University of Alberta

Color-Based Image Retrieval Using Compact Binary Signatures

by

Vishal Chitkara

Technical Report TR 01-08
May 2001

DEPARTMENT OF COMPUTING SCIENCE
University of Alberta
Edmonton, Alberta, Canada

Color-Based Image Retrieval Using Compact Binary Signatures

Vishal Chitkara

May 17, 2001

Abstract

Significant research has focused on determining efficient methodologies for retrieving images in large image databases. This thesis addresses the design and implementation of a new image abstraction technique based on compact signature bit-strings and an appropriate similarity metric. Performance evaluation of image retrieval on a large heterogeneous database of up to 50,000 images demonstrated that the proposed technique is able to outperform the use of Global Color Histograms by up to 55%, and that of Color Coherence Vectors by up to 20% in terms of *retrieval effectiveness* – this relative advantage was also observed when using classical Precision vs. Recall curves. Perhaps more important is the fact that the proposed approach saves 75% (87.5%) of storage space when compared to Global Color Histograms (Color Coherence Vectors). This makes it possible to store and search an image database of reasonable size, without the aid of (complex) disk-based access structures.

1 Introduction and Motivation

The enormous growth of image archives has significantly increased the demand for research efforts aimed at efficiently finding similar images within a large image database. One popular strategy of searching for images within an image database is called Query By Example (QBE) [6], in which the query is expressed as an image template or a sketch thereof, and is commonly used to pose queries in most Content-Based Image Retrieval (CBIR) systems such as IBM's QBIC[1], Virage's VIR Image Engine retrieval system[2], and IBM/NASA's Satellite Image Retrieval system[3].

Typically, a CBIR system extracts visual features from a given query image, which are then used for comparison with the features of other images stored in the database. The similarity function is thus based on the abstracted image content rather than the image itself. One should note that given the ever-increasing volume of image data that is available, the approach of relying on human-assisted annotations as a means of image abstraction is not feasible. The global color distribution of an image is one such feature that is extensively utilized to compute the abstracted image content. It exhibits desired features such as low complexity for extraction, invariance to scaling and rotation, and partial occlusion [61]. In fact, it is common to use a *Global Color Histogram* (GCH) to represent the distribution of colors within an image. Assuming a n-color model, a GCH is then an n-dimensional feature vector $(h_1,\ h_2,\ ...,\ h_n)$, where h_j represents the normalized percentage of color pixels in an image corresponding to each color element c_j. In this context, the retrieval of similar images is based on the similarity between their respective GCHs. A common similarity metric is based on the Euclidean distance between the abstracted feature vectors that represent two images.

The effectiveness of a CBIR system ultimately depends on its ability to accurately identify the relevant images. Our basic motivation is based on the observation that classical techniques based on GCHs often offer poor performance since they treat all colors equally, and take only their distributions into account. More specifically,

[1]http://wwwqbic.almaden.ibm.com/
[2]http://www.virage.com/products/vir-irw.html
[3]http://maya.ctr.columbia.edu:8080/

the relative density of a color is not taken into account by these approaches. Let us motivate this point through a simple example. Consider two different pictures of a single colorful fish, with two quite different and large backgrounds. The similarities/differences among the various less dense colors of the fish are likely more important than the large difference in the background. Hence, we believe that colors relatively less dense should be given more attention than the more prominent colors.

It should be noted though, that in using either the GCH or Grid-based approach, storing n- dimensional vectors of a color histogram for each image in the database may consume significant storage space. In order to minimize the space requirement, we propose the use of a compact representation of these vectors, thereby leading us to the utilization of binary signature bit-strings. An image's signature bit-string, hereafter referred to as *signature*, is an abstract representation of the color distribution of an image by bit-strings of a pre-determined size. To process a query, all image signatures are compared against the signature of the query image using a well-defined similarity metric. The candidate images are then retrieved and ranked according to their similarity with the query image.

One should note that, in order to avoid sequential scanning, the signatures need to be indexed if the image database is of a non-trivial size. An ideal index structure would then quickly filter out all irrelevant images. There exist various access structures for dynamic indexing of images based on feature vectors, such as the family of R-Trees [16]. The issue of efficiently indexing image signatures is however not the primary focus of this research. The main emphasis of this thesis is to demonstrate that the proposed signatures are an efficient and compact alternative to the use of GCHs. Indeed, we will argue later that the use of the proposed signatures for images may be useful for storing and searching an image database of reasonable size using a relative small amount of main memory. Hence, we might avoid, in some non-trivial cases, the need of a disk-based access structure. Nevertheless, if one aims at indexing several millions of images then a disk-based index is needed.

The thesis is organized as follows. Section 2 contains an overview of the current literature on existing image retrieval systems and techniques. Section 3 contains a description of the proposed image abstraction methodology. A majority of the current image retrieval systems and techniques have been directed towards representation of only those colors in the color histogram that have a significant pixel dominance. The proposed image abstraction methodology however emphasizes more on the colors which are less dominant while still taking major colors into account. In doing so, we have also introduced a metric for ranking the results, based on the comparison of signatures between a query image and the image collection. The methodology to evaluate the effectiveness of the retrieval procedure in the proposed and existing techniques, and the obtained experimental results, are presented in Section 4. Finally, Section 5 concludes the thesis and states directions for future work.

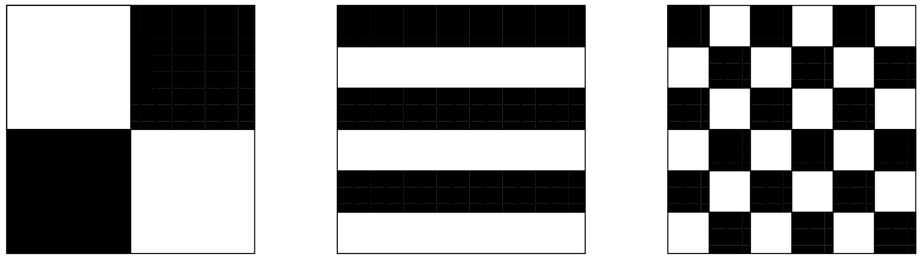
2 Related Work

The past few years have been directed to considerable research efforts in the field of image processing and retrieval. The primary focus of these efforts is to model the image contents as a set of pre-defined feature vectors. The similarity between images is thus based on the computation of the distance function on the extracted features; the most common form of the distance function being the Euclidean distance. Finally, the images are ranked based on the ascending order of distance and returned to the user as the query results.

Each of these efforts propose a different framework for generating the feature vectors. They also differ on their usage of the visual attributes and the color space models. In what follows, we give an overview of some of the commonly used visual attributes, color space models, and other representative work.

2.1 Representation of Visual Attributes

A CBIR system processes the information contained in an image, and creates an abstract representation of its content in terms of the visual attributes. Any query operations would then be solely performed on this abstracted



Block Pattern Striped Pattern Checkerboard

Figure 1: Three different textures with the same distribution of colors

content, rather than the actual image itself. The following contains a brief overview of some of the commonly used visual attributes.

Color is an extensively utilized visual attribute in image retrieval, that often simplifies object identification and extraction [18, 44]. Color is especially convenient, because it provides multiple measurements at a single pixel of the image, often enabling the classification to be done without the need of complex spatial decision-making [47]. The retrieval based on color similarity requires distances in color space to correspond to human perception [6]. Any resulting difference between colors is then evaluated as a distance between the corresponding color points. Traditionally, color histograms have been used to measure the weighted dissimilarity between any two images. Over the past few years, various other methodologies have been proposed to evaluate the similarity. The color representation and retrieval of these techniques is addressed in the subsequent sub-sections.

The visual attribute texture is a representation of the surface of an image object [22]. Intuitively, the term refers to properties such as smoothness, coarseness, and regularity of an image object [18]. In general, the structural homogeneity does not result from the presence of a single color or intensity, but requires the interaction of various intensities within a region [68]. To understand the necessity of texture in image retrieval, consider the image set shown in Figure 1. The first image comprises of four blocks: two black and two white. The center image forms a striped pattern with three black and three white stripes. The rightmost image comprises of 18 black and 18 white blocks forming a check-board pattern. The color histograms for each of these three images comprises of 50% white and 50% black pixels. Therefore, in order to differentiate between these three images, we require the usage of texture features. Over the years, researchers in Computer Vision have understood the need of texture in pattern recognition and image retrieval to derive potentially more useful semantic information [25, 51].

Shape refers to the characteristic contour of an object that identifies it in a meaningful form. Traditionally, shapes are represented through a set of features such as area, axis-orientation, certain characteristic points etc. [6]. Shape representations are broadly divided into two categories, boundary-based and region-based, and the most successful representatives of these are the Fourier Descriptor and Moment Invariants respectively [44]. Retrieval of a subset of images that satisfy certain constraints is a central problem in shape retrieval, with the degree of similarity between two images calculated as the distance between corresponding points [6, 27, 34].

Spatial relationship symbolizes the arrangement of objects within an image. The relationship between two objects can be broadly classified as either directional (also referred as projective) or topological [6]. Directional relationship is based on the relative location and metric distance between two image objects. Topological relationships, on the other hand, are not based on capturing the distance between any two objects, but are based on set-theoretical concepts like union, intersection, disjunction etc. [6, 29].

One of the most important challenges in building a image retrieval system is the choice and representation of the visual attributes [43]. As already discussed above, several domains of visual attributes are possible, but

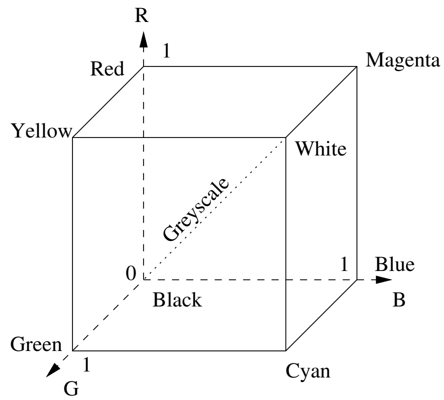


Figure 2: Color Cube for Normalized RGB Coordinates

amongst these color is most intuitive and straight-forward for the user [31, 55]. Undoubtedly, shape and texture are also important visual attributes to derive potentially more useful semantic information, but there exists comparatively less understanding of how best to implement these attributes as compared to color, for an efficient image retrieval [19]. Nevertheless, a survey about these can be found in ([6] Chapter 3,4).

The scope of this research is circumscribed to image abstraction and retrieval based only on the color content and in what follows, we give a brief overview of the various color models.

2.2 Color Space Models

There exist various color models, dictated by the means through which an image is intended to be used [6]. Most color space models define colors in three dimensions, such that each color is represented by three co-ordinates. Furthermore, the color models are classified as uniform/non-uniform depending upon the difference in color space, as perceived by an observer. The difference between any two colors is approximated to be the Euclidean distance in a uniform color space [6].

RGB is the most commonly used color space model, and is composed of three primary colors - Red, Green, and Blue. The primary colors are *additive*; that is by varying their combinations, other colors can be obtained [7, 13, 23]. The model is visualized as a unit cube (Figure 2), with corners of black, white, the three primary colors (red, green, blue), and the three secondary colors (cyan, magenta, yellow). The color model however bears the limitation that it is not perceptually uniform, meaning that the calculated distance in the RGB space does not truly correspond to the perceptual color difference [32, 50].

The CMY color model is based on the secondary colors of the RGB color space model, that is - cyan (green plus blue), magenta (red plus blue) and yellow (red plus green) [7, 13]. The subset of the cartesian coordinate system for the CMY color model is similar to that of the RGB color space, except that the white color occupies the origin (Figure 3). Other colors are obtained by performing either an addition or a subtraction on the white component [23]. The color model however bears the limitation that each of the three base colors are never available as pure colors, and are always adulterated by a certain proportion of each other. It is therefore impossible to create pure black color using this model and in order to overcome this problem, the color model has been extended to form another color model, referred to as the CMYK color model, which uses black as the fourth color [2].

The HSV (Hue, Saturation, Value) color model (also referred to as HSB model, with B for brightness) is suitably equipped to meet human perception of color [30]. The model can be visualized as being cylindrical, but usually represented by a hexagonal cone (Figure 4). Value (analogous to intensity) is represented along the

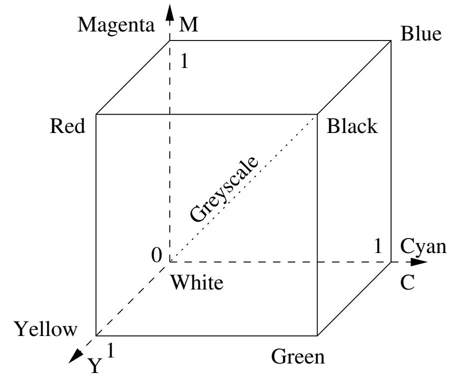


Figure 3: Color Cube for Normalized CMY Coordinates

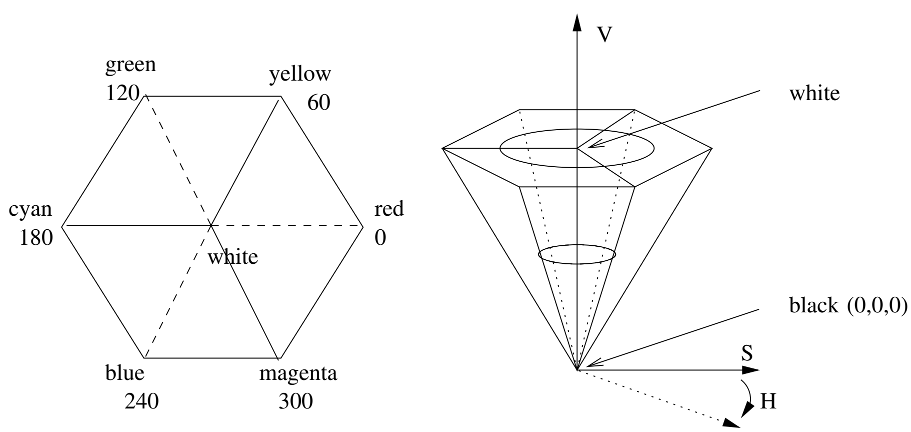


Figure 4: Hexagonal Cone for HSV Color Model

vertical axis, with V=1 containing the relatively bright colors. Saturation is represented on a side of the hexagonal cone, with hue being an angle around the vertical axis. As saturation varies from 0 to 1.0, the corresponding colors vary from unsaturated (gray) to saturated (no white component). Hue ranges from [0,360] degrees, with corresponding variation beginning from red, through yellow, green, cyan, blue, and magenta, back to red, such that there are red values both at 0 and 1.0. This is a device dependent, non-uniform color space, which intuitively corresponds to the RGB color space model, with the main diagonal of the RGB being the V-axis of the HSV color model.

The Munsell color space is a perceptually uniform model, represented in cylindrical coordinates with its dimensions being brightness, hue and saturation [6]. Brightness is represented along the vertical axis, with 11 distinct variations as the range changes from 0 to 1.0. Hue represents the angular displacement, with a representation of five primary (Blue, Green, Yellow, Red, Violet) and five secondary (Green-Blue, Green-Yellow, Red-Yellow, Red-Violet, Blue-Violet) colors [6]. Saturation, represented along the circular radius, varies from unsaturated to saturated colors.

$L^*u^*v^*$ color model is another uniform color space, based on the human perception. The component L^* is perceptually based on luminance (brightness), with u^* and v^* being the chromatic coordinates (Figure 6) [6]. Within the u^* and v^* perceptual coordinate system, there exists a triangle of visible spectrum. The triangle is represented with blue at the bottom, and green and red being the upper left and right vertices respectively [10].

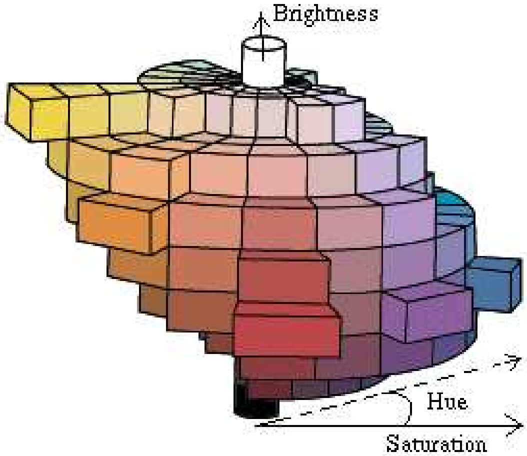


Figure 5: A 3D solid representation of the Munsell system (adapted from [1])

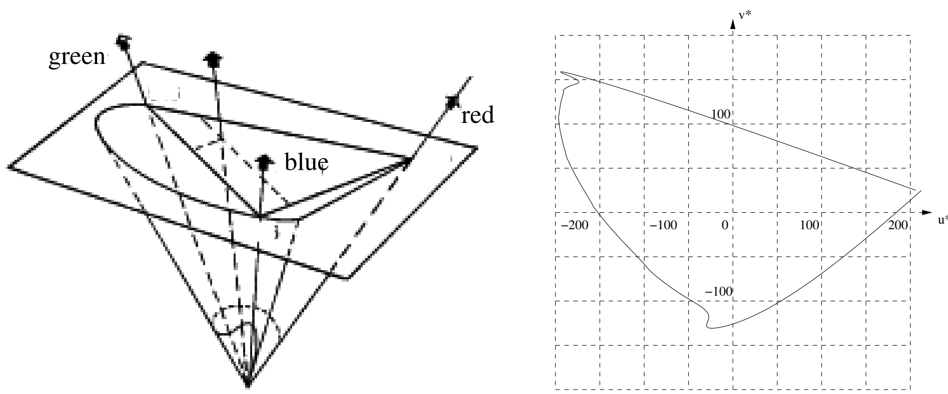


Figure 6: Color representation in the $L^*u^*v^*$ chromatic coordinate system

All visible colors with the same chromaticity, but different luminance map onto the same point in the triangle.

The selection of a color space is crucial for deriving potentially more useful object-related color information, and has been focus of various studies [64]. Several researchers have evaluated various color models for the purpose of image retrieval under varying sets of imaging conditions [17]. It has been argued that the RGB color model closely corresponds with the physical sensors of the human eye, although the human perception is more accurately reflected using the HSV color space [65]. Nevertheless, the RGB color space is most frequently used [48, 64, 65], and also forms the basis of this research.

2.3 Color-based Image Retrieval

The existing CBIR techniques can typically be classified as those that perform image retrieval based on color, texture, shape or a combination of these. Color is an extensively utilized visual attribute that plays an important role in image identification and extraction [31]. It has been observed that even though color plays a crucial role in image retrieval, when combined with other visual attributes it would yield much better results [26]. This is because, two images with entirely similar color compositions, may differ with respect to the position of the color clusters. The scope of this research is however circumscribed to image abstraction and retrieval based only on the color content.

2.3.1 Color-based Image Retrieval Systems and Techniques

A Global Color Histogram (GCH) is the most traditional way of describing the color property of an image [6]. The GCH for an image is constructed by computing the normalized percentage of color pixels in an image corresponding to each color element. Assuming an n-color model, a GCH is then an n-dimensional feature vector $(h_1,\ h_2,\ ...,\ h_n)$, where h_j represents the (usually) normalized percentage of color pixels in an image corresponding to each color element c_j. More formally, we can define the component h_j as a unique combination of the values - red, green, and blue (obtained after normalization). The histogram by itself does not include any spatial information about an image. For example, the difference between a large region in red color and a large number of scattered red pixels would not be captured by the color histogram. In this context, the retrieval of similar images is based on the similarity between their respective GCHs. A common similarity metric is based on the Euclidean distance between the abstracted feature vectors that represent two images, and it is defined as:

$$d(Q, I) = \sqrt{\sum_{j=1}^{n}(h_j^Q - h_j^I)^2}$$

where Q and I represent the query image and one of the images in the image set, and h_j^Q and h_j^I represent the coordinate values of the feature vectors of these images respectively. Note that a smaller distance reflects a closer similarity match. The above inference stems from the fact that color histograms are mapped onto points in an n-dimensional space, and similar images would therefore appear relatively close to each other.

In recent years, there has been considerable research published regarding CBIR systems and techniques. In what follows, we give an overview of some representative work related to ours, i.e., color-oriented CBIR.

A similarity measure based on color moments is proposed in [59]. The authors propose a color representation that is characterized only by the first three color moments namely color average, variance and skewness, thus yielding low space overhead. Each of these moments is part of an index structure and have the same units, which make them somewhat comparable to each other. The similarity function used for retrieval is based on the weighted sum of the absolute difference between corresponding moments of the query image and the images within the data set. A similar approach was also proposed by Appas *et al* [3], the main difference being that the image is segmented into five overlapping cells.

The authors in [26] describe a technique for integrating color information with spatial knowledge in order to obtain an overall impression of the image. The technique is based on the following steps: Heuristic rules are used to determine a set of representative colors. These colors are then used to obtain relevant spatial information using the process of maximum entropy discretization. Finally, the above computed information is utilized to retrieve the relevant similar images from the image set. The technique is based on computing the GCH and color histogram for the grid at the center. The histogram considers only those colors that contribute at least a certain percentage of the total region area. The color with the largest number of pixels in the GCH is presumed to be the background color of the image, and the color with the largest number of pixels in the central grid is presumed as being the color of the object in the image. The technique therefore addresses the problem of background influence of an image object.

A system for color indexing based on automated extraction of local regions is presented in [55]. The system first defines a quantized selection of colors to be indexed. Next, a binary color set for a region is constructed based on whether the color is present for a region. In order to be captured in the index by a color set, a region must meet the following two requirements: (i) There must be at least N pixels in a region, where N is a user-defined parameter, and (ii) Each color in the region must contribute with at least a certain percentage of the total region area; this percentage is also user-defined. Each region in the image is represented using a bounding box. The information stored for each region includes the color set, the image identifier, the region location and the size. The image is therefore queried, based not only on the color, but also on the spatial relationship and composition of the color region.

In [58] the authors attempt to capture the spatial arrangement of different colors in the image, based on using a grid of cells over the image and a variable number of histograms, depending on the number of distinct colors present. The paper argues that on an average, an image can be represented by a relatively low number of colors, and therefore some space can be saved when storing the histograms. The similarity function used for retrieval is based on a weighted sum of the distance between the obtained histograms. The experimental results have shown that such a technique yields 55% less space overhead than conventional partition-based approaches, while still being up to 38% more efficient in terms of the quality of image retrieval.

Pass *et al* [38] describe a technique based on incorporating spatial information with the color histogram using Color Coherence Vectors (CCV). The technique classifies each pixel in a color bucket as either coherent or incoherent, depending upon whether the pixel is a constituent of a large similarly-colored region. The authors argue that the comparison of coherent and incoherent feature vectors between two images allows for a much finer distinction of similarity than using color histograms. The authors compare their experimental results with various other techniques and show their technique to yield a significant improvement in retrieval.

A large number of published work can be found in this domain. For an extensive survey refer to [6, Ch. 2] and [32, Sec. 6.4].

2.3.2 Image Retrieval Systems and Techniques

Typically, the CBIR system extracts visual features from a given query image, which are then used for comparison with the features of other images stored in the database. The similarity function is thus based on the abstracted image content rather than the image itself. One should note that given the ever-increasing volume of image data that is available, the approach of relying on human-assisted annotations as a means of image abstraction is not feasible. The color distribution of an image is one such feature that is extensively utilized to compute the abstracted image content. It exhibits the desired features such as low complexity for extraction, invariance to scaling and rotation, and partial occlusion [61]. In the following, we briefly describe the visual feature extraction and retrieval in some of the popular image retrieval systems.

QBIC, standing for Query By Image Content [12, 36] is a classical example of an image retrieval system. It performs CBIR using several perceptual features, according to the user specifications. The system utilizes a partition-based approach for representing color. Since the quadratic measure of distance is computationally intensive, the average Munsell transformation is used to pre-filter the candidate images. The system also defines a metric for color similarity based on the k-element bin in the color histogram [11]. The texture patterns are modified versions of the following measures - coarseness, contrast and directionality. The system exhibits a limited ability to search for spatial features within an image and is based on the combination of heuristic shape features and algebraic moment invariants. The QBIC system also supports multi-dimensional indexing by using orthogonal transformation, such as the Karhunen-Loeve Transform (KLT) to perform dimension reduction with an R^*-tree used as an underlying indexing structure [11].

Virage [4, 20] by Virage Inc. is another CBIR system that supports visual querying based on color, texture and spatial-relationship, performed on the following four basic primitives - Global Color, Local Color, Structure, and Texture classification. The system presents the image data information to the user using four different layers namely: domain objects and relations, domain events and relations, image objects and relations, and image representations and relations in order to provide the flexibility of simultaneously viewing the data from various abstraction levels. The system also provides the user with an ability to manipulate the weight assignment amongst the visual features [21, 68].

Photobook system [40] is an interactive set of tools developed by the M.I.T. Media Laboratory on Perceptual Computing. The system interacts with the user through a Motif interface, and performs retrieval based on text annotations. The system further improves the quality of the retrieval set by considering other visual attributes of the images. The comparison is performed on the extracted feature vectors with consideration on invariance to scaling and rotation. The version 5 of Photobook provides a library of matching algorithms for calculating the linear distance between a set of images. The version 6 of Photobook goes a step further and allows for matching user-defined algorithms via dynamic code loading. The system includes a distinct interactive agent (referred to as FourEyes) which is capable of learning from the user selection [39].

VisualSEEk [54, 56] and WebSEEk [57] are academic visual information systems developed at the Columbia University. VisualSEEk is an hybrid image retrieval system that integrates image feature extraction based upon the representation of color, texture [53] and spatial layout. Using the binary feature set back-projection on color and texture, the system extracts salient regions from an image and computes an arbitrary spatial layout for these regions which includes both an absolute and relative location. The system differs from other systems in the sense that the user diagrammatically forms the queries based on the spatial arrangement. The system is capable of performing a wide variety of complex queries due to an efficient indexing, and also because spatial issues such as adjacency, overlap and encapsulation can be addressed by the system [56]. The retrieval process is accentuated by using binary-tree based indexing algorithms [44, 52, 53, 55].

Contrary to VisualSEEk, WebSEEk [57] is a catalog-based search engine for the World Wide Web. WebSEEk supports queries on both subject-catalog and visual properties such as color, spatial-layout and texture matching features [44]. The system collects images using several autonomous Web spiders, which automatically index and assign subject-catalogs to the images using HTML and URL tags. The user initiates the search procedure by selecting a catalog from the menu and then makes a selection on the visual sub-categories.

ImageRover [46] is an image retrieval tool developed at the Boston University. The system combines both visual and textual statistics for computing the image decompositions, textual associations, and indices. The extracted visual features are stored in a vector form using color and texture-orientation histograms, while the textual features are captured using *Latent Semantic Indexing* based on associating the related words in the containing HTML document [8]. The user guides a search procedure by refining the initial query using the relevance feedback of both visual and textual indexes to achieve a better search performance. The system performs relevance feedback using Minkowski distance metric, and the retrieval process is accentuated by using an approximate k-nearest neighbors indexing scheme [46, 62].

2.3.3 Partition-based Image Retrieval

The distribution of colors within an image can be extended to capture spatial knowledge about the color distribution of an image. A commonly used extension of this approach is its incorporation into a partition based system. The idea of *Partition* or *Grid-based* approaches is to segment the image into cells of equal or varying sizes, depending upon the implementation requirements, while treating each cell as a separate image. To embed spatial information about the color distribution of an image, each image is decomposed into cells with varying visual contents. Determining the suitable block size for decomposing an image requires a careful analysis, and has been the focus of many research studies, even though no single partitioning has been deemed appropriate for all image retrieval methodologies [58]. The Grid approach therefore utilizes each cell as a separate image and the overall distance between two images becomes the sum of the individual Euclidean distances between the corresponding cells. A common similarity metric used in a Grid based approach takes the following form:

$$d(Q,I) = \sum_{k=1}^{m} \sqrt{\sum_{j=1}^{n} (h_j^Q - h_j^I)^2}$$

where Q and I represent the query image and one of the images in the image set, and h_j^Q and h_j^I represent the coordinate values of the feature vectors of these images respectively. The variable k represents the cell number in the grid placed over the image and m is the total number of cells making up the grid itself.

It is important to stress however, that the use of such a grid approach has a serious drawback. The image representation is not invariant to rotation and/or translation any longer. This is important as an image and its upside-down copy would be deemed quite different by the grid-based approach, whereas most users would call them identical. Nevertheless, should the user be willing to lose rotation/translation invariance, such an approach delivers very good results.

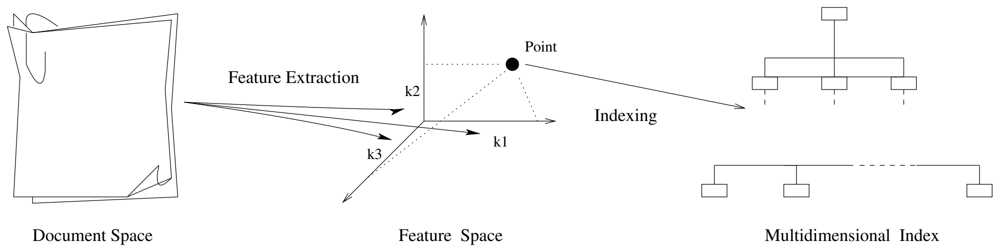


Figure 7: Mapping a document into a 3-dimensional space (adapted from [33])

2.4 Access Structures for Image Data

In general, image retrieval by content requires that for each image the visual features be extracted and mapped to a feature vector. If each feature of an image is treated as a dimension, then given the values for all these features, the image can be represented as a point in the n-dimensional vector space, say $\vec{k} = (k_1,\ k_2,\ k_3,\ k_4,\ ...,\ k_{n-1},\ k_n)$ (Figure 7) [29, 33]. A search for images with similar visual features as that of the query image, would then result in the retrieval of the actual objects that lie closest to the *query point* in a n-dimensional space [29]. For efficiency, these feature vectors are precomputed and stored [11]. The retrieval procedure for analyzing the image similarity comprises of sequentially comparing the feature vector of a query image, with those of all the other images in the database. For a small size image collection, this procedure is acceptable, however for large image collections, there is a need to index the feature vectors because sequential scanning then breaks down as an effective search procedure [6, 30].

An index structure ideally filters out the similar images by conducting nearest neighbor query search in the feature vector space [28]. The relevant images are retrieved without analyzing the entire database, a process which considerably speeds up the retrieval in a large image collection [6]. One should however note that the commonly used indexing techniques within a traditional DBMS (e.g. B^+-trees and hash tables [42]) are not suitable for indexing image feature vectors, as they deal with single-dimensional data [30]. In this regard, there has been a considerable amount of research devoted to the design of efficient indexing structures for high-dimensional data. Recent structures are the R^*-tree [5, 49], SS-tree [15, 66], and the SR-tree [28]. However, since the main scope of this thesis lies on the investigation of how effective the proposed approach is, we therefore do not include a discussion about indexing structures here. Nevertheless, we will revisit this issue briefly in Section 4 when performing some experiments regarding the scalability of linear scan for signature files.

2.5 Conclusion

This Section discussed the various issues dealing with querying and retrieval of image databases. Images are identified by various attributes and features crucial for deriving potentially more efficient retrieval. However, amongst these color is the most straight-forward and intuitive to the user. The scope of this research is circumscribed to image abstraction and retrieval based only on the color content. The Section also discusses the different techniques for color-oriented content-based retrieval of images. Content-based retrieval systems for images were also presented and discussed.

3 Color-based Image Retrieval using Compact Binary Signatures

3.1 The Proposed Model

In this sub-section, we describe the system design and implementation of *BSIm* [4], a prototype web-based image retrieval system that is based on the proposed techniques for image abstraction and retrieval. The prototype system is designed and implemented on the LINUX system using Perl script and JAVA programming language. The overall system architecture is shown in the Figure 8. It consists of two primary modules: the pre-processing and the retrieval modules. Initially, the image database is pre-processed by the *pre-processing module* to generate the binary signatures as an abstract representation for each image. The *retrieval module* accepts a sample image from the user as a query through the *user interface module*, and traverses the image metadata for matching the similar images. The user can also ask the system for a set of random images from those presented. Once the user selects an image to guide the search procedure, the retrieval of similar images is initiated. For the purpose of this study, we have thumbnailed the images such that the aspect ratio and the overall quality of the image are preserved. At time of this writing, the graphical interface is designed to display the top 15 retrieved images ranked in decreasing order of similarity, with the first image being the query image.

3.1.1 The Model Components

The system comprises of four networked components: a *user interface* for handling user input and output, a *pre-processing system* that extracts the color features from a given image to generate a binary signature, a *retrieval system* to process the query requests, and a *storage manager* responsible for maintaining the image database and its metadata.

The user interface is presented via a Web browser as a CGI document. As previously discussed, the user may initiate a search procedure by either entering the numerical identifier of a query image or by selecting an image from a a set of random images. Note that, in our implementation, we do not provide the user a flexibility to append or remove an image (and its corresponding signature) at a later stage as the sole focus of this research is to investigate the feasibility of using the proposed signatures for effective CBIR.

The pre-processing system is initiated when an image is added to the database, and begins by translating the submitted image into a common file format. Each pixel value in the image is first blurred with the average of its 8 adjacent neighbors, and then normalized in the RGB color space. For each image, the color distribution is then calculated using the GCH. The binary signature for an image is an abstract representation of the global color histogram. Finally, the binary image signature is appended to a text file containing the signatures of all the other images.

The queries are issued using the QBE paradigm. The retrieval system isolates the binary signature of the specified query image and produces a relative distance with the other image signatures. The image set is then re-ordered with respect to their ascending distances (relative to the query image), and the top retrieved images ranked in decreasing order of similarity are presented as the query's answer to the display manager.

As already discussed, the storage manager is responsible for maintaining the image database and its metadata. Any modifications to the database and metadata are handled by this module. The system comprises of two databases. The first being a flat file used for storing binary signatures of all the images, while the other contains the actual images.

3.2 Image Abstraction using Binary Signatures

Most of the summary methods described in the previous section improve upon image retrieval by incorporating other perceptual features with the color histogram, spatial information being the most common. A majority of

[4] `http://db.cs.ualberta.ca/BSIm/bsim.cgi/`

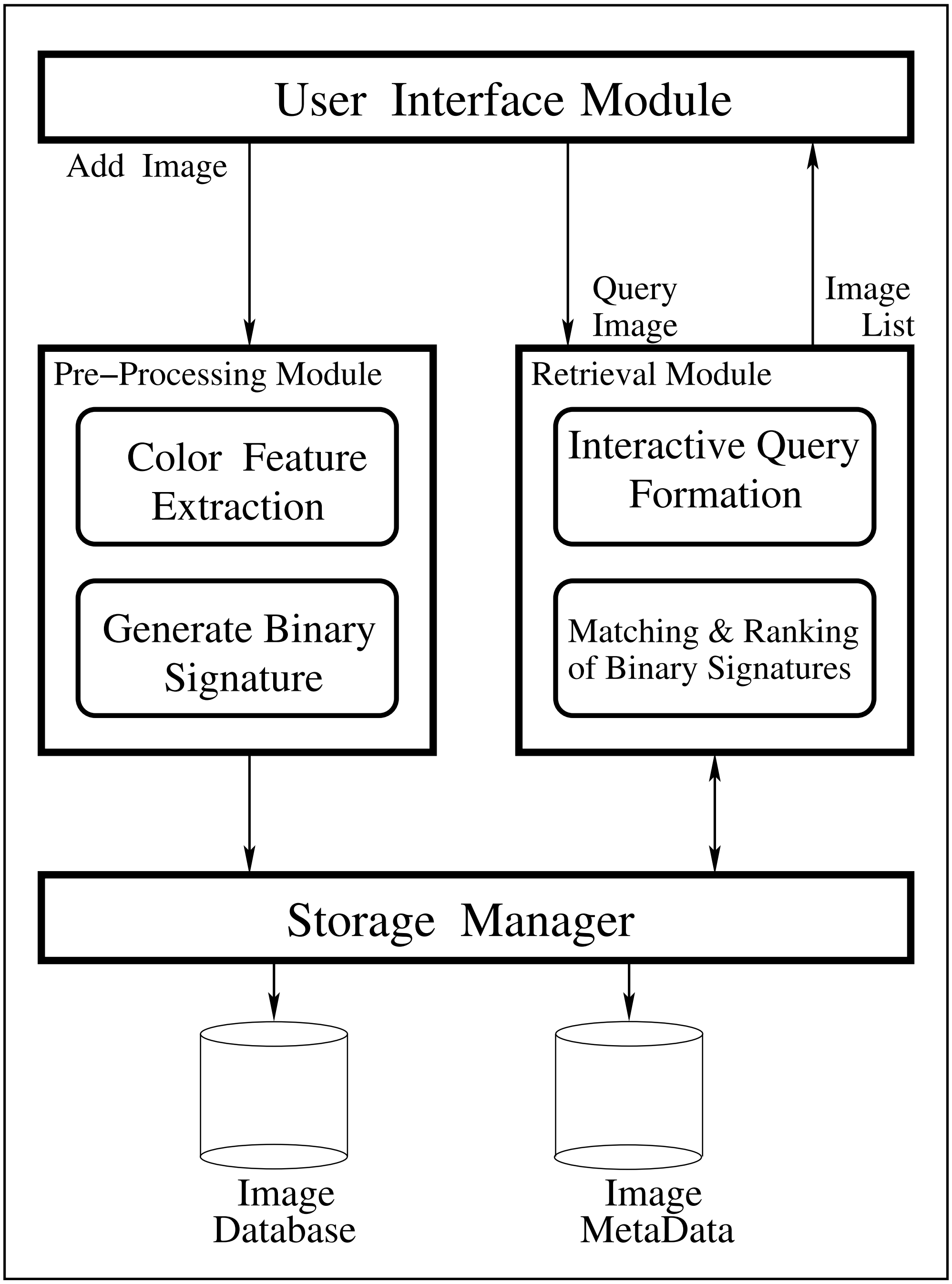


Figure 8: The system architecture of the proposed model

these efforts have been directed toward a representation of only those colors in the color histogram that have a significant pixel dominance. Our best approach differs exactly in this regard. We actually emphasize more on the colors which are less dominant while still taking major colors into account. In order to use signatures for image abstraction, we have designed the following scheme:

- Each image in the database is quantized into a fixed number of n colors, $C = (c_1,\ c_2,\ \ldots,\ c_n)$ to eliminate the effect of small variations within images and also to avoid using a large file due to the high resolution representation [37].

- Each color element c_j is then discretized into t binary bins $(B^j = b_1^j\ b_2^j\ \ldots\ b_t^j)$ of equal or varying capacities, referred to as the bin-size. If all bins have the same capacity, we say that such an arrangement follows a Constant-Bin Allocation (CBA) approach, otherwise it follows a Variable-Bin Allocation (VBA) approach. As an example, consider an image comprising of n colors and t bins. The signature of this image would then be represented by the following bit-string: $S = b_1^1\ b_2^1 \ldots\ b_t^1\ \ b_1^2\ b_2^2 \ldots\ b_t^2\ \ldots\ b_1^n\ b_2^n \ldots\ b_t^n$ where, b_i^j represents the i^{th} bin relative to the color element c_j. For simplicity, we refer to the sub-string $b_1^j\ b_2^j \ldots\ b_t^j$ as B^j $(1 \leq j \leq n)$. Therefore, the signature of an image I can also be represented as: $S_I = B_I^1\ B_I^2 \ldots\ B_I^n$.

- The normalized values obtained after automatic color extraction are used within the corresponding set of bins to generate a binary assignment of values indicating the presence or absence of a color within a particular density range. Using the CBA approach, each color c_j has its bins set according to the following condition:

$$b_i^j = \begin{cases} 1 & \text{if } i = \lceil h_j \times t \rceil \\ 0 & \text{otherwise} \end{cases}$$

  Note that we assume that the entries in the GCH are normalized with respect to the total number of pixels in the image.

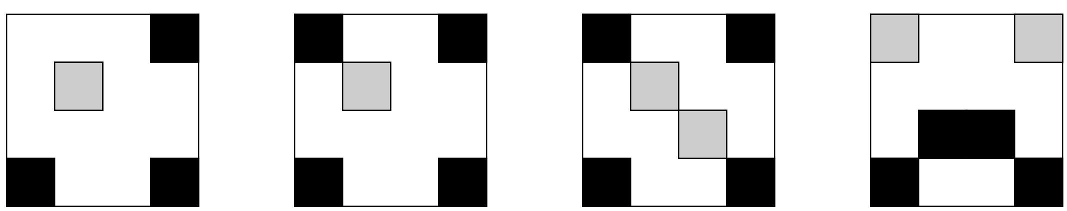


Image A　　Image B　　Image C　　Image D

Figure 9: Sample image set

As an example, consider image A in Figure 9 having three colors, and for simplicity, let us assume n=3, hence, $C = (c_1, c_2, c_3)$ = (black, gray, white). The normalized color densities can then be represented by the vector $H_A = (h_1, h_2, h_3)$ = (0.18, 0.06, 0.76) where each h_j represents the percentage pixel dominance of color c_j. Next, assume that the color distribution is discretized into $t = 10$ bins of equal capacities, that is, each bin accommodates one-tenth of the total color representation. Hence, b_1 would accommodate percentage pixel dominance from 1% to 10%, and b_2 accommodates from 11% to 20% and so on. Therefore, image A comprising of 18% of color c_1, 6% of color c_2, and 76% of color c_3 is represented by the following signature (as detailed in Table 1—which also shows the signatures for the other images in Figure 9): $S_A = 010000000010000000000000000100$.

Such an assignment within a set of bins which have equal capacities to accommodate the percentage pixel dominance is referred to as the *Constant-Bin Allocation* (CBA). Another approach, Variable-Bin Allocation (VBA), is based on an assignment where the bins within a set accommodate varying capacities and is presented

Color Set of Bins	Color Density	Binary Signature									
		b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
Image A											
c_1/B_A^1	18%	0	1	0	0	0	0	0	0	0	0
c_2/B_A^2	6%	1	0	0	0	0	0	0	0	0	0
c_3/B_A^3	76%	0	0	0	0	0	0	0	1	0	0
Image B											
c_1/B_B^1	24%	0	0	1	0	0	0	0	0	0	0
c_2/B_B^2	6%	1	0	0	0	0	0	0	0	0	0
c_3/B_B^3	70%	0	0	0	0	0	0	1	0	0	0
Image C											
c_1/B_C^1	24%	0	0	1	0	0	0	0	0	0	0
c_2/B_C^2	12%	0	1	0	0	0	0	0	0	0	0
c_3/B_C^3	64%	0	0	0	0	0	0	1	0	0	0
Image D											
c_1/B_D^1	24%	0	0	1	0	0	0	0	0	0	0
c_2/B_D^2	12%	0	1	0	0	0	0	0	0	0	0
c_3/B_D^3	64%	0	0	0	0	0	0	1	0	0	0

Table 1: Detailed Signatures using CBA

in the next sub-section. The parameters n and t play an important role in the retrieval efficacy of the proposed methodologies, as we shall see during the course of this section.

Recall that each bin is represented by a single bit, therefore the obtained signature is a compact, yet effective representation of the color content, with due emphasis being placed on the space overhead. To discuss this further, let us assume that the storage of a real number requires f bytes. Therefore, to store the GCH of an image comprising of n colors, $(n \times f)$ $bytes$ would be required, whereas the proposed image abstraction requires only $(n \times t)$ $bits$ for representing an image. Similarly, the Color Coherence Vector (CCV) [38] technique would require $(2 \times n \times f)$ $bytes$ to represent an image. The proposed technique is therefore much more space efficient than both CCVs and GCHs. For instance, by using $n = 64$, $f = 2$ bytes and $t = 10$, the CBA's signature requires 80 bytes per image, a GCH then requires 128 bytes, whereas the CCV requires 256 bytes. That is to say that CBA's space overhead is 37% smaller than that required by the corresponding GCH, and 68.75% than that required by the CCV. In addition, it is reasonable to expect that a large number of colors may not be present in an image, which may lead to a large sequence of 0's, which would allow the use of compression techniques, e.g., simple run-length encoding [14], to save even more space.

A more careful analysis of the signatures reveal that at most one single bit is set within any bin set B^j. Hence, one could simply record the *position*, within B^j, of that one bit which is set instead of the whole 10 bit signature for B^j. If each bin set comprises of t bins, only $\lceil \log_2 t \rceil$ bits are needed to encode the position of the set bit (if any). Thus, for $t = 10$, the CBA approach would require a mere 4 bits to encode each color. Revisiting the example above (where $n = 64$, $f = 2$ and $t = 10$), the CBA approach would require only 32 bytes when compared to to the 128 bytes required by the GCH and 256 bytes required by the CCV, a substantial savings of 75% over GCH and 87.5% over CCV in storage space.

However, it is not totally clear whether this approach will always yield better "compression" than, for instance, using run-length encoding on the full signature, as this is highly dependent on the number of consecutive 0's a signature would have. That would depend on the number of colors present in an image – which, as we shall see later, is usually low. In addition, the focus of this thesis is to investigate the feasibility of using the proposed signatures for effective CBIR. Even though the space savings feature is a major motivation, nevertheless, we know that at least the use of $\lceil \log_2 t \rceil$ per color is feasible and we shall use this as a lower bound for the savings in the space overhead.

For the sake of clarity, in the remainder of this section we shall display the full binary signatures for the images involved in the examples. The reader should be aware that such an explicit signature representation is not necessarily translated into the CBIR's storage space.

3.3 Image Retrieval using Binary Signatures

Once the signatures for each image in the data set is pre-computed and stored into the database, the retrieval procedure can take place. It is basically concerned with computing the associated similarity between the signatures of the user-specified query image and all the other images in the database. At the outset, we used the following measure to analyze the image similarity:

$$d_0(Q, I) \;=\; \textstyle\sum_{j=1}^{n} \mid (pos(B_Q^j) - pos(B_I^j) \mid$$

where $pos(B_R^k)$ gives the position of the set bit within (the set of bins) B^k of image R. For instance, using image A in Figure 9, we have $pos(B_A^1) = 2, pos(B_A^2) = 1$ and $pos(B_A^3) = 8$. Analytically, a color which is not present in an image has the position of the set bit as 0. This measure of similarity however is not robust, as illustrated in the following example.

Color/ Set of Bins	Color Density	Binary Signature									
		b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
Image X											
c_1/B_X^1	40%	0	0	0	1	0	0	0	0	0	0
c_2/B_X^2	30%	0	0	1	0	0	0	0	0	0	0
c_3/B_X^3	30%	0	0	1	0	0	0	0	0	0	0
Image Y											
c_1/B_Y^1	31%	0	0	0	1	0	0	0	0	0	0
c_2/B_Y^2	48%	0	0	0	0	1	0	0	0	0	0
c_3/B_Y^3	21%	0	0	1	0	0	0	0	0	0	0
Image Z											
c_1/B_Z^1	37%	0	0	0	1	0	0	0	0	0	0
c_2/B_Z^2	31%	0	0	0	1	0	0	0	0	0	0
c_3/B_Z^3	32%	0	0	0	1	0	0	0	0	0	0

Table 2: Signatures to exemplify the derivation of the similarity metric

Consider the signatures of images X, Y and Z as detailed in the Table 2. By simple inspection of the color densities (second column in the table) it is clear that images X and Z are more similar to each other than images X and Y. In fact, it seems reasonable to assume that a smaller difference in few colors is less perceptually critical than a larger difference in a single color. This is what is conceivable from the color distribution of the three images shown in Table 2. However, we have: $d_0(X, Y) \;=\; (4-4) \;+\; (5-3) \;+\; (3-3) \;=\; 2$, and $d_0(X, Z) \;=\; (4-4) \;+\; (4-3) \;+\; (4-3) \;=\; 2$, which suggests that both Y and Z are equally similar to X, thus contradicting our previous assumption.

Fortunately though, if we square the individual distances between the sets of bins, we can obtain a more robust description of the similarity metric. The new (and hereafter used) distance becomes:

$$d(Q, I) \;=\; \textstyle\sum_{j=1}^{n} [pos(B_Q^j) - pos(B_I^j)]^2$$

The reasoning behind the squaring is to further accentuate the distance between corresponding sets of bins. For instance, two corresponding bins whose set bits are two position apart, contributes more to the (dis)similarity metric than two occurrences of two corresponding bins whose set bits are only one position apart. Using this new definition of distance on the example discussed above, we have: $d(X,Y) = (4-4)^2 + (5-3)^2 + (3-3)^2 = 4$, and $d(X,Z) = (4-4)^2 + (4-3)^2 + (4-3)^2 = 2$, which reflects more closely our assumed perception of image similarity.

Finally, using the obtained similarity distances, the image set is then re-ordered with respect to their ascending distances (relative to the query image), and the top ranked ones are presented as the query's answer.

For the sake of illustration consider the images of Figure 9, and their respective signatures in Table 1. Using the similarity metric defined above yields: $d(A,B) = 2 < d(A,C) = d(A,D) = 3$, which matches our intuition. However, at this point it is worthwhile to single out the most important disadvantage of using only the color distribution as a basis to abstract an image. It is not unreasonable to argue that images C and D (Figure 9) are not similar. Even though they do have the same color distributions, the colors have very different *spatial* distribution, which is *not* captured when only the quantitative color density is used, in fact, $d(C,D) = 0$.

In such a case, superimposing a grid over the image, obtaining LCHs for each resulting cell and calculating the cumulative distance among such cells (as explained in [9]) would help to verify that both images are indeed not similar. To illustrate this, Figure 10 shows images C and D "exploded" so that the cells (assuming a 2×2 grid) are explicit (the arrows indicate how cells would be compared to each other). It should be obvious that the sum of the distances between signatures for corresponding cells is not null, therefore those images are color-wise but not spatially similar. It is important to note that the CBA approach and the similarity metric discussed above can be used within each cell as well. The issue of superimposing a grid over the image is however not addressed within the scope of this thesis, but is covered in [9].

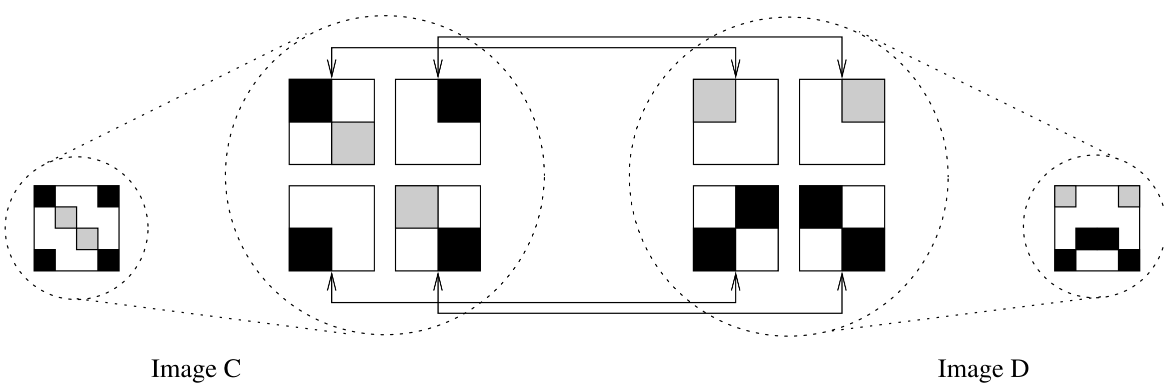


Figure 10: Images C and D from Figure 9 "exploded" in a 2×2 grid

We have observed that less dominant colors form a significant composition in an image. In fact, investigating 30,000 images used in our experiments, we verified that each image has, on average, 9.73 colors after being normalized to a 64-color representation using the RGB color model. In addition, most of such colors (7 on average) have a density of 10% or less. Hence, it is conceivable to conjecture that a substantial cover of an image is due to many colors which individually constitute a small portion of the whole image. This lead us to modify the signatures in order to emphasize colors with smaller densities. Towards that goal, the VBA scheme was developed, and it is discussed next.

Variable-Bin Allocation (VBA) is based on varying the capacities of the bins which represent the color densities. The design of VBA is based on the fact that distances due to less dominant colors are important for an efficient retrieval. The generated signature therefore lays a greater emphasis on the distance between the less dominant colors than on larger dominant colors. The experiments performed on VBA, assuming a color distribution discretized into $t = 10$ bins, were based on both b_1 and b_2 accommodating 3% of the total color density, b_3 accommodates 4%, and b_4 and b_5 accommodate 5% each. The remaining bins under consideration are based on a reasoning similar to the one used for the CBA approach, i.e. b_6 accommodates density representations from 21% to 30%, and b_7 accommodates the representations from 31% to 40%, and so on. Note that densities over 61% are all allocated to the same bin (b_{10}). This is due to the fact that an image rarely has more than half of itself covered by a single color. In other words, we enhance the granularity of color with smaller distributions, without completely ignoring those with a large presence. Table 3 depicts the above classification, and also lists the binary signatures for each possible color distribution under the VBA approach. In principle, both the discretized bin

Color Density		Binary Signature									
From	To	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}
01	03	1	0	0	0	0	0	0	0	0	0
03	06	0	1	0	0	0	0	0	0	0	0
07	10	0	0	1	0	0	0	0	0	0	0
11	15	0	0	0	1	0	0	0	0	0	0
16	20	0	0	0	0	1	0	0	0	0	0
21	30	0	0	0	0	0	1	0	0	0	0
31	40	0	0	0	0	0	0	1	0	0	0
41	50	0	0	0	0	0	0	0	1	0	0
51	60	0	0	0	0	0	0	0	0	1	0
61	100	0	0	0	0	0	0	0	0	0	1

Table 3: Classification of bins in a VBA

number and color density can be used to generate a wide range of binary signatures under the VBA approach. These possibilities are covered later under the discussion of query performance in Section 4.

As before, a color which is not present in an image will imply a set of all null-set bins. Assuming that two sets of bins are being compared, and if one of them has all null-set bins then the associated distance would be the position of the set bin from the other set of bins. Similarly, if both the sets have all null-set bins then the associated distance is null.

Similar to the previous example where we determined the signatures of images A,B and C using a CBA, the signature of these images using VBA are shown in Table 4. Now, we have $d(A, B) = 1 < d(A, C) = d(A, D) = 5$, which matches our intuition even better, i.e., the use of VBA further increased the distance between image A and both C and D (which are still deemed identical). In addition, note that when using CBA we would have had $d(A, B) = 2 > d(B, C) = 1$ but VBA yields $d(A, B) = 1 < d(B, C) = 4$, reflecting the fact that the same quantitative change (one single cell) within a color which is less dense is more important than the same quantitative change within a more predominant color. This indicates that by using VBA, the distance between dissimilar images is likely to be significantly increased. Our belief that this scheme would enable a more efficient retrieval than CBA was confirmed in the experiments.

3.4 Implementation Issues

The proper assessment of an image retrieval system requires a good understanding of the user needs. Nonetheless, an image abstraction and retrieval methodology in hand requires a properly structured and systematic plan for efficient retrieval. Such implementation issues are not only specific to their desired outcome, but also suggest the need for supplementary analysis that support implementation. This sub-section briefly describes some of the general implementation-specific details and their semantic outcomes. They have been born during the conceptual design of the proposed image abstraction and retrieval methodology.

Color Set of Bins	Color Density	Binary Signature									
		b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
Image A											
c_1/B_A^1	18%	0	0	0	0	1	0	0	0	0	0
c_2/B_A^2	6%	0	1	0	0	0	0	0	0	0	0
c_3/B_A^3	76%	0	0	0	0	0	0	0	0	0	1
Image B											
c_1/B_B^1	24%	0	0	0	0	0	1	0	0	0	0
c_2/B_B^2	6%	0	1	0	0	0	0	0	0	0	0
c_3/B_B^3	70%	0	0	0	0	0	0	0	0	0	1
Image C											
c_1/B_C^1	24%	0	0	0	0	0	1	0	0	0	0
c_2/B_C^2	12%	0	0	0	1	0	0	0	0	0	0
c_3/B_C^3	64%	0	0	0	0	0	0	0	0	0	1
Image D											
c_1/B_D^1	24%	0	0	0	0	0	1	0	0	0	0
c_2/B_D^2	12%	0	0	0	1	0	0	0	0	0	0
c_3/B_D^3	64%	0	0	0	0	0	0	0	0	0	1

Table 4: Detailed Signatures using VBA

3.4.1 Color Normalization

Color normalization or quantization refers to the irreversible transformation process of converting a true-color image with 256^3 possible color values into an image comprising of only a few representative colors. The objective of quantization is therefore to minimize the perceived difference between the original and the normalized images in an attempt to compromise between storage space and perceptual similarity. The transformation process can be visualized as dividing the RGB color cube (although other color space models from sub-section 2.2 can also be used) into a number of smaller sub-cubes such that each color in the RGB color space is mapped to a particular sub-cube, and is referred to as irreversible because it is impossible to regenerate the original cube. Heckbert [24] pioneered an adaptive approach to color normalization based on histogram statistics, and since then various authors have proposed different techniques for color normalization [45]. For the purpose of this study, we use uniform color quantization method to reduce the possible number of colors in an image due to the low computational load. The process of uniform color quantization is based on dividing the RGB unit cube into smaller equal-sized sub-cubes.

3.4.2 Bin Allocation

As already discussed, the bin-size plays an important role in the retrieval efficacy of a system and is primarily dependent on the color distribution of similar images that must be ideally retrieved. In a scenario, where the similar images in a database bear close recognition to each other then one may wish to discretize the color distribution into larger sizes ($e.g.$ $t = 10$) so that the retrieval system can easily retrieve them, while still saving significant storage space. However, if one wishes to obtain a very efficient retrieval, then the possibility of discretizing the color into bins of smaller sizes needs to be explored ($e.g.$ $t = 20$). A larger bin size would therefore mean that we lay a larger emphasis on the difference between the color distributions than the one with smaller bin sizes. However it is not evidently clear that bins with larger sizes would always yield better retrieval effectiveness than bins with smaller sizes, for instance using $t = 20$ bins, we would tend to discriminate between

5% and 6% of a color across two images, even though they are very similar. A more detailed discussion on this issue and subsequent experimental results are covered in the next sub-section.

3.4.3 Length of the Encoded Signature

The proposed image abstraction techniques emphasizes on less dominant colors in an image, rather than the most dominant colors as seen in the design of VBA where the bin-sizes are smaller for colors with smaller compositions, and colors with densities above 61% are allocated to the last bin. In this study, we also explore the possibility of completely eliminating from the binary signatures, the colors with larger compositions. The experimental results, discussed in the next sub-section support our assumption that the inclusion of less-dominant colors significantly improve the retrieval effectiveness than those with larger compositions and thus colors with larger compositions may also be completely ignored. Such a design would further expedite the saving in the storage space.

3.5 Conclusion

This sub-section described the design and implementation of a image retrieval system that utilizes binary signatures for image retrieval. In general, the signatures are an abstract representation of the information contained in a GCH. The proposed methodologies result in a significant saving of storage space due to the bit-wise representation of the color content.

4 Experiments

4.1 Experimental Setup

The experiments were performed on a large heterogeneous database containing 50,000 images (obtained from two commercially available collections: *PrintArtist Platinum* by Sierra Home, and *Master Photos 50,000 - Premium Photo Collection* by COREL) with 15 images, used as queries. The query images, as well as a pre-defined set of similar images were obtained from another collection of images (*Corel Gallery Magic 65,000*) which diminishes the probability of biasing the results obtained in our experiments. The images in each of these subsets resemble each other based on the color distribution and also on the image semantics (all subsets can be seen at: `http://www.cs.ualberta.ca/~mn/CBIRdataset/`). These images subsets were then mixed with the original datasets to compose the database to be queried.

The color constituents were normalized to 27, 64 and 125 colors from the 256^3 possible color values in the RGB color space. To normalize an image into 64 colors, each of the R, G and B channels is quantized into 4 equal intervals thereby resulting in a 64 color representation. Similarly, to obtain a 27 and 125 color representation, we would have to quantize the R, G and B channels into equal intervals of length 3 and 5 respectively.

Apart from varying the normalized number of colors in an image, the length of the binary signature bit-string and the bin-size were also varied for both the CBA and the VBA. Results obtained using GCH and CCV were used to serve as a benchmark.

4.2 Evaluation of Query Results

The evaluation of a retrieval system can be performed in many different ways, depending on the ranking of the relevant image set, the effectiveness of indexing, and the operational efficiency, among other factors. The operational efficiency of a system is often measured in terms of computer resources such as processing time, and is a vital component in conjunction to effectiveness for evaluating a retrieval system [63]. However, it should be noted that obtaining a single measure using the various parameters for evaluating a retrieval system has proven

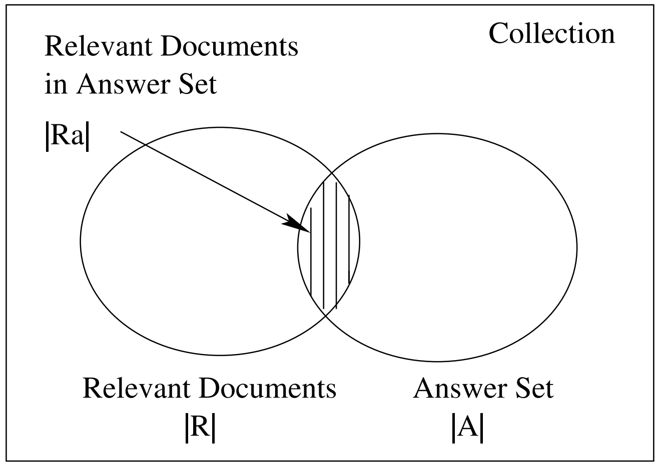


Figure 11: Recall and Precision for a sample answer set (adapted from [69])

to be extremely difficult [41]. We would like to evaluate the system in a comparative way to measure how much do certain changes, such as varying the normalized number of colors in an image or the length of the binary signature bit-string lead to an improvement in performance.

The effectiveness of an information retrieval system is a measure of the ability of the system to satisfy the user in terms of the relevance of the retrieved documents [63], and is often measured using *recall* and *precision* [6, 69] where, recall measures the ability of a system to retrieve the relevant documents, and conversely precision is a measure of the ability to reject the irrelevant ones. The following sub-section contains a more detailed explanation of recall and precision.

4.2.1 Recall and Precision

The most common way to evaluate the performance of a retrieval system is to calculate the number of relevant documents that have been retrieved, and their relative ordering among all the documents. From a user's perspective, a retrieval system should retrieve as many relevant documents as it can, with minimum number of non-relevant ones. Roughly speaking, the former corresponds to the concept of recall, while the latter pertains to the notion of precision [41].

The *Recall* of an information retrieval system refers to the percentage of the total relevant documents retrieved [6, 67], and is mathematically defined as the total number of retrieved relevant documents from all the relevant documents. Figure 11 shows a graphical representation for a sample answer set, where $|R_a|$ denotes a subset of $|R|$ relevant documents, that appear in the retrieved document list $|A|$ [69]. The value of recall for such a representation is given by the function:

$$\mathrm{Recall} = \frac{\text{Number of relevant documents that are retrieved}}{\text{Total number of relevant documents}} = \frac{|\mathrm{R_a}|}{|\mathrm{R}|}$$

For example, if the database comprises of 100 relevant documents for a query, and the search procedure was able to retrieve only 10 of these relevant documents, then the recall of the system for this particular query would be 10%.

Conversely, *precision* is a measure of the information retrieval system to retrieve only the relevant documents [6]. Mathematically, precision P_r is the fraction of the relevant documents for a query document amongst all the r retrieved documents [67]. The value of precision for the Figure 11 would then be given by the following

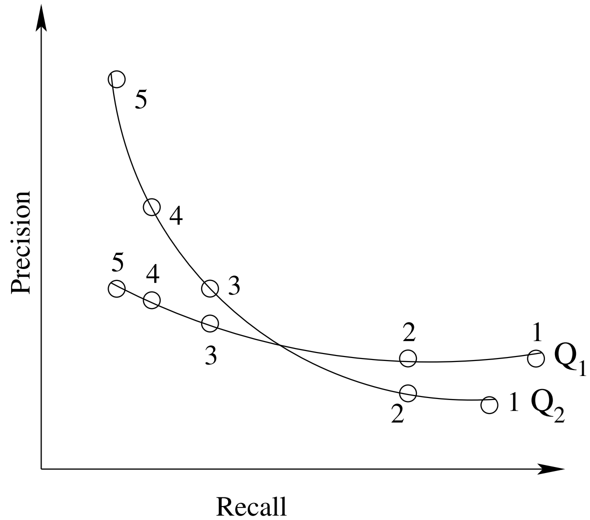


Figure 12: The recall and precision curves for two queries (adapted from [63])

function [69]:

$$\text{Precision} = \frac{\text{Number of relevant documents that are retrieved}}{\text{Total number of retrieved documents}} = \frac{|\mathrm{R_a}|}{|\mathrm{A}|}$$

For example, if the number of retrieved documents is 100 and 20 of these are relevant, then the precision of the system for this particular query would be 20%. Both recall and precision have values in the interval [0,1]. The desired values of both recall and precision however must be as close to 1 as possible for an efficient retrieval system [29]. Geometrically, a set of ordered recall and precision values can be joined to make up a *recall-precision curve*. The recall and precision for a query set are inversely related to each other, and generate a curve similar to the ones shown in the Figure 12. The performance of a retrieval system can be measured by generating an average curve for all the queries [63]. In general, a curve with maximum value of recall and precision that is, a curve closest to the upper right-hand corner indicates the best performance.

In a non-experimental database, it is usually impossible to distinguish between the relevant and non-relevant documents. Therefore to measure recall, we require prior knowledge of the total number of relevant documents in the collection with respect to each query. Both these measures of effectiveness are based on the ordering of the documents in response to a user query. A higher ranking query reflects a closer similarity match with the query image. However, a problem arises if the retrieval system generates a weak ordering of the documents in the output, i.e., two or more documents are ranked the same in response to a user query [41]. This implies that even though the images are equally similar to the query image, there is still a relative difference between their ordering. Various other authors have proposed different methodologies in order to overcome the above limitation. As already discussed in the previous sub-section, we square the individual distances between the set of bins to address this issue. Another problem associated with this measure of retrieval is the need to equate the database collection size with query relevance properties. In other words, the recall and precision values for a query item would proportionally detoriate with a considerable increase in the document collection size. This would therefore lead us to an incorrect evaluation of the retrieval system. To deal with the above limitation, the precision and recall values of the first few images in the retrieved set are examined.

4.2.2 Retrieval Effectiveness

The usefulness of precision and recall measures has been challenged though. It is a "double" measure which makes it harder to compare the results of two different approaches, as, for instance, one particular user may be more interested in precision whereas another may be only concerned with recall. In fact, Su suggests that neither precision nor recall are highly significant in evaluating a retrieval system [60].

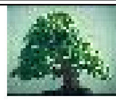
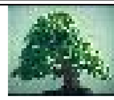
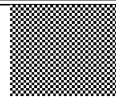
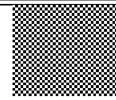
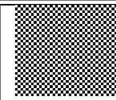
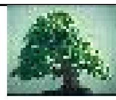
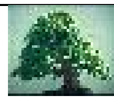
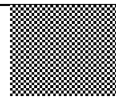
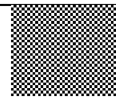
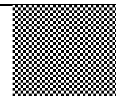
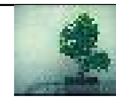
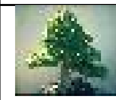
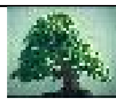
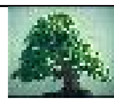
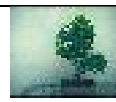
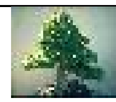
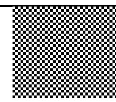
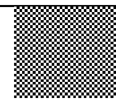
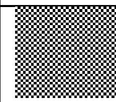
System / Avg. Rank	Query Image	Answer Set					
		1	2	3	4	5	6
A / 3							
B / 4							
Desired / 2							

Table 5: Comparison of two retrieval systems on the basis of retrieval effectiveness

In order to address this, we have also used a methodology closely related to the one presented in [11] to evaluate QBIC's performance. The chosen metric is based on the ratio of the average ranking of the similar images (determined *a priori*) to the ideal ranking of these images in the result set:

$$\text{Retrieval Effectiveness} = \frac{\text{Average ranking of all images determined similar}}{\text{Ideal average ranking of all images determined similar}}$$

As an example, consider an image set comprising of ten images including two images that closely resemble a given query image. Now, assume that methodology A places the desired images from the subset at positions 1, 3 and 5, thus yielding an average rank of 3; and say that the methodology B places them at position 1, 5 and 6, resulting in an average rank of 4 (see Table 5). Knowing that the ideal average rank would be 2 – the desired images would be the first three that appears in the answer set – the retrieval effectiveness of methodology A is 1.5, and that of methodology B equals 2. Thus, methodology A is more effective than methodology B, since the desired results are ranked better in the answer set. The smaller the obtained value, the better the effectiveness. More formally, for a given query image, we have:

$$\text{Retrieval Effectiveness} = \frac{2\ \sum_{i=1}^{s}\ s_i}{(s\ (s+1))}$$

where, s is the number of images deemed beforehand, similar to the query image, and s_i represents the ranks of such images in the result set. In practice, this ratio attempts to measure how quickly (in relation to the expected answer set size) one is able to find all relevant images. Note that this measure is normalized with respect to the size of the answer sets, thus one can use an average performance as a representative measurement.

4.3 Experimental Results and Analysis

The experiments were performed on a large heterogeneous database of up to 50,000 JPEG images with 15 images, used as queries. The images are deemed heterogeneous based on not only the visual attributes but also the image dimensionality. The set of the 50,000 images were merged from two commercially available collections: *PrintArtist Platinum* (nearly 20,000 images) by Sierra Home, and *Master Photos 50,000 - Premium*

Photo Collection (about 30,000 images) by COREL. Furthermore, the query images and their similarly subsets were obtained from another image collection: *Corel Gallery Magic 65,000*, and were grouped into 15 different subsets, each subset containing between 6-24 similar images, that resemble each other based on the color distribution and also on the image semantics. For instance, there is a set of distinct Bonzai trees, "Halloween" pumpkins, the Queen's red army, and so forth. The appendix lists four such similarity subsets (all subsets can be seen at: `http://www.cs.ualberta.ca/~mn/CBIRdataset/`). While it is quite possible that different researchers may come up with different categorizations, our experiments indicate that the actual classification will have little effect on the overall performance of the system. Furthermore, to the best of our knowledge, there is no standard benchmark collection for evaluating the effectiveness of a CBIR system, thus the ad-hoc nature of our evaluation method. The database was further categorized into five lots from 10,000 to 50,000 random images with increments of 10,000 such that each lot comprises of all the query images and the similar subsets. To reduce the storage overhead, while still preserving the aspect ratio and the overall quality of the image, both the image height and the width were reduced to 70% of their original size.

The initial stages in computing the signature in the proposed technique are similar to the computation of a GCH and the CCV. In fact, both metadata were generated and used for retrieval, serving as a benchmark. Although researchers may dispute the suitability of such a comparison, we contend that the three approaches are related because they are all histogram based, describe color features and share the same similarity metrics. As already discussed in Section 2, the CCV technique allows for a finer distinction that color histograms by classifying each pixel value as either coherent or incoherent, depending upon whether the pixel is a constituent of a large similarly-colored region [38]. On the contrary, the GCH merely describes the colors which are present in the image, and their quantities. We can therefore, construct the GCH for an image by simply adding the coherent and incoherent values for each discretized color. The proposed technique on the other hand is an abstract representation obtained by discretizing the colorspace of a GCH.

Corresponding to the technique proposed by the authors [38] for computing the CCVs, each pixel value in an image is first blurred with the average of its 8 adjacent neighbors. Next, for each of the categorized lots of 10,000 to 50,000 images, the color constituents were normalized to 27, 64 and 125 colors from the 256^3 possible color values in the RGB color space. In our experiments, the 64-color RGB color space demonstrated the best results[5], and is used as a benchmark. We also performed numerous analysis experiments by varying the length of the binary signature bit-string for both the CBA and the VBA. This was done to evaluate how certain changes such as varying the normalized number of colors in an image, the length of the binary signature bit-string etc. will lead to an improvement in performance. For the sake of illustration, we emphasize primarily on the intermediate lot of 30,000 images to exemplify the experimental results and analysis. We however present the experimental results for each of the five lots to demonstrate that the proposed technique outperforms the conventional approaches and also the usage of CCVs for multiple heterogeneous databases.

We have compared our results with the GCH and CCV based on the query performance and the storage space. Our first experiment investigated retrieval performance comparisons based on recall and precision, and retrieval effectiveness. To compare the approaches fairly, we bring them to a common ground by using the average recall-precision curve, average retrieval effectiveness for all the queries. For each document collection, the retrieval results based on the retrieval effectiveness, recall and precision are obtained. Each of the obtained values are then averaged for an overall performance comparison. Next, we evaluate the retrieval results by varying the normalized number of colors in an image, bin-sizes, and the length of the binary signature bit-string to analyze their effect on the performance. We also compare the SR-tree performance when indexing the GCH against VBA with sequential scan, since SR-trees are arguably the most efficient access method for multi-dimensional feature space [35]. As such, we did not use the SR-tree/CCV combination because the expected size of the index structure would then be twice the size of the SR-tree/GCH structure for the same dataset [35]. Furthermore, a significant goal of this research is to reduce the space overhead for image retrieval. Recall that the proposed

[5] Further discussed in sub-section 4.3.2

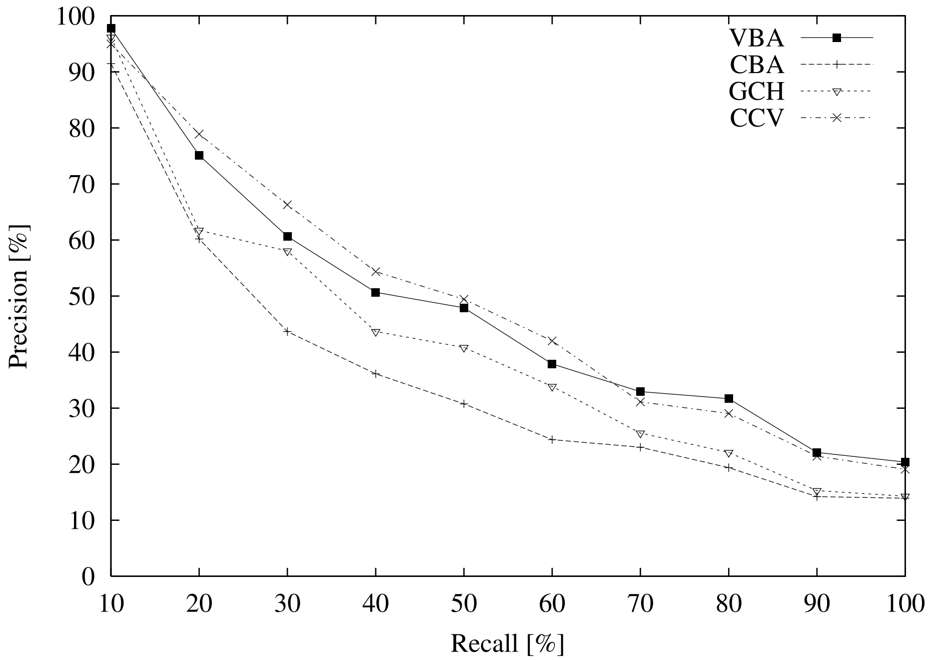


Figure 13: Recall and Precision for 10,000 images

approach saves 75% of storage space when compared to GCH, and 87.5% for CCV. This discussion is further extended in the subsequent sub-sections.

4.3.1 Query Effectiveness

Experiments were performed using the five datasets mentioned in the previous sub-section. We run queries for each of the datasets and compute the average recall-precision curves and retrieval effectiveness for an overall performance comparison. In order to compute the above measures, we rank the entire list of retrieved documents until all the similar images for a query are part of the answer set. The recall-precision curves for the five datasets are shown (Figures 13 to 17), while the success rate (%) for the various techniques are summarized in the Table 6.

The experimental results show that VBA performs better than both CCV and GCH for most query images. This relative advantage was observed using both the classical precision-recall curves, and even more visible using the measure of retrieval effectiveness. For smaller values of recall (between 20% to 40%), CCV slightly outperforms VBA, though the curve drops down below VBA shortly afterwards. Although, CBA does not demonstrate a better recall-precision curve than both CCV and GCH, the ordering of the retrieved documents in CBA is better than GCHs. The retrieval effectiveness measures shown in the Figure 18 indicate that the relative ordering of the retrieved document set is the best for VBA, followed by CCV, CBA and GCH (in order). The figure also indicates that the linear curves tend to move apart with increase in the dataset size, suggesting that the relative ordering observed between the various techniques is preserved, although further values were not computed. We also analyzed the success rate for each of the investigated techniques, such that for a given query image, which of them performs better in terms of the retrieval effectiveness. Table 6 indicates that, the probability that VBA performs better than other investigated techniques is higher. The results also indicate that CBA outperforms the GCH in terms of the success rate by more than twice.

Consider now the case where most of the images relevant to a given query are ranked high, and very few of them are ranked low. Then the performance measures discussed above, would likely not lead us to a correct evaluation of the retrieval system. In order to overcome the above limitation, we examine the performance

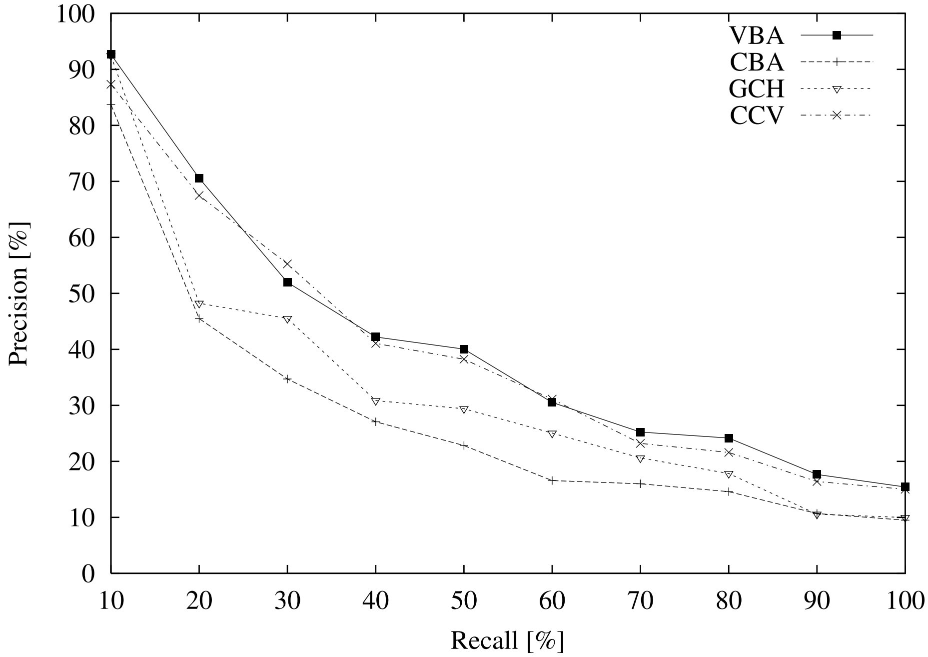


Figure 14: Recall and Precision for 20,000 images

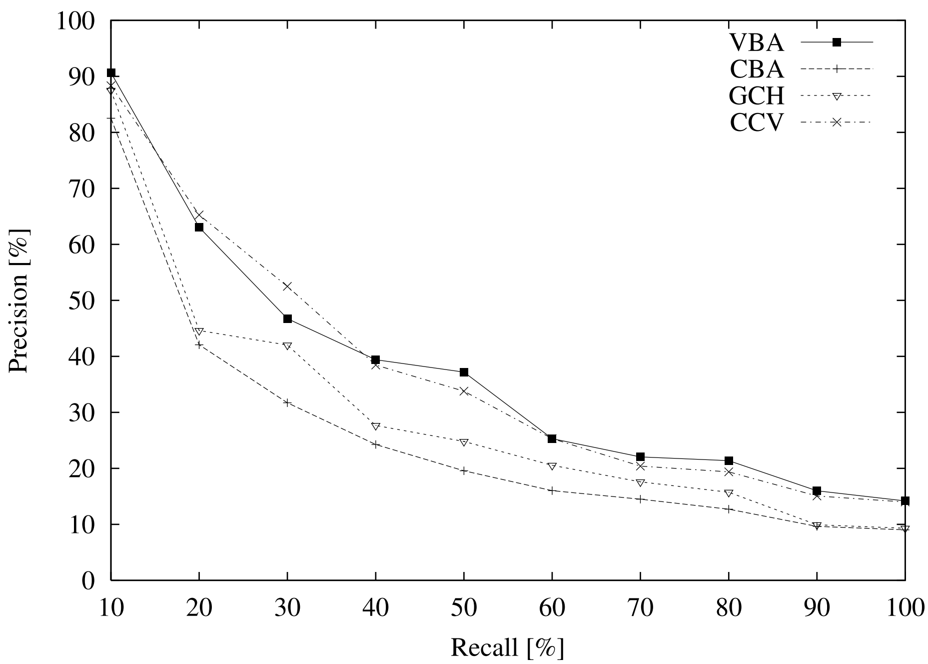


Figure 15: Recall and Precision for 30,000 images

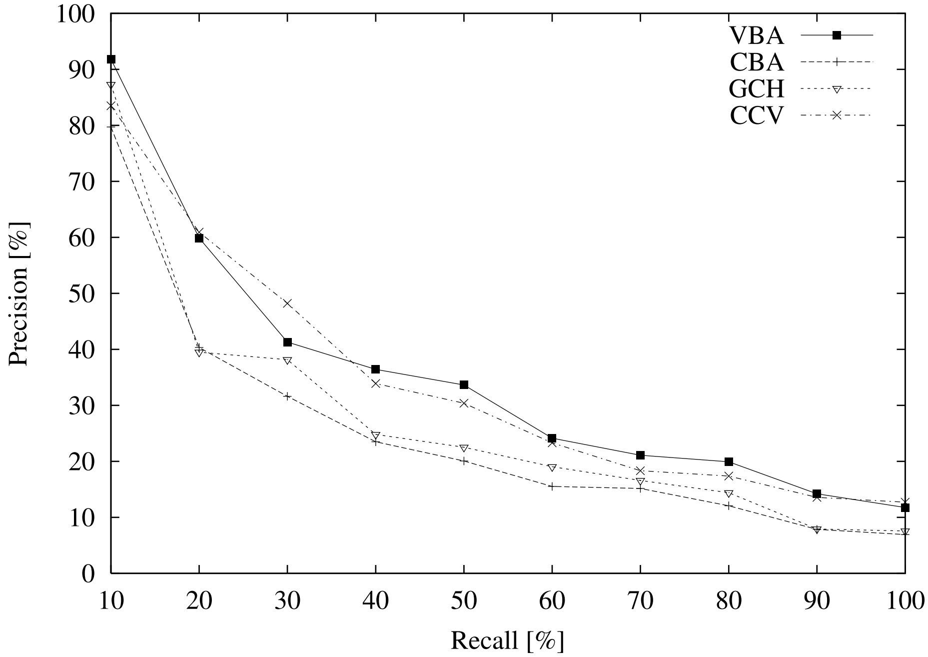


Figure 16: Recall and Precision for 40,000 images

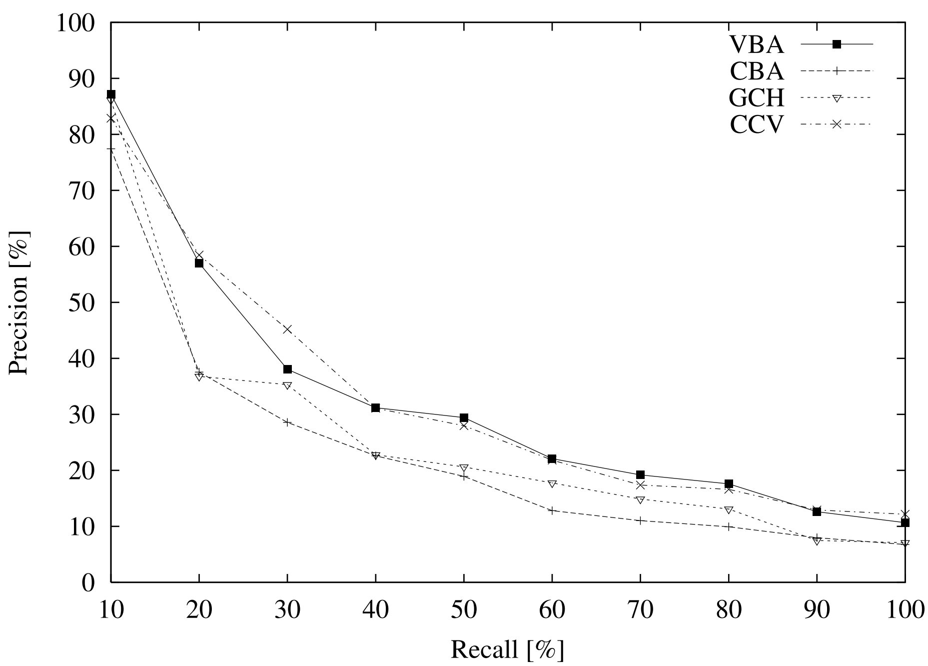


Figure 17: Recall and Precision for 50,000 images

Dataset Size	10,000	20,000	30,000	40,000	50,000
Percentage - Success Rate					
VBA	53.3%	53.3%	53.3%	46.6%	53.3%
CCV	26.6%	20.0%	20.0%	26.6%	26.6%
CBA	13.3%	20.0%	20.0%	20.0%	13.3%
GCH	6.6%	6.6%	6.6%	6.6%	13.3%

Table 6: Summary of the success rate using a 64-color RGB color space

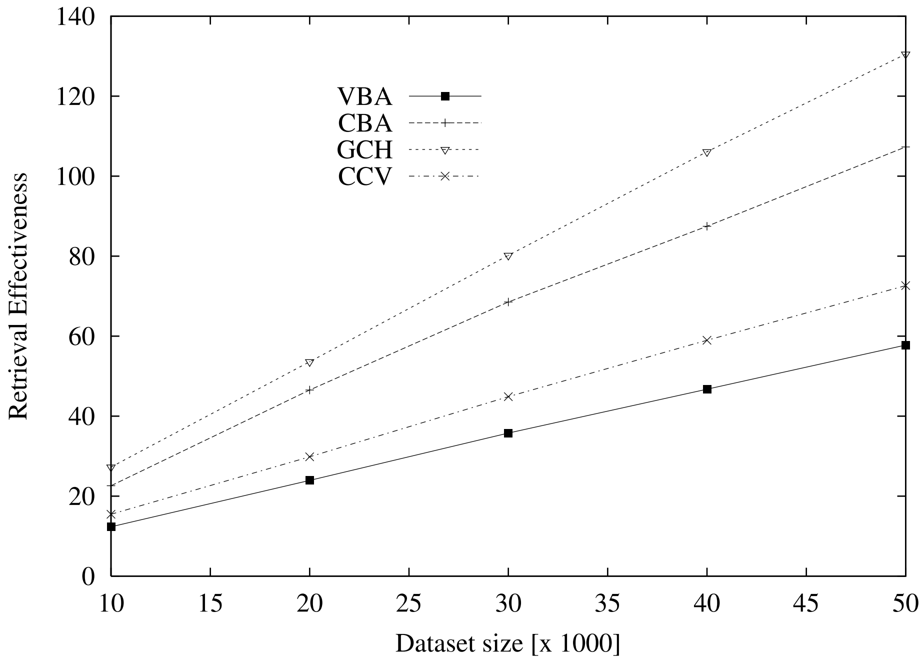


Figure 18: Retrieval Effectiveness using a 64-color RGB color space

measures at various ceilings for the retrieved documents. There exists no general agreement to determine the cut-off point of the retrieved answer set for analyzing the performance of a retrieval system. To analyze this effect, we compute the performance measures for 30,000 images at 80% of the required answer set (as done by [58]). The result of this analysis are summarized in Figure 19 for the precision vs. recall curve, while the average retrieval effectiveness and percentage of success rate for the datasets is shown in the Table 7. The above study is based on the assumption that only the first 80% of the similar images are part of the dataset and this attempts to eliminate the effect of the trailing images from the similarity subset. The experimental results indicate that VBA closely competes with CCVs in terms of the recall and precision. The probability that either of these techniques would perform better than the other is similar, even though VBA outperforms CCVs in terms of the overall desired ranking of the retrieved similar images (average retrieval effectiveness) by approximately 20%.

Since it is a well-known belief that the user is primarily interested in the first few retrieved documents and very rarely examines the entire set, we also analyze the performance measures for the first 100 images. This differs from the previously stated experiments in the sense that we now focus on only the first few retrieved documents. The assumption that a fixed number of relevant images amongst the first 100 retrieved documents are part of the dataset is invalid in the current scenario, because different techniques would then obtain entirely different number of similar images in the respective retrieved answer sets. As an example, it might happen

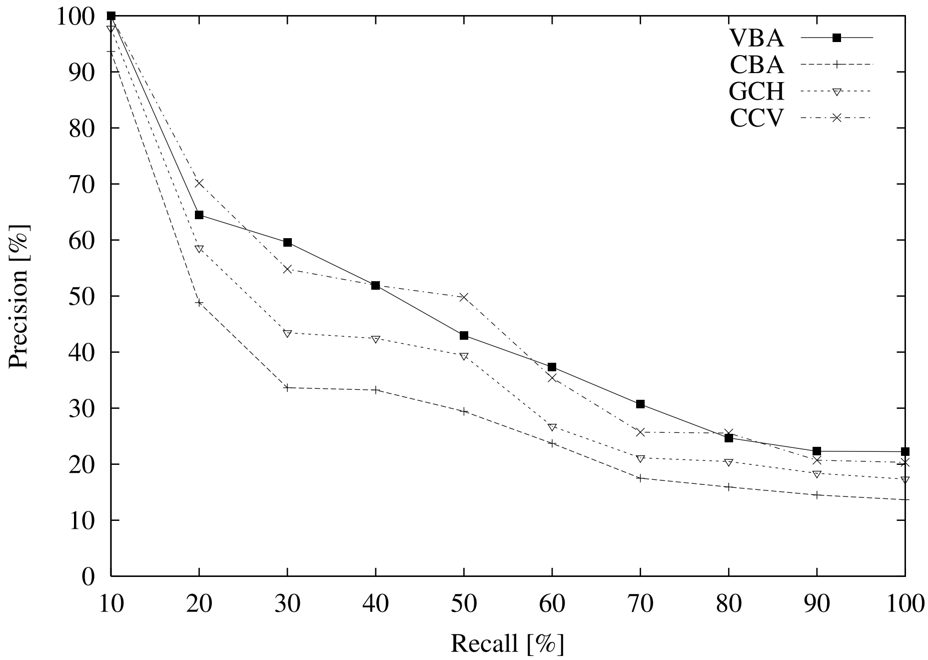


Figure 19: Recall and Precision for 30K images computed using first 80% of the deemed similar set

Technique	Retrieval Effectiveness	Success Rate
VBA	18.96	46.6%
CCV	23.51	46.6%
CBA	42.66	06.6%
GCH	42.19	06.6%

Table 7: Summary of the success rate for 30K images computed using first 80% of the deemed similar set

a large number of similar images are part of the retrieved documents set for one technique, and very few of them occur in the other set. The results of this analysis are summarized in Figure 20 for the precision vs. recall curve. The result of retrieval effectiveness are not stated for the current scenario because the definition of retrieval effectiveness considerably changes under the present assumption (as explained above). The experiments indicate that CCV closely competes with VBA in terms of the recall and precision. The ordering of the retrieved document sets is higher in VBA, although the percentage of queries for which either approach delivered the best performance is the same. The above experiments also indicate that GCH produces a better precision than CBA for a restricted retrieval set, but delivers a weak ordering in terms of the overall desired ranking of the retrieved similar images. For the sake of illustration, four sample queries and the respective first ten returned images for the most representatives approaches investigated in this thesis are shown in the Appendix.

4.3.2 Miscellaneous Analysis on Query Effectiveness

Experiments were performed to investigate how the retrieval measures respond to variations in the normalized number of colors. To investigate this, we normalize the color representation to 27 and 125 colors from the 256^3 possible color values in the RGB color space. The result of this analysis for 30,000 images is summarized in Figure 21 and Figure 22 for the recall-precision curves, while the summary of success rate is shown in the Table 8 i.e., the percentage of queries for which each approach delivered the best performance. Figure 23 illustrates

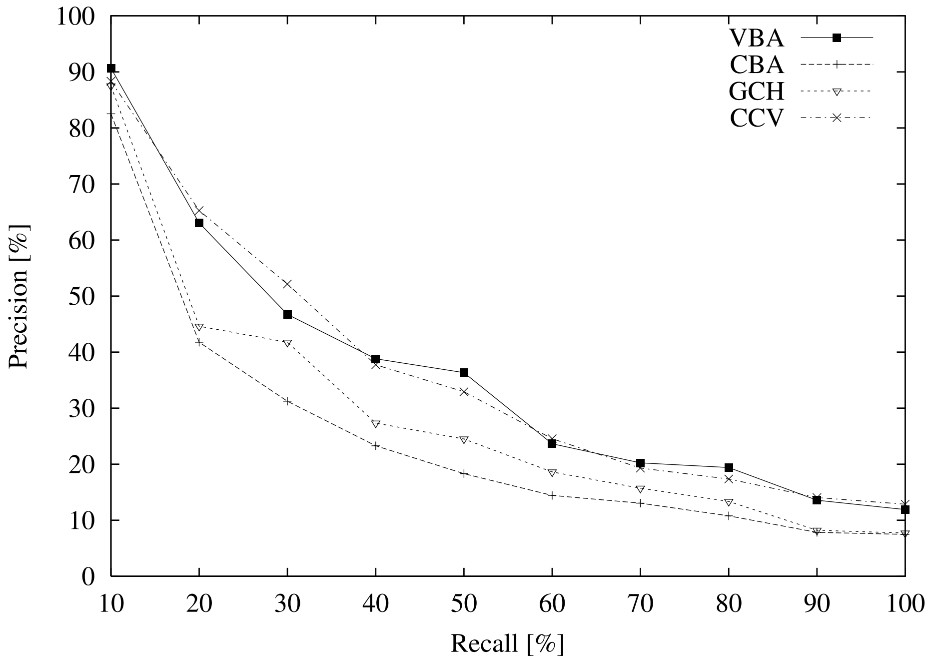


Figure 20: Recall and Precision for 30K images computed using the first 100 retrieved answer set

the summary of the retrieval effectiveness performed by varying the color quantization. In our experiments, the 64-color RGB color space demonstrated the best results, and is used as a benchmark. Various other authors also commonly use the 64-color RGB color space as a benchmark in their studies [38, 58]. The experimental results show that the CCVs demonstrate more efficient retrieval than variable-bin allocation for more color values, namely 125 colors. This relative advantage was observed using both the average precision vs. recall curves and retrieval effectiveness. Conversely, VBA delivers higher precision (especially in the mid-range of recall values) and retrieval effectiveness for fewer color values. Both VBA and CCVs illustrate better precision vs. recall curves than CBA and GCH in all color normalizations. Note, that by using more color values, one would obtain a larger color representation for an image. Therefore, we need to draw a line between the usage of space and quantization of the color values.

Color Quantization	27	64	125
Percentage - Success Rate			
VBA	53.3%	53.3%	26.6%
CCV	26.6%	20.0%	40.0%
CBA	13.3%	20.0%	26.6%
GCH	6.6%	6.6%	6.6%

Table 8: Summary of the success rate by varying the color quantization

Other experiments were performed for analyzing the retrieval measures by varying the size of the signature bit-string in the proposed methodologies. The first experiment alters the image signature length by varying the capacity of the bin-size. As an example, instead of discretizing the color distribution of an image into $t = 10$ bins of equal capacities for CBA we explored the possibility of discretizing the color into $t = 5$ and $t = 20$ bins. The analysis of these experiments are summarized in the Table 9, and the experimental results indicates that the current approach of discretizing the color distribution of an image into $t = 10$ bins of equal capacities for CBA

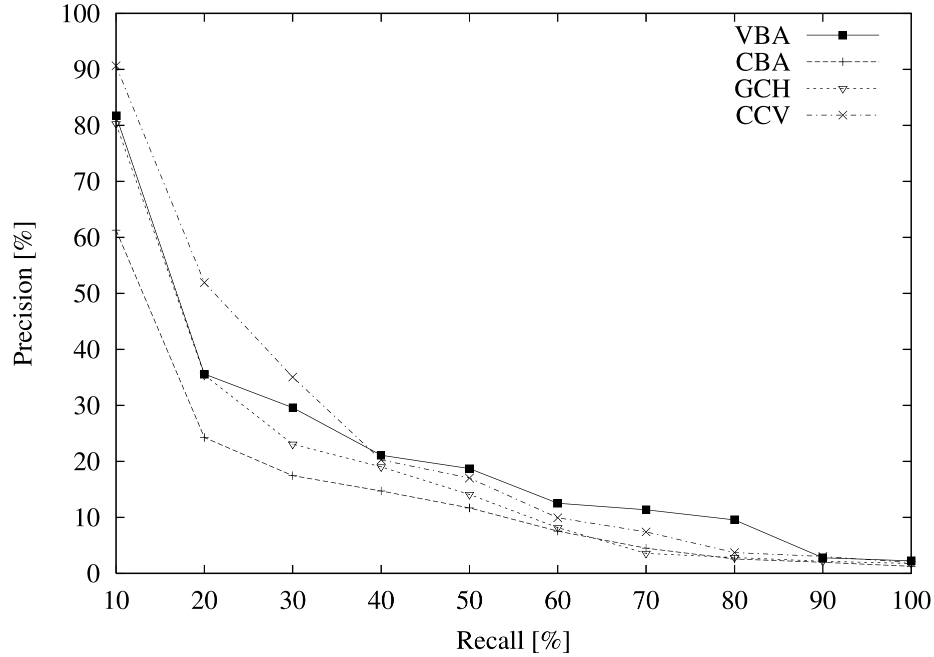


Figure 21: Recall and Precision for 30K images with normalization to 27 colors

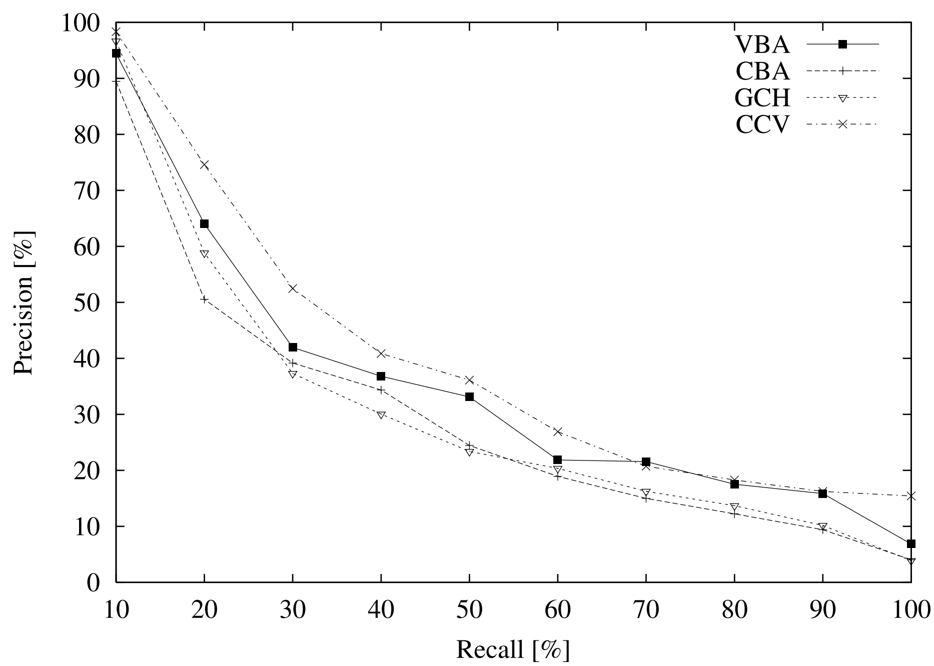


Figure 22: Recall and Precision for 30K images with normalization to 125 colors

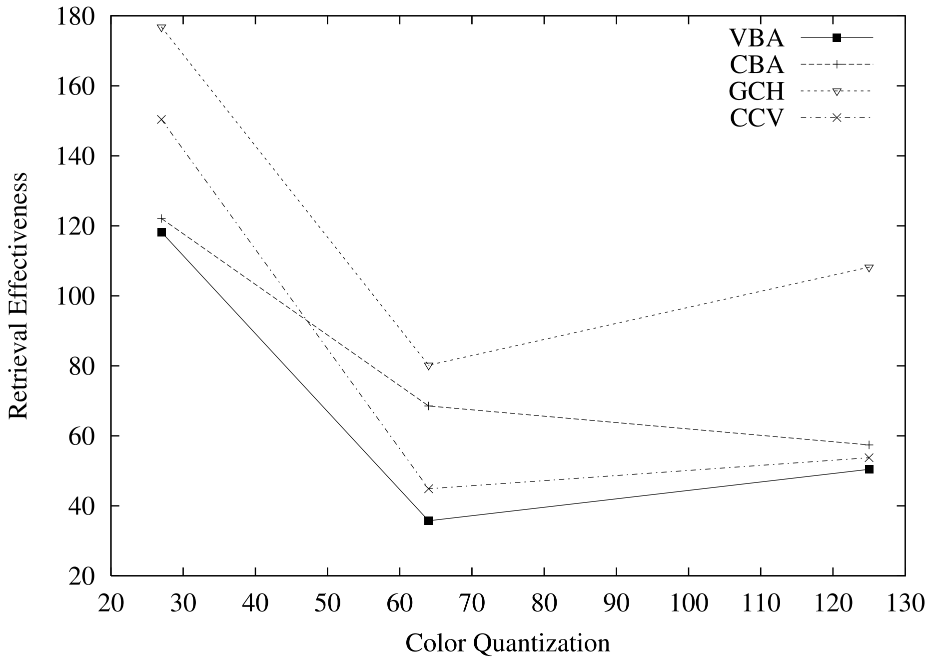


Figure 23: Summary of the *retrieval effectiveness* by varying the color quantization

Bin size of CBA	Retrieval Effectiveness
05	130.72
10	68.53
20	70.50

Table 9: Summary of the experimental results on retrieval effectiveness based on varying the capacity of the bin-size

is more adequate than any other distributions based on both the storage space and effectiveness.

We also performed experiments on 30,000 images by ignoring the trailing bins from the color sub-string. Our motivation then was to demonstrate the reduction in space overhead by taking advantage of the fact that a substantial cover of an image is due to the many colors which individually constitute a small portion of the whole image. Table 10 lists the retrieval effectiveness of these experiments for analyzing the overall performance. Clearly, VBA performs best in the current scenario ($length = 10$). The experiments performed on 30,000 images using a CBA, by truncating the trailing bins from the color sub-string demonstrate that the retrieval effectiveness for $length = 6$, excels the retrieval effectiveness of other values. This would mean that colors above 60% are completely ignored, and supports our assumption that the inclusion of less-dominant colors significantly improves the retrieval effectiveness than those with larger compositions.

Finally, note that with VBA we obtain approximately 55% better retrieval effectiveness than using GCH, while still saving 75% of storage space. It also outperforms the use of CCV by 20% in terms of retrieval effectiveness while still saving 87.5% of storage space. The CBA approach using 6 bins, on the other hand requires a mere 38% of the storage space required by the GCH, while still generating about 40% better retrieval effectiveness.

Substring Length (after truncation)	VBA	CBA
05	53.4	58.6
06	47.4	57.1
07	43.4	57.5
08	38.9	64.1
09	36.3	68.9
10	35.7	68.5

Table 10: Summary of the experimental results on retrieval effectiveness based on truncating the color sub-string

4.3.3 GCH/SR-Tree versus VBA with Linear Scan

We have seen in design of the proposed model (Section 3) that the retrieval procedure for analyzing the image similarity comprises of sequentially comparing the feature vector of a given query image, with those of all the other images in the database. Even though this is feasible for small to medium size image sets, it is arguable whether this would be acceptable for very large databases. At this point the need for access structures (indices) arises. We therefore compare the performance of the sequential search when indexing the VBA signatures against the SR-tree indexing of the GCHs, since SR-trees are arguably the most efficient access methods for multi-dimensional indexing [35]. The implementation libraries of the SR-tree[6] were provided to us by N. Katayama and S. Satoh.

An SR-tree [28] derives its structure from a combination of SS-tree [15, 66] and an R^*-tree [5, 49]. The proposed index structure refines the disjointness among regions by partitioning them, and in doing so they demonstrate that an SR-tree outperforms both the basic building index structures. A region in an SR-tree is identified by an intersection of the bounding spheres and rectangles of the underlying points, as illustrated in the Figure 24. Figure 25 demonstrates an SR-tree formed from Figure 24. Each non-leaf node of an SR-tree comprises of a pointer to the child node, the number of points, a bounding sphere and a bounding rectangle. The leaf nodes comprise of a point and an attribute data. The insertion algorithm for an SR-tree derives itself from an SS-tree, but requires that the entries for both the bounding spheres and rectangles be updated for a new insertion, as compared to updating only the bounding spheres in case of an SS-tree. The retrieval algorithm for an SR-tree is based on traversing the tree as an order of the distance function, from each child node. A closer child node is visited prior to visiting the further ones. The authors of [28] argue that an SR-tree outperforms both the SS-tree and an R^*-tree, but requires a higher construction cost for building the tree structure.

To perform the evaluation on the access structures, we indexed the GCH for each of the categorized lots of 10,000 to 50,000 images. CCVs did not seem to a feasible alternative for this evaluation, due to the very high dimensionality of their feature vectors. A sequential search in a file of n records is said to be of the order $O(n)$, because the time taken to search a particular record is proportional to the number of records in the file [14]. Experiments were performed to investigate the storage-overhead and query processing time between the two combinations. A label attribute is used to uniquely identify a data point in the index file [28]. Now, if the label can be represented in 8 bytes, then using the VBA signatures, each data point can be represented in 40 bytes (32 bytes for the binary signature).

To analyze the impact of query processing time on the size of the access structure for determining the 20 nearest neighbors, we fix the page size at 8 Kbytes. A page size of 4Kbytes did not seem to be a practical choice due to the high-dimensionality of the color histogram vectors. Given that our similarity metrics are fairly simple, the query processing can be thought of as an I/O bound process [35]. We compared the query processing time for GCH/SR-tree against VBA/Linear-scan using the variables: indexed images, number of nearest neighbors and

[6] `http://www.rd.nacsis.ac.jp/ katayama/homepage/research/srtree/English.html`

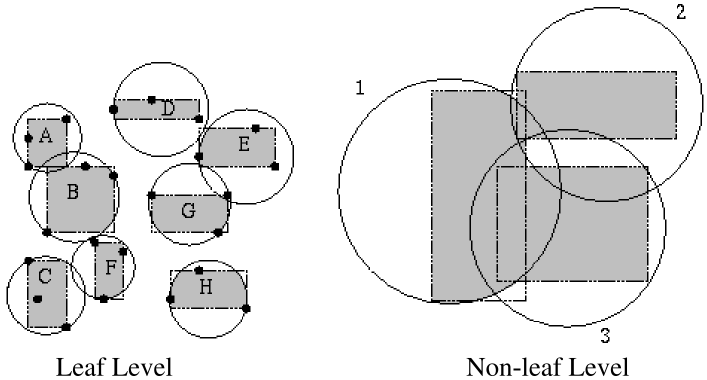


Figure 24: Intersection of bounding rectangles and spheres (adapted from [28])

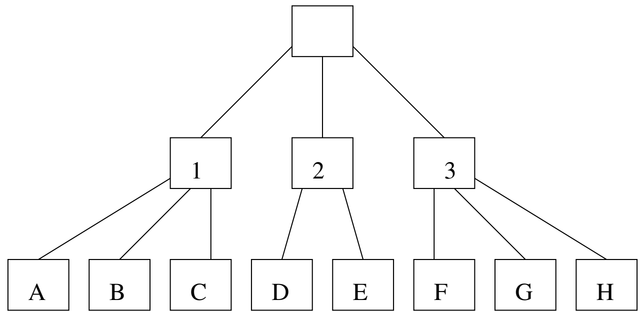


Figure 25: An SR-tree formed from 24 (adapted from [28])

page size. Figure 26 shows the effect of the number of indexed images on the query processing time. As already discussed, each signature can be represented in 40 bytes (32 bytes for the binary signature, and 8 bytes required to represent the label). Therefore, 1 Kbyte holds 25 records, hence, each I/O brings in 200 records for a page-size of 8 Kbytes, such that the number of I/Os required for an average search of 10,000 records comes down to 50. The results indicate that the number of I/Os by the GCH/SR-tree grows faster than using VBA/Linear-scan. Note that although the blocking of records does result in substantial improvement in performance, it however does not change the number of I/Os required for a sequential search operation on determining the nearest neighbors. Figure 27 illustrates this discussion, where the number of indexed images was kept constant at 30,000 images, and that each call brings in 8 Kbytes worth of records. A larger page-size increases the transfer rate with the disk, and results in less I/Os. Figure 28 shows that as we increase the page size, the curve of the number of I/Os for both VBA/Linear-scan and GCH/SR-tree drop down. The results also indicate the fewer number of I/Os required by VBA/Linear-scan in comparison to GCH/SR-tree for page sizes between 4 Kbytes to 16 Kbytes.

In order to compare the size of the index structure for an image collection, we index each of the categorized lots of 10,000 to 50,000 images. Suppose we have a file of 10,000 images, and as already discussed, each image using the VBA technique can be represented in 40 bytes. Therefore, to index 10,000 images, the file size would be 0.381 MB. We also index the various image sets for the combination GCH/SR-tree. Figure 29 shows that the SR-tree for GCH grows much more faster than the file used for indexing the VBA binary signatures. We thus conclude that storage-wise, the combination VBA/Sequential scan is much more efficient than an SR-tree indexing the GCH. The blocking of records does result in substantial improvement in performance, but does not change the size of the access structure for VBA/Linear-scan. Figure 30 illustrates this discussion, whereas the curve for GCH/SR-tree drops down with an increase in the page size.

We do recognize though that for very large image sets the GCH/SR-tree combination could become faster than VBA/Linear-scan. Unfortunately, we were unable to obtain enough data to further test this hypothesis.

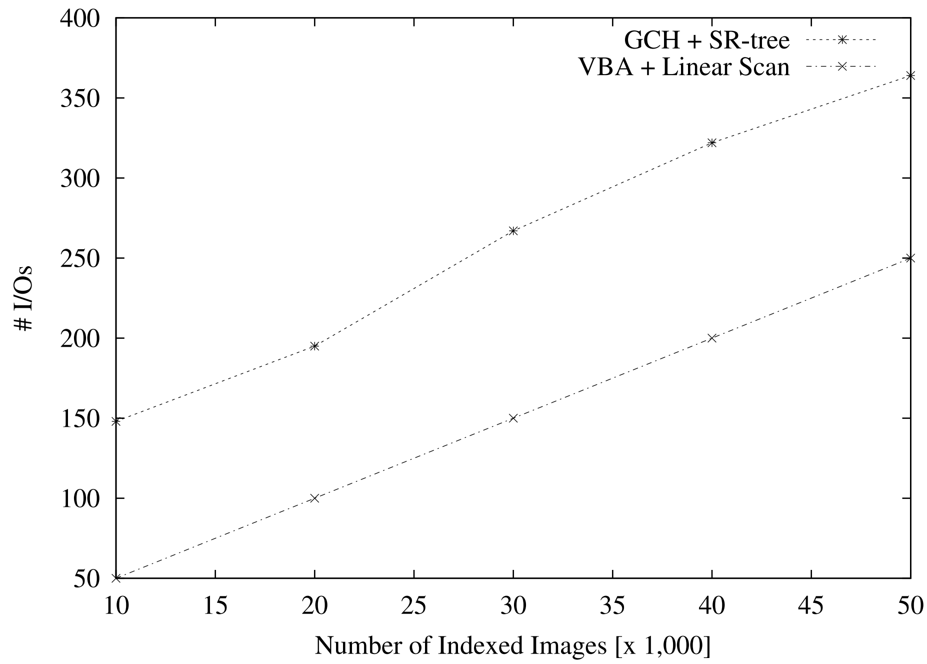


Figure 26: Number of I/Os for different indexed images

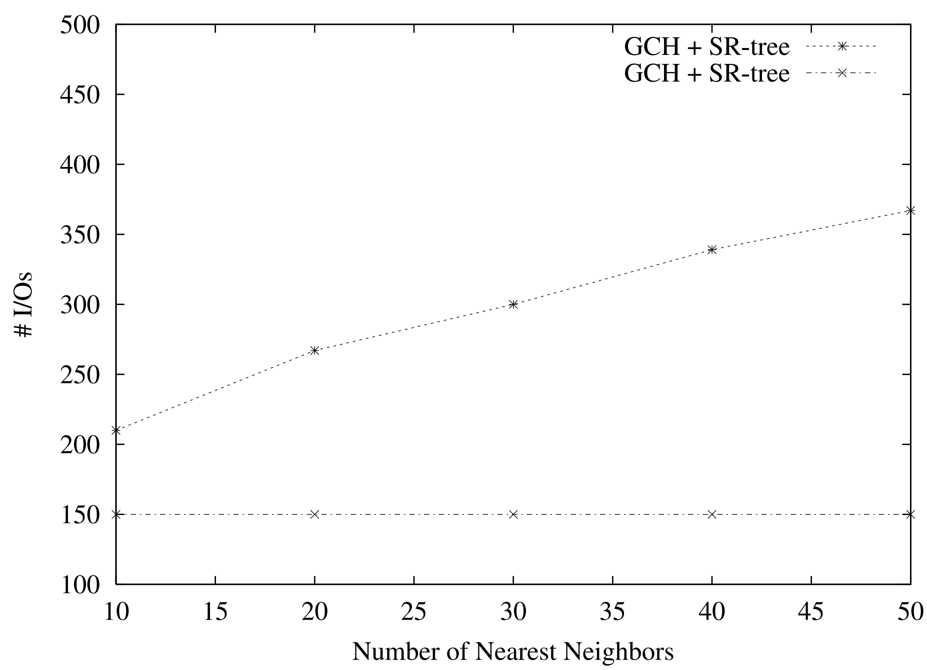


Figure 27: Number of I/Os for different number of nearest neighbors

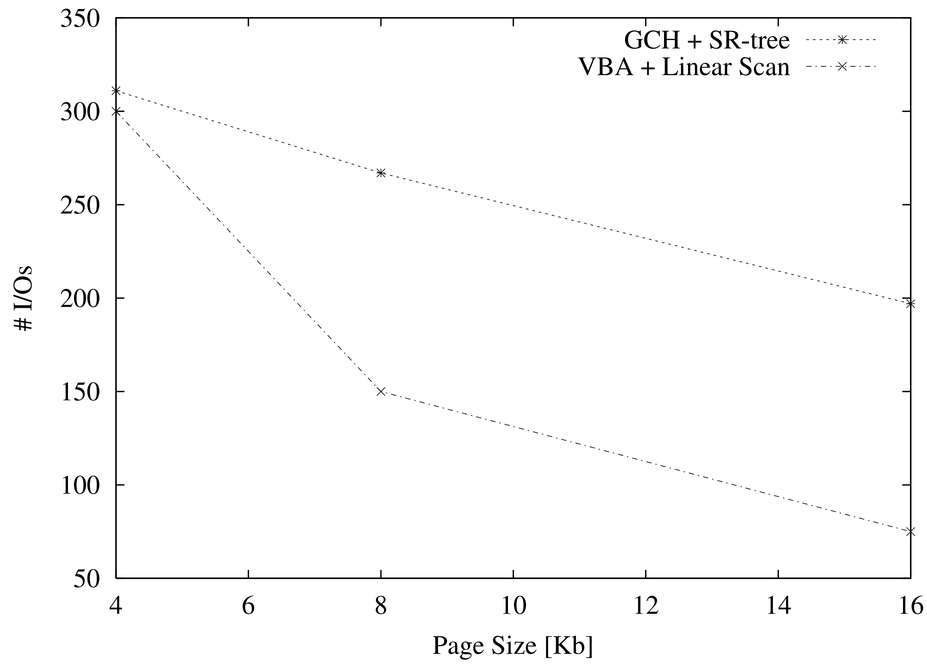


Figure 28: Number of I/Os for different page sizes

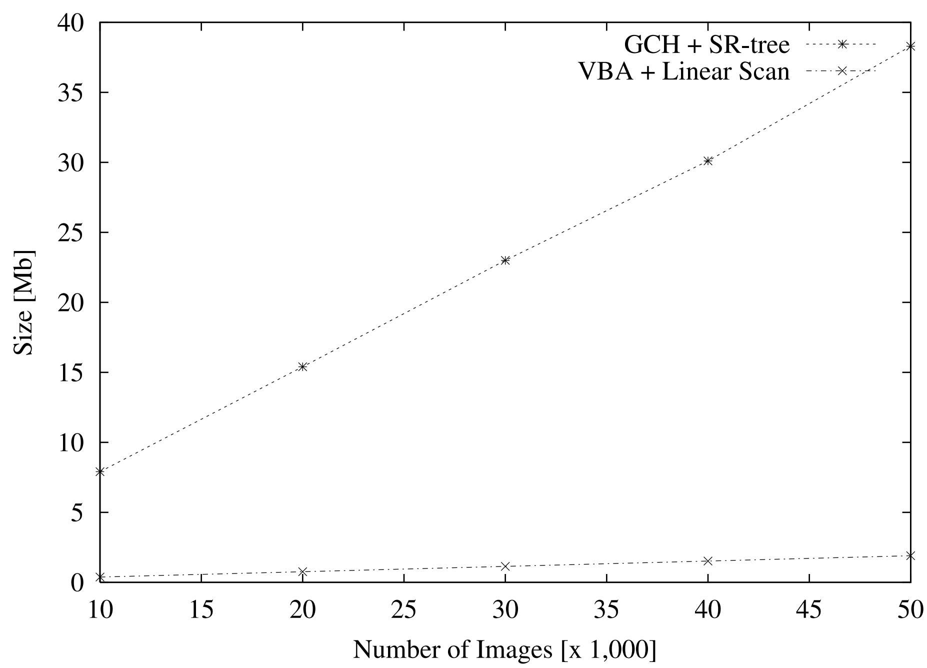


Figure 29: Size of the signature metadata for different number of images

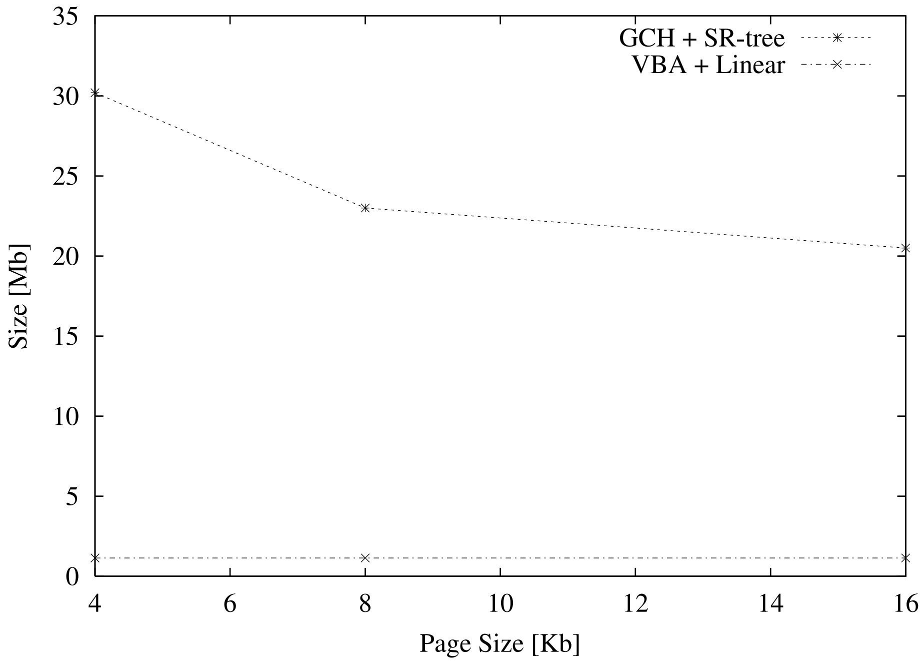


Figure 30: Size of the signature metadata for different page sizes

4.4 Conclusion

The investigated image retrieval techniques are tested under different conditions (example, varying the normalized number of colors in an image, the length of the binary signature bit-string etc.) to determine under what circumstances does one perform better than the another. The experiments performed on a heterogeneous database of 30,000 images with 15 images used as queries, demonstrate that both the allocation techniques perform better than the GCH, while VBA also outperforms the use of CCV by 20% in terms of the retrieval effectiveness. The retrieval process using a VBA produces the best results (approximately 55% better than the GCH), while still saving three-quarters of storage space. CBA (using 6 bins) requires a mere 48 bytes to represent an image as against 80 bytes required by the VBA (using 10 bins), 128 bytes required by the GCH, and 256 bytes required by the CCV. Therefore, the VBA approach not only performs better than GCH and CCV, but also consumes less resources, which is an important feature of the proposed approach.

5 Conclusions and Future Work

5.1 Conclusions

The main contribution of this thesis is the design of a new image abstraction methodology to generate binary signature for an image content. We have also introduced a metric for ranking the results, based on the comparison of signatures between a query image and the image collection. In general, the image abstraction technique is based on a compact representation of the information in a GCH by decomposing the color distributions of an image into bins for each normalized color. If all bins for a color distribution have the same capacity, we say that such an arrangement follows CBA, otherwise it follows a VBA approach. The design of VBA is based on the fact that distances due to the less dominant colors are important for an efficient retrieval. The use of the proposed methodology also results in space reduction due to the bit-wise representation of the image content.

The experiments performed on a heterogeneous database of 50,000 images with 15 images used as queries, demonstrate that both the derived allocation techniques perform better than the GCH, in terms of the retrieval effectiveness, while the usage of variable bins in image abstraction also outperforms the CCV. The above experiments also indicate that the GCH produces a better recall-precision curve than CBA, but delivers a weak ordering in terms of the overall desired ranking of the retrieved similar images. Experiments were also performed on the proposed image abstraction methodology by collapsing the last few bins in a binary signature to argue the importance of colors which are less dominant that major colors. The retrieval process using VBA produces the best results (approximately 55% better than the GCH and 20% better than CCV in terms of the retrieval effectiveness), while still saving three-quarters of storage space. The proposed image abstraction requires 80 bytes to represent an image as against 128 bytes required by the GCH and 256 bytes required by the CCV. Therefore, the VBA approach not only performs better, but also consumes less resources, which is an important feature of the proposed approach.

We also argue that superimposing a grid on the image would improve the retrieval effectiveness for all the approaches. Using such a grid approach has the drawback that the CBIR system would then become sensitive to image rotation and thus may detract from the overall retrieval ability of the system.

5.2 Future Work

The scope of this thesis was limited to address the design and implementation of a new image abstraction technique based on signature bit-strings and an appropriate similarity metric. There exist various other promising and challenging areas of research and development where the present work can be extended. These include:

- In the current design and implementation of the proposed methodology for image retrieval, the file containing the binary signatures is sequentially scanned and all image signatures are compared against the signature of the query image. Further enhancements can be made by designing a hash or tree-based access structure to support the signatures proposed in this paper and thus speedup the query processing (this has been addressed in [35]).

- As already discussed in Section 2, using a grid-based approach has the drawback that the CBIR is then sensitive to image rotation. Another interesting opportunity for research would be devising new ways to make a grid-based approach less sensitive to rotations.

- Finally, a more comprehensive set of tests including the signature encoding/ compression issues and comparisons to other proposed CBIR techniques must be performed in the near future.

It is not evidently clear that the framework presented in this thesis will always yield better results than the other investigated approaches. However, the arguments that the proposed signatures are a compact alternative, and that the less dominant colors are fairly important for an efficient retrieval have been verified.

References

[1] The Munsell color system, color and color management technical guides. URL: http://www.adobe.com/support/techguides/color/colormodels/munsell.html.

[2] The theory and practice of color - from a technical point of view. URL: http://xata.lviv.ua/˜color/technical.htm.

[3] A. R. Appas, A. M. Darwish, A. I. El-Desouki, and S. I. Shaheen. Image indexing using composite regional color channels features. In *Storage and Retrieval for Image and Video Databases VII*, pages 492–500, 1999.

[4] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. F. Shu. The Virage image search engine: An open framework for image management. In *Proceedings of the Storage and Retrieval for Still Image and Video Databases IV*, pages 76–87, San Jose, CA, USA, February 1996.

[5] N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, pages 322–331, Atlantic City, New Jersey, USA, May 1990.

[6] A. D. Bimbo. *Visual Information Retrieval*, chapter 3 and 4. Morgan Kaufmann Publishers, Inc., 1999.

[7] John F. Koegel Buford. *Multimedia Systems*. Addison-Wesley Publishing Co. - New York:ACM Press; Reading, Mass., 1994.

[8] M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 24–28, Santa Barbara, CA, USA, June 1998.

[9] V. Chitkara, M. A. Nascimento, and C. Mastaller. Content-based image retrieval using binary signatures. In *Technical Report TR-00-18, Department of Computing Science, University of Alberta*, Edmonton, Alberta, Canada, 2000.

[10] M. D. Fairchild. *Color Appearance Models*. Addison-Wesley Publishing Company, Reading, MA, 1998.

[11] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, and R. Barber. Efficient and effective querying by image content. In *Journal of Intelligent Information Systems*, pages 231–262, July 1994.

[12] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image content and video content: The QBIC System. In *IEEE Computer*, pages 23–32, September 1995.

[13] J. D. Foley, A. v. Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice in C*, chapter 13, Achromatic and Colored Light, pages 563–604. Addison-Wesley Publishing Company, Inc., 2nd edition, 1990.

[14] M. J. Folk and B. Zoellick. *File Structures*. Addison-Wesley Publishing Company, Inc., 2nd edition, June 1992.

[15] Y. Fu and J. C. Teng. Improving high-dimensional indexing with heuristics for content-based image retrieval. In *International Workshop on Integrated Spatial Databases*, pages 249–267, Maine, USA, June 1999.

[16] V. Gaede and O. Günther. Multidimensional access methods. *Computing Surveys*, 30(2):170–231, 1998.

[17] T. Gevers and A. W. M. Smeulders. A comparative study of several color models for color image invariant retreival. In *Proceedings of the First International Workshop on Image Database and Multimedia Search*, pages 17–23, Amsterdam, Holland, August 1996.

[18] R. C. Gonzalez and R. E. Wood. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1993.

[19] A. A. Goodrum. Image information retrieval: An overview of current research. *Informing Science*, 3(2):63–66, February 2000.

[20] A. Gupta and R. Jain. Visual information retrieval. *Communications of the ACM*, 40(5):71–79, 1997.

[21] A. Gupta, T. E. Weymouth, and R. Jain. An extended object-oriented data model for large image bases. In *Proceedings of the 2nd International Symposium on Design and Implementation of Large Spatial Databases (SSD)*, pages 45–61, Barcelona, Spain, September 1991.

[22] R.M. Haralick, K. Shanmugan, and I. Dinstein. Textural features for image classification. In *Proceedings of the IEEE Transactions on Systems, Man and Cybernetics*, pages 610–621, November 1973.

[23] J. Y. Hardeberg. Color management: Principles and solutions. *NORSIGnalet, Norwegian Signal Processing Society*, (3), 1999.

[24] P. Heckbert. Color image quantization for frame buffer display. In *Proceedings of the annual ACM conference on Computer Graphics*, pages 297–307, July 1982.

[25] P. B. Heidorn and B. Sandore. *Digital Image Access and Retrieval*. 1996.

[26] W. Hsu, T. S. Chua, and H. K. Pung. An integrated color-spatial approach to content-based image retrieval. In *Proceedings of the 3rd ACM Multimedia Conference*, pages 305–313, San Francisco, CA, USA, November 1995.

[27] A. K. Jain and A. Vailaya. Shape-based retrieval: A case study with trademark image databases. In *Pattern Recognition*, pages 1369–1390, 1998.

[28] N. Katayama and S. Satoh. The SR-tree: An index structure for high-dimensional nearest neighbor queries. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 369–380, Tucson, Arizona, USA, May 1997.

[29] S. Khoshafian and A. B. Baker. *Multimedia and Imaging Databases*. Morgan Kaufmann Publishers, Inc., 1996.

[30] S. Lin. An extendible hashing structure for image similarity searches. Master's thesis, Department of Computing Science, University of Alberta, Fall 2000.

[31] W. C. Low and T. S. Chua. Color-based relevance feedback for image retrieval. In *Proceedings of the International Workshop on Multimedia Database Management Systems*, pages 116–123, Dayton, OH, USA, August 1998.

[32] G. Lu. *Multimedia Database Management Systems*. Artech House Publishers, August 1999.

[33] D. Manolescu. Feature extraction - a pattern for information retrieval. In *Proceedings of the 5th Pattern Languages of Programming*, Monticello, Illinois, USA, August 1998.

[34] R. Mehrotra and J. E. Gary. Similar-shape retrieval in shape data management. In *IEEE Computer Magazine*, pages 57–62, September 1995.

[35] M. A. Nascimento, E. Tousidou, V. Chitkara, and Y. Manolopoulos. Color based image retrieval using signature trees. In *Technical Report TR-01-02, Department of Computing Science, University of Alberta*, Edmonton, Alberta, Canada, 2001.

[36] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content, using color, texture, and shape. In *Storage and Retrieval for Image and Video Databases I*, volume 1908 of *SPIE Proceedings*, pages 173–187, San Jose, California, USA, 1993.

[37] D. S. Park, J. S. Park, T. Y. Kim, and J. H. Han. Image indexing using weighted color histogram. In *Proceedings of the 10th International Conference on Image Analysis and Processing*, pages 909–914, Venice, Italy, September 1999.

[38] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the 4th ACM International Conference on Multimedia*, pages 65–73, Boston, Massachusetts, USA, November 1996.

[39] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *In Storage and Retrieval for Image and Video Databases II (SPIE)*, pages 34–47, Bellingham, Washington, USA, February 1994.

[40] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. In *International Journal of Computer Vision*, pages 233–254, San Jose, CA, USA, June 1996.

[41] V. Raghavan, P. Bollmann, and G. S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. In *ACM Transactions on Information Systems*, pages 205–229, 1989.

[42] R. Ramakrishnan and J. Gehrke. *Database Management Systems*. Number ISBN: 0-07-232206-3. McGraw-Hill, August 1999.

[43] S. Ravela, R. Manmatha, and E. M. Riseman. Retrieval from image databases using scale-space matching. In *CMPSCI Technical Report UM-CS-95-104, University of Massachusetts*, Amherst, MA, 1995.

[44] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Past, present, and future. *Journal of Visual Communication and Image Representation*, 10:1–23, 1999.

[45] P. Scheunders. A comparison of clustering algorithms applied to color image quantization. In *Pattern Recognition Letters*, pages 1379–1384, November 1997.

[46] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content-based image browser for the world wide web. In *In Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 2–9, San Juan, Porto Rico, June 1997.

[47] L. Shapiro and G. Stockman. *Computer Vision, 1/e*. Prentice Hall, 2001.

[48] G. Sharma and H. J. Trusell. Digital color imaging. In *In Proceedings of the IEEE Transactions on Image Processing*, pages 901–932, July 1997.

[49] M. K. Shin, S. Y. Huh, S. H. Lee, J. S. Yoo, K. H. Jo, and J. S. Lee. An efficient index structure for high dimensional image data. In *International Journal of Information Technology*, 2000.

[50] J. R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression*. PhD thesis, Graduate School of Arts and Sciences, Columbia University, 1997.

[51] J. R. Smith and S.-F. Chang. Automated image retrieval using color and texture. In *Technical Report CU/CTR 408-95-14, Columbia University*, 1995.

[52] J. R. Smith and S.-F. Chang. Automated binary texture feature sets for image retrieval. In *n Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2239–2242, Atlanta, GA, USA, May 1996.

[53] J. R. Smith and S.-F. Chang. Local color and texture extraction and spatial query. In *In Proceedings of the IEEE International Conference on Image Processing*, pages 1011–1014, Lausanne, Switzerland, September 1996.

[54] J. R. Smith and S.-F. Chang. *Querying by Color Regions using the VisualSEEk Content-Based Visual Query System, Intelligent Multimedia Information Retrieval*. AAAI/MIT Press, 1996.

[55] J. R. Smith and S.-F. Chang. Tools and techniques for color image retrieval. In *Storage and Retrieval for Image and Video Databases IV*, pages 426–437, San Diego/La Jolla, California, USA, 1996.

[56] J. R. Smith and S.-F. Chang. VisualSEEK: a fully automated content-based image query system. In *Proceedings of the 4th ACM International Conference on Multimedia*, pages 87–98, Boston, Massachusetts, USA, November 1996.

[57] J. R. Smith and S.-F. Chang. Visually searching the web for content. *IEEE Multimedia Magazine*, 4(3):12–20, 1997.

[58] R. O. Stehling, M. A. Nascimento, and A. X. Falcao. On 'shapes' of colors for content-based image retrieval. In *Proceedings of the International Workshop on Multimedia Information Retrieval (MIR '2000)*, pages 171–174, Los Angeles, USA, November 2000.

[59] M. A. Stricker and M. Orengo. Similarity of color images. In *Proceedings of the Storage and Retrieval for Image and Video Databases III*, pages 381–392, San Diego/La Jolla, California, USA, February 1995.

[60] L.T. Su. The relevance of recall and precision in user evaluation. In *Journal of the American Society for Information Science*, pages 207–217, New York, USA, 1994.

[61] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[62] L. Taycher, M. La Cascia, and S. Sclaroff. Image digestion and relevance feedback in the imagerover www search engine. In *In Proceedings of the 2nd International Conference on Visual Information Systems*, pages 85–94, San Diego, CA, USA, December 1997.

[63] C. J. van Rijsbergen. *Information Retrieval, 2/e*. Butterworths, London, 1979.

[64] C. Vertan and N. Boujemaa. Color texture classification by normalized color space representation. In *In Proceedings of the International Conference on Pattern Recognition*, Barcelona, 2000.

[65] S. Wesolkowski and E. Jernigan. Color edge detection in RGB using jointly euclidean distance and vector angle. In *Proceedings of the IAPR Vision Interface Conference*, pages 9–16, Trois-Rivires, Canada, 1999.

[66] D. A. White and R. Jain. Similarity indexing with the SS-tree. In *Proceedings of the 12th International Conference on Data Engineering*, pages 516–523, New Orleans, Louisiana, USA, 1996.

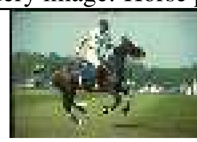
[67] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes, Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing, 2nd edition, 1999.

[68] B. Xu. A visual query facility for DISIMA image database management system. Master's thesis, Department of Computing Science, University of Alberta, April 2000.

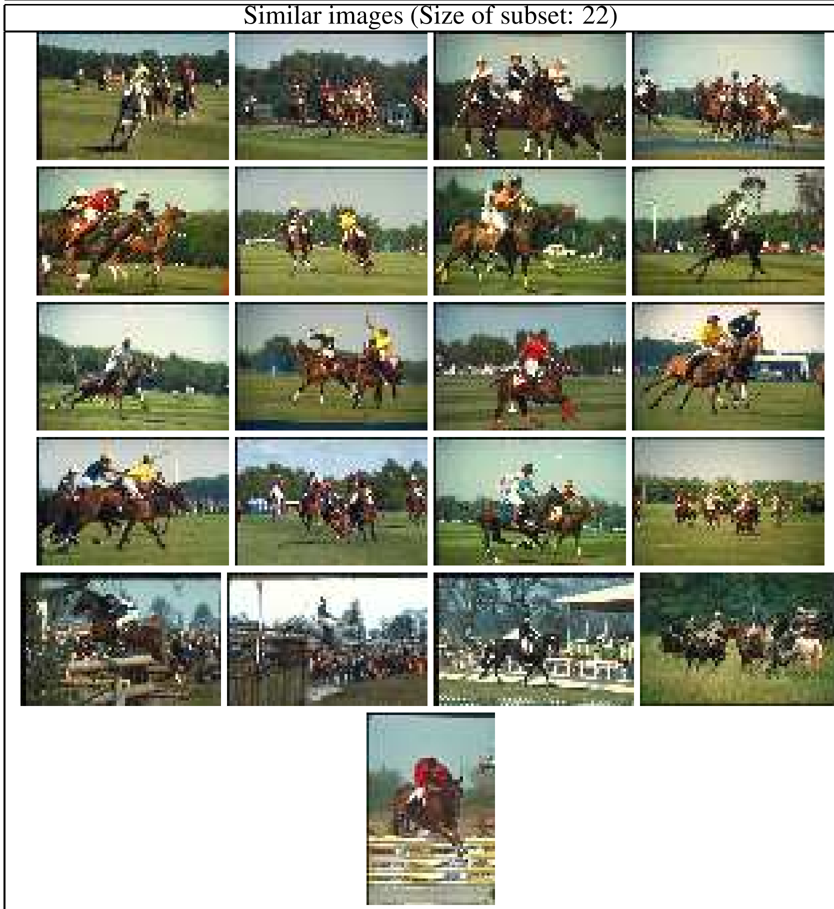
[69] R. B. Yates and B. R. Neto. *Modern Information Retrieval*. Addison-Wesley Publishing Company, 1999.

A Sample Query Images and Subsets

Query image: Horse polo



Similar images (Size of subset: 22)



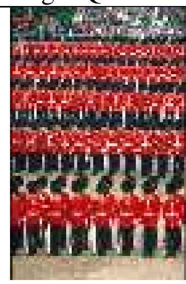
Query image: Old boats

Similar images (Size of subset: 8)

Query image: Halloween pumpkins

Similar images (Size of subset: 11)

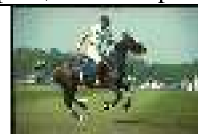
Query image: Queen's red army



Similar images (Size of subset: 18)

B Sample Query Images and Results

Query Image (Horse polo, size of expected answer: 22 images)

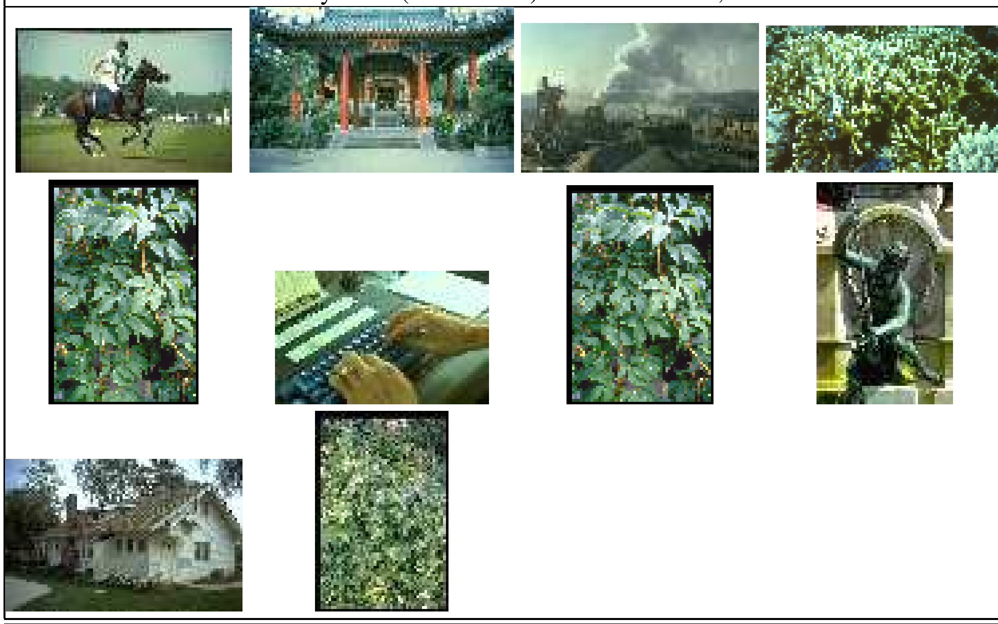


10 best matches by CCV – Recall: 13.63%, Precision: 30%

10 best matches by VBA (Bin size:10) – Recall: 18.18%, Precision: 40%

Query Image (Horse polo, size of expected answer: 22 images)

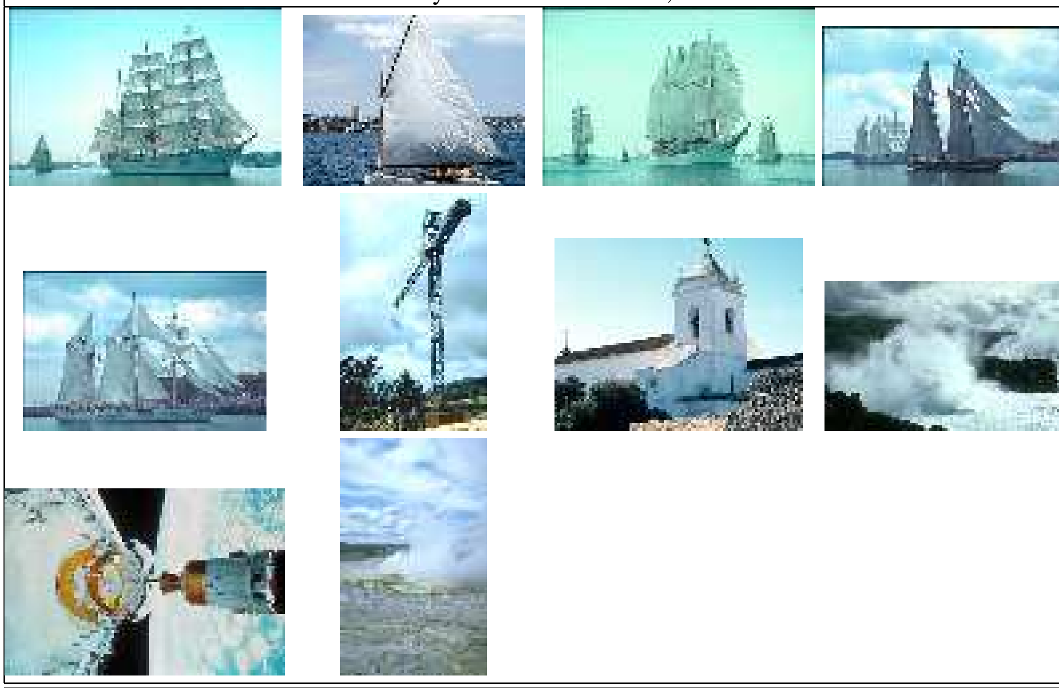
10 best matches by CBA (Bin size:10) – Recall: 4.54%, Precision: 10%



10 best matches by GCH – Recall: 9.09%, Precision: 20%

Query Image (Old boats, size of expected answer: 8 images)

10 best matches by CCV – Recall: 50%, Precision: 40%



10 best matches by VBA (Bin size:10) – Recall: 75%, Precision: 60%

Query Image (Old boats, size of expected answer: 8 images)

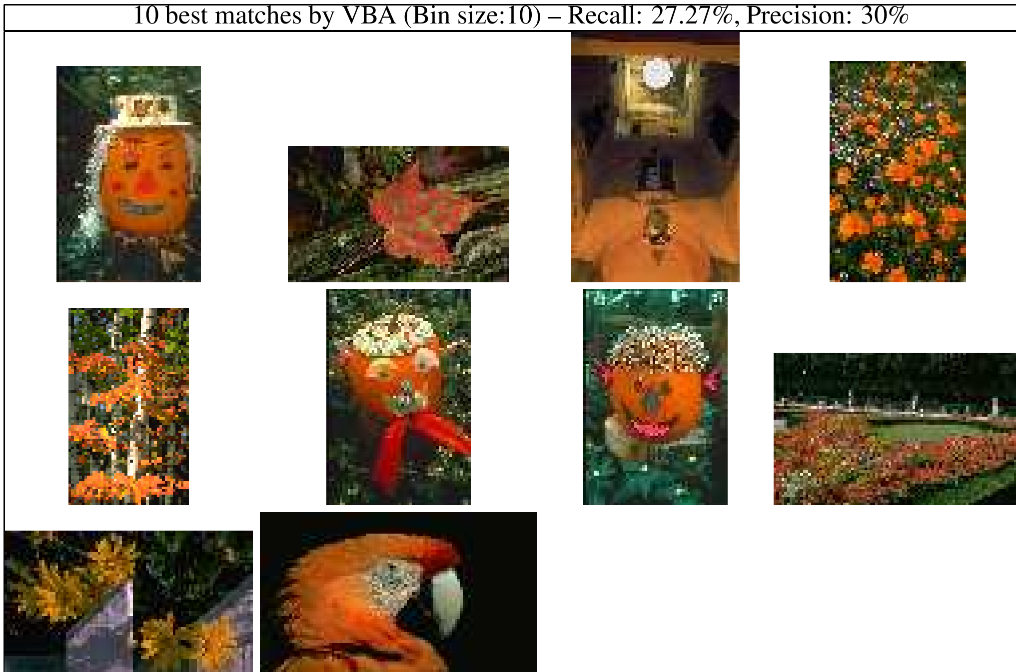
10 best matches by CBA (Bin size:10) – Recall: 62.5%, Precision: 50%

10 best matches by GCH – Recall: 37.5%, Precision: 30%

Query Image (Halloween pumpkins, size of expected answer: 11 images)

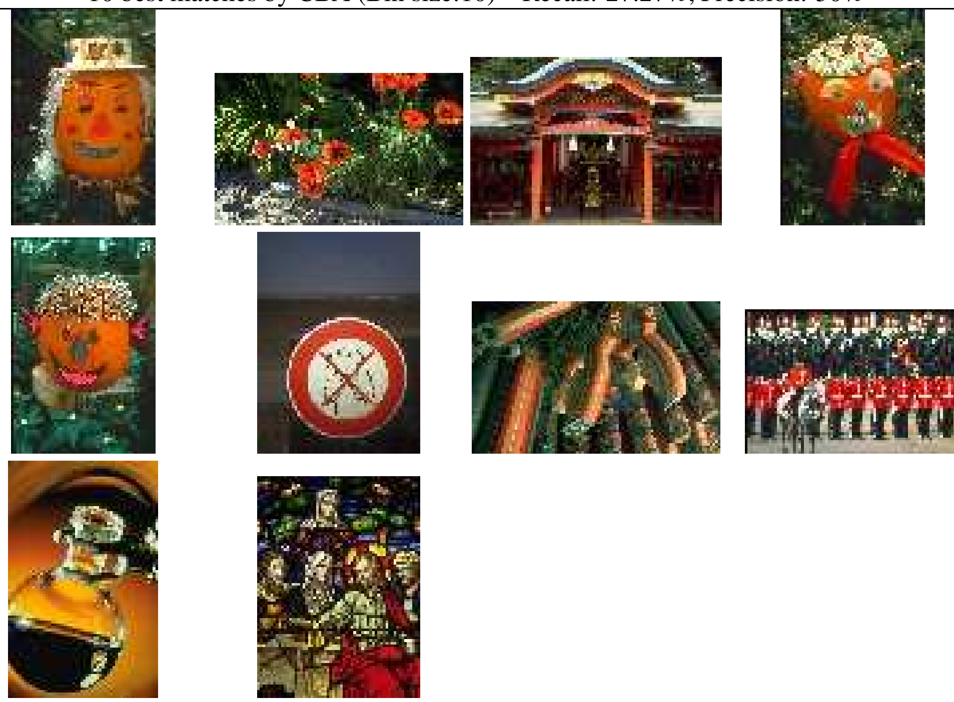
10 best matches by CCV – Recall: 18.18%, Precision: 20%

10 best matches by VBA (Bin size:10) – Recall: 27.27%, Precision: 30%

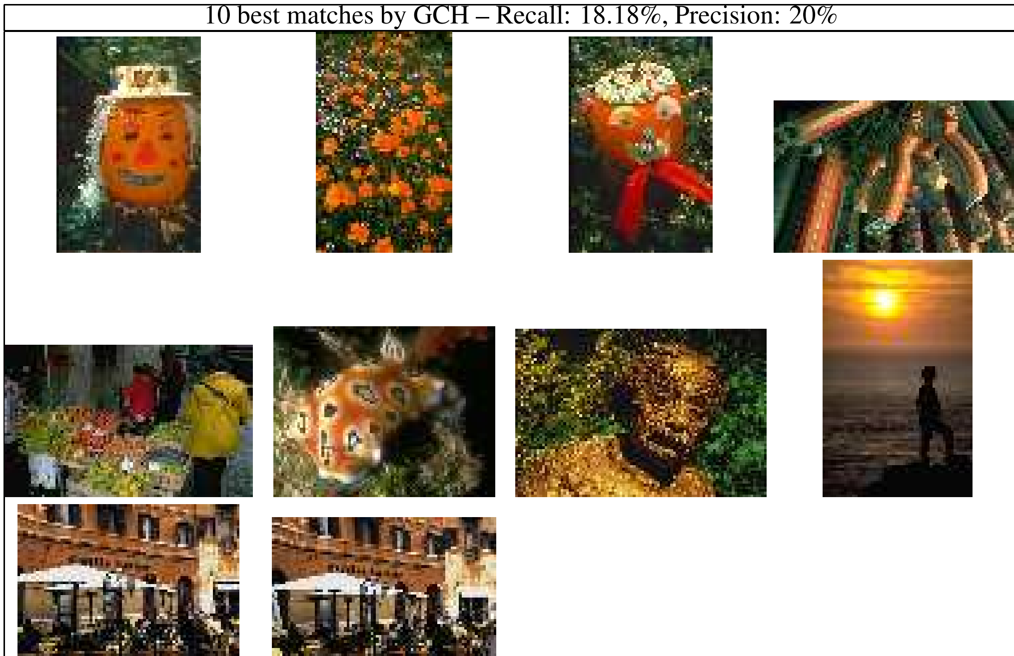


Query Image (Halloween pumpkins, size of expected answer: 11 images)

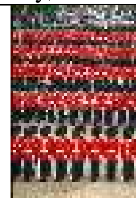
10 best matches by CBA (Bin size:10) – Recall: 27.27%, Precision: 30%



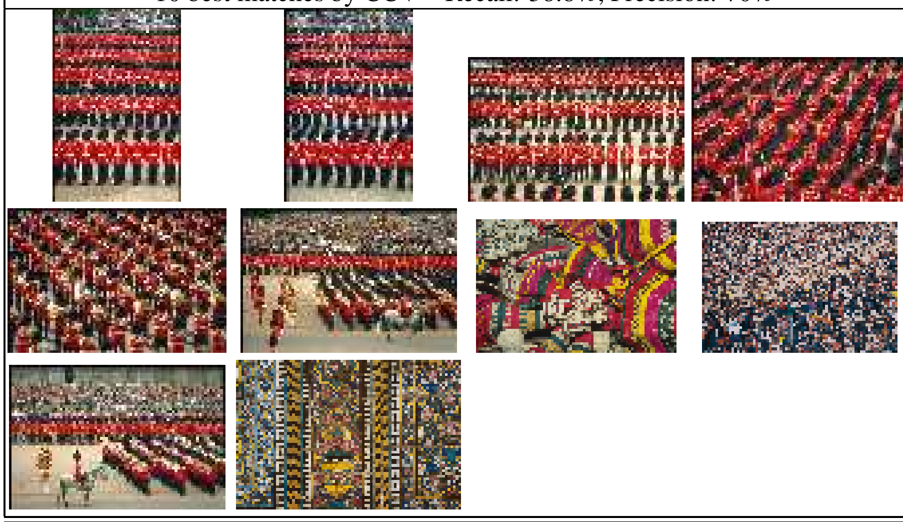
10 best matches by GCH – Recall: 18.18%, Precision: 20%



Query Image (Queen's red army, size of expected answer: 18 images)

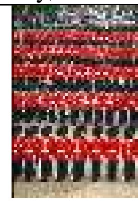


10 best matches by CCV – Recall: 38.8%, Precision: 70%

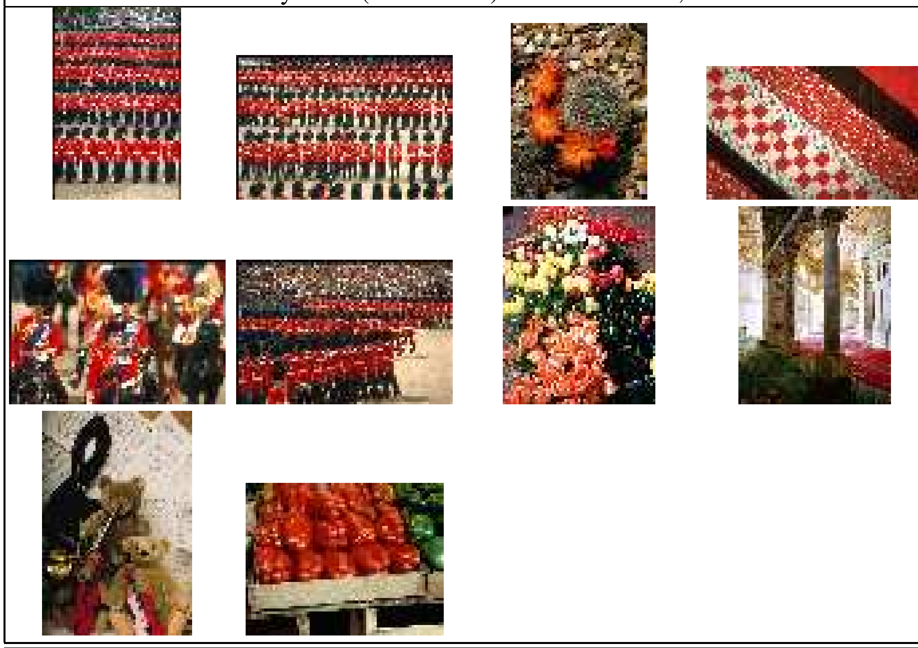


10 best matches by VBA (Bin size:10) – Recall: 33.3%, Precision: 60%

Query Image (Queen's red army, size of expected answer: 18 images)



10 best matches by CBA (Bin size:10) – Recall: 22.2%, Precision: 40%



10 best matches by GCH – Recall: 16.6%, Precision: 30%