting with the motivations of this research, which is using CBIR to overcome the weakness of ABIR. Accordingly, more efforts need to be carried out to apply these techniques to DC-based CBIR and use it to different colour-based applications.

v. Fusion are achieved in CBIR either in feature level (by combination of different type of features (Gehler & Nowozin, 2009; Zhang et al., 2010)) or in rank level (by combination of the ordered results of different methods (it also called rank aggregation) (Fagin, Kumar & Sivakumar, 2003; Jegou, Schmid, Harzallah & Verbeek, 2010)). Accordingly, the two DC-based methods that are proposed in this work (DC-based *Correlogram* and DC-based BIC) can be fused in a feature or rank level to improve the retrieval accuracy. These two methods have different view of the image where the *Correlogram* has information about image texture in addition to the colour. Whereas, the BIC has information about shape in addition to the colour. Therefore, fusing these methods can lead to improve the CBIR accuracy without increasing the time complexity especially in feature level because the time of extracting the DCs of the image need to be computed once to the both methods.

vi. There are many applications that can use only the colour feature for efficient retrieval of object images such as applications for retrieving flag, trademarks, manufactured objects, postal stamps and textile patterns. Additionally, another application can benefit from the contributions of this work, which is “Children Search Engine” that can search cartoon images (Talib, Mahmuddin & Husni, 2010).

REFERENCES

Aboulmagd, H., El-Gayar, N., & Onsi, H. (2009). A new approach in content-based image retrieval using fuzzy. *Telecommunication Systems, 40*(1-2), 55-66. doi: 10.1007/s11235-008-9142-9.

Achanta, R., Estrada, F., Wils, P., & Susstrunk, S. (2008). Salient region detection and segmentation. In *Computer Vision Systems* (pp. 66-75). Springer Berlin Heidelberg.

Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009). Frequency-tuned salient region detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 1597–1604.

Alaoui, R., Ouatik, S., Alaoui, E., & Meknassi, M. (2009). Spatial Color Indexing: An Efficient and Robust Technique for Content- Based Image Retrieval. *Journal of Computer Science, 5* (2), 109-114.

Alexandrov, A. D., Ma, W.Y., Abbadi, A. El, & Manjunath, B. S. (1995). Adaptive Filtering And Indexing For Image Databases. *Proceedings of International Society for Optical Engineering (SPIE)- Storage and Retrieval for Image and Video Databases*.

Arampatzis, A., Zagoris, K., & Chatzichristofis, S. A. (2013). Dynamic two-stage image retrieval from large multimedia databases. Information Processing & Management, 49(1), 274-285.

Arslan, S., Yazıcı, A., Saçan, A., Toroslu, I. H., & Acar, E. (2013). Comparison of feature-based and image registration-based retrieval of image data using multidimensional data access methods. Data & Knowledge Engineering, 86(0), 124-145. doi: http://dx.doi.org/10.1016/j.datak.2013.01.007.

Attig, J., Copeland, A., & Pelikan, M. (2004). Context and Meaning: The Challenges of Metadata for a Digital Image Library within the University. *College & Research Libraries*, *65*(3) 251-261.

Aulia, E. (2005). *Hierarchical Indexing for Region Based Image Retrieval*. M.Sc thesis, Louisiana State University.

Babu, G. P., Mehtre, B. M., & Kankanhalli, M. S. (1995). Color indexing for efficient image retrieval. *Multimedia Tools and Applications, 1* (2), 327–348.

Beckmann, N., Kriegel, H., Schneider, R., & Seeger, B. (1990). The R* -tree, an efficient and robust access method for points and rectangles. *Proceedings ACM Special Interest Group on Management of Data, International Conference Management Data*, 322–331.

Bentley, J. L. (1979). Multidimensional Binary Search Trees in Database Applications. *IEEE Transactions on Software Engineering, SE-5*(4), 333-340. doi: 10.1109/tse.1979.234200

Bohm, C., Berchtold, S., & Keim, D. A. (2001). Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Survey, 33*(3), 322-373. doi: 10.1145/502807.502809

Boykov, Y. Y., & Jolly, M. P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *Proceedings of Eighth IEEE*

International Conference on Computer Vision ICCV 2001 (Vol. 1, pp. 105-112).
Broek, E. V. D., Kisters, P., & Vuurpijl, L. (2004). The utilization of human color categorization for content-based image retrieval. In *Proceedings of International Society for Optical Engineering (SPIE), 5292,* 351–362.
Buckley, C., & Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. Paper presented at the *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*.
Cai, W., Song, Y., & Feng, D. D. (2012). Regression and classification based distance metric learning for medical image retrieval. Paper presented at *the 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, 2012.
Carneiro, G., & Vasconcelos, N. (2005). A database centric view of semantic image annotation and retrieval. Paper presented at *the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval.*
Carson, C., Belongie, S., Greenspan, H., & Malik, J. (2002). Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24* (8), 1026-1038.
Catalan, J. A., & Jin, J. S. (2000). Dimension reduction of texture features for image retrieval using hybrid associative neural networks. Paper presented at *IEEE International Conference on the Multimedia and Expo, ICME 2000.*
Celebi, E., & Alpkocak, A. (2000). Clustering of texture features for content-based image retrieval. In *Advances in Information Systems* (pp. 216-225). Springer Berlin Heidelberg.
Chang, S.-F., Sikora, T., & Purl, A. (2001). Overview of the MPEG-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology, 11*(6), 688-695.
Chang, B.-M., Tsai, H.-H., & Chou, W.-L. (2013). Using visual features to design a content-based image retrieval method optimized by particle swarm optimization algorithm. *Engineering Applications of Artificial Intelligence, 26*(10), 2372-2382. doi: http://dx.doi.org/10.1016/j.engappai.2013.07.018.
Chen, T., Cheng, M.-M., Tan, P., Shamir, A., & Hu., S.-M. (2009). Sketch2photo: Internet image montage. *ACM Transaction Graphics, 28*(5), 1–10.
Chen, Y., Wang, J. Z., & Krovetz, R. (2005). CLUE: Cluster-based Retrieval of Images by Unsupervised Learning. *IEEE Transactions on Image Processing, 14,* 1187-1201.
Chen, Y., Zhou, X. S., & Huang, T. S. (2001). One-class SVM for learning in image retrieval. Paper presented at *International Conference on the Image Processing, 2001.*
Cheng, M.-M., Zhang, G.-X., Mitra, N. J., Huang, X., & Hu, S.-M. (2011). Global Contrast based Salient Region Detection. *24th IEEE Conference on Computer Vision and Pattern Recognition,* 409-416.
Chinlek, S., & Premchaiswade, W. (2001). Image retrieval using AC/CDC. Paper presented at *the Proceedings of International Symposium on Communications and Information Technologies, 2001*.

Chun, Y., Kim, N., & Jang, I. (2008). Content-based image retrieval using multi-resolution color and texture features. *IEEE Transactions on Multimedia, 6*(10), 1073–1084.

Clough, P., Grubinger, M., Deselaers, T., Hanbury, A., & Muller, H. (2007). Overview of the ImageCLEF 2006 photographic retrieval and object annotation tasks. *In Evaluation of Multilingual and Multi-modal Information Retrieval* (pp. 579-594). Springer Berlin Heidelberg.

Cour, T., & Shi, J. (2007). Recognizing Objects by Piecing Together the Segmentation Puzzle. *IEEE Conference on Computer Vision and Pattern Recognition.*

Das, M., Riseman, E. M., & Draper, B. A. (1997). FOCUS: Searching for Multi-colored Objects in a Diverse Image Database. *Proceedings of Computer Vision and Pattern Recognition,* 756-761.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR), 40*(2), 5.

Datta, R., Li, J., & Wang, J. Z. (2005). Content-based image retrieval: approaches and trends of the new age. Paper presented at *the Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, Hilton, Singapore.

Deng, H., Zhang, W., Mortensen, E., Dietterich, T., & Shapiro, L. (2007). Principal curvature-based region detector for object recognition. Paper presented at *the IEEE Conference on Computer Vision and Pattern Recognition, 2007 (CVPR'07).*

Deng, Y., Kenney, C., Moore, M. S., & Manjunath, B. S. (1999). Peer group filtering and perceptual color quantization. *IEEE International Symposium of Circuits System VLSI (ISCAS'99), 4*, 21–24.

Deng, Y., Manjunath, B. S., Kenney, C., Moore, M. S., & Shin, H. (2001). An efficient color representation for image retrieval. *IEEE Transaction on Image Processing, 10 (1)*, 140–147.

Deza, E., & Deza, M. M. (2009). Encyclopedia of Distances. Springer Book, page 94.

Do, M. N., & Vetterli, M. (2002). Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Transactions on Image Processing, 11*(2), 146-158.

Dobrescu, M., Stoian, M., & Leoveanu, C. (2010). Multi-modal CBIR Algorithm Based on Latent Semantic Indexing. Paper presented at *the Fifth International Conference on Internet and Web Applications and Services*, Barcelona, Spain.

Duaimi, M. G. (2006). *Development of A Content-based Image Retrieval System.* PhD Thesis, Al-Nahrain University, Baghdad, Iraq.

Park, D.-S., Park, J.-S., Kim, T. Y., & Han, J. H. (1999). Image indexing using weighted color histogram. Paper presented at *the Proceedings International Conference on Image Analysis and Processing.*

Eakins, J. P., & Graham, M. E. (1999). *Content-Based Image Retrieval: A Report to the JISC Technology Applications Programme*. Institute for Image Data Research, University of Northumbria, Newcastle.

Eitz, M., Hildebrand, K., Boubekeur, T., & Alexa, M. (2010). An evaluation of descriptors for large-scale image retrieval from sketched feature lines. Computers & Graphics, 34(5), 482-498.

ElAlami, M. E. (2011). Unsupervised image retrieval framework based on rule base system. Expert Systems with Applications, 38(4), 3539-3549.

Everingham, M., Zisserman, A., Williams, C. K., Van Gool, L., Allan, M., Bishop, C. M., . . . Dorkó, G. (2006). The 2005 pascal visual object classes challenge Machine Learning Challenges. In *Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment* (pp. 117-176): Springer.

Fagin, R., Kumar, R., & Sivakumar, D. (2003). Efficient similarity search and classification via rank aggregation. Paper presented at *the Proceedings of the 2003 ACM SIGMOD international conference on Management of data.*

Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., & Equitz, W. (1994). Efficient and effective querying by image content. *Journal of intelligent information systems, 3*(3-4), 231-262.

Fang, Y., Geman, D., & Boujemaa, N. (2005). An interactive system for mental face retrieval. In *ACM Special Interest Group on Multimedia (SIGMM), International workshop on Multimedia information retrieval*, 193-200.

Fauqueur, J., & Boujemaa, N. (2002). Region-Based Image Retrieval: Fast Coarse Segmentation and Fine Color Description. In *Proceedings of IEEE International Conference on Image Processing (ICIP'2002)*, Rochester, USA.

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. Paper presented at *the Computer Vision and Pattern Recognition Workshop, 2004, CVPRW'04.*

Ferrari, V., Tuytelaars, T., & Gool, L. V. (2004). Simultaneous object recognition and segmentation by image exploration. In *Proceedings of the 8th European Conference on Computer Vision*, Prague, Tcheque Republic, 40–54.

Gehler, P., & Nowozin, S. (2009). On feature combination for multiclass object classification. Paper presented at *the IEEE 12th International Conference on Computer Vision*, 2009.

Gervautz, M., & Purgathofer, W. (1990). A simple method for color quantization: octree quantization. In S. G. Andrew (Ed.), *Graphics gems* (pp. 287-293): Academic Press Professional, Inc.

Geusebroek, J. (2006). Compact Object Descriptors from Local Colour Invariant Histograms. *British Machine Vision Conference.*

Gevers, T., & Smeulders, A. W. M. (1999). Color-based object recognition. *Journal of Pattern recognition*, 32(3), 453-464.

Ghoshal, A., Ircing, P., & Khudanpur, S. (2005). Hidden Markov models for automatic annotation and content-based retrieval of images and video. Paper presented at *the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval.*

Goferman, S., Zelnik-Manor, L., & Tal, A. (2010). Context-aware saliency detection. *IEEE Conference on Computer Vision and Pattern Recognition.*

Gong, Y., Chuan, C. H., & Xiaoyi, G. (1996). Image indexing and retrieval using color histograms. *Multimedia Tools and Applications, 2*(1996), 133–156.

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset.* Technical Report UCB/CSD-04-1366, California Institute of Technology.

Grubinger, M. (2007). *Analysis and Evaluation of Visual Information Systems Performance.* (PhD Thesis), Victoria University, Melbourne, Australia.

Guldogan, E. (2008). *Improving content-based image indexing and retrieval performance.* (Doctoral Dissertation), Tampere University of Technology.

Guttman, A. (1984). R-trees: A dynamic index structure for spatial search. *Proceedings ACM SIGMOD International Conference Management Data,* 47–57.

Hafner, J., Sawhney, H.S., Esquitz, W., Flickner, M., & Niblack, W. (1995). Efficient color histogram indexing for quadratic form distance functions. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 17* (1995), 729–736.

Hanjalic, A., Lienhart, R., Ma, W.-Y., & Smith, J. R. (2008). The Holy Grail of Multimedia Information Retrieval: So Close or Yet So Far Away? *IEEE Proceedings, Special Issue on Multimedia Information Retrieval, 96* (4), 541-547.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 28*(1), 100-108.

He, X. (2004). Incremental semi-supervised subspace learning for image retrieval. Paper presented at *the Proceedings of the 12th annual ACM international conference on Multimedia.*

Hegazy, D., & Denzler, J. (2008). Boosting colored local features for generic object recognition. *Pattern Recognition and Image Analysis*, *18*(2), 323-327.

Hoi, C.-H., & Lyu, M. R. (2004). A novel log-based relevance feedback technique in content-based image retrieval. Paper presented at *the Proceedings of the 12th annual ACM international conference on Multimedia.*

Hou, A. Lin, Zhao, L.-Q., & Shi, D.-C. (2010). Garment image retrieval based on multi-features. Paper presented at *the International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*, 2010.

Howarth, P. D. (2007). *Discovering images: features, similarities and subspaces.* (PhD Thesis), University of London.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., & Zabih, R. (1997). Image indexing using color correlograms. *Proceedings of Computer Vision and Pattern Recognition*,17–19 June, 762–768.

ImageCLEF. (2003). *The CLEF Cross Language Image Retrieval Track.* Retrieved from http://imageclef.org/.

Jacobs, C. E., Finkelstein, A., & Salesin, D. H. (1995). Fast multiresolution image Querying. Paper presented at *the ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH),* Los Angeles, CA.

Jaswal, G., & Kaul, A. (2009). Content Based Image Retrieval – A Literature Review. *National Conference on Computing, Communication and Control (CCC-09),* 198-201.

Jegou, H., Schmid, C., Harzallah, H., & Verbeek, J. (2010). Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(1), 2-11.

Jeon, J., Lavrenko, V., & Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. Paper presented at *the Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*.

Jiebo, L., & Crandall, D. (2006). Color object detection using spatial-color joint probability functions. *IEEE Transactions on Image Processing, 15*(6), 1443-1453. doi: 10.1109/tip.2006.871081

Jones, K. S., & van Rijsbergen, C. J. (1976). Information retrieval test collections. *Journal of documentation, 32*(1), 59-75.

Jouili, S., & Tabbone, S. (2012). Hypergraph-based image retrieval for graph-based representation. *Pattern Recognition, 45*(11), 4054-4068.

Kankanhalli, M. S., & Rui, Y. (2008). Application potential of multimedia information retrieval. In *Proceedings of the IEEE, 96*(4), 712-720.

Kaski, S., Kangas, J., & Kohonen, T. (1998). Bibliography of self-organizing map (SOM) papers:1981-1997. *Neural computing surveys, 1*(3&4), 1-176.

Khan, F. S., Rao, M. A., Weijer, J. V. D., Bagdanov, A. D., Vanrell, M., & Lopez, A. (2012). Color Attributes for Object Detection. *Twenty-Fifth IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*.

Kim, S., Park, S., & Kim, M. (2003). Central Object Extraction for Object-Based Image Retrieval. *CIVR 2003 : International Conference on Image and Video Retrieval*, Urbana Champaign, IL, 39-49.

Kiranyaz, S., Birinci, M., & Gabbouj, M. (2010). Perceptual color descriptor based on spatial distribution: A top-down approach. *Journal of Image and Vision Computing 28* (8), 1309–1326.

Kiranyaz, S., Birinci, M., & Gabbouj, M. (2012). *Perceptual Color Descriptors*. Foveon, Inc. / Sigma Corporation., San Jose, California, USA: Boca Raton, FL, CRC Press.

Ko, B., & Byun, H. (2002). Integrated region-based image retrieval using region's spatial relationships. Paper presented at *16th International Conference of Pattern Recognition*, 2002.

Krishnan, N., Banu, M. Sheerin, & Christiyana, C. Callins. (2007). Content Based Image Retrieval Using Dominant Color Identification Based on Foreground Objects. *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, 3, 190-194.

Kunttu, I., Lepistö, L., Rauhamaa, J., & Visa, A. (2003). Image correlogram in image database indexing and retrieval. *Proceedings of 4th European Workshop on Image Analysis for Multimedia Interactive Services*, London, UK, April 9-11, 88-91.

Kushki, A., Androutsos, P., Plataniotis, K. N., & Venetsanopoulos, A. N. (2004). Retrieval of images from artistic repositories using a decision fusion framework. *IEEE Transactions on Image Processing, 13*(3), 277-292.

Laaksonen, J., Koskela, M., & Oja, E. (2002). PicSOM-self-organizing image retrieval with MPEG-7 content descriptors. *IEEE Transactions on Neural Networks, 13*(4), 841-853.

Lee, S.-J., Lee, Y.-H., Ahn, H., & Rhee, S.-B. (2008). Color image descriptor using wavelet correlogram. *The 23rd International Technology Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 1613–1616.

Lee, Y. H., Lee, K. H., & Ha, H. Y. (2003). Spatial Color Descriptor for Image Retrieval and Video Segmentation. *IEEE Transaction on Multimedia, 5*(3), 358–367.

Leibe, B., Mikolajczyk, K., & Schiele, B. (2006). Efficient Clustering and Matching for Object Class Recognition. *Proceedings of the British Machine Vision Conference*, 789-798.

Leung, C. H. C., & Hibler, D. (1991). *Architecture of a Pictorial Database Management System.* Technical report, British Library Research, London, UK.

Leung, C. H. C., & Ip, H. (2000). Benchmarking for Content-Based Visual Information Search. *Fourth International Conference On Visual Information Systems (VISUAL'2000)*, Lecture Notes in Computer Science (LNCS), Springer, Lyon, France, 1929, 442–456.

Lew, M. S., Sebe, N., Djeraba, C., & Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), 2*(1), 1-19.

Li, J., & Wang, J. Z. (2008). Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(6), 985-1002.

Li, J., Wu, W., Wang, T., & Zhang, Y. (2008). One step beyond histograms: Image representation using markov stationary features. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

Li, J. (2005). A mutual semantic endorsement approach to image retrieval and context provision. Paper presented at *the Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*.

Li, J., & Lu, B.-L. (2009). An adaptive image Euclidean distance. *Pattern Recognition, 42*(3), 349-357.

Li, R., Bhanu, B., & Dong, A. (2005). Coevolutionary feature synthesized EM algorithm for image retrieval. Paper presented at *the Proceedings of the 13th annual ACM international conference on Multimedia*.

Li, Y. (2005). *Object and Concept Recognition for Content-Based Image Retrieval.* PhD Thesis, University of Washington.

LiFang, Y., XiangLin, H., Rui, L., & Hui, L. (2012). An Effective Similarity Measurement Algorithm for Dominant Color Feature Matching in Image Retrieval. *Applied Mechanics and Materials, 182*, 1169-1173.

Lightstone, S., Teorey, T., & Nadeau, T. (2007). *Physical Database Design: The Database Professional's Guide to Exploiting Indexes, Views, Storage, and More.* Morgan Kaufmann Publishers.

Lin, Y.-Y., Liu, T.-L., & Chen, H.-T. (2005). Semantic manifold learning for image retrieval. Paper presented at *the Proceedings of the 13th annual ACM international conference on Multimedia*.

Liu, G.-H., & Yang, J.-Y. (2013). Content-based image retrieval using colour difference histogram. Pattern Recognition, 46(1), 188-198.

Liwei, W., Yan, Z., & Jufu, F. (2005). On the Euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1334-1339. doi: 10.1109/tpami.2005.165.

Lloyd, S. P. (1982). Least Squares Quantization in PCM. *IEEE Transaction Information Theory, 28* (2), 129–137.

Long, F., Zhang, H., & Feng, D. D. (2003). Fundamentals of content-based image retrieval. In *Multimedia Information Retrieval and Management* (pp. 1-26). Springer Berlin Heidelberg.

Luoni, C. (2000). *Development of an Interface to a Database Storing the Features of a Multimedia Retrieval System.* License thesis (BSc), Computer Vision and Multimedia Laboratory at the University of Geneva, Geneva, Switzerland.

Ma, W. Y., & Manjunath, B. S. (1997). Edge flow: A framework of boundary detection and image segmentation. *Proceedings IEEE Conference Computer Vision Pattern Recognition*, 744–749.

Ma, W. Y., & Manjunath, B. S. (1997). Netra: A toolbox for navigating large image databases. *International Conference on IEEE Image Processing*, 1997, 568-571.

Ma, W. Y., & Manjunath, B. S. (1999). NeTra: A Toolbox for Navigating Large Image Databases. *Multimedia Systems, 7*(3), 184-198.

Ma, W. Y., & Zhang, H. J. (1998). Benchmarking of image features for content-based retrieval. In *Signals, Systems & Computers*, *1998. Conference Record of the Thirty-Second Asilomar Conference on IEEE* (Vol. 1, pp. 253-257).

Ma, W. Y., Deng, Y., & Manjunath, B. S. (1997). Tools for texture/color based search of images. *Proceedings of SPIE Conference on Human Vision and Electronic Imaging II,* 3106 (1997), 496–507.

Maimon, O., & Rokach, L. (2005). *Data Mining and Knowledge Discovery Handbook*. Springer.

Malisiewicz, T., & Efros, A. A. (2007). Improving spatial support for objects via multiple segmentations. In *British Machine Vision Conference.*

Manjunath, B. S., Ohm, J. -R., Vasudevan, V. V., & Yamada, A. (2001). Color and texture descriptors. *IEEE Transaction on Circuits and Systems for Video Technology, 11* (June), 703–715.

Matas, J., Koubaroulis, D., & Kittler, J. (2000). *Colour Image Retrieval and Object Recognition Using the Multimodal Neighbourhood Signature*. D. Vernon (Ed.): ECCV 2000, LNCS 1842,Springer-Verlag Berlin Heidelberg, 48-64.

Mauro, J. (2008). *Oracle interMedia Managing Multimedia Content*. ORACLE, USA.

Mehyar, T., & Atoum, J. O. (2012). An Enhancement on Content-Based Image Retrieval using Colour and Texture Features. Journal of Emerging Trends in Computing and Information Sciences, 3(4).

Mejdoub, M., Fonteles, L., BenAmar, C., & Antonini, M. (2009). Embedded lattices tree: An efficient indexing scheme for content based retrieval on image databases. *Journal Visual Communivation Image Representation, 20*(2), 145-156.

Milanese, R. (1993). *Detecting Salient Regions in an Image: from Biological Evidence to Computer Implementation*. PhD Thesis, University of Geneva.

Moghaddam, H., & Saadatmand-Tarzjan, M. (2006). Gabor wavelet correlogram algorithm for image indexing and retrieval. *The 18th International Conference on Pattern Recognition, ICPR 2006*, 2, 925–928.

Mojsilovic, A., Hu, J., & Soljanin, E. (2002). Extraction of perceptually important colors and similarity measurement for image matching, retrieval, and analysis. *IEEE Transaction of Image Processing, 11*(11), 1238–1248.

Mojsilovic, A., Kovacevic, J., Hu, J., Safranek, R., & Ganapathy, K. (2000). Matching and retrieval based on the vocabulary and grammar of color patterns. *IEEE Transactions on Image Processing, 9*(1), 38–54.

Moosmann, F., Triggs, W., & Jurie, F. (2006). Randomized clustering forests for building fast and discriminative visual vocabularies. In *Neural Information Processing Systems (NIPS) Conference.*

Mustaffa, M. R., Ahmad, F., Mahmod, R., & Doraisamy, S. (2012). Multi-resolution Joint Auto Correlograms: Determining the distance function. Paper presented at *the International Conference on Information Retrieval & Knowledge Management.*

Müller, H., Marchand-Maillet, S., & Pun, T. (2002). The truth about corel-evaluation in image retrieval. In *Image and Video Retrieval* (pp. 38-49). Springer Berlin Heidelberg.

Müller, H., Michoux, N., Bandon, D., & Geissbuhler, A. (2004). A review of content-based image retrieval systems in medical applications-clinical benefits and future directions. *International journal of medical informatics, 73*(1), 1-24.

Müller, H., Müller, W., Squire, D. M., Marchand-Maillet, S., & Pun, T. (2001). Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters, 22*(5), 593-601.

Nagasaka, A., & Tanaka, Y. (1992). Automatic video indexing and full video search for objects. *Visual Database Systems II: Proceedings of the IFIP*, 113–127.

Neumann, D., & Gegenfurtner, K. R. (2006). Image retrieval and perceptual similarity. *ACM Transactions on Applied Perception (TAP)*, *3*(1), 31-47.

Nister, D., & Stewenius, H. (2006). Scalable Recognition with a Vocabulary Tree. Paper presented at *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Nölle, M., Rubik, M., & Hanbury, A. (2006). Results of the muscle cis coin competition 2006. Paper presented at *the Proceedings of the Muscle CIS Coin Competition Workshop*, Berlin, Germany.

Ortega-Binderberger, M., & Mehrotra, S. (2004). Relevance feedback techniques in the MARS image retrieval system. *Multimedia systems, 9*(6), 535-547.

Pass, G., & Zabih, R. (1999). Comparing images using joint histograms. *Springer. Multimedia Systems, 7*(3), 234-240.

Pauleve, L., Jegou, H., & Amsaleg, L. (2010). Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letter, 31*(11), 1348-1358.

Pass, G., Zabih, R., & Miler, J. (1997). Comparing Images Using Color Coherence Vectors. *Proceeding ACM on Multimedia,* 65-73.

Pavlidis, T. (2008). Limitations of CBIR. *International Conference on Pattern Recognition*, December 8-11 in Tampa, Florida.

Pedronette, D. C. G., & Torres, R. D. S. (2012). Exploiting pairwise recommendation and clustering strategies for image re-ranking. *International Journal of Information Sciences, 207*, 19-34.
Penatti, O. A., Valle, E., & Torres, R. D. S. (2012). Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation, 23*(2), 359-380.
Peng, J. (2003). Multi-class relevance feedback content-based image retrieval. *Computer Vision and Image Understanding, 90*(1), 42-67.
Po, L. M., & Wong, K. M. (2004). A new palette histogram similarity measure for MPEG-7 dominant color descriptor. *IEEE International Conference Image Processing (ICIP'04)*, 3 (2004), 1533–1536.
Poursistani, P., Nezamabadi-pour, H., Askari Moghadam, R., & Saeed, M. (2013). Image indexing and retrieval in JPEG compressed domain based on vector quantization. Mathematical and Computer Modelling, 57(5–6), 1005-1017. doi: http://dx.doi.org/10.1016/j.mcm.2011.11.064.
Powell, G. (2006). *Beginning Database Design and Implementation.* Wrox Press.
Premchaiswadi, W., & Tungkasthan, A. (2011). A Compact Auto Color Correlation using Binary Coding Stream for Image Retrieval. *Proceedings of the 15th WSEAS international conference on Computers*, 430-436.
Qiu, G. (2003). Color Image indexing using BTC. *IEEE Transaction on Image Processing, 12*(1), 93–101.
Rafiee, G., Dlay, S. S., & Woo, W. L. (2010). A review of content-based image retrieval. Paper presented at *the 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP)*, 2010.
Rahman, M. M. (2008). *Semantical Representation and Retrieval of Natural Photographs and Medical Images Using Concept and Context-based feature spaces.* Ph.D. Dissertation, Computer Science And Software Enggineering, Concordia University, Canada, 13-18.
Rahmani, R., Goldman, S. A., Zhang, H., Krettek, J., & Fritts, J. E. (2005). Localized content based image retrieval. Paper presented at *the Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval.*
Rasiwasia, N., Moreno, P. J., & Vasconcelos, N. (2007). Bridging the Gap: Query by Semantic Example. *IEEE Transactions Multimedia, 9*(1), 923-938.
Renato, O. S., Mario, A. N., & Alexandre, X. F. (2002). A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. *Proceedings Information and Knowledge Management,* 102-109.
Rodhetbhai, W. (2009). *Preprocessing for Content-Based Image Retrieval.* (PhD Thesis), University Of Southampton.
Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision, 40*(2), 99-121.
Rui, Y., & Huang, T. (2000). Optimizing learning in image retrieval. Paper presented at *the IEEE Conference on Computer Vision and Pattern Recognition.*

Rui, Y., Huang, T. S., & Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation, 10*(1), 39-62.

Russell, B. C., Freeman, W. T., Efros, A. A., Sivic, J., & Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. Paper presented at *IEEE Computer Society Conference on the Computer Vision and Pattern Recognition*, 2006.

Rutishauser, U., Walther, D., Koch, C., & Perona, P. (2004). Is bottom-up attention useful for object recognition?. Paper presented at *the IEEE Conference on Computer Vision and Pattern Recognition*, 37–44.

Salton, G., & McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill Inc., New York City, NY, USA.

Samet, H. (1990). *The Design and Analysis of Spatial Data Structures.* Reading, MA: Addison-Wesley.

Sande, K., Gevers, T., & Snoek, C. (2008). Evaluation of color descriptors for object and scene recognition. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 1–8.

Sande, K. E., Gevers, T., & Snoek, C. G. (2010). Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1582-1596.

Schettini, R., Ciocca, G., & Zuffi, S. (2001). A survey of methods for colour image indexing and retrieval in image databases. *Color Imaging Science: Exploiting Digital Media*, 183-211.

Schmid, C., Mohr, R., & Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision, 37(2)*, 151–172.

Sha'ashua, A., & Ullman, S. (1988). Structural saliency: the detection of globally salient structures using a locally connected network. *International Conference on Computer Vision (ICCV)*, 321-327.

Sikora, T. (2001). The MPEG-7 visual standard for content description-an overview. *IEEE Transactions on Circuits and Systems for Video Technology, 11*(6), 696-702.

Smeulders, A., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(12), 1349-1380.

Smith, J. R., & Chang. (1996). Visualseek: A Fully Automated Content-Based Image Query System. *Proceedings of ACM Multimedia*, Boston, 87-98.

Spath, H. (1980). *Cluster Analysis Algorithms for Data Reduction and Classification.* Ellis Horwood Publishers, England.

Stehling, R. de O., Nascimento, M. A., & Falcao, A. X. (2003). Cell histograms versus color histograms for image representation and retrieval. *Knowledge and Information Systems, 5*(3), 315–336.

Stehling, R. de O., Nascimento, M. A., & Falcao, A. X. (2001). An adaptive and efficient clustering-based approach for content-based image retrieval in image databases. *International Database Engineering & Applications Symposium*, 356–365.

Stricker, M. A., & Orengo, M. (1996). Similarity of color images. *Proceedings SPIE, Storage Retrieval Still Image Video Databases IV*, 2420, 381–392.

Sudhamani, M. V., & Venugopal, C. R. (2007). Grouping and indexing color features for efficient image retrieval. *International Journal of Applied Mathematics and Computer Sciences, 4*(3), 150-155.

Sudhamani, M. V., & Venugopal, C.R. (2006). Non-parametric classification of image data through clustering: An application for image Retrieval. *Proceedings of IEEE International Conference Image and signal processing*.

Swain, M., & Ballard, D. (1991). Color Indexing. *International Journal of Computer Vision, 7*(1), 11–32.

Sykora, D., Burianek, J., & Zara, J. (2003). Segmentation of black and white cartoons. *Conference on Computer Graphics*, 223-230.

Sykora, D., Burianek, J., & Zara, J. (2005). Sketching cartoons by example. *2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 27-34.

Tahaghoghi, S. M. M., Thom, J. A., & Williams, H. E. (2001). Are two pictures better than one? In *Australasian Database Conference*, Queensland, Australia, 138-144.

Talib, A., Mahmuddin, M., Husni, H., & George, L. E. (2013a). Efficient, Compact, and Dominant Color Correlogram Descriptors for Content-based Image Retrieval. Paper presented at *the MMEDIA 2013: Fifth International Conference on Advances in Multimedia*, Venice, Italy, 22-26 April 2013.

Talib, A., Mahmuddin, M., Husni, H., & George, L. E. (2013b). A weighted dominant color descriptor for content-based image retrieval. *Journal of Visual Communication and Image Representation, 24*(3), 345-360.

Talib, A., Mahmuddin, M., & Husni, H. (2010). Using content-based image retrieval for accessing images on the web for children. Paper presented at *the IEEE Second IITA International Joint Conference on Artificial Intelligence (IITA-JCAI 2010)*, 25-26 December , Shanghai, China.

Tao, D., Li, X., & Maybank, S. J. (2007). Negative samples analysis in relevance feedback. *IEEE Transactions on Knowledge and Data Engineering, 19*(4), 568-580.

Tao, D., Tang, X., Li, X., & Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(7), 1088-1099.

Taranto, C., Mauro, N. D., Ferilli, S., & Esposito, F. (2010). Approximate image color correlograms. *Proceedings of the international conference on Multimedia*, Firenze, Italy, 1127-1130.

Taycher, L. (1997). *Image Feature Extraction Subsystem of the ImageRover WWW Image Search System.* Mater thesis, Boston University.

Thomée, B. (2010). *A picture is worth a thousand words: content-based image retrieval techniques.* (PhD Thesis), Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden University.

Tollari, S., & Glotin, H. (2007). Web image retrieval on ImagEVAL: Evidences on visualness and textualness concept dependency in fusion model. Paper presented at *the Proceedings of the 6th ACM international conference on Image and video retrieval*.

Tong, S., & Chang, E. (2001). Support vector machine active learning for image retrieval. Paper presented at *the Proceedings of the ninth ACM international conference on Multimedia*.

Torres, R., & Falcão, A. X. (2006). Content-based image retrieval: Theory and applications. *Journal of Theoretical and Applied Informatics (RITA), 13*(2), 161-185.

TRECVID. (2003). *TREC Video Retrieval Evaluation*. Retrieved from http://trecvid.nist.gov/.

Tungkasthan, A., Intarasema, S., & Premchaiswadi, W. (2009). Spatial color indexing using ACC algorithm. *7th International Conference on ICT and Knowledge Engineering*, 1-2 Dec. 2009, 113 – 117.

Tuytelaars, T., & Mikolajczyk, K. (2008). Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision, 3* (3), 177-280.

Vailaya, A., Figueiredo, M. A. T., Jain, A. K., & Zhang, H. J. (2001). Image classification for content-based indexing. *IEEE Transactions on Image Processing, 10*(1), 117-130.

van de Weijer, J., Schmid, C., Verbeek, J., & Larlus, D. (2009). Learning Color Names for Real-World Applications. *IEEE Transactions on Image Processing, 18*(7), 1512-1523. doi: 10.1109/tip.2009.2019809

Vidal, M. L. A., Cavalcanti, J. M. B., de Moura, E. S., da Silva, A. S., & da Silva Torres, R. (2012). Sorted dominant local color for searching large and heterogeneous image databases. Paper presented at *the 21st International Conference on Pattern Recognition (ICPR).*

Voorhees, E. M. (1998). Variations in relevance judgments and the measurement of retrieval effectiveness. Paper presented at *the Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.*

Voorhees, E. M., & Buckley, C. (2002). The effect of topic set size on retrieval experiment error. Paper presented at *the Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*.

Voorhees, E. M., & Harman, D. (2000). Overview of the sixth text retrieval conference (TREC-6). *Information Processing & Management, 36*(1), 3-35.

Wan, X., & Kuo, C.-C. J. (1996). Color distribution analysis and quantization for image retrieval. In *Security Professionals Information Exchange SPIE proceedings*, 2670.

Wan, X., & Kuo, C. J. (1998). A multiresolution color clustering approach to image indexing and retrieval. *Proceedings IEEE International Conferencec Acoustics, Speech, Signals Processing*, 6, 3705–3708.

Wang, C., Zhang, L., & Zhang, H. J. (2008). Learning to Reduce the Semantic Gap in Web Image Retrieval and Annotation. *The Annual International ACM SIGIR Conference on Research and development in information retrieval*, Singapore, 355-362.

Wang, J. Z., Li, J., & Wiederhold, G. (2001). SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(9), 947-963.

White, D. A., & Jain, R. (1996). Similarity indexing with the SS-tree. *Proceedings 12th IEEE Confernce on Data Engineering*, 516–523.

Williams, A., & Yoon, P. (2007). Content-based image retrieval using joint correlograms. *Multimedia tools and Applications, 34*(2), 239-248.

Wong, K. M., Po, L. M., & Cheung, K. W. (2007). A compact and efficient color descriptor for image retrieval. *Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2007)*, Beijing, China, 611–614.

Wong, K.-M., Po, L.-M., & Cheung, K.-W. (2007). Dominant Color Structure Descriptor For Image Retrieval. *IEEE International Conference on Image Processing, 2007*, 6, 365-368.

Xia, W., & Kuo, C. C. J. (1998). A new approach to image retrieval with hierarchical color clustering. *IEEE Transactions on Circuits and Systems for Video Technology, 8*(5), 628-643. doi: 10.1109/76.718509

Yamada, A., Pickering, M., Jeannin, S., & Jens, L.C. (2001). *MPEG-7 Visual Part of Experimentation Model Version 9.0 - Part 3 Dominant Color.* ISO/IEC JTC1/SC29/WG11/N3914, Pisa.

Yanai, K., Shirahatti, N. V., Gabbur, P., & Barnard, K. (2005). Evaluation strategies for image understanding and retrieval. Paper presented at *the Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval.*

Yang, N.-C., Chang, W.-H., Kuo, C.-M., & Li, T.-H. (2008). A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval. *Journal of Visual Communication and Image Representation, 19* (2008), 92–105.

Yi, T., & William, I. G. (1999). Spatial Color Indexing: A Novel Approach for Content-Based Image Retrieval. *Proceedings of International Conference on Multimedia Computing and Systems ICMCS,* 530 – 535.

Yi, T., & William, I. G. (2007). Content-based image retrieval using joint correlogram. *Multimedia Tools and Applications, 34*(2), 239-248, Springer.

Yildizer, E., Balci, A. M., Jarada, T. N., & Alhajj, R. (2012). Integrating wavelets with clustering and indexing for effective content-based image retrieval. *Journal of Knowledge-Based System, 31*, 55-66.

Yu, J., & Seah, H.-S. (2011). Fuzzy diffusion distance learning for cartoon similarity estimation. *Journal of Computer Science and Technology, 26*(2), 203-216.

Yu, J., Cheng, J., & Tao, D. (2012). Interactive cartoon reusing by transfer learning. *Journal of Signal Processing, 92*(9), 2147-2158.

Zagoris, K., Chatzichristofis, S. A., Papamarkos, N., & Boutalis, Y. S. (2009). img(Anaktisi): A Web Content Based Image Retrieval System. *International Workshop on Similarity Search and Applications*, 154-155.

Zhai, Y., & Shah, M. (2006). Visual attention detection in video sequences using spatiotemporal cues. *ACM Multimedia*, 815-824.

Zhang, H., Gong, Y., Low, C. Y., & Smoliar, S.W. (1995). Image retrieval based on color features: An evaluation study. *Proceedings of SPIE Digital Image Storage Archiving Systems,* 2606, 212–220.

Zhang, H., & Zhong, D. (1995). Scheme for visual feature-based image indexing. Paper presented at *the IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*.

Zhang, Jun. (2011). *Robust content-based image retrieval of multi-example queries.* (PhD Thesis), University of Wollongong.

Zhang, L., Lin, F., & Zhang, B. (2001). Support vector machine learning for image retrieval. Paper presented at *the International Conference on Image Processing*, 2001.

Zhang, M., & Alhajj, R. (2009). Content-Based Image Retrieval: From the Object Detection/Recognition Point of View. Book chapter in *Artificial Intelligence for Maximizing Content Based Image Retrieval*, 115-144.

Zhang, Q., & Tai, X. Y. (2008). Endoscope Image Retrieval Based on Color Feature Fusion. Paper presented at *the Congress on Image and Signal Processing, IEEE CISP'08*.

Zhang, S., Huang, J., Huang, Y., Yu, Y., Li, H., & Metaxas, D. N. (2010). Automatic image annotation using group sparsity. Paper presented at *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, S., Yang, M., Cour, T., Yu, K., & Metaxas, D. (2012). Query Specific Fusion for Image Retrieval. *The 12th European Conference on Computer Vision (ECCV)*.

Zhang, Y. (2002). *On The Use of CBIR in Image Mosaic Generation.* Technical Report, Dept. of Computing Science, University of Alberta, Edmonton, Alberta, Canada.

Zhou, X. S., & Huang, T. S. (2003). Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems, 8*(6), 536-544.

Zhou, Z.-H., Chen, K.-J., & Dai, H.-B. (2006). Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Transactions on Information Systems (TOIS), 24*(2), 219-244.