Murat Birinci

A PERCEPTUAL APPROACH FOR IMAGE RETRIEVAL BASED ON SPATIAL COLOR DISTRIBUTION

Master of Science Thesis

Examiners: Prof. Moncef Gabbouj and Dr. Serkan Kiranyaz
Examiners and topic approved in the Information Technology Department Council meeting on 6 June 2007
Preface

This Thesis has been carried out in the Institute of Signal Processing, Tampere University of Technology, Finland as part of the MUVIS project.

First of all, I would like to express my sincere gratitude to my supervisors Professor Moncef Gabbouj and Dr. Serkan Kiranyaz for giving me the opportunity for this work and their continuous teachings, guidance and support. I would also like to extend my special thanks to all MUVIS Team members, especially Esin Güldogan, for providing such a productive and friendly environment.

I would also like to thank to my friends for their all kinds of support and motivation. Special thanks go to Emre Kapucu, Uygar Tuna and finally Baran Aydogan for letting his photographic performances to be published in the thesis.

Finally I would like to thank my dear family for their endless love and support even though we were kilometers apart.

Yaptığım her işte örnek aldığım, hayatımıla ilgili aldığım her kararda beni koşulsuz destekleyen aileme sonsuz teşekkürler.

Tampere, August 2007.

Murat Birinci

Mustanlahdenkatu 1 B59

33210 Tampere, FINLAND
ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Masters Degree Programme in Information Technology

Birinci, Murat: A Perceptual Approach for Image Retrieval Based on Spatial Color Distribution

Master of Science Thesis, 95 pages, 3 Appendix pages

August 2007

Major Subject: Image and Video Signal Processing
Examiners: Prof. Moncef Gabbouj and Dr. Serkan Kiranyaz
Keywords: content-based image indexing and retrieval, perceptual color descriptor, human visual system, spatial color distribution.

Color features are key-elements which are widely used in image and video content-analysis and retrieval. However, most of them show severe limitations and drawbacks due to their inefficiency of modeling the human visual system with respect to color perception. Moreover, they cannot characterize all the properties of the color composition in visual scenery. In this thesis a perceptual color feature is presented, which describes all major properties of prominent colors both in the spatial and color domains. In accordance with the well-known Gestalt law, a top-down approach is adopted in order to model (see) the whole color composition before its parts, and in this way, problems of pixel-based approaches could be avoided. In the color domain, dominant colors are extracted along with their global properties and quad-tree decomposition partitions the image so as to characterize its spatial color distribution (SCD). Two efficient SCD descriptors are proposed; the proximity histograms, which distils the histogram of inter-color distances and the proximity grids, which cumulate the spatial co-occurrence of colors in a 2D grid. Both approaches are configurable and provide means of modeling SCD in a scalar and directional way. A combination of the extracted global and spatial properties forms the final descriptor. This descriptor is neither biased nor does it become noisy from the presence of such color elements that are not visible for humans in both the spatial and the color domains. Finally a penalty-trio model fuses all color properties in a similarity distance computation during retrieval. Experimental results approve the superiority of the proposed technique against well-known global and spatial descriptors.
Contents

1 Introduction .................................................................................................................. 1

2 Content Based Image Retrieval ................................................................................... 4
   2.1 MPEG-7 Multimedia Content Description Interface ........................................... 5
   2.2 Visual Descriptors ............................................................................................... 7
   2.3 Similarity-based Indexing Methods ...................................................................... 11
   2.4 Similarity Metrics ............................................................................................... 14
   2.5 Query Methods ................................................................................................... 16
   2.6 Performance Evaluation Metrics ......................................................................... 17
   2.7 A Sample Content Based Indexing & Retrieval Framework: MUVIS ............... 19
      2.7.1 MUVIS Overview ....................................................................................... 20
      2.7.2 Main MUVIS Applications ......................................................................... 21
      2.7.3 MUVIS Extended Framework ...................................................................... 24

3 Color as a Feature ......................................................................................................... 26
   3.1 Human Color Vision ............................................................................................ 26
      3.1.1 Optical Vision .............................................................................................. 27
      3.1.2 Perceptual Vision ......................................................................................... 30
      3.1.3 Color Perceptual Rules ............................................................................... 34
      3.1.4 Gestalt Theory on Visual Perception ........................................................... 36
   3.2 Color Spaces ......................................................................................................... 39
   3.3 Color in CBIR ....................................................................................................... 45
      3.3.1 Global Color Descriptors .............................................................................. 45
      3.3.2 Spatial Color Descriptors ............................................................................ 48

4 A Novel Perceptual Color Descriptor .......................................................................... 56
   4.1 Descriptor Formation ............................................................................................ 57
      4.1.1 Proximity Histograms ................................................................................... 60
      4.1.2 Proximity Grids ............................................................................................. 63
   4.2 Similarity Distance – The Penalty Trio Model ....................................................... 65

5 Experimental Results .................................................................................................. 72
   5.1 Color-Based Retrievals in Synthetic Databases .................................................. 73
   5.2 Color-Based Retrievals in Natural Databases ...................................................... 76

6 Conclusion ................................................................................................................... 83

References ....................................................................................................................... 85

Appendix .......................................................................................................................... 96
List of Tables

Table 2-1: Query Types based on matching criteria................................................................. 17
Table 2-2 Supported Audio/Video/Image formats in MUVIS.................................................. 21
Table 5-1: Similarity distances and ranks of A and B in Figure 4-6 when C is queried in Corel_1K.................................................................................................................. 76
Table 5-2: ANMRR scores of the proposed and the competing descriptors for three Corel databases....................................................................................................................... 77
List of Figures

Figure 2-1: MPEG-7 Scope........................................................................................................... 6
Figure 2-2: MPEG-7 Structure.................................................................................................. 6
Figure 2-3: Sample Textures ...................................................................................................... 8
Figure 2-4: Different view angles of a human body (left) Occlusion (right) (modified from [138]).................................................................................................................. 10
Figure 2-5: Six spatial relationships defined in [106]. Top row is *left/right, up/down* and *touch* respectively. Bottom row is six possible *front* relationships................. 11
Figure 2-6: Kd-Tree partitioning of the space and associated Kd-Tree [83]......................... 12
Figure 2-7: R-Tree (left) and SS-Tree (right) Partitioning [83].............................................. 13
Figure 2-8: M-Tree partitioning and associated M-Tree [83]............................................... 14
Figure 2-9: A Precision-Recall Curve....................................................................................... 18
Figure 2-10: MUVIS Overview.............................................................................................. 20
Figure 2-11: DbsEditor User Interface ................................................................................... 22
Figure 2-12: MBrowser User Interface .................................................................................. 23
Figure 2-13: MBrowser Feature & Query Dialog ..................................................................... 23
Figure 2-14: FeX Module Interaction....................................................................................... 24
Figure 3-1: Newton's Experiment on Color Spectrum........................................................... 27
Figure 3-2: Electromagnetic Spectrum ................................................................................... 28
Figure 3-3: Structure of the Human Eye (left), Photoreceptor Distribution on the Retina  (right)................................................................................................................................. 28
Figure 3-4: Photoreceptor Sensitivities (S: Short, M: Middle and L: Long) [93]............... 29
Figure 3-5: Additive (left) and Subtractive (right) Mixing of Colors......................................... 30
Figure 3-6: Opponent Nature of Colors (left) and What Human Eye Perceives (right)  [32]............................................................................................................................... 31
Figure 3-7: The Effect of Simultaneous Contrast....................................................................... 32
Figure 3-8: Bartleson-Breneman Effect. The squares horizontally have the same color. ... 32
Figure 3-9: Bezold-Brücke Effect [12] ................................................................................. 33
Figure 3-10: Formation of the Afterimage ............................................................................. 33
Figure 3-11: SPD Distributions of two perceptually identical colors.................................... 34
Figure 3-12: Color Representations in English .................................................................... 35
Figure 3-13: Utilization of Dominant Colors ....................................................................... 35
Figure 3-14: Proximity Grouping .......................................................................................... 37
Figure 3-15: Similarity Grouping ........................................................................................... 37
Figure 3-16: The Factor of Common Fate ............................................................................ 37
Figure 3-17: The Factor of Closure (left & middle) and Good Curve (right) ....................... 38
Figure 3-18: Object and Background ..................................................................................... 38
Figure 3-19: RGB Color Space. Axes (left) and Illustration (right).................................. 39
Figure 3-20: Different RGB Spaces with different Spans [26] ............................................. 40
Figure 3-21: CMYK Color Space Axis (left), Typical color span in the Visible Spectrum (right) (modified from [26]) ................................................................. 41
Figure 3-22: HSV Color Space (left), HSV Color Wheel (right) ............................................ 42
Figure 3-23: CIE-XYZ Color Space [26] ............................................................................. 43
Figure 3-24: CIE-XYZ vs. CIE-Luv [122] .......................................................................... 44
Figure 3-25: CIE-Lab Color Space ....................................................................................... 44
Figure 3-26: Different color compositions of red, blue and white with same proportions (weights). .............................................................................................................. 48
Figure 3-27: Sample Image Partitioning Approaches ............................................................ 49
Figure 3-28: Same Content with Different Spatial Locations ............................................... 50
Figure 3-29: First 6 ranks of Correlogram retrieval (via QBE) in a 20K database. Top-left is the query image .................................................................................................................. 54
Figure 4-1: Overview of the proposed color descriptor formation ........................................ 58
Figure 4-2: Spatial Outliers and Quad-Tree Decomposition of different maximum depths (4, 6, 7) .............................................................................................................. 58
Figure 4-3: $N(b_i,k)$ templates in 8x8 block grid ($D_{Q^T}^{max} = 3$) for 4 range values in $L_v$ (top) and $L_i$ (bottom) norms. ................................................................. 62
Figure 4-4: The process of proximity grid formation for the block (X) for $L=4$. ............... 64
Figure 4-5: Proximity grid vs. histogram for a sample color pair: red-blue. ....................... 65
Figure 4-6: One-to-one matching of DC pairs among 3 images (A, B and C). ....................... 68
Figure 5-1: Query of a 3-color object (top-left) in synthetic database. ............................ 74
Figure 5-2: Comparison of the retrievals of a 4-block image (top-left in both images) between Correlogram (left) and proposed color histogram (right) ...................... 74
Figure 5-3: Synthetic Retrievals via Correlogram, Proximity histogram and Proximity Grid. ................................................................................................................................. 75
Figure 5-4: 4 typical queries using 4 descriptors in Corel_10K database. Top-left is the query image ......................................................................................................................... 78
Figure 5-5: 4 typical queries using 4 descriptors in Corel_20K database. Top-left is the query image ......................................................................................................................... 79
Figure 5-6: NMRR Results for Corel_10K (left) and Corel_20K (right)......................... 80
Figure 5-7: Two queries in Corel_10K (left) and Corel_20K (right) databases where (Auto-) Correlogram performs better than the proposed descriptor. Top-left is the query image ......................................................................................................................... 81
Figure 5-8: Dominant Color and QuadTree back-projections of some texture dominant images ................................................................................................................................. 81
# Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>2 Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>3 Dimensional</td>
</tr>
<tr>
<td>AAC</td>
<td>Advanced Audio Codec</td>
</tr>
<tr>
<td>AFeX</td>
<td>Audio Feature Extraction</td>
</tr>
<tr>
<td>ANMRR</td>
<td>Average Normalised Modified Retrieval Rank</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>AVR</td>
<td>Average Rank</td>
</tr>
<tr>
<td>CBIR</td>
<td>Content Based Image Retrieval</td>
</tr>
<tr>
<td>CBMR</td>
<td>Content Based Multimedia Retrieval</td>
</tr>
<tr>
<td>CCV</td>
<td>Color Coherence Vector</td>
</tr>
<tr>
<td>CIE</td>
<td>Commission Internationale de L'éclairage - International Commission on Illumination</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>D</td>
<td>Descriptor</td>
</tr>
<tr>
<td>DC</td>
<td>Dominant color</td>
</tr>
<tr>
<td>DCD</td>
<td>Dominant Color Descriptor</td>
</tr>
<tr>
<td>DLL</td>
<td>Dynamic Linked Library</td>
</tr>
<tr>
<td>DS</td>
<td>Description Scheme</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Video Disc</td>
</tr>
<tr>
<td>EMD</td>
<td>Earth Movers Distance</td>
</tr>
<tr>
<td>FeX</td>
<td>Feature Extraction</td>
</tr>
<tr>
<td>FV</td>
<td>Feature Vector</td>
</tr>
<tr>
<td>GLA</td>
<td>General Lloyd Algorithm</td>
</tr>
<tr>
<td>GLCM</td>
<td>Gray Level Co-occurrence Matrix</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>GoF</td>
<td>Group of Frames</td>
</tr>
<tr>
<td>GoP</td>
<td>Group of Pictures</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HCT</td>
<td>Hierarchical Cellular Tree</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>ISO/IEC</td>
<td>International Organization for Standardization/ International Electrotechnical Commission</td>
</tr>
<tr>
<td>MAM</td>
<td>Metric Access Method</td>
</tr>
<tr>
<td>MM</td>
<td>Multimedia</td>
</tr>
<tr>
<td>MP3</td>
<td>MPEG-1 Audio Layer 3</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>Mpel</td>
<td>Mega pixel</td>
</tr>
<tr>
<td>NMRR</td>
<td>Normalised Modified Retrieval Rank</td>
</tr>
<tr>
<td>NQ</td>
<td>Normal Query</td>
</tr>
<tr>
<td>PAM</td>
<td>Point Access Method</td>
</tr>
<tr>
<td>PQ</td>
<td>Progressive Query</td>
</tr>
<tr>
<td>QbE</td>
<td>Query by Example</td>
</tr>
<tr>
<td>QbR</td>
<td>Query by Region</td>
</tr>
<tr>
<td>QbS</td>
<td>Query by Subject</td>
</tr>
<tr>
<td>QT</td>
<td>Quad-Tree</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>SAM</td>
<td>Spatial Access Method</td>
</tr>
<tr>
<td>SBD</td>
<td>Shot Boundary Detection</td>
</tr>
<tr>
<td>SCD</td>
<td>Spatial color Distribution</td>
</tr>
<tr>
<td>SD</td>
<td>Similarity Distance</td>
</tr>
<tr>
<td>SEG</td>
<td>Segmentation</td>
</tr>
<tr>
<td>SPD</td>
<td>Spectral Power Distribution</td>
</tr>
<tr>
<td>VCD</td>
<td>Video Compact Disc</td>
</tr>
</tbody>
</table>
1 Introduction

Effective management of large data collections necessitates a successful Information Retrieval (IR) system. While it used to concern only professionals such as librarians and paralegals in the past; together with the developments in the hardware and networking technologies, IR has become a crucial part of daily life. Searching an e-mail through mailboxes, using a search engine in order to search the Internet or even searching personal collections to find images, videos or any kind of documents are all examples of IR. In addition to personal use, it is also exploited in various professional domains such as journalism, fashion, publishing, medicine, architecture, criminology etc. Earliest attempts in IR, dating back to 1970’s, include textual annotations (descriptions) of the content, which are then used for retrieval. However, such an approach did not find large acceptance due to its annotator dependency and requirement of massive human labor. Moreover, annotations are usually made in a single language, estranging it further from being a generic approach. Thus, together with the tremendous increase in the content to be retrieved, automatic description of the content is utilized letting the innovation of Content Based Multimedia Retrieval (CBMR). CBMR deals with the retrieval of multimedia (MM) items such as image, audio and video clips; where based on the specifications presented in a query, similar items are retrieved from a database. Content Based Image Retrieval (CBIR), on the other hand, focuses on description and retrieval via visual features only. In [33] Eakins defined three levels of abstraction for CBIR in order to characterize image queries. These are retrieval via:

- **Primitive** features such as color, texture or shape.
- **Logical** features such as the identity of objects shown.
- **Abstract** attributes such as the significance of the scenes depicted.

Most state of art CBIR systems are working with primitive features; even provide required semantics (logical or abstract features) through them. In addition to being objective, primitive features can be extracted directly from the media itself, without any external knowledge requirement. Logical features, on the other hand, require some level of semantic inference and outside knowledge in order to define and understand objects. Finally, abstract features require sophisticated reasoning in order to recognize meaning and purpose of objects and scenery.

The color composition is one of the discriminative properties of visual scenery due to its ease of interpretability and conspicuousness. Various color descriptors are utilized in
CBIR systems in order to describe the content of an image (or any visual media); however their retrieval performance is usually limited especially on large databases due to lack of the discrimination power of such color descriptors. One of the main reasons for this is because most of them are designed based on some heuristics or naïve rules that are not formed with respect to what humans or more specifically human visual system (HVS) finds **relevant** in color similarity. The word *relevance* is described as “the ability (as of an IR system) to retrieve material that satisfies the needs of the user”. Therefore, it is of decisive importance that human color perception is respected whilst modeling and describing any color composition of an image. In other words, if and only when a particular color descriptor is designed based entirely on HVS and human’s color perception rules, further discrimination power and hence certain improvements on the retrieval performance can be achieved.

Accordingly, the study of human color perception and similarity measurement within the color domain become crucial and there is a wealth of research performed in this field. For example in [14], Broek et. al. claimed that humans tend to focus on 11 basic color categories instead of the whole spectrum. In [85], Mojsilovich et. al. proposed five perceptual criteria (called as “basic color vocabulary”) important for comparing the color patterns as well as a set of rules (called as “basic color grammar”) governing the use of these criteria in similarity judgment. Following the fact that human eye cannot perceive a large number of colors at the same time, nor able to distinguish similar (close) colors well, they show that at the coarsest level of judgment, HVS primarily uses **dominant colors** (i.e. the few colors prominent in the scenery) to judge similarity. Additionally, spatial distribution of such color compositions is also of significant importance while evaluating the similarity of two color patterns. Several approaches tried utilizing color distribution as the spatial location and used fixed partitions of the image. However, the concept of *an eagle flying in the sky* is independent of the position of the eagle; whether on the left, right, upper or bottom of the image. This is following from the fact that human color perception is mostly affected from the relative proximities of the colors. In [51] Huang et. al. defined the color spatial distribution via a table called *color correlogram* indicating the spatial correlations of the colors. On the other hand, Ren et. al. in [106] incorporated directional positioning of two color regions as left, right, up, down, touch or front. Yet, for example, a bluish region next to a green region may represent either a lake in front of a hay field or sky over a hay field depending on their positions. However in an image where two people are walking together, it does not matter which one is on the left/right. While direction information may or may not be informative, design of a color descriptor should abide by some perceptual rules in order to reflect human color perception and increase its discrimination power as stated above.

Henceforth throughout this thesis, a systematic approach to extract such a perceptual color descriptor is presented and then an efficient similarity metric is proposed to achieve
the highest discrimination power possible for the color-based retrieval in general-purpose image databases. In order to remove the un-perceivable elements (henceforth referred as outliers) and to secure the global (perceptual) color properties, one alternative is to apply non-linear filters (e.g. median or Bilateral [123]); however there would be no guaranty that such a filter will remove all or the majority of the outliers and yet several filter parameters are needed to be set appropriately for an acceptable performance, which is not straightforward to do so especially for large databases. Instead, a top-down approach both in dominant color (DC) extraction and modeling their global spatial distribution is adopted in this thesis. This approach is in fact phased from the well-known Gestalt rule of perception, [131]: “Humans see the whole before its parts”, therefore, the method strives to extract what is the (next) global element both in color and spatial domain, which are nothing but the DCs and their spatial distribution within the image. In order to achieve such a (global) spatial representation within an image, starting from the entire image, quad-tree decomposition is applied to the current (parent) block only if it cannot host the majority of a particular DC, otherwise it is kept intact (non-decomposed) representing a single, homogeneous DC presence in it. So this approach tries to capture the “whole” before going through “its parts” and whenever the whole body can be perceived with a single DC, it is kept “as is”. Hence the outliers (few alien pixels, which cannot anyway be perceived in a big block of a particular DC) can be suppressed from the spatial distribution and furthermore, the resultant (block-wise) partitioned scheme can be efficiently used for a global modeling and due description of the spatial distribution. Finally a penalty-trio model uses both global and spatial color properties and performs an efficient similarity metric. After the image is (quad-tree) decomposed, this global spatial distribution is represented via inter-proximity statistics of the DCs, both in scalar and directional modes. These modes of spatial color distribution (SCD) can both describe the distribution of a particular DC with itself (auto SCD) and with other DCs (inter SCDs). The proposed method is fully automatic (i.e. without any supervision, feedback or training involved). Forming the whole process as a Feature eXtraction (FeX) module into MUVIS framework [90], allows testing the mutual performance in the context of multimedia indexing and retrieval.

The rest of the thesis is organized as follows. Before going into the details of the proposed approach, Chapter 2 describes the current state-of-the-art in CBIR and Chapter 3 explains color as a feature and presents the related studies in the area of color based CBIR, stressing particularly their limitations and drawbacks under the light of earlier discussion on human color perception. In Chapter 4 a generic overview of the proposed color descriptor is introduced together with the extraction, formation of the feature vector and calculation of the similarity distances. Chapter 5 presents the retrieval results of the proposed color descriptor on both synthetic and natural image databases. Chapter 6 concludes the thesis and suggests topics for future research.
2 Content Based Image Retrieval

Developments in multimedia technology in the past century altered the media world from analog to digital, resulting in not only putting every individual into a content creator position, but also transferring all the media already at hand into digital format, which significantly increases the amount of digital media available. Visual, aural and textual information can now be created, stored and modified digitally via everyday devices or software such as cameras, audio players, text editors, implying every single user may have a vast amount of such media. Moreover, improvements in network technology, especially in Internet, give rise to distribution of this content over an indefinitely large population, yet bringing forth the problem of management of this information for efficient accessibility.

For many years, libraries constructed library catalogues extracting significant content from the books such as title, author and published year. Similar indexing approaches are still being used wherever a large database is in question. The very first approaches on multimedia (MM) indexing utilize text-based annotations, analogous to library catalogues, where a text based description is linked to every database element. However the reliability of such methods is entirely annotator dependent and furthermore, they require vast amount of work especially with increasing database sizes. Such drawbacks are also encountered for any kind of manual indexing method which is why automatic indexing is essential. Automatic schemes intend to overcome any subjectivity and represent the content effectively, defining the scope of Content Based Multimedia Retrieval (CBMR). CBMR is an extensive research field comprising a tremendous amount of ongoing research. It mainly deals with the retrieval of similar MM items over large databases on the basis of their content. Here, three important expressions are ‘retrieval’, ‘similar’ and ‘content’, each of which outlines the nature of CBMR as follows: MM content is described via various features (visual, aural, etc.) and the similarity between each item is then estimated as (dis-)similarity distances (SDs) computed over the resultant feature vectors. Finally, images are retrieved according to their SDs via any querying scheme. In short, a similarity-based retrieval process mainly involves:

- Content description via Feature Extraction (FeX)
- SD calculation between feature vectors (FVs)
- Retrieval via a querying scheme
Considering the span of this thesis, the focus is narrowed down to Content Based Image Retrieval (CBIR), which deals with the retrieval of images from large collections based on their visual content.

CBIR systems aspire to extract an objective description of the content, still, as the Chinese proverb “One picture is worth ten thousand words” implies, content is subjective. In other words, the semantic interpretation of an image may not match its description via features, since the projection of a 3D view onto a feature space leads a considerable information loss. This discrepancy is referred as the semantic gap [113], which is an inevitable problem for all CBIR systems.

2.1 MPEG-7 Multimedia Content Description Interface

Starting from late 80’s MPEG (Moving Picture Experts Group), which is a working group of ISO/IEC, is suggesting solutions for the multimedia commerce. As mentioned earlier, multimedia collections are growing at a remarkable rate and MPEG proposed several standards to cope with both handling and management of this data. Most of them try to handle the storage or transportation problem via compression. MPEG-1 [53] deals with the coding of moving pictures and associated audio for digital storage media. It is employed in Video CD (VCD) and the well-known audio format MP3 is a part of MPEG-1 standard (MPEG-1 Audio Layer 3). MPEG-2 [54] takes care of generic coding of moving pictures and associated audio information by extending MPEG-1 to support broadcast (TV) quality transmission of digital video, thus making digital TV and DVD possible. MPEG-4 [55] represents aural, visual and audiovisual content called audio-visual objects being either natural or synthetic. They can be images, audio, video, text, 2D and 3D meshes, and synthetic face and body objects. MPEG-4 provides also efficient interactive multimedia transport at any bandwidth over a range of. MPEG-21 [15], on the other hand, defines a multimedia framework that supports the delivery and consumption of multimedia content in a secure environment. It is also referred as the Rights Expression Language. Among all the MPEG standards, MPEG-7 [56] is the only one concerning the content of the MM items, though, it does not standardize the content extraction; it standardizes how to represent the content (e.g. see Figure 2-1: MPEG-7 Scope). MPEG-7 is formally referred as Multimedia Content Description Interface and it basically defines the syntax for the description of the MM items via Descriptors (Ds) and Description Schemes (DSs). In other words it defines the language with its grammar and leaves the writing task to the programmer. Ds define the syntax and the semantics of each feature whereas the DSs specify the structure and semantics of the relationships between Ds and/or DSs. They are together referred as the Description Tools of MPEG-7.
The MPEG-7 standard consists of the following parts:

- **Systems**: The binary format for encoding MPEG-7 descriptions for efficient transport and storage.
- **Description Definition Language (DDL)**: The language for defining the syntax. It also allows definition of new DSs
- **MPEG-7 Visual**: Deals only with visual descriptions
- **MPEG-7 Audio**: Deals only with aural descriptions
- **Description Schemes (DSs)**: Generic features and multimedia descriptions
- **MPEG-7 Reference Software**: A software implementation of relevant parts of the MPEG-7 Standard with prescriptive status
- **Conformance Testing**: Guidelines and procedures for testing conformance of MPEG-7 implementations
CBIR applications are mainly interested in MPEG-7 Visual Description Tools, which consist of basic structures and Ds and utilize visual information such as color, texture, shape, motion and localization to extract elementary and sophisticated Ds. Additionally, MPEG-7 provides high-level description tools for face recognition applications. MPEG-7 Visual covers the following basic features and corresponding Ds:

- **Basic Elements**: Grid layout, Time series, 2D–3D multiple view, Spatial 2D Coordinates and Temporal interpolation.

- **Color**: Color space, Color quantization, Dominant colors, Scalable color, Color layout, Color structure and GoF/GoP color.

- **Texture**: Homogeneous Texture, Non-Homogeneous Texture (Edge histogram) and Texture Browsing.

- **Shape**: Object bounding box, Region-based shape, Contour-based shape, 3D shape descriptor.

- **Motion**: Camera motion, Object motion trajectory, Parametric object motion, Motion activity, Motion trajectory features (e.g., speed, direction, acceleration).

- **Localization**: Region Locator and Spatio–Temporal Locator.

### 2.2 Visual Descriptors

In order to describe the visual content, CBIR systems approach the problem from the perspective of human visual system (HVS). How HVS perceives a scene and how the relevant information is extracted leads the definition and classification of visual features. In that sense visual descriptors are classified into two groups as **low-level** and **high-level** features. Low-level features are extracted directly from the digital representation of the scene without any interpretation, such as color, texture and shape. On the contrary, high level features require some semantic understanding and subjectivity, which construes the objects and their interrelations. At this point, the aforementioned term, *semantic gap*, can also be described as the gap between high-level and low-level features. Current state of the art focuses mainly on narrowing this semantic gap. Thus, even in designing low-level descriptors, human perception plays an essential role. Color, texture and shape are the three basic low-level features standardized in MPEG-7. They merely form the basis of CBIR
systems and many sophisticated algorithms have been proposed for describing the visual content via these features.

**Color** composition of an image can turn out to be a powerful feature for the purpose of CBIR, if extracted in a perceptually oriented way and semantically intact. Furthermore, color structure in a visual scenery is robust to noise, image degradations, changes in size, resolution and orientation. Eventually most of the CBIR systems use various color descriptors in order to retrieve relevant images (or generally speaking the visual multimedia material). A detailed discussion of color feature and descriptors is left to Chapter 3.

**Texture**, whilst lacking a solid and unique definition, can still be described roughly as ‘the representation of the feeling of an object’s surface’.

The dominant characteristics of textures in Figure 2-3 can distinctly be observed either as repetitive or regular patterns; however, due to the intuitive nature of texture there are vast amount of definitions. Zucker et al. in [140] states that “Texture is an apparently paradoxical notion. On the one hand, it is commonly used in the early processing of visual information, especially for practical classification purposes. On the other hand, no one has succeeded in producing a commonly accepted definition of texture. The resolution of this paradox, we feel, will depend on a richer, more developed model for early visual information processing, a central aspect of which will be representational systems at many different levels of abstraction. These levels will most probably include actual intensities at the bottom and will progress through edge and orientation descriptors to surface, and perhaps volumetric descriptors. Given these multi-level structures, it seems clear that they should be included in the definition of, and in the computation of, texture descriptors.”. On the other hand Tamura in [120] says “We may regard texture as what constitutes a macroscopic region. Its structure is simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule.”

Even though no precise definition for textures exists, as in [21] there are still some properties they are assumed to obey:

- “Texture is a property of areas; the texture of a point is undefined. So, texture is a contextual property and its definition must involve gray values in a spatial
neighborhood. The size of this neighborhood depends upon the texture type, or the size of the primitives defining the texture.

- Texture involves the spatial distribution of gray levels. Thus, two-dimensional histograms or co-occurrence matrices are reasonable texture analysis tools.

- Texture in an image can be perceived at different scales or levels of resolution. For example, consider the texture represented in a brick wall. At a coarse resolution, the texture is perceived as formed by the individual bricks in the wall; the interior details in the brick are lost. At a higher resolution, when only a few bricks are in the field of view, the perceived texture shows the details in the brick.

- A region is perceived to have texture when the number of primitive objects in the region is large. If only a few primitive objects are present, then a group of countable objects is perceived instead of a textured image. In other words, a texture is perceived when significant individual “forms” are not present.”

Depending upon the particular application, vast amount of texture descriptors are present at the current state of art. One group of descriptors, which are called statistical methods, arises from the statistical properties of textures and tries to describe them by relying on the distribution of gray-level intensities. Probably the most well known and widely used example of this group is the Gray Level Co-Occurrence Matrix (GLCM) [98], which estimates the distribution of intensity level pairs at a certain distance and direction. Higher orders statistics are still applicable in the expense of computational complexity but still first and second order statistics are most commonly used. Another perspective in describing the texture is getting use of the texture elements and their geometric properties, the so called geometric methods. Even the definition of texture element is vague that there exists several terms such as textons [57] or tokens [79]. Model based methods try to construct mathematical methods in order to describe the textures. Mandelbrot [77] proposed the term ‘fractals’ and stated that an image is self-similar if it has similar structures at different scales. One last method for texture description is signal processing methods, which analyze the texture in different domains. Following from the fact that humans perform a frequency analysis over an image [16], [43], frequency domain approaches such as Fourier, Gabor and Wavelet transforms are also used.

Shape can be defined as the surface arrangement that outlines an object giving it a definite form, but this is still lacking a precise definition. Defining the shape of an object may be much more difficult when the flexibility of the object is considered. A running and a standing man may differ is shape but they still represent the same object, or a folded paper also differs in shape from its original form. On top of that, when projections are
considered, shape descriptors have to deal also with problems such as occlusions and projection angle (e.g. see Figure 2-4).

![Figure 2-4: Different view angles of a human body (left) Occlusion (right) (modified from [138])](image)

Describing a shape is mainly done in two ways: contour-based or region-based. Contour-based methods, such as Chain code [40], Shape Signature [29], Polygonal Approximation [22], Autoregressive Models [62], Fourier Descriptors [102], [136] and Curvature Scale Space Descriptors [87], [88], exploit the boundary information to describe the shape which is an essential part of human perception in defining the shape. Region-based methods use the whole pixels inside the region assuming that the object may be composed of a set of regions or may have holes in it. Geometric Moments [82], Zernike Moments [121], [80], Grid Representation [71] exploits the region based approach. Zhang et. al claimed in [137] that a shape representation should be affine invariant, robust, compact, easy to derive and matching, and perceptually meaningful. Thus Fourier Descriptors, Curvature Scale Space Descriptors, Zernike Moments and Grid Representation have been acknowledged to be suitable for CBIR. MPEG-7 utilizes Zernike moments for region-based description and Curvature Scale Space Descriptors for contour-based approach.

Since the shape feature describes the shape of an object, it requires a successful object extraction beforehand. However, this issue is still an unresolved problem in image processing field. In other words the inaccuracy in object extraction or segmentation algorithms at the current state diminishes the usage of shape descriptors in CBIR.

In addition to these features, spatial layout can also be considered as a visual feature. Even if the color, texture and shape of two objects/regions are similar, their spatial location may be a discriminative factor. For example, a blue region with certain texture and shape may represent either sky or sea, while sea is most likely in the lower portion of the image.
Content Based Image Retrieval

and sky is in the upper portion. Several studies tried to exploit the spatial layout according to the location of the region in an image. Song et. al. [116] and Mojsilovic et. al [86] utilized spatial positions of the regions simply as upper, bottom, top to convey spatial layout. In another approach, Ma et. al. [73] used the minimum bounding rectangle and Mezaris et. al. [81] used the region centroid to provide spatial position of a region. However, the relative location, in other words the interpositions of regions is of more importance from the HVS perspective. For instance Egnehofer and Franzosa [34] suggested the relationship between objects in eight categories, which are equal, inside, cover, overlap, touch, disjoint, covered-by and contains. Chang et al. [19], [20] represented this relationship in a directional way such as such as left/right, below/above. Ren et al. [106] claimed that direction alone is not sufficient to describe region interrelations and defined them in six categories according to region centroids as in Figure 2-5: left, right, up, down, touch and front. Smith et. al. defined composite region templates (CRTs) that define prototypal spatial orderings of the regions throughout the image. They further claimed that the relative horizontal position is less important that the relative vertical position of the regions.

![Figure 2-5: Six spatial relationships defined in [106]. Top row is left/right, up/down and touch respectively. Bottom row is six possible front relationships.](image)

### 2.3 Similarity-based Indexing Methods

CBIR systems are required to retrieve the images based on their visual content, which is achieved via mapping the database primitives to a high dimensional (feature) space. This brings along the problem of accessing the information through their high dimensional representation, particularly for large databases. The aim of all access methods is to achieve the fastest retrieval by minimizing the file access and similarity computations during searching (query). Most of the database access methods are based on partitioning and they are mainly distinguished by the type and method of partitioning. Methods, which partition the feature space are called Point Access Methods (PAMs), ones which consider the partitioning of the database are called Spatial Access Methods (SAMs) and finally Metric Access Methods (MAMs) partition the feature space based on the SDs.
Attempts in PAMs include Kd-Trees and some hashing approaches. Hashing based approaches try to store the objects closer, which are closer in the original space; however they only handle the case of point queries (Section 2.5). In designing Kd-Trees, in [83] it is stated that “the idea is to hierarchically partition the data space by an isooriented hyperplane, passing through a data point. The orientation of the separating hyperplane is chosen by alternating between dimensions at each level of the hierarchy”. Figure 2-6 shows a sample Kd-Tree partitioning. A range of Kd-Tree variants are present such as Adaptive (Balanced) Kd-Trees [6] and BSP-Trees (Binary Space Partitioning Trees) [41].

![Figure 2-6: Kd-Tree partitioning of the space and associated Kd-Tree [83]](image)

While PAMs deal with data points, SAMs extended the focus to database objects via grouping the data points. Database objects are modeled as spheres, rectangles or any other geometric object. In that sense, SAMs are extended versions of PAMs via transformation (object mapping), overlapping regions (object bounding), clipping (object duplication) or multiple layers [42]. R-Tree [48] is an example of SAMs, trying to partition the data as minimum bounding rectangles. SS-Tree [132], on the other hand, uses minimum bounding spheres for partitioning (Figure 2-7). A compound method, SR-Tree [59], utilizes both minimum bounding rectangles and spheres while using rectangles for constructing the tree and spheres for increasing the selectivity. TV-Tree [69] and X-Tree [9] are other models of SAMs.
MAMs deal only with the relative distances of data; meaning that utilization of MAMs as an indexing scheme requires only the definition of a SD satisfying some metric requirements as stated in Section 2.4. As the data space is not of any concern, any kind of distance metric is applicable. Despite PAMs and SAMs, MAMs try to minimize the SD calculation time, which can be significant in high dimensions. However, the biggest advantage of MAMs over other indexing methods is that while PAMs and SAMs works on a single feature space, MAMs can work on databases consisting of several features; which, in fact, is usually the case for multimedia databases where several features (visual, aural etc.) are present. M-Tree (Metric Tree) [25] is an effective example of MAMs, where a set of points are chosen and every data point is associated to its closest representative (Figure 2-8). The radiuses of the partitions in Figure 2-8 are the distances of the representatives to their farthest data point. The data points are partitioned until a maximum number of points per leaf is obtained.

Figure 2-7: R-Tree (left) and SS-Tree (right) Partitioning [83]
Kiranyaz and Gabbouj state in [65] that even though M-tree (and its variants [124], [139], [23], [24]) provide dynamic access to the database, their performances may significantly vary depending on the position of the object of interest. Thus they proposed a MAM based, dynamic and self-organized indexing scheme, the so-called Hierarchical Cellular Tree (HCT). It is mainly a hierarchical clustering method where items are partitioned depending on their relative distances and stored within cells on the basis of their similarity proximity.

The curse of dimensionality, stated by Bellman [4], is a bottleneck that any high dimensional access method should remedy. It is simply the fact that the number of partitions increases exponentially with increasing dimensionality. The efficiency of access methods may even lie below sequential indexing [83], which is the straightforward searching of data points. Numerous methods have been proposed to cope with curse of dimensionality and achieve high efficiency; however they rely on either data approximations [128], [129], [7], [110], [8] or query approximations [5], [68], [45], [52], [35] resulting in worse performance since they tolerate some error in the result set.

2.4 Similarity Metrics

Regardless of the feature representation and indexing methodology exploited, the similarity between two database primitives, for instance images, should somehow be formulated. In CBIR in order to reflect human perception, a similarity metric should yield smaller SD for similar images and vice versa for the opposite. Numerous approaches exist for similarity distance formulation between two data points or feature vectors but still, just like any other metric, similarity metrics need to satisfy a number of properties; namely a metric must be nonnegative, reflexive and symmetric. Moreover, it must satisfy the triangle inequality.
Let \( R \) denote the real numbers and \( R^d \) be a real \( d \)-dimensional vector space. If a metric is defined as a function \( m( , , ) \) from \( R^d \times R^d \) to \( R \); then for any vector \( x, y, z \) inside the feature space, \( m \) should satisfy:

- \( m(x, y) \geq 0 \) \hspace{1cm} \text{(nonnegative)}
- \( m(x, y) = 0 \ \text{iff} \ x = y \) \hspace{1cm} \text{(reflexive)}
- \( m(x, y) = m(y, x) \) \hspace{1cm} \text{(symmetric)}
- \( m(x, z) \leq m(x, y) + m(y, z) \) \hspace{1cm} \text{(triangle inequality)}

The requirements of nonnegativity and reflexivity signify the fact that high similarity means low SD and, therefore, identical images should have the lowest possible SD (i.e. SD=0). The idea that images are mutually similar is stated by the symmetry requirement. In other words, if an image \( A \) is similar to image \( B \), than \( B \) is similar to \( A \). The triangle inequality ensures that no image can be similar to two dissimilar images at the same time. Still early researches in psychology claim that human perception may not necessarily follow triangular inequality and symmetry properties, e.g. see [2], [125].

Various similarity metrics are exploited in CBIR, such as Minkowski-Form Distances [83], Quadratic Distance [36] and Earth Movers Distance (EMD) [100]. The Minkowski distance between two vectors \( x = \{x_1, x_2, \ldots, x_N\} \) and \( y = \{y_1, y_2, \ldots, y_N\} \), which is also named as \( L_p \)-norm, is defined as

\[
L_p(x,y) = \left(\sum_{k=1}^{N}|x_k - y_k|^p\right)^{1/p}
\]  

(1)

Special cases of Minkowsky distances are popularly used, such as \( L_1 \) (Manhattan distance), \( L_2 \) (Euclidean distance). As \( p \to \infty \) it forms the Maximum distance or Chebyshev distance, which can be formulated as \( \max\{|x_k - y_k|\} \). However, Minkowski distances do not take into account the relation across dimensions. This issue is addressed in [36], and quadratic distance between two vectors \( x \) and \( y \) (let them be the abovementioned vectors) is proposed as follows:

\[
d_{quad}(x,y) = [(x-y)A(x-y)]^{1/2}
\]

\[
A = [a_{ij}], \quad a_{ij} = 1 - \frac{d_{ij}}{d_{max}} \quad \text{and} \quad d_{ij} = |x_i - y_i|
\]  

(2)

where \( a_{ij} \) is the similarity coefficient between dimensions. This formulation allows the comparison of different dimensions with a certain amount of inter-similarity between them in the expense of computational complexity; however it underestimates distances because it tends to emphasize the dimension similarity [118]. Furthermore Po and Wong in a recent study [103] show that the quadratic distance formulation has serious limitations in finding
color SDs: it does not match the human color perception well enough. Hence it gives even worse results than the naïve $L_p$-norm on some particular cases. Mahalanobis distance is a special case of quadratic distance, where the similarity matrix is the inverse of the covariance matrix ($A = \sum^{-1}$). Note also that if the covariance is equal to the identity matrix ($\sum = I$), the resultant metric is the Euclidean.

EMD is first proposed by Peleg, et. al. [100] then applied to CBIR by Rubner et al. [109], and defined as “the minimal cost that must be paid to transform one distribution into the other”. In order to capture the feature distribution well, distributions are represented as weighted features, so called signatures. In accordance with Rubner’s definition, EMD can be formulated roughly as,

$$EMD = \sum (\text{distance moved}) \times (\text{amount moved})$$

or more formally,

$$EMD(X, Y) = \frac{\sum \sum d_{ij} f_{ij}}{\sum \sum f_{ij}}$$

Measurement of the distance between two distributions entails the definition of the distance between two single features, which is called ground distance ($d_{ij}$). For example, in case of color, $d_{ij}$ are distances between individual colors $i$ and $j$ whereas $f_{ij}$ is chosen to minimize the numerator in Eq. (4), which is the paid cost. Rubner used EMD to compute a perceptual distance between images; however, the most apparent drawback of EMD is its high computational complexity.

2.5 Query Methods

Query is the process of requesting information from a database, in other word it is ‘how you search’ for the information. According to Yoshitaka [134], there are two methods to make a search (query): Query by Subject/Object (QbS) and Query by Example (QbE).

QbS requires knowledge assistance to provide semantic description of the content. A simple way to do so is the annotation of database items, where each database item is associated with a text description and queried via a text box. Due to their deficiency in describing the content adequately, retrieval based on the semantics meant by the annotation is proposed [130], [18]. This approach utilizes explanation of queries via semantically equivalent expressions and evaluation of these expressions. Yet, annotation based approaches suffer from impracticality considering the current multimedia database volumes.
QbE provides the user a way to submit the query criteria via an example. Many choices for QbE are present, depending on the application and user needs. If the point of issue is CBIR, the example is presented either as another image or a sketch by the user. Many systems, such as QBIC [36], Photobook [101], MUVIS [90], MARS [96], utilize QbE approach. Leung [67] presented a trademark retrieval approach via query by sketch and a sketch retrieval method for color images is presented in [60]. For audio databases, Ghias et al. [44] presented an approach where the sample audio is presented as humming of the user. Also popular audio-streaming websites such as Last.fm [66] and Pandora [97], attempts to utilize QbE scheme by getting the song or artist name as the query. Still QbE has some defects in reflecting the visual content. For example a view of sea, a ship or sunset may all be present in the same image yet they all reveal particular interests. Assuming that the objects in an image are extracted appropriately, the user may also select an object or region as an example forming a higher level querying which is also referred as Query by Region (QbR), used by systems such as Blobworld [17] and NeTra [75].

Apart from the method used, queries can also be classified depending on their matching criteria. Given that \( m(.,.) \) is a distance metric, \( q \) is the query item and \( x \) is the target item, Table 2-1 explicates query types and their matching criteria.

<table>
<thead>
<tr>
<th>Query Type</th>
<th>Matching Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Point Query:</strong></td>
<td>( m(q,x) = 0 )</td>
<td>Exact match case(s) are retrieved</td>
</tr>
<tr>
<td><strong>Range Query:</strong></td>
<td>( m(q,x) \leq r \quad r \in \mathbb{R}^+ )</td>
<td>Items inside a distance range are retrieved</td>
</tr>
<tr>
<td><strong>K-Nearest Neighbor:</strong></td>
<td>( m(q,x) \leq r_k )</td>
<td>( K \in \mathbb{R}^+ ) closest items are retrieved</td>
</tr>
</tbody>
</table>

\( r_k \): the distance to the K nearest neighbors

2.6 Performance Evaluation Metrics

CBIR systems are mainly not concerned in the abovementioned point queries which can retrieve only exact matches, as they aspire to retrieve a number of similar (in content) images among a database. Since the notion of similarity is subjective and feature dependent, if a quantitative assessment on the performance of a CBIR system is to be made, the database should be constructed such that it comprises meaningful and semantically intact classes of similar images. Assuming that such appropriate data, namely the expected output or the ground truth data, is present, a quantitative performance metric should be defined in order to evaluate the competence of the system.
One of the favored metrics for performance evaluation in information retrieval is over the terms \textit{precision} and \textit{recall}. Recall is defined as the probability of retrieving an image given that it is similar, and likewise precision is defined as the probability that an image is similar given that it is retrieved [72]. They can be formulated as follows:

\[
\text{precision} = \frac{\text{Number of relevant items retrieved}}{\text{Number of items retrieved}}
\]

\[
\text{recall} = \frac{\text{Number of relevant items retrieved}}{\text{Total number of relevant items}}
\]

Graphical representation of such terms as \textit{Precision-Recall Curves} is commonly used in CBIR area. In case of an ideal retrieval, \textit{precision} value is equal to 1 regardless of the \textit{recall} value. Conversely, the worst case is equivalent to a random retrieval where the precision value is equal to the probability of finding a relevant item among the whole set. Any retrieval system having an equal or lower precision rate than this probability is simply incompetent. As the number of retrievals increase, the increase in irrelevant retrievals is practically larger than relevant retrievals, yielding a trade-off between precision and recall. A sample \textit{Precision-Recall} curve is shown in Figure 2-9.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure29.png}
\caption{A Precision-Recall Curve}
\end{figure}

\textit{Precision} and \textit{recall} work in a binary manner, regarding only the information of relevancy. However, an important issue that should be considered in performance of a retrieval process is the rank information. This problem is addressed in MPEG-7 by defining the \textit{Average Normalized Modified Retrieval Rank} (ANMRR) as the retrieval performance criteria, which can be formulated as follows:
Content Based Image Retrieval

\[ \text{AVR}(q) = \frac{\sum_{k=1}^{N(q)} R(k)}{N(q)} \quad \text{and} \quad W = 2N(q) \]
\[ \text{NMRR}(q) = \frac{2\text{AVR}(q) - N(q) - 1}{2W - N(q) + 1} \leq 1 \]
\[ ANMRR = \frac{\sum_{q=1}^{Q} \text{NMRR}(q)}{Q} \leq 1 \]  

where \( N(q) \) is the minimum number of relevant (via ground-truth) images in a set of \( Q \) retrieval experiments, \( R(k) \) is the rank of the \( k^{th} \) relevant retrieval within a window of \( W \) retrievals, which are taken into consideration during per query, \( q \). If there are less than \( N(q) \) relevant retrievals among \( W \) then a rank of \( W+1 \) is assigned for the remaining (missing) ones. \( \text{AVR}(q) \) is the average rank obtained from the query, \( q \). Note that if all \( N(q) \) retrievals are relevant, then \( \text{NMRR}(q) = 0 \), the best retrieval performance is thus achieved. On the other hand, if none of relevant items can be retrieved among \( W \) then \( \text{NMRR}(q) = 1 \), as the worst case. Therefore, the lower \( \text{NMRR}(q) \) is the better (more relevant) the retrieval is, for the query, \( q \).

2.7 A Sample Content Based Indexing & Retrieval Framework: MUVIS

A CBIR system lets the user to index and retrieve media databases via any of the aforementioned schemes. While most of such systems emphasize only one aspect of CBIR, few provide any variety or the opportunity of feature selection. Princeton’s PicHunter [27] deals with the color features only and likewise Columbia’s VisualSEEK [114] utilizes only a DC approach together with elementary shape descriptors. However, systems like IBM’s QBIC [36] makes use of all color/shape/texture features and also allows queries based on example images, user-constructed sketches and/or selected color and texture patterns. MARS [96] also provides queries over combination of the same features and textual descriptions. A detailed comparison of various CBIR systems can be found in [127].

Accompanied by the improvements in related technologies, development of CBIR systems has received significant attention from the academic and commercial community in recent years. MUVIS [90] is an emerging CBIR system developed in the Institute of Signal Processing at Tampere University of Technology. It was initially implemented for indexing and retrieval of image databases over visual features in late 90’s. As of the beginning of 21st century, it revolved into a unified solution for content based indexing and retrieval.
2.7.1 MUVIS Overview

Besides being mainly designed for multimedia management (indexing, browsing, querying, summarization, etc.), MUVIS framework offers also implementation and testing of third party algorithms over FeX, AFeX, SEG and SBD modules (Section 2.7.3); thus, it can be used as a test-bed platform to develop and test new techniques in the context of CBMR. Additionally it proposed a novel query technique named Progressive Query (PQ) [64] together with an indexing scheme called Hierarchical Cellular Tree (HCT) [65].

MUVIS is developed for Windows platform (except 3.1 and NT), and there is also a JAVA based implementation, M-MUVIS, for mobile devices [1]. MUVIS supports various MM types in various formats (Table 2-2) and stores them in related databases together with associated indexing information. A hybrid database contains both images and video clips whereas the other database types only carry a single media type (see Figure 2-10).
2.7.2 Main MUVIS Applications

MUVIS primarily employs two applications: *DbsEditor* (Indexing) and *MBrowser* (Retrieval). Additionally, another application called *AVDatabase* creates real-time audio/video databases by capturing from an external device and using various compression techniques.

1. **DbsEditor**

MUVIS handles database indexing, management and MM conversions via DbsEditor. Any kind of MUVIS database can be created and any supported MM file (Table 2-2) can be inserted into the database. Feature extraction is another task of DbsEditor via integration and management of *FeX, AFeX, SEG* (Segmentation) and *SBD* (Shot Boundary Detection) modules (*Section 2.7.3*). It lets the selection of features together with their parameter values; thus extraction of different sub-features via setting different values to parameters of a certain feature is possible (Figure 2-11 and Figure 2-13). It also provides preview for any clip or image in a database. DbsEditor’s GUI is shown in Figure 2-11 together with some potential actions that it is capable.

### Table 2-2 Supported Audio/Video/Image formats in MUVIS

<table>
<thead>
<tr>
<th>Audio</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MUVIS Audio</strong></td>
<td><strong>MUVIS Video</strong></td>
</tr>
<tr>
<td><strong>Codec</strong></td>
<td><strong>Sampling Freq.</strong></td>
</tr>
<tr>
<td>WMV</td>
<td>16, 22, 800</td>
</tr>
<tr>
<td>AAC</td>
<td>24, 44.1, 1 KHz</td>
</tr>
<tr>
<td>QuickTime</td>
<td></td>
</tr>
<tr>
<td>PCM</td>
<td>8 - 16 KHz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>MUVIS Image Types</strong></th>
<th><strong>Frame Formats</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitmap</td>
<td>JPEG</td>
</tr>
</tbody>
</table>

206x393

21
II. MBrowser

MBrowser is mainly in charge of MM retrieval and browsing of any MUVIS database created by DbsEditor. It utilizes QbE scheme for querying and allows two query types: Normal Query and Progressive Query [64]. Any database item can be displayed and posed for querying in a MUVIS database with a weighted combination of existing features (Figure 2-13). While exclusive images are allowed for querying the active database, an external audio/video clip should first be appended to the MUVIS database before querying. MBrowser supports display for single frame, shot frames, scene frames, partial video (ROI) and entire video; also enables video summarization through shot and scene frames. The GUI of MBrowser is shown in Figure 2-12 where the query image for a QbE operation, the first page of the retrieval results and feature vectors of the query and any target image are shown. The GUI also supports segmentation masks for images and video clips, shot boundaries and key-frames for video clips.
Figure 2-12: MBrowser User Interface

Figure 2-13: MBrowser Feature & Query Dialog
2.7.3 MUVIS Extended Framework

As mentioned earlier, MUVIS provides a framework structure for the implementation of the third party applications for visual/aural feature extraction, segmentation and shot boundary detection algorithms over distinct modules:

\[
\begin{align*}
\text{FeX} & : \quad \text{Visual Feature Extraction} \\
\text{AFeX} & : \quad \text{Aural Feature Extraction} \\
\text{SBD} & : \quad \text{Shot Boundary Detection and Key-Frame Extraction} \\
\text{SEG} & : \quad \text{Spatial Segmentation}
\end{align*}
\]

As the scope of this thesis falls into visual feature extraction, particular attention is given to \textit{FeX} framework in the rest of this section. The other modules, as well as the \textit{FeX} module, are detailed in [47].

Any \textit{FeX} algorithm is implemented as a DLL (Dynamic Linked Library) using the API (Application Programming Interface) header file \textit{FeX.API.h}, which defines the required functions in order to manage the feature extraction operations. Both DbsEditor and MBrowser interact with the module through these API functions, as illustrated in Figure 2-14.

![Figure 2-14: FeX Module Interaction](image)

- \textit{FeX.Bind}: Handshaking between MUVIS applications and the module
- \textit{FeX.Init}: Initialization of the module
- \textit{FeX.Extract}: Implementation of the feature extraction algorithm
- \textit{FeX.Exit}: Termination of the module operation
- \textit{FeX.GetDistance}: Implementation of the SD between two feature vectors
After \texttt{FeX\_Bind} provides the necessary handshaking with a MUVIS application (DbsEditor for instance), \texttt{FeX\_Init} initializes the feature parameters. \texttt{FeX\_Extract} function is then called for every database item and returns the associated feature vector. Once all the features are extracted, \texttt{FeX\_Exit} resets the module and terminates the operation concluding the indexing (feature extraction) procedure. \texttt{FeX\_GetDistance} function is called by the MBrowser application in order to calculate the SD between two feature vectors. If the queried item does not belong to the current database, MBrowser calls \texttt{FeX\_Extract} to acquire the feature vector for that item.

MUVIS has several FeX modules already implemented within: RGB, HSV, YUV Histograms, MPEG-7 Dominant Color [103], Color Correlogram [51], GLCM [98], MPEG-7 Edge Histogram [133] and 2D-Walking Ant Histogram [38].
3 Color as a Feature

Color is a key-element in describing visual scenery. Although the sensory processing of color is well understood (e.g. see Section 3.1.1), less is known about how it is processed in the brain. One theory assumes that color is processed independent of other visual features (such as shape, texture, depth, motion) and that the function of color in image processing is rather limited [70]. Other approaches have proposed that color plays a particular role in some image segmentation tasks, such as detecting berries amongst bushes [89], [104]. In any case, color appears to be an important cue for the rapid identification and recognition of a visual scene.

Owing to its significance in human visual perception, color is one of the most widely used low-level features in CBIR purposes. Accurate description of a color composition of an image entails a proficient color representation. Color spaces are defined for such representation based on the desired application (as explained in Section 3.2). In order to keep up with the application demands, they are usually based on human color vision; either optical (Section 3.1.1) or perceptual (Section 3.1.2). The right choice of color space increases both descriptive and discriminative power of any color descriptor in CBIR (Section 3.3).

3.1 Human Color Vision

Modern comprehension of color dates back to 17th century and particularly to the era of Isaac Newton (1642-1726). He is the first one to propose that light is responsible from color, by obtaining different colors of light via refracting it through a prism. At that time, people believed that prism was coloring the light, so to disprove this, he refracted the light back together again through a prism forming a white light as seen in Figure 3-1: Newton's Experiment on Color Spectrum. So he claimed that sunlight comprises individual rays with different refrangibilities and color is an innate property of light.
In 1810 Johann Wolfgang von Goethe (1749-1832), the famous writer, objected Newton’s idea that light alone is responsible from color; and he proposed the effect of perception on color vision. He claims that physically same colors may appear different to humans or conversely, different colors may be perceived physically same by the human brain under certain circumstances. According to Goethe, color vision depends on the object, the lighting and perception [32]. He claims that colors arise at the borders, where light and dark flow together. As stated in [117], “Goethe pictures to himself that light and darkness relate to each other like the north and south pole of a magnet. The darkness can weaken the light in its working power. Conversely, the light can limit the energy of the darkness. In both cases color arises”. According to his theory, yellow is a light, which has been dampened by darkness and blue is a darkness weakened by the light. Although Goethe’s reification of darkness resulted in rejection of his work by most of the physicists, he still was able to draw the distinction between optical and perceptual vision which will be detailed in the following sections. Optical vision involves the study of physics and physiology while perceptual vision is mostly connected with psychology.

### 3.1.1 Optical Vision

Physics of color is generally explained by Newton’s experiments which accept light as the source of color. According to him, color is observed as the light is reflected or refracted by objects and reaches the eye. He extended his experiments to further split the spectrum he observed by letting only one color pass through the prism; and observing no further splitting, he finally concluded that it was the whole color spectrum.

In fact, the spectrum Newton observed was only the *visible spectrum* that is a minuscule fraction of the entire electromagnetic spectrum as illustrated in Figure 3-2. Sun light is a continuous distribution of wavelengths and different refraction of each color it acquires through the Newton’s prism is rooted in different wavelengths of each color. The wavelength range of visible spectrum is approximately from 400*nm* to 700*nm*, varying from red to violet and forming the so called *spectral colors*. 

---

**Figure 3-1: Newton’s Experiment on Color Spectrum**
Figure 3-2: Electromagnetic Spectrum

The detection of the visual spectrum takes place in the eye and subsequently the information is sent to the brain to be processed. The eye has photoreceptors located in the retina that are responsible for the reception of the light. There are two kinds of such receptors that serve different functions: rods and cones. Rods provide vision at low light levels and thus serve vision when little light is available, such as at night. They are very sensitive to light and they let detection of light that is over a billion times dimmer than bright daylight; however they provide a coarse and colorless image. On the other hand, cone photoreceptors provide sharp vision under daily life illumination. Most of our daily activities are performed in daylight and at room light levels; thus, most of the time our vision is based on cones.

There are around 120 million rods and 6-7 million cones on the retina. Cones are concentrated on the fovea where there are no rods present. It is a small portion of the retina and the eye tends to focus the light onto that region. However, as depicted in Figure 3-3, while cones show an enormous density on the fovea, they almost seize to exist outside that region yielding a poor peripheral vision.

Figure 3-3: Structure of the Human Eye (left), Photoreceptor Distribution on the Retina (right)
The ability to ‘see’ color is rooted in cones, which requires multiple classes of cone photoreceptors. Human color vision is trichromatic and requires three types of cones, which differs in their spectral sensitivities. They are commonly referred as Red, Green and Blue (R, G and B) receptors however naming them after single colors may be misleading since they span a range of spectral wavelengths. So a naming such as Short, Middle and Long (S, M and L) is more convenient and well accepted in the field of color vision (Figure 3-4). In terms of population, about 64% of the cones are L, about 32% M, and about 2% are S-cones. The S-cones have the highest sensitivity and are mostly found outside the fovea. The response of cones is not binary. No matter how complex the composition of wavelengths is, it is reduced to three color components over cones. Light of a given wavelength (color), say 500 nm (green), stimulates all three types of cones, but the M-cones will be stimulated most strongly since their sensitivity is the highest for that wavelength (Figure 3-4). So each cone produces a signal to be transmitted to the brain. The values that each cone produces are referred as tristimulus values.

![Figure 3-4: Photoreceptor Sensitivities (S: Short, M: Middle and L: Long) [93].](image)

The trichromatic theory, formulated in Eq. (7), was put forward first by Young in [135], and later expanded by Helmholtz in [50], way before the physiological discovery of the three photoreceptors in the eye. The theory indicates that any color $C$ can be obtained by linear combination of three separate light sources known as primary colors ($C_1$, $C_2$ and $C_3$); known as additive mixing. It should be noted that the magnitudes of $C_i$ may not be of the same size since the sensitivity of the eye would be different for each of the primary colors.

$$C = C_1I_1 + C_2I_2 + C_3I_3$$

(7)
$I_i$ can be referred as the relative intensities of the primary colors. Although any color could be used as a primary color, use of monochromatic colors enables the broadest range of colors to be obtained by additive mixing.

Using the same idea but a different approach, one can obtain colors via subtractive color mixing. While additive mixing can be thought as shedding different lights onto a dark background, subtractive mixing is putting different color filters in front of a white illuminating background. Subtractive mixing is mostly used by artists and designers since paints and pigments can be thought as a filter absorbing some of the light and reflecting the rest. The more paints are mixed, the more colors are absorbed and they end up with black. Additive and subtractive mixing are depicted in Figure 3-5.

![Figure 3-5: Additive (left) and Subtractive (right) Mixing of Colors](image)

It is important to realize that a yellow stimulus produced by the additive or subtractive mixture of appropriate red and green lights does not simply match monochromatic yellow light but is indistinguishable from it; this fact brings forth the necessity to shed light on the effect of perception on color vision.

### 3.1.2 Perceptual Vision

The processing of the signals that are provided from the tristimulus values is quite different in the brain. In 1966, by monitoring the neural pathway between eye and the brain, De Valois et. al. claimed in [30] that the brain constructs opponent channels from tristimulus values: red-green channel, blue-yellow channel and black-white channel. A similar theory, based on the opponent nature of colors, was proposed in 19th century by Edwald Hering (1834-1918) way before the opponent channels in the brain were discovered. Hering suggested in *Opponent Theory of Color Vision*, stated in [92], that all colors arise from a combination of green-or-red, blue-or-yellow and brightness, and proposed his famous color wheel illustrated in Figure 3-6.
The color circle (wheel) that Hering defined consists of colors formed by mixing of four primary colors. These colors, in his color circle, are also referred as *hues*. Hue of a color can be defined in many ways, such as the dominant wavelength of the color or the color in its purest form. This brings along the term purity or *saturation* of a color, which can be understood as the dominance of the hue in the color. As saturation decreases, all colors become a value of gray. However it should not be considered as light or dark since that effect is revealed, in accordance with the black-white channel in Hering’s model, as the *value* of a color. It is simply the intensity or the strength of the light on the color that can be viewed as a lightness/darkness measure.

Definition of the terms hue, saturation and value explains the process of ‘seeing’ in a more perceptual way; however there is still a gap between what the eyes ‘see’ and what the brain ‘sees’. In other words what we see may not necessarily be what we get; since after the reception of the light (color) in the eye, the way it is processed in the brain may considerably modify the information. Goethe tried to derive this fact in his studies and claimed that the sensation of opponent colors does not originate physically from the actions of light on our eyes but perceptually from the actions of our visual system. Starting from the experiments Newton performed, Goethe systematically varied the experimental conditions such as the shape, size, color, and orientation of the images viewed to determine how they influenced what he saw.

All the factors that Goethe experimented affect human perception in a different way. Two adjacent colors interact with one another and change our perception in different ways as shown in Figure 3-7: The Effect of Simultaneous Contrast which is called *simultaneous contrast*. Considering the fact that colors are never present alone in the nature, simultaneous contrast is a noteworthy effect for human color perception. It fairly amplifies the distinctions between two adjacent colors.
Figure 3-7: The Effect of Simultaneous Contrast shows the effect of simultaneous contrast from two perspectives. The image on the left shows the color difference of small rectangles which is obvious when a background with a value between them is present. For the image on the right, although both inner squares have the same value, they are perceived to be different due to their backgrounds. The later effect can be explained in accordance with the well-known Bartleson-Breneman Effect that is illustrated in Figure 3-8. A darker background causes all the values to appear lighter while a lighter background has exactly the opposite effect. Both have the greatest impact on darker values as the former compresses the gray value steps, the later widens that range.

Figure 3-8: Bartleson-Breneman Effect. The squares horizontally have the same color.
In 1873 Bezold and later in 1878 Bruche stated another important fact in [11] that as light becomes more intense, the perception of colors (hues) also changes as illustrated in Figure 3-9, which is also referred as the Bezold-Brücke Effect. As illumination increases from 100 to 1000 trolands (troland is the unit of conventional retinal illuminance), an expansion in the colors that appear yellow and blue is observed together with a contraction in the range of colors that appear red or green. This is an outcome of the adaptation of the opponent channels yellow/blue and red/green with respect to light.

One phenomenon worth to mention that affects human color perception is the afterimage. Simplest example of an afterimage is the white spot observed when humans close their eyes right after staring at an illumination source. This is a black and white example; however, color afterimages are also present and they are mainly due to the opponent nature of the visual system. When the visual system is adapted for a period of time to a visual stimulus (color), red for instance, the associated cones are weakened. Thus right after that adaptation if the visual scene is changed to a white surface for example, the weakened cones yield a weaker response. Due to the opponent nature of the visual system, absence of red is interpreted as presence of green and a green afterimage is seen. In Figure 3-10, if one stares at the black dot on the left for 30 seconds and look at the black dot on the right, an afterimage composed of the opponent colors will be observed.
3.1.3 Color Perceptual Rules

It was stated that a color produced by the additive or subtractive mixture of other primaries does not simply match that color but is indistinguishable from it; meaning that although they are physically different, they appear to be perceptually same as in Figure 3-11, where two perceptually identical colors’ spectral power distributions (SPD) are shown.

The color-matching experiment is a well-known method for investigating the additive property of human color matching. Subject is given a region illuminated by a certain color and asked to match the adjacent region’s color via three primary light sources, following the fact that human color vision is based on three primary receptors. Even though our vision is based on three primaries, Hering utilized six colors working in opponent pairs for human color perception; which are red, green, blue, yellow, white and black. These colors are usually referred as Hering primaries. Still, psychological and linguistic studies revealed that humans have a small number of basic color terms for specifying colors. In addition to Berlin and Kay’s findings in [10] that perception of Hering primaries is universal, it is claimed in [107] by Robertson et. al that each culture (language) has its own primaries for defining colors. They categorized colors for English, for instance, in 11 primaries which include orange, purple pink, brown and gray additional to Herring primaries. A demonstration of their categorization is given in Figure 3-12.
Studies of Berlin and Kay also demonstrated that humans’ capability of defining the primaries is much better than establishing boundaries between them. So, accordingly, in [14] Broek et. al. focused on the utilization of color categorization for CBIR purposes and introduced a new color matching method, which takes human cognitive capabilities into account. They have exploited the fact that people tend to think and define colors over the aforementioned 11 categories, and referred them as focal colors.

It should be noted that colors are never present alone in the nature; instead they appear as color compositions or color patterns. Considering the fact that color perception is strongly dependent on the variations in their adjacency as discussed in Section 3.1.2; instead of matching two colors alone, similarity of color patterns should also be inspected. However it is also a fact that human eye can not perceive a large number of colors at the same time, nor able to distinguish similar (close) colors well. This fact also agrees with the concept of focal colors, leading to the conclusion that a small number of colors are sufficient to represent a multicolored pattern; which are typically called dominant colors. Utilization of dominant colors (DCs) for representation of a multicolored pattern (an image) is given in Figure 3-13: Utilization of Dominant Colors, where a 24 bit (16,777,216) colored image can be represented via 6 DCs.
Mojsilović et. al. in [85] proposed five categories governing humans’ judging similarity and matching of color patterns. These categories are called color vocabulary and together with some hierarchy of rules, the so-called color grammar. The proposed categories are:

1) Overall Color: Presence or absence of a DC. Either one color has a high percentage or a multicolored pattern gives the impression of a DC.
2) Color Purity: The notion of colorfulness, overall chroma or overall saturation.
3) Regularity and Placement: Describes the pattern in terms of its uniformity and repetition as uniform/nonuniform and repetitive/nonrepetitive.
4) Directionality: Presence or absence of a dominant orientation in the distribution.
5) Complexity and Heaviness: General impression of the pattern as ‘light’, ‘soft’, ‘sharp’ etc. This category is stated as optional.

Among the rules suggested in [85], two of them are particularly related for modeling the similarity metrics of human color perception. The first one indicates that the two color patterns that have similar DCs are perceived as similar. The second rule states that two multicolored patterns are perceived as similar if they possess the same (dominant) color distributions regardless of their content, directionality, placement or repetitions of a structural element.

Innovation of focal colors and DCs both suggest that human perception tends to group similar colors together instead of seeing them individually. This grouping manner of human visual perception is well discussed and explained by Gestalt psychology (theory) of visual perception.

3.1.4 Gestalt Theory on Visual Perception

Getting its roots from Johann Wolfgang von Goethe, the idea behind the Gestalt theory is to approach the perception problem as a whole. Being motivated by the statement, “The whole is bigger than the sum of its parts”, Gestalt theory claims that human perception is based on well organized patterns rather than individual components. When humans hear a musical composition, they do not perceive the notes separately; instead they perceive the melody formed by these notes. Or in case of visual scenery, they do not see different intensity values. Textural patterns or different colors; they see the visual composition such as sky, trees, etc. In conclusion, when humans are subject to a number of stimuli, they do not perceive them individually; instead those stimuli are perceived as a part of a larger whole. The theory is being criticized for being merely descriptive, since it does explain the question how and merely states the facts about perception. Still, it is commonly accepted and being used in fields related to perceptual psychology.
The theory is mainly based on six principles as stated in [131], called *Gestalt Laws of Grouping* (or *Prägnanz* in German):

**I. Proximity**
Humans tend to group the objects that are closer to each other. As Figure 3-14 states, a grouping such as \{(1,2)(3,4)\ldots\} is more natural than \{(1)(2,3)(4,5)\ldots\}.

![Figure 3-14: Proximity Grouping](image)

**II. Similarity**
Another factor that affects natural grouping is the resemblance of the parts. Keeping the proximity equal in that case, in Figure 3-15 our vision prefers grouping the dots with the same color and seeing vertical lines instead of a horizontal grouping.

![Figure 3-15: Similarity Grouping](image)

**III. Common Fate**
When a movement takes place, a new grouping occurs, such that the elements with the same moving direction are inclined to be grouped together. Sometimes the new grouping opposes the predominant factor, changing the common fate of the organization. Suppose such movement occurred in Figure 3-16 and the distribution is changed from the upper to the lower. Even the moving components confront the predominant factor, they constitute a new grouping.

![Figure 3-16: The Factor of Common Fate](image)
IV. Closure
The factor of closure states how items are grouped together if they tend to complete a pattern. In Figure 3-17, left image is nothing but a curved line, still it is perceived as a circle with a missing end point. Similarly, the image in the middle is not perceived as three deformed shapes, instead it is perceived as a rectangle and an ellipse overlapped.

![Figure 3-17: The Factor of Closure (left & middle) and Good Curve (right)](image)

V. Good curve
It is however, important to notice that closure may not be the dominant factor in all cases. The image on the right in Figure 3-17 can be seen as a curve joining three squares or as three deformed shapes touching.

VI. Area
The idea is hat humans tend to perceive the smaller of two overlapping items as object and the larges as the background. The item in Figure 3-18 can be viewed as a large black square with a hole in the middle, however humans tend to perceive it as a white circle on a black square.

![Figure 3-18: Object and Background](image)

Additionally, it is stated in [131] that earlier experience has significant effects on human visual perception. The principle is that if humans are accustomed to seeing a certain formation, they tend to organize the parts in a similar way.

Gestalt Theory on human perception is applicable to both perception types, visual and aural. Particularly in visual case it signifies the fact that humans tend to impose a structure or meaning on what they see, and claims that they see the whole before its parts, and follow a procedure from top to bottom.
3.2 Color Spaces

A color space is simply a model to represent colors in terms of color components defined in accordance with a point of interest. While some color spaces are designed for the ease of color reproduction in various environments, some are designed to reflect human color perception. Although there are several color spaces in color industry, the typical color spaces (e.g. RGB, CMYK, HSV, CIE-XYZ, CIE-Lab and CIE-Luv) are typically used in image processing applications. Still, there are many other color spaces based on different criteria. For example YUV color spaces are based on human perception and models range of colors that can be displayed on digital video.

I. *RGB Color spaces*

Relying on the additive mixing of colors discussed in Section 3.1.1 and choosing the primary colors as Red, Green and Blue; RGB color spaces aim to reproduce colors as shown in Figure 3-19. The choice of primary colors is following from the fact that human eye is strongly perceptive to these primaries. Since additive mixing utilizes addition of lights to obtain the color spectrum, hardware devices employing lights to obtain colors, such as display devices, prefer RGB color space.

![RGB Color Space. Axes (left) and Illustration (right)](image)

Depending on the application area and choice of primary color values, different RGB spaces are defined spanning different ranges of visible colors. sRGB is designed to be used in the display devices, such as monitors, in order to match the home or office viewing conditions. Adobe RGB [39] is another RGB space that covers a larger portion of the visible spectrum in order to produce the colors represented by printers on the computer display. Adobe defined another RGB space in 1998 that covers even larger range of color values called Adobe Wide Gamut RGB. Comparison of mentioned RGB spaces with respect to visible colors, defined by CIE (Commission Internationale de L'éclairage - International Commission on Illumination) and detailed under CIE-XYZ Color Space, is given in Figure 3-20: Different RGB Spaces with different Spans. D65 and D50, as shown in the figure, are the reference white points.
One question at this point might be “Why choose a space with smaller range?”. The main reason is that all colors under a specific color space should also be produced in some medium such as monitors or printers; otherwise they would be wasted or redundant.

II. CMYK Color Space

Similar to RGB color space, CMYK uses three primary colors. However, it utilizes subtractive mixing and uses Cyan, Magenta, and Yellow as the primary colors. Additionally Black (Key) is used to provide darkening or desaturation of colors. As mentioned earlier, subtractive mixing is mostly exploited by painters due to the absorbing nature of paints (pigments); thus, hardware devices exploiting color pigments, such as printers, get use of CMYK color space. Addition of black color is simply to save ink, since black is formed of mixing all primary colors and used frequently. Furthermore, although theoretically mixing three primary colors cyan, magenta and yellow forms the black color, in practice it is far away from a perfect (pure) black since forming pure pigments of the primary colors is almost impossible.

\[
\begin{bmatrix}
C \\
M \\
Y
\end{bmatrix} = 
\begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix} - 
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]  

(8)

Most of the devices utilizing CMYK color space need a conversion between RGB and CMYK spaces, which is straightforward as expressed in Eq. (8) that assumes color values in
the normalized range, $[0,1]$. \textit{CMYK} color space forms the same color cube with \textit{RGB} except its axis changes as in Figure 3-21. Moreover, since the choice of primary colors changes, the span of visible colors is also different.

![Figure 3-21: CMYK Color Space Axis (left), Typical color span in the Visible Spectrum (right) (modified from [26])](image)

Although \textit{RGB} and \textit{CMYK} color spaces are well suited for hardware implementations as they form a color by combining three primary colors, human perception is not that straightforward since nobody refers to a color as a mixture of three colors. One can say that \textit{RGB} and \textit{CMYK} are well suited for color generation but quite limited for color description. Thus more perceptual color spaces are defined in order to signify human color perception, as detailed next.

III. \textit{HSV Color Space}

\textit{HSV} color space utilizes the aforementioned terms \textbf{Hue}, \textbf{Saturation} and \textbf{Value} to form a more intuitive and natural representation of colors than \textit{RGB} and \textit{CMYK} spaces. Providing a separation between luminance and chrominance (color) information has advantages for image processing applications. It is represented in the form of a circle having the \textit{hues} on the outer ring and \textit{saturating} towards the center of the circle. The effect of \textit{value} is illustrated as a cone effect in Figure 3-22, since decreasing \textit{value} diminishes the representable colors, i.e. distinct saturation and hue levels. Consequently, \textit{hue} is referred as an angle varying from $0^\circ$ to $360^\circ$ degrees and \textit{saturation} and \textit{value} are given in terms of percentages ($0\text{-}100\%$).
HSV color space is usually preferred for computer graphics applications where the user is asked to define or pick a color as it well matches how humans describe a color. For such purposes, usually a HSV Color wheel as in Figure 3-22 is preferred.

![HSV Color Space (left), HSV Color Wheel (right)](image)

**Figure 3-22: HSV Color Space (left), HSV Color Wheel (right)**

RGB, CMYK and HSV color spaces can be referred as *device dependent* color spaces since RGB space varies with display characteristics, CMYK space varies with printer, ink or paper characteristics and HSV space is simply a nonlinear transformation of RGB space. Such spaces allow manufacturers to specify their own colors related to their products. However, human color perception remains uniform across devices. *Device independent* colors, on the other hand, are meant to be the true colors perceived by the human eye, which are results of the studies carried out by CIE.

IV. CIE-XYZ Color Space

In 1931, CIE defined a color space based on the trichromatic theory stated in Section 3.1.1. Three hypothetical primaries $X$, $Y$ and $Z$ are chosen such that the whole visible spectrum could be mapped via a positive mixture of them. Figure 3-23 shows the color range of CIE-XYZ space, which can also be referred as the chromacity diagram of CIE-XYZ since the lightness value is excluded. CIE-XYZ is designed in such a way that $Y$ corresponds approximately to the lightness value. The colors in Figure 3-23 are represented via two parameters $x$ and $y$ as in Eq. (9), which are functions of the tristimulus values.

\[
x = \frac{X}{X + Y + Z} \quad \text{and} \quad y = \frac{Y}{X + Y + Z}
\]  

(9)
V. CIE-Luv Color Space

CIE-Luv color space is founded in 1960 and modified in 1975 to get its final form; that is why some sources make use of CIE-Lu’v’ notation, referring to its modified form. It has been accepted for many applications that require measurement of lights, since it represents the additive mixture of two lights as a straight line across the color space. CIE-Luv is quite useful for predicting the appearance or mixture of colors produced by luminous displays, such as computer monitors and TV screens.

CIE-Luv color space is a perceptually uniform transformation of the CIE-XYZ space, meaning that two color pairs that are perceptually equally different are equally distant in the CIE-Luv color space. Figure 3-24 shows the transition from CIE-XYZ color space to CIE-Luv color space. The lines on both graphs represent color difference of equal proportion. It is easy to observe from the varying lengths of the lines that CIE-XYZ space is not uniform, which is modified in CIE-Luv space to obtain uniform perceptual distances shown by lines with almost equal lengths (not perfectly uniform). Moreover, although the lightness value $Y$ is uniform in CIE-XYZ (equal steps between lightness values), it is transformed into a new lightness measure, $L$. The reason behind this is the fact that humans do not perceive the differences between higher and lower lightness (intensity) values the same. In fact, their ability to distinguish between higher degrees of lightness is much better than lower values.
VI. CIE-Lab Color Space

Similar to CIE-Luv, CIE-Lab is designed to match the human color perception in a uniform way. However, it uses a different approach than CIE-Luv, arising from the opponent nature of human color perception discovered in 1966 and detailed in [30]. It uses the same lightness channel $L$ with CIE-Luv, but uses two opponent channels $a$ and $b$ to model the opponent color structure. As Figure 3-25 depicts, channel $a$ represents the position of the color between red and green, while the position of the color between yellow and blue is represented by channel $b$.

CIE-Lab space is capable of representing every color in human vision and more, which means it includes some imaginary colors that humans cannot perceive and none of the display or printing devices is able to produce. Even though it is capable of representing the colors of any such device accurately, it needs to be converted to another color space to be displayed since its span of colors is much larger than any rendering device.
CIE-Lab and CIE-Luv spaces are mostly preferred for measuring color differences due to their approximate perceptual uniformity according to Eq. (10). However, this approximate uniformity only holds for low (spatial) frequencies because their design was based on large color patches [105]. At higher frequencies, the relative sensitivity to $L$ component of color is much greater, and both CIE-Lab and CIE-Luv color spaces become somewhat non-uniform. Still, the fact that they decouple the luminance (lightness) from chrominance (color) makes them useful in image processing and compression applications.

Kasson and Plouffe in [58] indicate that decoupling of CIE-Lab is much better than other color spaces like XYZ and CIE-Luv.

$$
\Delta E_{\text{Lab}} = \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2}
$$

$$
\Delta E_{\text{Luv}} = \sqrt{(\Delta L)^2 + (\Delta u)^2 + (\Delta v)^2}
$$

(10)

\[ 3.3 \text{ Color in CBIR} \]

Most CBIR systems frequently utilize color as a discriminative feature among images, due to its robustness to noise, image degradations, changes in size, resolution and orientation. However, although color is an important factor in judging the similarity of images, it should always be noted that color alone does not provide complete information for content-based retrieval over general, broad-context image databases. Instead, it has been shown in [108] that color properties correlate with the true content only in certain extend, but can not be used as a single cue to characterize the entire content.

Color description of visual scenery may or may not include spatial distribution information. If not, the descriptors are said to represent the image globally, where questions such as “which color?” and “how much?” are only of interest. Spatial color descriptors on the other hand, takes into account the distribution of colors over the image and answers the questions such as “where?” and “how?”.

\[ 3.3.1 \text{ Global Color Descriptors} \]

In one of the earlier works of color descriptors, Kato et. al. [61] used the color of every corresponding pixel in two images for comparison and the number of corresponding pixels having the same color determines the similarity between them. However, it is obvious that humans can neither see individual pixels nor perceive large amount of color levels; hence this approach did not provide robust solutions, i.e. slight changes in camera position, orientation, noise or lightning conditions may cause significant degradations in the similarity computation. Swain and Ballard [119] proposed the first Color Histogram, which solves this sensitivity problem. In their work color histograms are extracted and
histogram intersection method is utilized for comparing two images (i.e. the query and next image in the database). Since this method is quite simple to implement and gives reasonable results especially in small to medium size databases, several other histogram-based approaches emerged, such as [28], [46], [49], [90], [94], [101], [112], [115] and [126]. The primary feature of such histogram-based color descriptors (be it in RGB, CIE-Lab, CIE-Luv, or HSV) is that they cluster the pixels into fixed color bins, which are quantizing the entire color space using a pre-defined color palette. This two-fold approach, clustering all the pixels having similar color and reducing the color levels from millions to (usually) thousands or even hundreds via quantization, is the main reason behind the limited success that the color histograms achieved since both operations are indeed the small steps through obtaining the perceivable elements (the true DCs and their global distributions); yet their performance is still quite limited and usually degrades drastically in large databases due to several reasons. First and the foremost, they apply static-quantization where the color palette boundaries are determined empirically or via some heuristics –yet nothing based on human color perception rules. If, for example, number of bins are set too high (fine quantization) then similar color-pair will end up in different bins. This will eventually cause erroneous similarity computation whenever using any of the naïve metrics such as $L_1$, $L_2$ or using the histogram intersection method as in [119]. On the other hand if number of bins is set too low (coarse quantization) then there is an imminent danger of completely different colors falling into the same bin and this will obviously degrade the similarity computation and reduce the discrimination power. No matter how the quantization level (number of bins) is set, pixels with such similar colors but happens to be opposite sides of the quantization boundary, separating two consecutive bins will be clustered into different bins and this is an inevitable source of error in all histogram-based methods. The color quadratic distance [49] proposed in the context of QBIC system provides a solution to this problem by fusing the color bin distances into the total similarity metric. Let $X$ and $Y$ be two color histograms with total number of $N$ bins and if they are written as pairs of color bins and weight: $X = \{(c_1, w_1^x), (c_2, w_2^x), \ldots, (c_N, w_N^x)\}$ and $Y = \{(c_1, w_1^y), (c_2, w_2^y), \ldots, (c_N, w_N^y)\}$ then the quadratic distance between $X$ and $Y$ is as follows:

$$D_Q(X,Y)^2 = (X - Y)^T A (X - Y)$$

$$= \sum_{i}^{N} \sum_{j}^{N} (w_i^x - w_i^y)(w_j^x - w_j^y) a_{ij}$$

(11)

where $A = [a_{ij}]$ is the matrix of color similarities between the bins $c_i$ and $c_j$. Although this formulation allows the comparison of different histogram bins with some inter-similarity between them; however it has already mentioned in Section 2.4 that it does not match the human perception well enough.
Besides the aforementioned clustering drawbacks and the resultant erroneous similarity computation, color histograms have computation deficiencies due to the hundreds (or even thousands) of redundant bins created for each image in a database, although ordinary images usually contain few DCs (i.e. <8), and more than that cannot anyway be perceived by HVS [84] according to second color perception rule mentioned in Section 3.1.3. Therefore, color histograms not only create a major computational deficiency in terms of storage, memory limit and computation (CPU) time due to spending hundreds or thousands of bins for the few DCs present, moreover their similarity computations will be biased by the outliers hosted within those redundant bins. Recall that two images with similar color composition will have similar DC properties; however there is no such requirement for the outliers as they can be entirely different. Hence including color outliers into similarity computation may cause misinterpreting two similar images as dissimilar or vice versa, and usually reduce the discrimination power of histogram-based descriptors, which eventually makes them unreliable especially in larger databases.

In order to solve the problems of the color histograms applying such a static quantization scheme, various Dominant Color Descriptors, e.g. [3], [31], [37], [78] and [84], have been developed using dynamic quantization with respect to image color content. DCs, if extracted properly according to the aforementioned color perception rules, can indeed represent the prominent colors in any image as in Figure 3-13. They have a global representation, which is compact and accurate and they are also computationally efficient since only few colors that are usually present in a natural image and perceivable by a human eye are described. A top-down DC extraction scheme is proposed in [31] where the method is entirely designed with respect to HVS color perceptual rules. For instance HVS is more sensitive to the changes in smooth regions than in detailed regions. Thus in this work colors are quantized more coarsely in the detailed regions and smooth regions have more importance respectively. To exploit this fact, a smoothness weight \( w(p) \) is assigned to each pixel \( p \) based on the variance in a local window. Afterwards, the General Lloyd Algorithm (GLA, also referred to as Linde-Buzo-Gray and it is equivalent to the well-known \( K \)-means clustering method [76]) is used for color quantization. For a color cluster \( C_i \), its centroid \( c_i \) is calculated by

\[
c_i = \frac{\sum w(p)x(p)}{\sum w(p)}, \quad x(p) \in C_i
\]  

and the initial clusters for GLA is determined by using a weighted distortion measure, defined as

\[
D_i = \sum w(p)\|x(p) - c_i\|^2, \quad x(p) \in C_i
\]
This is used to determine, which clusters to split until either a maximum number of clusters (DCs), $N^\text{max}_{\text{DC}}$, is achieved or a maximum allowed distortion criteria, $\varepsilon_D$, is met. Hence pixels with smaller weights (detailed sections) are assigned fewer clusters so that the number of color clusters in the detailed regions where the likelihood of outliers’ presence is high, is therefore suppressed. As the final step, an agglomerative clustering (AC) is performed on the cluster centroids to further merge similar color clusters so that there is only one cluster (DC) hosting all similar color components in the image. A similarity threshold $T_s$ is assigned to the maximum color distance possible between two similar colors in a certain color domain (CIE-Luv, CIE-Lab, etc.) used. Another merging criteria is the color area, that is, any cluster should have a minimum amount of coverage area, $T_A$, so as to be assigned as a DC; otherwise it will be merged to the closest color cluster since it is just an outlier. Another important issue is the choice of the color space since a proper color clustering scheme for DC extraction tightly relies on the metric. Therefore, a perceptually uniform color space should be used and the most common ones are CIE-Luv and CIE-Lab as discussed in Section 3.2, which are designed such that color distances perceived by HVS are also equal in $L_2$ (Euclidean) distance in these spaces.

![Figure 3-26: Different color compositions of red, blue and white with same proportions (weights).](image)

### 3.3.2 Spatial Color Descriptors

Although the true DCs, which are extracted via such perceptually oriented scheme with the proper metric can address the aforementioned problems of color histograms, global color properties (DCs and their coverage areas) alone are not enough for characterizing and describing the real color composition of an image since they all lack the crucial information of spatial relationship among the colors. In other words, describing “what” and “how much” color is used will not be sufficient without specifying “where” and “how” the (perceivable) color components (DCs) are distributed within the visual scenery. For
example all the patterns shown in Figure 3-26: Different color compositions of red, blue and white with same proportions (weights), have the same color proportions (be it described via DCs or color histograms), but different spatial distributions and thus cannot be perceived as the same. Especially in large image databases, this is the main source of erroneous retrievals, which makes “accidental” matches between images with “similar” global color properties but different in the color distribution. There are several approaches to address this problem. The segmentation-based methods can be counted as one example; they are nevertheless not feasible since in most cases segmentation implies a great deal of laborious work and user interaction during database creation. Some studies used the local positions of color blocks for characterizing the spatial distributions. For instance in an earlier study, Gong et. al. [46] divided the image into nine equal sub-images and representing each of them by a color histogram. In a similar work, Stricker and Dimai [118] split the image into five regions: an oval central region and four corners. Both approaches are illustrated in Figure 3-27.

They tried to combine color similarity from each region whilst attributing more weight to the central region. Due to fixed partitioning and usage of the block positions alone for matching, such methods become strictly domain dependant solutions since such large and fixed-block partitioning along with the utilization of color histograms cannot provide any reliable description for SCD in general. Ooi et. al. [95] enhances the idea of using a statistically derived quad-tree decomposition to obtain homogeneous blocks and comparing the matching blocks (in the same position) to obtain SCD similarity. However the local position of a certain color in an image cannot really describe the true SCD due to several reasons. First the image dimensions, resolution and their aspect ratio can vary significantly. So an object with a certain size can fall (perhaps partially) into different blocks in different locations. Furthermore, imagine the following simple scene, a colored object with a background. The content is basically a picture of this particular object regardless of its
location, whether it is in the left or right, up or down. Hence any rotation or translation of
the similar color elements, as in Figure 3-28: Same Content with Different Spatial
Locations, will cause drastic mismatches on such methods.

![Figure 3-28: Same Content with Different Spatial Locations](image)

Pass et al. [99] presented Color Coherence Vector (CCV), which partitions the
histogram bins based on the spatial coherence of the pixels, whereas a pixel is “coherent” if
it contains the similar color of a colored-region and “incoherent” otherwise. For each color
c_i, let \( \alpha(c_i) \) and \( \beta(c_i) \) be the number of coherent and incoherent pixels, thus the pair
\((\alpha(c_i), \beta(c_i))\) is called a coherence pair for the \( i^{th} \) color then the coherent vector can be
defined for an image \( I \) as:

\[
CCV(I) = \{(\alpha(c_1), \beta(c_1)), (\alpha(c_2), \beta(c_2)),..., (\alpha(c_N), \beta(c_N))\}
\]  

(14)

and they use \( L_1 \) metric to compare two images. A nice property of this method is the
classification of the outlier (color) pixels in spatial domain (i.e. incoherent) from the
prominent ones (i.e. coherent). In other words, whenever a color pixel is an outlier
(spatially isolated or alone and thus not perceivable by HVS), it is separated from the
perceivable (in a sufficiently big group) ones during the color based similarity computation.
They report a better retrieval performance than the traditional histogram-based methods;
yet apart from the aforementioned drawbacks of the histogram-based method applied on
individual pixels, classifying color pixels alone, without any metric or characterization for
the SCD will not describe the real color composition of an image. Another variation of this
approach is characterizing the adjacent color pairs, in other words the color boundaries.
Nagasaka and Tanaka [91] developed a color matching technique to model color
boundaries. Thus two images are expected to be similar if they have similar sets of color
pairs. In a similar approach, Stricker [118] used the boundary histograms to describe the
length of the color boundaries. Such a heuristic approach of using color adjacency
information might be more comprehensible than the ones using local positions by dividing
image into fixed blocks, since they at least used “relative” features instead of “static” ones, yet they are likely to suffer from the changes in the background color or the relative translations of the objects or color elements. The former case implies to a strong dissimilarity although only the background color is changed whilst the rest of the object(s) or color elements stay intact. In the latter case there is no change in the adjacent colors, however the inter-proximities of the color elements (hence the entire color composition) are changing and hence a certain dis-similarity should occur. Therefore, the true characterization of SCD lies in the inter-proximities (the relative distances) of color elements with respect to each other. In other words, characterizing inter- or self-color proximities (e.g. the relative distances of the DCs) shall be a reliable and discriminative cue about the color composition. This property is robust to translations, rotations and variations in image properties (dimensions, aspect ratio and resolution) and hence forms the basis of the proposed descriptor in Chapter 4.

One of the most promising approaches among all SCD descriptors is the color Correlogram [51], which is a table where the \( k \)th entry for the color histogram bin pair \((i, j)\) specifies the probability of finding a pixel of color-bin \( j \) at a distance \( k \) from a pixel of color-bin \( i \) in the image. Let \( I \) be an \( W \times H \) image quantized with \( m \) colors \((c_1, \ldots, c_i, \ldots, c_m)\) via RGB color histogram. For a pixel \( p = (x, y) \in I \), let \( I(p) \) denote its color value and let \( I < c_i \equiv \{ p \mid I(p) = c_i \} \). So the color histogram value of a quantized color \( c_i \), \( h(c_i, I) \), can be defined as:

\[
h(c_i, I) = WH \Pr(p \in I < c_i)
\] (15)

Accordingly the color Correlogram \( \gamma_{c_i,c_j}^{(k)} \), for the quantized color pair \((c_i, c_j)\) and a pixel distance \( k \leq d \) can be expressed as:

\[
\gamma_{c_i,c_j}^{(k)} = \Pr \left( p_2 \in I < c_j \bigg| p_1 \in I < c_i, \|p_1 - p_2\| = k \right)
\] (16)

where \( c_i, c_j \in \{c_1, \ldots, c_m\} \), \( k \in \{1, \ldots, d\} \) and \( |p_1 - p_2| \) is the distance between pixels \( p_1 \) and \( p_2 \) in \( L_\infty \) norm. In order to compute \( \gamma_{c_i,c_j}^{(k)} \), it is sufficient to compute the inter-color proximity count, \( \Gamma_{c_i,c_j}^{(k)} \), which can be expressed as follows:

\[
\Gamma_{c_i,c_j}^{(k)}(I) = \left| \left\{ p_1 \in I < c_i, p_2 \in I < c_j \mid |p_1 - p_2| = k \right\} \right|
\]

\[
\gamma_{c_i,c_j}^{(k)}(I) = \frac{\Gamma_{c_i,c_j}^{(k)}(I)}{h(c_i, I)h(c_j, I)}
\] (17)
Therefore, the Correlogram $\gamma_{c_i,c_j}^{(k)}$ is in fact a measure of inter-color proximity, which is normalized by the amount of color, $h(c_i, I)$, and total number of pixels at distance $k$, $8k$. The computation complexity is a critical factor for the feasibility of Correlogram. The naïve algorithm to compute $\gamma_{c_i,c_j}^{(k)}$ takes $O(W.H.d^2)$, which is a massive complexity. In order to obviate it, a pair of quantities $\lambda_{(x,y)}^{c, h}(k)$ and $\lambda_{(x,y)}^{c, v}(k)$ must be first computed as follows:

$$
\lambda_{(x,y)}^{c, h}(k) = \left\{(x + i, y) \in I < c > \left| 0 \leq i \leq k \right. \right\}
$$
$$
\lambda_{(x,y)}^{c, v}(k) = \left\{(x, y + j) \in I < c > \left| 0 \leq j \leq k \right. \right\}
$$

both of which counts number of pixels of color $c$, within a certain distance $k$, from a fixed pixel position, $(x,y)$ in horizontal, ($h$) and vertical, ($v$) directions. Afterwards, $\Gamma_{c_i,c_j}^{(k)}$ can be derived from these pair of quantities as follows:

$$
\Gamma_{c_i,c_j}^{(k)}(I) = \sum_{(x,y) \in F < c_i >} \left( \lambda_{x-k,y+k}^{c, h}(2k) + \lambda_{x-k,y-k}^{c, h}(2k) + \lambda_{x-k,y-k+1}^{c, v}(2k-2) + \lambda_{x+k,y-k+1}^{c, v}(2k-2) \right)
$$

and this time it takes $O(5.W.H.d)$ since $\Gamma_{c_i,c_j}^{(k)}$ computation uses two quantities, $\lambda_{(x,y)}^{c, h}(k)$ and $\lambda_{(x,y)}^{c, v}(k)$, which are computed beforehand and stored for each pixel in the image. This will however requires $O(16.W.H.d.m)$ memory space (in bytes) to store them, which is the mandatory cost to pay in order to speed up the process from $O(W.H.d^2)$ to $O(5.W.H.d)$. Another computational issue is the storage of the feature vector. Since the feature vector size of Correlogram (in bytes) is $O(16m^2d)$, a simplified version, the so-called Auto-Correlogram, which only captures the spatial correlation between the same colors and thus reduces the feature vector size to $O(16md)$ bytes, was proposed in [51].

In the spatial domain and pixel level, Correlogram can characterize and thus describe the relative distances of distinct colors between each other and thus such a description can indeed reveal a high resolution model of SCD. Accordingly Ma and Zhang [74], conducted a comprehensive performance evaluation among several global/spatial color descriptors for CBIR and reported that Correlogram achieves the best retrieval performance among the others, such as color histograms, CCV, color moments, etc. However, the color Correlogram computation first of all, has serious feasibility problems. Nowadays digital image technology offers several mega-pixel (Mpel) image resolutions. For a conservative assumption, consider a small size database with only 1000 images each of which in only 1 Mpel resolution. Without any loss of generality, assume that $W=H=1000$. In such image dimensions, a reasonable setting for $d$ would be $100<d<500$, corresponding to $\sim 10\%-50\%$ image dimension range. Any $d$ setting less than 100 pixels would be too small for
characterizing the true SCD of the image—probably describing only a thin layer of adjacent colors (i.e. colors that can be found within a small range). Assume the lowest range setting: \(d=100\) (yet a Correlogram working over only a 10% range of the image dimension is hardly a “spatial” color descriptor). Even with such “minimal” settings, the naïve algorithm will require \(\sim O(10^{10})\) computations (including divisions, multiplications and additions). Even with fast computers, this will require several hours of computation per image and infeasible time is required to index even the smallest databases. In order to achieve a feasible computational complexity for the naïve algorithm, the range has to be reduced drastically (i.e. \(d\sim 10\)) and the images should be decimated by 3 to 5 times in each dimension. Such a solution unfortunately changes (decimates) the color composition of the scheme and with such limited range, the true SCD cannot anymore be characterized by the Correlogram. The other alternative is to use the fast algorithm. A typical quantization for RGB color histogram can be 8 partitions in each color dimension (i.e. \(8\times8\times8=512\) bins RGB histogram), the fast algorithm will speed up the process around 25 times; however, it will also require a massive memory, (> 400Gb per image) and this time neither decimation, nor drastic reduction on the range will make it feasible and practically speaking, one can hardly make it work only for thumbnail size images and only when \(d<10\) and much coarser quantization (e.g. using \(4\times4\times4\) RGB histogram) is used. It is otherwise not applicable for any other (typical) cases. Furthermore, its massive storage requirement is another serious bottleneck of the Correlogram. Note that for the minimal range (\(d=100\)) and typical quantization settings (i.e. \(8\times8\times8\) RGB partitions), the amount of required space for the feature vector storage of a single image is above 400Mb. This allows the Correlogram barely applicable only for small size databases, i.e. for 1000 image database the storage space required is above 400Gb. To make it work, the range value has to be reduced drastically along with using a much coarser quantization (\(4\times4\times4\) bins or less). Unfortunately with such settings, recall the problems of coarse quantization of color histograms and such a diminished range setting. The only alternative is computing Auto-Correlogram instead of Correlogram, which is eventually recommended and used in [51]; however without characterizing spatial distribution of distinct colors with respect to each other, the true SCD cannot be accurately described.
Apart from all such feasibility problems, a Correlogram has several limitations and drawbacks. The first and the foremost is its pixel-based structure, which characterizes the color proximities at a pixel level. Such a high-resolution description not only makes it too complicated and infeasible to perform, it also becomes meaningless with respect to HVS color perception rules simply because individual pixels do not mean much for the human eye. As an example, consider a Correlogram description such as “the probability of finding a red pixel within a 43 pixel proximity of a blue pixel is 0.25” and so what difference does it make to have this probability in 44 or 42 pixels proximity for the human perception? Another similar image might have the same probability but in 42 pixels proximity, which makes it indifferent or even identical for the human eye; however a significant dissimilarity will occur via Correlogram’s naïve (dis-)similarity computation. Furthermore, since Correlogram is a pixel level descriptor working over RGB color histogram, the outliers, both in color and spatial domains have an imminent affect both over computational complexity and the retrieval performance of the descriptor. Hundreds of color outliers hosted in the histogram, even though not visible for human eye, will cause the aforementioned feasibility problems on computational (memory and speed) cost and storage, which makes the Correlogram inapplicable in many cases or significantly degraded and limited (e.g. due to coarse color quantization, insufficient range setting, decimation, etc.) so as to make it feasible again. Yet the real problem lies on degradation caused by the outliers directly over the description accuracy such as their bias (shift) over the true (perceivable) probabilities (inter-color proximities). Imagine, for example, two colors, red and blue, where all the red pixels are far away from the ones in blue, except one (or a few) that are not really visible (and so a spatial outlier), however there will be a definite probability bias (or shift) due to its proximity to all blue pixels. Depending on the amount of red pixels, this can be a significant (if red color is also outlier) or insignificant (red is sufficiently dominant) biasing due to the normalization by the amount of color, $h(c_i, I)$, in
the calculation of Correlogram as in Eq. (17). Note that in a RGB histogram, there will inevitable be several (hundreds, if not thousands) color outliers depending on the size, and as well as the spatial outliers of both dominant and non-dominant (outlier) colors spread over the entire image and thus their total cumulative effect would be significant and hence degrade the accuracy of the SCD characterization of the descriptor via proximity probabilities. Finally using the probability alone makes the descriptor insensitive to the dominance of a color or its area (weight) in the image. This is basically due to the normalization by the amount (weight or area) of color, \( h(c, I) \), and such an important perceptual cue is lacking in the Correlogram’s description. This might be a desirable property to find the similar images simply “zoomed” as in [51], and hence the color areas significantly vary but the distance probabilities do not; however it also causes severe mismatches especially in large databases since the probability of the pair-wise color distances might be the same or close independent of their area in the image and hence regardless of their dominance (whether they are DCs or outliers). An example of such descriptor deficiency can be seen in a query of sample image as shown in Figure 3-29. In short, these properties make Correlogram a colored texture descriptor rather than color since its pixel-level, area insensitive, co-occurrence description is quite similar to texture descriptors based on co-occurrence statistics (e.g. Gray-Level Co-occurrence Matrix (GLCM) [98]) only with a major difference of describing color co-occurrences instead of gray level (intensity) values.
4 A Novel Perceptual Color Descriptor

Under the light of the earlier discussion, humans focus on a few (dominant) colors and their (spatial) distributions while judging the color similarity between images and our ability to extract such a global color view out of visual scenery, no matter it is a digital image or a natural 3D view, is indeed amazing. However, it is not that straightforward to accomplish this while dealing with digital images for CBIR purpose. Note that on a standard 24 bit representation, there is a span of 16 million colors, which can be assigned on thousands of individual pixels. Such a “high resolution” representation might be required for the nowadays digital image technologies; however it is not too convenient for the purpose of describing of the color composition or performing a similarity measurement based on the aforementioned human color perceptual rules.

Recall the HVS fact mentioned earlier about humans inability to see individual pixels, or to perceive even a tiny fracture of such a massive amount of color levels; thus it is crucial to perform certain steps in order to extract the true “perceivable” elements (the true DCs and their global distributions). In other words the un-perceivable elements (henceforth referred as outliers), which does not have significant contribution or weight over the present color structure, in both color and spatial (pixel) domain, should be suppressed or removed. Recall that according to two color perception rules presented in [84], two images that are perceived as similar in terms of color composition have a similar DC properties; however the color properties of their outliers might be entirely different and hence this can affect (degrade, bias or shift) any similarity measurement if not handled accordingly. For example in the well-known perceptual audio coding schemes such as MP3 and AAC [13], in order to maximize the coding efficiency such outliers (the sound elements that humans cannot hear) in both spatial (time) and spectral (frequency) domains are removed and thus more bits can be spent for the “dominant” sound elements. In a similar fashion, the outliers both in color and spatial domain should be removed for description efficiency.

The proposed color descriptor is designed to address the drawbacks and problems of the other color descriptors. In order to achieve this, it is mainly motivated from the human color perception rules. Therefore, global and spatial color properties are extracted and described in a way HVS perceives them and the outliers, both in color and spatial domain, are suppressed and eliminated by adopting a top-down approach in both domains during feature extraction.

In the following sections, first the indexing scheme will be detailed. Two efficient SCD descriptors are proposed; the proximity histograms, which extract the histogram of inter-
color distances and the \textit{proximity grids}, which cumulates the spatial co-occurrence of colors in a 2D grid. Both approaches are configurable and provide means of modeling SCD in a scalar and directional way. Next, the retrieval scheme will be discussed; utilizing a penalty-trio model which basically penalizes the individual differences in global and spatial color properties in a similarity distance computation.

4.1 Descriptor Formation

As explained in Chapter 3, the DCs represent the prominent colors in an image whilst the unperceivable color components (outliers) are discarded. As a result, they have a global representation, which is compact and accurate, and they represent the few (dominant) colors that are perceivable and present in an image. Therefore, the DC scheme is purposefully used to represent global color properties and spatial color properties are extracted thereafter, as shown in Figure 4-1. A DC extraction algorithm similar to the one in [70] is adopted where the method is entirely designed with respect to HVS color perceptual rules and configurable with few thresholds:

\begin{align*}
T_S &: \text{Color similarity} \\
T_A &: \text{Minimum area} \\
\varepsilon_D &: \text{Minimum distortion} \\
N_{DC}^{\text{max}} &: \text{Maximum number of DCs}
\end{align*}

As the first step, the true number of DCs present in the image (i.e. \(1 \leq N_{DC} \leq N_{DC}^{\text{max}}\)) is extracted in CIE-Luv color domain and back-projected to the image for further analysis involving extraction of the spatial properties (SCD) of DCs. Let \(C_i\) represents the \(i^{th}\) DC class (cluster) with the following members: \(c_i\) is the color value (centroid), \(w_i\) is the weight (unit normalized area) and \(\sigma_i\) is the standard deviation obtained from the distribution of (real) colors clustered by \(C_i\). Due to the DC thresholds set beforehand, \(w_i > T_A, |c_i - c_j| > T_S \) for \(1 \leq i, j \leq N_{DC}\).
During the back-projection phase, the DC, which has the closest centroid value to a particular pixel color, will be assigned to that pixel. As a natural consequence of this process, spatial outliers, i.e. isolated pixel(s), which are not populated enough to be perceivable, can emerge (see the example in Figure 4-2) and should thus be eliminated. Due to the perceptual approach based on the Gestalt rule (Section 3.1.4), “Humans see the whole before its parts”, a top down approach such as quad-tree decomposition can process the “whole” first, meaning the largest blocks possible, which can be described (and perceived) by a single DC, before going into its “parts”. Due to its top-down structure, the proposed scheme does not suffer from the aforementioned problems of some pixel-based approaches (Section 3.3).

Two parameters are used to configure the quad-tree:
The minimum weight (dominance) within the current block required from a DC not to go down for further partition

\( D_{QT}^{\text{max}} \) : The depth limit indicating the maximum amount of partition (decomposition) allowed.

Note that with the proper setting of \( T_w \) and \( D_{QT}^{\text{max}} \), QT decomposition can be carried out to reach the pixel level; however such an extreme partitioning should not be permitted to avoid the aforementioned problems of pixel level analysis. Since \( D_{QT}^{\text{max}} \) determines when to stop the partitioning abruptly, it should not be set too low so that it does not cause inhomogeneous (mixed) blocks and on the other hand, extensive experimental results suggest that \( D_{QT}^{\text{max}} > 6 \) is not required even for the most complex scenes since the results are almost identical to the one with \( D_{QT}^{\text{max}} = 6 \). Therefore, the typical range is \( 4 \leq D_{QT}^{\text{max}} \leq 6 \). Using a similar analogy \( T_w \) can be set in accordance with \( A_T \), i.e. \( T_w \geq 1 - A_T \). Therefore, for the typical \( A_T \) setting (between 2-5%), \( T_w \) can be conveniently set as \( \geq 95\% \).

Let \( B^p \) corresponds to the \( p \)th partition of the block \( B \) where \( p = 0 \) is the entire block and \( 1 \leq p \leq 4 \) represents the \( p \)th quadrant of the block. The 4 quadrants can be obtained simply by applying equal partitioning to the parent block or via any other partitioning scheme, which is optimized to yield most homogenous blocks possible. For simplicity the former case is used and accordingly a generic QT algorithm, \textit{QuadTree}, can be expressed as follows:

\textit{QuadTree} (parent, depth)

- Let \( w_{\text{max}} \) be the weight of the DC, which has the maximum coverage in \textit{parent} block.
- If \( w_{\text{max}} > T_w \) then \textbf{Return}.
- Let \( B^0 = \text{parent} \)
- For \( \forall p \in [1..4] \) do:
  - QuadTree \( (B^p, \text{depth}) \)
- Return.

The QT decomposition of a (back-projected) image \( I \) can then be initiated by calling \textit{QuadTree} \( (I, 0) \) and once the process is over, each QT block carries the following data: its depth \( D \leq D_{QT}^{\text{max}} \), where the partitioning is stopped, its location in the image and the major DC, which has the highest weight in the block (i.e. \( w_{\text{max}} > T_w \)) and perhaps some other
DCs, which are eventually some spatial outliers. In order to remove those spatial outliers, a QT back-projection of the major DC into its host block is sufficient. Figure 4-2 illustrates the removal of some spatial outliers via QT back-projection on a sample image. The final scheme where outliers in both color and spatial domains are removed and the (major) DCs are assigned (back-projected) to their blocks can be conveniently used for further (SCD) analysis to extract spatial color features. Note that QT blocks can vary in size depending on the depth, yet even the smallest (highest depth) block is large enough to be perceivable and carry a homogenous DC. So instead of performing pixel-level analysis, the uniform grid of blocks in the highest depth \( D = D_{\text{max}}^{\text{QT}} \) can be used for characterizing the global SCD and extracting the spatial features in an efficient way.

As shown in Figure 4-1, two modes, which performs two different approaches to extract spatial color features can be used. The first is the scalar mode, over which inter-DC proximity histograms are computed within the full range of image. These histograms indicate the amount of a particular DC that can be found from a certain distance of another DC; however this is a scalar measure where the direction information is lacking. For example such a measure can state “17% of red is 8 units (blocks) away from blue” but without any directional information. Therefore, the second mode is designed to represent inter-occurrence of one DC with respect to another over a 2D (proximity) grid from which both distance and direction information can be obtained. Note that inter-color distances are crucial for characterizing the SCD of an image; however, the direction information may or may not be useful depending on the content. For example, the direction information in “17% of red is 8 units (blocks) right of blue” is important for describing a national flag (and hence the content) but “One black and one white horse are running together on a green field” is sufficient to describe the content without any need to know the exact directional order of black, white and green. In the following sub-sections both modes will be detailed.

### 4.1.1 Proximity Histograms

Once the QT back-projection of major DCs into their host blocks are completed, all QT blocks hosting a single (major) DC with a certain depth \( D \leq D_{\text{max}}^{\text{QT}} \) are further partitioned into the blocks in highest depth (i.e. \( D = D_{\text{max}}^{\text{QT}} \)) so as to achieve a proximity histogram in the highest block-wise resolution. Therefore, in such a uniform block-grid, the image \( I \) will have \( N \times N \) blocks where \( N = 2^{D_{\text{max}}^{\text{QT}}} \) and each of which hosts a single DC. Accordingly the problem of computing inter-DC proximities turns out to be block distances and hence the block indices in each dimension (i.e. \( \forall x, y \in [1, N] \)) can directly be used for distance (proximity) calculation. Since the number of blocks does not change with respect to image dimension(s), the resolution invariance is, therefore, achieved (e.g. the same image in different resolutions will have identical proximity histograms/grids as opposed to
significantly varying Correlograms due to its pixel-based computations). As shown in Figure 4-1, either $L_1$ or $L_\infty$ norms can be used for block-distance calculations. Let $b_1 = (x_1, y_1)$ and $b_2 = (x_2, y_2)$ be two blocks, the distance in $L_1$ norm can be defined as,

$$L_1: \|b_1 - b_2\| = |x_1 - x_2| + |y_1 - y_2|,$$

and for the $L_\infty$ norm,

$$L_\infty: \|b_1 - b_2\| = \max\{|x_1 - x_2|, |y_1 - y_2|\},$$

respectively. Using the block indices in both norms, the block distances become integer numbers and note that for a full range histogram, the maximum (distance) range will be $[1, L]$ where $L$ is $N-1$ in $L_\infty$ and $2N-2$ in $L_1$ norms respectively. A block-wise proximity histogram for a DC pair $c_i$ and $c_j$ stores in its $k^{th}$ bin the number of blocks hosting $c_j$ (i.e. $\forall b_i | I(b_i) = c_j$, equivalent to amount of color $c_j$ in $I$) from all blocks hosting $c_i$ (i.e. $\forall b_i | I(b_i) = c_i$, equivalent to amount of color $c_i$ in $I$) in a distance $k$. So such a histogram clearly indicates how close or far two DCs and their spatial distribution with respect to each other. Yet the histogram bins should be normalized by the total number of blocks, which can be found $k$ blocks away from the source block $b_i$ hosting the DC $c_i$ because this number will significantly vary with respect to the distance ($k$), the position of source block ($b_i$) and the norm ($L_1$ or $L_\infty$) used. Therefore, the $k^{th}$ bin of the normalized proximity histogram, $\Phi^{c_i}_{c_j}(k)$, between the DC pair $c_i$ and $c_j$ can be expressed as:

$$\Phi^{c_i}_{c_j}(k) = \sum_{b_i} \sum_{b_j} \Delta(b_i, b_j, k)$$

where

$$\Delta(b_i, b_j, k) = \begin{cases} N(b_i, k)^{-1} & \text{if } b_i \in I(c_i), b_j \in I(c_j), \|b_i - b_j\| = k \\ 0 & \text{else} \end{cases} \quad (20)$$

Note that the normalization factor, $N(b_i, k)$, by the total number of neighbor blocks in distance $k$, is independent from the DC distribution and hence it can be only computed once and used for all images in the database. Figure 4-3 presents $N(b_i, k)$ templates computed for all blocks ($\forall b_i \in I$), both norms and some range values. In the figure for illustration purposes $N$ is kept as 8 ($D_{\max} = 3$) and note that normalization cannot be applied for those blocks where $N(b_i, k) = 0$ since the range ($k$) is out of image boundaries and hence $\Phi^{c_i}_{c_j}(k) = 0$ for $\forall c_j$. 
Once the $N(b_i,k)$ templates are formed, normalized proximity histogram computation takes $O(N^4)$. Note that this is basically independent from original image dimensions, $W$ and $H$, and it is also a full-range computation (i.e. $k \in [1..L]$), which may not be necessary in general (say, half image range may be quite sufficient since above this range most of the (central) blocks will have either out-of-boundary case where $\Phi^r_i(k) = 0$ for $\forall c_i$ or only few blocks in the range, which is too low for obtaining healthy statistics). For $D_{QT}^{\max} = 5 \Rightarrow N=32$ and $N_{DC}^{\max} = 8$, as a typical setting, it requires $\sim O(10^6)$ computations, which are 10000 times less when compared to Correlogram with a minimal range setting (i.e. 10% of image dimension range). In fact the real speed enhancement is much more than that since the computations in Correlogram involves several additions, multiplications and worst of all, divisions for probability computations whereas only additions are sufficient for computing $\Phi^r_i(k)$ as long as $N(b_i,k)^{-1}$ is initially computed and stored as the template. The memory requirements for the full-range computation are $O(16N^2L)$ bytes for storing $N(b_i,k)^{-1}$ and plus $O(16N_{DC}^2L)$ for computing $\Phi^r_i(k)$ respectively. The memory space required for the typical settings given earlier will thus be $\sim$500Kb, an insignificant amount compared to Correlogram. Finally, note that $\Phi^r_i(k) = \Phi^r_i(k)$ and thus the space requirement for the storage of the proximity histogram within the feature vector is not...
A Novel Perceptual Color Descriptor

$O(16N_{DC}^2L)$ but $O(16\left(\frac{N_{DC}+1}{2}\right)L)$. So the typical storage space required per database image will be less than 17Kb (with $L_\infty$, <33Kb with $L_1$ norm), which is eventually 50 times less for the Auto-Correlogram ($O(16md)$) with minimal $m$ and $d$ settings, where $m=512$ (8x8x8) and $d=100$.

4.1.2 Proximity Grids

This is an alternative approach for characterizing the inter-DC distribution not only by the respective proximities, but also their inter-occurrences cumulating on a 2D proximity grid as explained in [63]. The process starts from the same configuration outlined earlier, i.e. let the image $I$ have $N\times N$ blocks, each of which hosts a single DC. 2D proximity grid, $\Psi^{\xi}(x,y)$, is formed by cumulating the co-occurrence of blocks hosting $c_j$ (i.e. $\forall b_j|I(b_j)=c_j$) in a certain vicinity of the blocks hosting $c_i$ (i.e. $\forall b_i|I(b_i)=c_i$) on a 2D (proximity) grid. In other words, via fixing the block $b_i$ (hosting $c_i$) in the center bin of the grid (i.e. $x=y=0$), the particular bin, which corresponds to the relative position of block $b_j$ (hosting $c_j$) is incremented by one and this process is repeated for all blocks hosting $c_j$ in a certain vicinity of $b_i$. Then the process is repeated for the next block (hosting $c_i$) until the entire image blocks are scanned for the color pair $c_i-c_j$. As a result the final grid bins represent the inter-occurrences of the $c_j$ blocks with respect to the ones hosting color $c_i$, within a certain range $L$ (i.e. $\forall x,y\in[-L,L], L\leq N-1$). Although $L$ can be set as $N-1$ for a full-range representation, it is, however, a highly redundant setting since $L \geq N/2$ cannot be fit exactly for any block without exceeding the image (block) boundaries. Therefore, $L < N/2$ would be a reasonable choice for $L$.

The computation of $\Psi^{\xi}(x,y)$ can be performed in a single pass through all image blocks. Let $b_i=(x_i,y_i)$ be the next block hosting the DC $c_i$. Fixing the $b_i$ in the center (i.e. $\Psi^{\xi}(0,0)$), all image blocks within the range $L$ from $b_i$ (i.e. $\forall b_j=(x_i+x_i+y_i+y_i)\in I | \forall x,y\in[-L,L]$) are scanned and the corresponding (proximity) grid bin, $\Psi^{\xi}(x,y)$, for a color $c_j$ in a block $b_j=(x_i+x,y_i+y)\in I$ is incremented by one. This process is illustrated on a sample image shown in Figure 4-4. During the raster-scan of uniform blocks, the block with white DC updates only three proximity grids (white to white, brown and blue) since those DCs can only be found within the range of $\pm L$. For illustration purposes the parameters are kept as $D_{\psi}^{\text{max}}=5 \Rightarrow N=32$ and $L=4$. 


Figure 4-4: The process of proximity grid formation for the block (X) for $L=4$.

As a result such a proximity grid characterizes both inter-DC proximities and the relative spatial position (inter-DC direction) between two DCs. This is straightforward to see in the sample images in Figure 4-5 where proximity grid distinguishes the relative direction of a DC pair, (red-blue) whilst proximity histogram cannot due to its scalar metric. Note that $\Psi_{c_i}^{x,y}(0,0) = 0$ for $i \neq j$ and $\Psi_{c_i}^{x,y}(0,0)$ indicates the total number of blocks hosting $c_i$. Since this is not a SCD property –rather a local DC property showing a noisy approximation of $w_i$ (weight of $c_i$), it can be conveniently excluded from the feature vector and the remaining $(2L+1)^2 - 1$ grid bins are (unit) normalized by the total number of blocks, $N^2$, to form the final descriptor, $\overline{\Psi}_{c_i}^{x,y}(x,y)$ where $\overline{\Psi}_{c_i}^{x,y}(x,y) \leq 1$, $\forall x,y \in [-L,L]$. 
Proximity grid computation takes $O(4N^2L^2)$. Similar to proximity histogram this is also independent from original image dimensions, $W$ and $H$, and for a full range process, $(L = N/2)$, the same number of computations, $O(N^2)$, is obtained; however, instead of regular addition operations required for proximity histogram or multiplications and divisions for Correlogram, proximity grid computation only requires incrimination, which is less costly compared even with addition operations. So for a typical grid dimension range, e.g. $N/8 < L < N/4$, the computation of proximity grid takes the shortest time. The memory space requirement is in $O(16N_{dc}^2L^2)$ and for a full range process $(L = N/2)$ with the typical settings $D_{dr}^{max} = 5 \Rightarrow N = 32$ and $N_{dc}^{max} = 8$, the memory required per database image will be 256Kb, still less than the Auto-Correlogram ($O(16md)$) even with the minimal $m$ and $d$ settings and half of the memory requirement of the proximity histogram. Since $\Psi_{e_i}^{r}(x,y) = \Psi_{e_i}^{r}(-x,-y)$ (symmetric with respect to origin), the storage (disc) space requirement is even lesser, $O(16\left(\frac{N_{dc}+1}{2}\right)L^2)$; however it requires 8 times more space than the proximity histogram. This is the cost of computing full-range proximity grid and therefore, it is recommended to employ the typical grid dimension range (e.g. $N/8 < L < N/4$) to reduce this cost to an acceptable level.

4.2 Similarity Distance – The Penalty Trio Model

Recall that in a retrieval operation in MUVIS, a particular feature of the query image, $Q$, is used for (dis-) similarity measurement with the same feature of a database image, $I$. Repeating this process for all images in the database, $D$, and ranking the images according to their similarity distances yield the retrieval result. As shown in Figure 4-1, the proposed
A Novel Perceptual Color Descriptor

color descriptors of $Q$ and $I$ contain both global and spatial color properties. Let $C_i^Q$ and $C_j^I$ represent the $i^{th}$ and $j^{th}$ ($i \leq N_{DC}^Q$, $j \leq N_{DC}^I$) DC classes where $N_{DC}^Q$ and $N_{DC}^I$ are the number of DCs in $Q$ and $I$ respectively. Along with these global properties, the proposed SCD descriptors of $Q$ and $I$ contain either proximity histogram ($\Phi^c_\psi(k)$) or grid ($\Psi^c_\psi(x,y)$) depending on the spatial color mode as shown in Figure 4-1. Henceforth for the similarity distance computation over the proposed color descriptor, both global and spatial color properties are used within a penalty-trio model, which basically penalizes the following mismatches between $Q$ and $I$:

- $P_\phi$: The amount of different (mismatching) DCs
- The differences of the matching DCs in:
  - $P_G$: Global color properties
  - $P_{SCD}$: SCD properties

So the penalty-trio over all color properties can be expresses as,

$$P_\Sigma(Q,I) = P_\phi(Q,I) + (\alpha P_G(Q,I) + (1-\alpha)P_{SCD}(Q,I))$$  \hspace{1cm} (21)$$

where $P_\Sigma \leq 1$ is the (unit) normalized total penalty, which corresponds to (total) color similarity distance and $0 < \alpha < 1$ is the weighting factor between global and spatial color properties. Note that all global color descriptors mentioned in Section 3.3.1 use only the first two (penalty) terms whilst discarding $P_{SCD}$ entirely. Correlogram, on the other hand, works only over $P_{SCD}$ without considering any global properties. Therefore, the proposed penalty-trio model fuses both approaches to compute a complete distance measure from all color properties.

Color (DC) matching is the key factor here. For this a two-level color partitioning is proposed: the first level partitions the group of color elements, which are too close for the human eye to distinguish, using a minimum (color) threshold, $T_{c}^{\min}$. Recall from the earlier discussion on DC extraction process that such close color elements are clustered into DC classes, i.e. $|c_i - c_j| \leq T_{S}$; $\forall c_j \in C_i$ and using the same analogy $T_{c}^{\min}$ can conveniently set as $T_{S}$. Another threshold, $T_{c}^{\max}$, is used for the second level above which no color similarity can be perceived. Finally for two particular DCs where inter-color distance is between the two levels, i.e. $T_{c}^{\min} < |c_i - c_j| < T_{c}^{\max}$, there exists a certain level of (color) similarity but not too close so as to be perceived as one.

To start with those colors, which show some similarity are defined as “matching” and
\( T_c^{\text{max}} \) can thus be used to partition the mismatching colors from the matching ones. One can form two sets: matching \( (S^M) \) and mismatching \( (S^\phi) \) DC classes from \( C^Q \) and \( C^I \) by assigning each DC, \( c_i \in C_i \), in one set, which cannot match any DC, \( c_j \in C_j \), in the other (i.e. \( |c_i - c_j| > T_c^{\text{max}} \) for \( \forall i,j \)) into \( S^\phi \) and the rest (with at least one match) into \( S^M \). Note that \( S^M + S^\phi = C^Q + C^I \) and using the DCs in \( S^\phi \), \( P_\phi \) can directly be expressed as,

\[
P_\phi(Q,I) = \frac{\sum (w_i | C_i \in S^\phi)}{2} \leq 1
\]  

(22)

The dissimilarity (penalty, \( P_\phi \)) increases proportional with the total amount (weight) of mismatching DCs. In the worst case where there are no colors matching, \( S^M = \{\phi\} \Rightarrow P_C = P_\phi = 1 \) makes sense since color-wise two images have nothing in common and hence entirely dissimilar. In another extreme case where all DCs are matching, so \( S^\phi = \{\phi\} \Rightarrow P_\phi = 0 \) and color (dis-)similarity will only emerge from global \( (P_G) \) and spatial \( (P_{SCD}) \) color properties of the (matching) DCs. Typically \( P_\phi \) contributes a certain color distance as a natural consequence of mismatching colors between \( Q \) and \( I \), yet the rest of the distance will result from the cumulated difference of color matching. This is, however, not straightforward to compute since one DC in \( Q \) can match one or more DCs in \( I \) (or \textit{vice versa}). One solution is to apply color quadratic distance \[49\] to fuse DC distances into the total similarity metric; however besides its serious drawbacks mentioned earlier, this formulation can be applied only to distance calculation from \textit{global} DC properties and hence cannot address how to fuse SCD distances (from proximity grid or histogram of each individual DC pair).

Another option is enforcing one-to-one DC matching, i.e. one DC alone in \( Q \) can match a single DC in \( I \) by choosing the best match and discarding the other matches. This, as well, induces serious errors due to the following fact: DC extraction is nothing but a dynamic clustering algorithm in color domain and due to the variations in color composition of the scenery or its pre-fixed parameters (thresholds), it can result in over- or under-clustering. Therefore, similar color compositions can be clustered into different number of DCs and enforcing one-to-one matching misses a certain part of matching DCs from both global and spatial similarity computation and erroneous results occur. A typical example of such a consequence can be seen in Figure 4-6 where there are three images with highly similar content, i.e. “An elephant under cloudy sky”. In two images (B and C), the \textit{cloud} and \textit{sky} are distinguished during DC extraction with separate \textit{blue} and \textit{white} DCs; however in image A, only one DC (light blue) is extracted with the same parameters. Consequently there is no (one-to-one) matching problem between B and C and such a matching will
naturally reflect similar global and spatial DC properties, but between A and B or C, if the single DC (light blue) is matched only with one DC (white or blue) this will obviously yield an erroneous result on both global and spatial similarity computations since neither DC (white or blue) properties (weight, distribution, proximities to other DCs, etc.) are similar to one in A (light-blue).

![Figure 4-6: One-to-one matching of DC pairs among 3 images (A, B and C).](image)

In order to compute $P_G$ and $P_{SCD}$ while considering the DC sets in $Q$ (or $I$), which are in a close vicinity of a single DC in $I$ (or $Q$) should be first fused into a single DC, e.g. in Figure 4-6, the DC light-blue in image A is close to both white and blue in image B (and C), therefore, both colors in B should be fused into a new DC (probably a similar light-blue color) and then $P_G$ and $P_{SCD}$ can be computed accurately between A and B. In order to accomplish this, $T_c^{\min} = T_s$ is used for matching the close DCs and a twofold matching process is performed via calling TargetMatch function, which first verifies and then fuses some DCs in target set, $T$ if required by any DC in the source set, $S$. Let $S_Q^M \subset S^M$ and $S_I^M \subset S^M$ be the sets of matching DCs for $Q$ and $I$ respectively. Since any DC in any set can request fusing two or more DCs in the other set, It is called twice, i.e. first $TargetMatch(S_Q^M, S_I^M)$, then $TargetMatch(S_I^M, S_Q^M)$. Accordingly, $TargetMatch$ can be expressed as follows:
The function, \textit{FuseDCs}, fuses all DCs in the list, \(L_i^M\), reforms the SCD descriptors of all (updated) DC pairs (\(\Phi_{c_i}^j(k)\) or \(\Psi_{c_i}^j(x, y)\)) and finally returns a new (fused) DC, \(C_x\). Then the target set, \(T\), is updated accordingly. Let \(X_{c_i}^c\) be the SCD operator (i.e. \(\Phi_{c_i}^j\) or \(\Psi_{c_i}^j\) depending on the SCD mode as shown in Figure 4-1) and \(X_{c_i}^c + X_{c_i}^\sigma\) can be defined as:
A Novel Perceptual Color Descriptor

$$X_{c_i}^\wedge + X_{c_i}^\wedge = \left\{ \begin{array}{ll}
\Phi_{c_i}^\wedge (k) + \Phi_{c_i}^\wedge (k) & \forall k \in [1, L] \\
\Psi_{c_i}^\wedge (x, y) + \Psi_{c_i}^\wedge (x, y) & \forall x, y \in [-L, L]
\end{array} \right\}$$

(23)

Let $\oplus$ be the fusing operator over DC classes. It is simple to show that $X_{c_i}^{\wedge \oplus c_i} = X_{c_i}^\wedge + X_{c_i}^\wedge$, $c_{i,2} \neq c_i$. Once the DCs in $L_0^M$ are fused, then they are removed along with their SCD descriptors whilst keeping the DCs (and their internal SCD descriptors) in $L_i^N$ intact. The new (fused) DC, $C_x$ (along with its SCD descriptors) is inserted into the target set, $T$. Recall from the earlier remarks on SCD descriptor properties, i.e. $\Phi_{c_i}^\wedge (k) = \Phi_{c_i}^\wedge (k)$ and $\Psi_{c_i}^\wedge (x, y) = \Psi_{c_i}^\wedge (-x,-y)$, once $X_{c_i}^\wedge$, $\forall c_j \in L_i^N$ are formed, it is, therefore, straightforward to compute $X_{c_i}^\wedge$, $\forall c_j \in L_i^N$.

After the consecutive calls of TargetMatch function, all DC sets in each set, which are close (matching) to a particular DC in the other set are fused and thus one-to-one matching can be conveniently performed by selecting the best matching pair in both sets. As a result the number of DCs in both (updated) sets, $S_0^M, S_1^M$ become equal (i.e. $|S_0^M| = |S_1^M| = N_M$) and one-to-one matching prevails. Assume without loss of generality that $i^{th}$ DC class in set $C_i^Q : \{c_i^Q, w_i^Q, \sigma_i^Q\} \in S_0^M$ matches the $i^{th}$ DC in set $C_i^I : \{c_i^I, w_i^I, \sigma_i^I\} \in S_1^M$ (i.e. via sorting one set with respect to other). So the penalties for global and SCD properties can be expressed as:

$$P_G(Q, I) = \beta \sum_{i=1}^{N_M} w_i^Q - w_i^I + (1 - \beta) \frac{\sum_{i=1}^{N_M} (c_i^Q - c_i^I)^2}{T_c \max N_M} \leq 1$$

$$P_{SCD}(Q, I) = \left\{ \begin{array}{l}
\sum_{i=1}^{N_M} \sum_{j=i}^{N_M} \sum_{x=1}^{L} \sum_{y=-L}^{L} \Delta \left( \Psi_{c_i}^Q (x, y) - \Psi_{c_i}^I (x, y) \right) \\
N_M^2 (2L + 1)^2 \leq 1
\end{array} \right\}$$

(24)

where $$\Delta(x - y) = \begin{cases} 0 & \text{if } x = y = 0 \\ \frac{|x - y|}{(x + y)} & \text{else} \end{cases}$$
where $0 < \beta < 1$, similar to $\alpha$, is the weighting factor between the two global color properties: DC weights and centroids. $\Delta$ is the normalized difference operator, which emphasizes the difference from zero – nonzero pairs (e.g. $=1$). This is a common consequence when the DC pairs’ area is relatively small but their SCD is quite different. It also suppresses the bias from similar SCDs of two DCs with large weights. Note that $P_{SCD}$ computation should be independent from the effect of DC weights since this is already taken into consideration within $P_G$ computation. As a result the combination of $P_G$ and $P_{SCD}$ represent the amount of dissimilarity present in all color properties and the unit normalization allows the combination in a configurable way with weights $\alpha, \beta$, which can favor one color property to another. With the combination of $P_\phi$, which represents the natural color dissimilarity due to mismatching, the penalty trio models a complete similarity distance between two color compositions.
5 Experimental Results

The experiments are performed to evaluate the proposed color descriptor efficiency with respect to HVS perceptive criteria (subjective test) and to compare retrieval (via QbE) performances within image databases indexed by the proposed and competing (Correlogram and MPEG-7 Dominant Color Descriptor) FeX modules. Both color and shape based retrievals are evaluated using the ground-truth methodology whilst providing both visual and numerical results. In the experiments performed in this chapter, 4 sample (MUVIS) databases are used:

1) **Corel_1K Image Database**: There are 1000 medium resolution (384x256 pixels) images from diverse contents of 10 classes such as wild life, city, buses, horses, mountains, beach, food, African natives, etc.

2) **Corel_10K Image Database**: There are 10000 images of 100 classes from Corel database bearing similar content as in Corel_1K.

3) **Corel_20K Image Database**: There are 20000 images of 200 classes from Corel database bearing similar content as in Corel_10K.

4) **Synthetic Image Database**: There are 1089 synthetic images covering various color compositions that are artificially created.

Note that all sample databases containing images had to be selected with mediocre resolutions; otherwise it is not feasible to apply Correlogram method and especially for Corel_10K and Corel_20K, as severe feasibility problems were witnessed due to its computational complexity. The reason of using three Corel databases with different sizes is to test and compare the scalability of the proposed color descriptor against the (increasing) database size. Finally the performance evaluation is presented over Synthetic database is to demonstrate the true description power of the proposed technique whenever color alone entirely characterizes the content of the image. Moreover, the robustness of the proposed descriptor is also evaluated against the changes of resolution, aspect ratio, color variations, translation, etc.

All experiments are carried out on a Pentium-5 1.8 GHz computer with 1024 MB memory. If not stated otherwise, the following parameters are used for all the experiments performed throughout this section: \( N_{DC}^{\text{max}} = 6, T_A = 2\%, T_s = 15 \) for DC extraction, \( T_w = 96\%, D_{QT}^{\text{max}} = 6 \) for QT decomposition and \( T_C^{\text{max}} = 45, T_C^{\text{min}} = T_s, \alpha = \beta = 0.5 \) for
penalty-trio model. For Auto-Correlogram, RGB color histogram quantization is set as 8x8x8 ($m=512$ colors) with $d=20$ for Corel_1K but 4x4x4 ($m=64$ colors) with $d=10$ for Corel_10K and Corel_20K. For Correlogram, 4x4x4 bins are used for Corel_1K and 3x3x3 bins for Corel_10K with $d=10$. Only Auto-Correlogram had to be used for Corel_20K due to Correlogram’s infeasible memory requirement for this database size. Same DC extraction parameters are used for MPEG-7 DCD and the proposed descriptor.

The proposed descriptor and the competing descriptors are implemented as FeX modules into the MUVIS framework in order to achieve better comparison (Section 2.7). A MUVIS application, DbsEditor, dynamically uses the respective FeX modules for feature extraction to index sample databases with the aforementioned parameters. Afterwards, MBrowser application is used to perform similarity-based retrievals via QBE operations. A query image is chosen among the database items to be the “Example” and a particular FeX module (e.g. MPEG-7 DCD) is selected to retrieve and rank the similar (based on color) images using only the respective (MPEG-7 DCD) features and an appropriate distance metric implemented within the FeX module. The recommended distance metrics are implemented for each FeX module, i.e. quadratic distance for MPEG-7 DCD and $L_1$-norm for Correlogram.

In order to measure the performance quantitatively, an unbiased and limited formulation of Average Normalized Modified Retrieval Rank, which is defined in MPEG-7 as the retrieval performance evaluation criteria, is used (see Section 2.6).

5.1 Color-Based Retrievals in Synthetic Databases

The images in Synthetic database contain colored regions in geometric and arbitrary shapes within which uniform samples from entire color space are represented. In this way the color matching accuracy can be visually evaluated and the first two penalty terms, $P_\phi$ and $P_G$ can be individually tested. Furthermore, the same (or matching) colors form different color compositions by varying their region’s shape, size and/or inter-region proximities. Hence this allows us to test both individual and mutual penalty terms $P_G$ and $P_{SCD}$. Finally the penalty-trio’s cumulative accuracy and robustness against the variations of resolution, translation and rotation can also be tested and compared against Correlogram.

Figure 5-1 presents a snapshot of the query of an image with 3-color squares on a white background. The proposed color descriptor is used with proximity histogram as the SCD descriptor and the retrieval results are ranked from left to right and top to bottom and the similarity distances are given on the bottom of the images. Among the first 6 retrievals, the same amounts of identical colors are used and hence $P_\phi = P_G = 0$, which allows us to test the accuracy of $P_{SCD}$ alone. The first three retrievals have insignificant similarity distances and this demonstrates the robustness of $P_{SCD}$ against the variations of rotation and
translation. The 4\textsuperscript{th}, 5\textsuperscript{th} and 6\textsuperscript{th} ranks present cases where proximity between the three colors starts to differentiate and hence SCD descriptor reflects the proximity differences successfully. For the 7\textsuperscript{th} ($P_{\phi} \neq 0$) and 8\textsuperscript{th} ranks ($P_{G} \neq 0$), $P_{z}$ starts to build up significantly since the color composition changes drastically due to emerging and missing color components.

Figure 5-1: Query of a 3-color object (top-left) in synthetic database.

Figure 5-2 presents retrieval results in a database consisting of 5 images where the proximity between red and blue blocks changes gradually, starting from a close union of four blocks to a separation of each block to the corners of the image. Intentionally one image is left with one block missing in order to sense the effect of both global and spatial color distribution ($P_{G}$ and $P_{SDC}$) Apparently, Correlogram with 4x4x4 RGB partitioning and $d=10$ fails to capture the high similarity between the query and the last retrieved image (blocks with the closest proximity). Note further that its retrieval occurs in a lower rank than the image with one missing block (blue). This is following from the fact that Correlogram fails to reach from one block to another due to its limited range.

Figure 5-2: Comparison of the retrievals of a 4-block image (top-left in both images) between Correlogram (left) and proposed color histogram (right)
Figure 5-3: Synthetic Retrievals via Correlogram, Proximity histogram and Proximity Grid. Top Left is the Query Image.

Figure 5-3 shows three queries (qA, qB and qC) in Synthetic database with different color compositions and resolutions. Some dimensions are tagged in yellow boxes. In qA, both proximity histogram and grid successfully retrieves images with the similar color compositions whereas Correlogram cannot due to its invariance to weight (area) and limited range. The area invariance effect can be easily seen in 2nd and particularly 3rd ranks where entirely different red and green weights occur. The same comments can be made for qB for 5th and all ranks above 7th. Moreover in qB, it is obvious that Correlogram cannot retrieve
Experimental Results

the image with identical color composition among the first 11 ranks due to its resolution (pixel-based) sensitivity. Note further that the proposed descriptor with both proximity histogram and grid first retrieves the color compositions where all colors are perfectly matching \( P_G = 0 \) with the weights in a close vicinity \( P_G \neq 0 \) and then balances between mismatching colors and weight differences of the matching ones. \( qC \) is particularly shown here to emphasize the effect of image resolution over Correlogram and the proposed descriptor. The query of the largest image among the others with dimensions in 5 different resolutions logarithmically scaled from 60 to 960 but the same color composition (4 red squares over white background), result the accurate ranking for the proposed descriptor; however Correlogram retrieved accurately only one whilst the other two are shifted to lower ranks and the one (with 60x60 dimension) is missed within 11 ranks.

5.2 Color-Based Retrievals in Natural Databases

Three sample databases (Corel_1K, Corel_10K and Corel_20K) are indexed using each \( FeX \) module and each individual (sub-) feature is used for retrieval. As presented in Table 5-1, the first retrieval experiment is performed to demonstrate the effect of DC fusing, detailed in Section 4.2, over the retrieval accuracy. Similar results of several retrieval experiments approve that DC fusing becomes the key factor for the success of the proposed descriptor. Therefore, DC fusing is applied for the rest of the experiments presented in this section.

Table 5-1: Similarity distances and ranks of A and B in Figure 4-6 when C is queried in Corel_1K.

<table>
<thead>
<tr>
<th>Query: C</th>
<th></th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P_G )</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>Fusing</td>
<td>0.176</td>
<td>0.156</td>
</tr>
<tr>
<td>Without Fusing</td>
<td>0.585</td>
<td>0.205</td>
</tr>
</tbody>
</table>

Table 5-2 presents ANMRR results and the query dataset size of each of the three Corel databases respectively. The query dataset is prepared a priori by regarding a certain degree of color-content coherency, that is, the content similarity can mostly be perceived by color similarity; however a unique, one-to-one correspondence between content and color similarities, as in the synthetic images given in the previous section, can never be guaranteed in such natural images due to the presence of other visual cues, such as texture, shape, etc. Nevertheless, according to ANMRR scores presented in the table, in all Corel databases the proposed descriptor with either SCD modes achieves superior retrieval performance than the competing methods, i.e. Correlogram, Auto-Correlogram and MPEG-7 Dominant Color combined with the quadratic distance computation. Note further that the performance improvement depends on the database size, e.g. see ANMRR scores of
Corel_1K vs. Corel_10K or Corel_20K. This indicates that the proposed color descriptor scales better than the other two with respect to database size and thus achieves a higher discrimination power respectively. Moreover, it is observed that in the majority of the queries (between 58-78%), the proposed method outperforms (auto-) Correlogram whereas the figure is even higher (76-92%) with Dominant Color. Finally for shorter descriptor size with proximity histograms, $L_\infty$-norm is used since comparative retrieval results promise no significant gain of using $L_1$ (e.g. for Corel_10K, ANMRR score of the proposed method with proximity histogram using $L_1$ is 0.254).

<table>
<thead>
<tr>
<th>Descriptors</th>
<th>Corel_1K (34 queries)</th>
<th>Corel_10K (176 queries)</th>
<th>Corel_20K (222 queries)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dominant Color</td>
<td>0.18</td>
<td>0.458</td>
<td>0.461</td>
</tr>
<tr>
<td>Auto-Correlogram</td>
<td>0.222</td>
<td>0.381</td>
<td>0.444</td>
</tr>
<tr>
<td>Correlogram</td>
<td>0.195</td>
<td>0.357</td>
<td>NA</td>
</tr>
<tr>
<td>Proposed (Prox.Histogram)</td>
<td><strong>0.154</strong></td>
<td><strong>0.263</strong></td>
<td><strong>0.357</strong></td>
</tr>
<tr>
<td>Proposed (Prox.Grid)</td>
<td><strong>0.162</strong></td>
<td><strong>0.291</strong></td>
<td><strong>0.39</strong></td>
</tr>
</tbody>
</table>

For visual evaluation, 4 retrieval results are presented in both Corel_10K and Corel_20K databases using all descriptors. For the queries as shown in Figure 5-4, both proximity histogram and grid are used against Correlogram and Dominant Color. In the 1st, 2nd and 4th queries, one can easily notice the erroneous retrievals of Correlogram due to its color area insensitivity (e.g. compare the amount of red, white and black colors between the query and 5th ranked image in the 1st query). As explained earlier, the probabilities can (accidentally) match between images with significantly different color proportions. Particularly in the 1st and 4th queries, the erroneous retrievals of Dominant Color occur due to the lack of SCD description, which also makes accidental matches between (dissimilar) images with close color proportions (e.g. in the 1st query, the amount of white, red and black colors is quite close between the query and 6th, 7th and 8th ranks; however their SCDS are not).
Figure 5-4: 4 typical queries using 4 descriptors in Corel_10K database. Top-left is the query image.
Figure 5-5: 4 typical queries using 4 descriptors in Corel_20K database. Top-left is the query image.

The retrieval results shown in Figure 5-5 are from Corel_20K database. As mentioned earlier Auto-Correlogram is used due to Correlogram’s feasibility problems. Similar arguments can be made over the retrieval results and note that the amount of erroneous retrievals is increased particularly on 2\textsuperscript{nd} and 3\textsuperscript{rd} queries since the database size is doubled and hence accidental matches occur more often than before. A more detailed illustration of retrieval results is given in Figure 5-6 which shows the individual NMRR results retrievals in both Corel_10K (50 queries) and Corel_20K (100 queries).
Figure 5-6: NMRR Results for Corel_10K (left) and corel_20K (right)
Experimental Results

Figure 5-7: Two queries in Corel_10K (left) and Corel_20K (right) databases where (Auto-) Correlogram performs better than the proposed descriptor. Top-left is the query image.

Figure 5-8: Dominant Color and QuadTree back-projections of some texture dominant images.
However, on both databases (Auto-) Correlogram may occasionally perform better than the proposed descriptor, such as the queries shown in Figure 5-7 where significant (color) texture is present in all query images. Note in Figure 5-8 that even DC back-projection may induce significant amount of loss in texture information; and since QT stops partitioning after reaching $D_{QT}^{\text{max}}$, additional texture loss occurs in QT back-projection. In accordance with the earlier remark in Section 3.3.2 stating that Correlogram is indeed a colored texture descriptor, Correlogram may catch the texture information well enough due to its pixel based structure and hence it can outperform any color descriptor whenever a textural structure is dominant.
6 Conclusion

CBIR techniques utilize various descriptors in order to describe the visual media content. Although they are claimed to lack the ability of describing the semantics in visual scenery, low-level features, such as color or texture, still form the basis of CBIR systems. Color content of an image is considered to be eloquent and easy to identify, thus representation of color composition is a very important issue in color image processing. As suggested in [111], a color descriptor should be in acceptable dimensions, computed rapidly, robust, should not be influenced with the database dimensions, and should reflect perceptual similarity. Wealth of research has been done and still going on in defining such color descriptors; however, at the current state of art, they fail to meet these requirements especially reflecting human color perception, thus discrimination power of such descriptors becomes inadequate especially with increasing database sizes.

The color descriptor presented in this thesis characterizes the perceptual properties of the color composition in a visual scene in order to maximize the description power. In other words, the so-called outliers, which are the un-perceivable color elements, are discarded for description efficiency using a top-down approach during extracting global and spatial color properties. In this way severe problems and limitations of traditional pixel based methods are effectively avoided and in spatial domain only the perceived (visible) color components can be truly extracted using QT decomposition. In order to reveal the true SCD properties either proximity histogram or grid, which represents the inter-proximity statistics in scalar and directional modes, is used.

During the retrieval phase, one-to-many DC matching is performed in order to apply the penalty-trio model over matching (and possibly fused) DC sets. This greatly reduces the faulty mismatches and erroneous similarity distance computations. The proposed penalty-trio model computes the normalized differences in both spatial and global color properties and combines all so as to yield a complete comparison between two color compositions.

The proposed method is implemented as a FeX module into the MUVIS framework [90] in order to test the retrieval performance and compare it to the other competitive color descriptors (MPEG-7 Dominant Color and Color Correlogram). Experimental results approve the superiority of the proposed descriptor over the competing methods in terms of discrimination power and retrieval performance especially on large databases. It has a major advantage of being applicable to any database size and image resolution. Thus it does not suffer from the infeasibility problems and severe limitations of Correlogram. Finally, it
achieves a significant performance gain on ANMRR scores; however, this is below the expectations particularly when compared with Correlogram due to two reasons: first and the foremost Correlogram has the aforementioned advantage on describing the texture in color images thanks to its pixel-level analysis via co-occurrence probabilities. Yet the major reason is that color alone does not provide complete information for content-based retrieval over general, broad-context image databases. It has been shown that color properties correlate with the true content only in certain extend, but cannot be used as a single cue to characterize the entire content [111].

Current research work includes tuning the proposed method as a shape descriptor since the distribution of colors may also describe the shape of an object. Whenever an object is entirely represented by a single DC—or a certain gray-level value as in the case of binary images or gray-level segmentation masks, then the proposed SCD may turn out to be an efficient shape feature via describing the distribution of the object parts (via auto SCD) and their relative distribution with respect to the background parts (via inter SCD). Additional feature work includes configuring the penalty-trio model dynamically and adaptively according to color compositions of the images compared and integrating second order statistics from both global and spatial properties into the descriptor.
References


[66] Last.fm, [www.last.fm](http://www.last.fm)


[97] Pandora, [www.pandora.com](http://www.pandora.com)


References


Appendix

- **RGB to HSV conversion:**
  \[
  \begin{align*}
  \text{define } & \quad \text{MAX} = \max(R, G, B), \text{MIN} = \min(R, G, B) \text{ and } K = \text{MAX} - \text{MIN}; \\
  H' &= \begin{cases} 
  \frac{G - B}{K} & \text{if MAX} = R \\
  2 + \frac{B - R}{K} & \text{if MAX} = G \\
  4 + \frac{R - G}{K} & \text{if MAX} = B \\
  \frac{\text{MAX}}{\text{NA}} & \text{if MAX} = \text{MIN} 
  \end{cases} \\
  S &= \begin{cases} 
  0 & \text{if MAX} = 0 \\
  \frac{K}{\text{MAX}} & \text{else} 
  \end{cases} \\
  V &= \text{MAX} \\
  \end{align*}
  \]
  \[
  \text{and } H = \begin{cases} 
  60 \times H' + 360 & \text{if } H < 0 \\
  60 \times H' & \text{else} 
  \end{cases}
  \]

- **HSV to RGB Conversion:**
  \[
  \begin{align*}
  \hat{H} &= \left\lfloor \frac{H}{60} \right\rfloor \text{ and } h = \frac{H}{60} - \hat{H}; \\
  a &= (1 - S) \times V; \\
  b &= (1 - (S \times h)) \times V; \\
  c &= (1 - (S \times (1 - h))) \times V; \\
  \end{align*}
  \]
  \[
  \begin{align*}
  &A \quad \hat{H} = 0 \\
  a &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  b &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  R &= \begin{bmatrix} a \\ b \\ V \\ a \end{bmatrix} \\
  G &= \begin{bmatrix} c \\ a \\ V \\ a \end{bmatrix} \\
  B &= \begin{bmatrix} c \\ a \\ V \\ b \end{bmatrix} \\
  &f \hat{H} = 1 \\
  \end{align*}
  \]
  \[
  \begin{align*}
  &A \quad \hat{H} = 1 \\
  a &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  b &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  R &= \begin{bmatrix} a \\ b \\ V \\ a \end{bmatrix} \\
  G &= \begin{bmatrix} c \\ a \\ V \\ a \end{bmatrix} \\
  B &= \begin{bmatrix} c \\ a \\ V \\ b \end{bmatrix} \\
  &f \hat{H} = 2 \\
  \end{align*}
  \]
  \[
  \begin{align*}
  &A \quad \hat{H} = 2 \\
  a &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  b &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  R &= \begin{bmatrix} a \\ b \\ V \\ a \end{bmatrix} \\
  G &= \begin{bmatrix} c \\ a \\ V \\ a \end{bmatrix} \\
  B &= \begin{bmatrix} c \\ a \\ V \\ b \end{bmatrix} \\
  &f \hat{H} = 3 \\
  \end{align*}
  \]
  \[
  \begin{align*}
  &A \quad \hat{H} = 3 \\
  a &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  b &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  R &= \begin{bmatrix} a \\ b \\ V \\ a \end{bmatrix} \\
  G &= \begin{bmatrix} c \\ a \\ V \\ a \end{bmatrix} \\
  B &= \begin{bmatrix} c \\ a \\ V \\ b \end{bmatrix} \\
  &f \hat{H} = 4 \\
  \end{align*}
  \]
  \[
  \begin{align*}
  &A \quad \hat{H} = 4 \\
  a &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  b &= \begin{bmatrix} V \\ c \\ a \end{bmatrix} \\
  R &= \begin{bmatrix} a \\ b \\ V \\ a \end{bmatrix} \\
  G &= \begin{bmatrix} c \\ a \\ V \\ a \end{bmatrix} \\
  B &= \begin{bmatrix} c \\ a \\ V \\ b \end{bmatrix} \\
  &f \hat{H} = 5 \\
  \end{align*}
  \]
• **AdobeRGB to XYZ Conversion:**

\[
[X \ Y \ Z] = [R \ G \ B] \begin{bmatrix}
0.5767 & 0.297361 & 0.0270328 \\
0.185556 & 0.627355 & 0.0706879 \\
0.188212 & 0.07528 & 0.991248
\end{bmatrix}
\]

• **XYZ to AdobeRGB Conversion:**

\[
[R \ G \ B] = [X \ Y \ Z] \begin{bmatrix}
2.04148 & -0.969258 & 0.0134455 \\
-0.564977 & 1.87599 & -0.118373 \\
-0.344713 & 0.0415557 & 1.01527
\end{bmatrix}
\]

• **XYZ to Luv Conversion:** (reference white: \([X_r, Y_r, Z_r]\))

\[
L = \begin{cases} 
116\sqrt{\frac{y_r}{Y_r}} & \text{if } y_r > \varepsilon \\
\frac{\varepsilon}{Y_r} & \text{if } y_r < \varepsilon 
\end{cases}, \\
u = 13L(\hat{u} - \hat{u}_r); \\
u = 13L(\hat{v} - \hat{v}_r); \\
\text{where } \hat{u} = \frac{4X}{X + 15Y + 3Z}, \quad \hat{v} = \frac{9Y}{X + 15Y + 3Z}, \quad \hat{u}_r = \frac{4X_r}{X_r + 15Y_r + 3Z_r}, \quad \hat{v}_r = \frac{9Y_r}{X_r + 15Y_r + 3Z_r}; \\
\varepsilon = \frac{216}{24389} \text{ and } \kappa = \frac{24389}{27}
\]
• **Luv to XYZ Conversion:** (reference white: \([X_r, Y_r, Z_r]\))

\[
X = \frac{d - b}{a - c}, \quad Y = \begin{cases} \left(\frac{(L + 16)}{116}\right)^3 & L > \kappa e \\ \frac{L}{\kappa} & L \leq \kappa e \end{cases}, \quad Z = Xa + b;
\]

where
\[
a = \frac{1}{3}\left(\frac{52L}{u + 13Lv_r} - 1\right), \quad b = -5Y, \quad c = \frac{1}{3}, \quad d = Y\left(\frac{39L}{v + 13Lv_r} - 5\right);
\]
\[
\hat{u}_r = \frac{4X_r}{X_r + 15Y_r + 3Z_r}, \quad \hat{v}_r = \frac{9Y_r}{X_r + 15Y_r + 3Z_r}, \quad e = 216/24389 \quad \text{and} \quad \kappa = 24389/27
\]

• **XYZ to Lab Conversion:** (reference white: \([X_r, Y_r, Z_r]\))

\[
L = 116f_x - 16, \quad a = 500(f_x - f_y) \quad \text{and} \quad b = 200(f_y - f_z)
\]

where
\[
f_x = \begin{cases} \sqrt[3]{x_r} & x_r \geq e \\ \frac{\kappa x_r + 16}{116} & x_r < e \end{cases}, \quad f_y = \begin{cases} \sqrt[3]{y_r} & y_r \geq e \\ \frac{\kappa y_r + 16}{116} & y_r < e \end{cases}, \quad f_z = \begin{cases} \sqrt[3]{z_r} & z_r \geq e \\ \frac{\kappa z_r + 16}{116} & z_r < e \end{cases};
\]
\[
x_r = \frac{x}{X_r}, \quad y_r = \frac{Y}{Y_r}, \quad z_r = \frac{Z}{Z_r};
\]
\[
e = 216/24389 \quad \text{and} \quad \kappa = 24389/27
\]

• **Lab to XYZ Conversion:** (reference white: \([X_r, Y_r, Z_r]\))

\[
X = x_r X_r, \quad Y = y_r Y_r, \quad Z = z_r Z_r,
\]

where
\[
x_r = \begin{cases} \frac{f_x^3}{(116f_x - 16)/\kappa} & f_x^3 \geq e \\ \frac{a}{500} + f_y & f_x^3 \leq e \end{cases}, \quad f_x = \frac{(L+16)/116}{y_r \geq e}; \quad f_y = \begin{cases} \frac{L}{\kappa} & L > \kappa e \\ \frac{L}{\kappa} & L \leq \kappa e \end{cases}, \quad f_z = \begin{cases} \frac{f_z^3}{(116f_z - 16)/\kappa} & f_z^3 \geq e \\ \frac{b}{200} & f_z^3 \leq e \end{cases};
\]
\[
e = 216/24389 \quad \text{and} \quad \kappa = 24389/27
\]