University of Alberta

Region-based Image Retrieval Using Multiple Features

by

Veena Sridhar ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 2002

0-612-81478-5

University of Alberta

Library Release Form

Name of Author: Veena Sridhar

Title of Thesis: Region-based Image Retrieval Using Multiple Features

Degree: Master of Science

Year this Degree Granted: 2002

S. Veena

Veena Sridhar
3A-9011, 112 Street, 89 Avenue
Edmonton, Alberta
Canada, T6G 2C5

Date: MAY 28 2002

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Region-based Image Retrieval Using Multiple Features** submitted by Veena Sridhar in partial fulfillment of the requirements for the degree of **Master of Science**.

Dr. Bin Han

Dr. Osmar R. Zaïane

Dr. Mario A. Nascimento

Xiaobo Li

Dr. Xiaobo Li

Date: May 21, 2002

Vidhya dadathi vinayam, Vinayat yaathi patratham
Patrathvath Dhanam Aapnothi, Dhanath dharma thathaha sukham

To my family especially my mom and dad who have sacrificed their todays for my tomorrows

Abstract

Large image databases are becoming popular due to the ease with which images are being created/digitized and stored. Content Based Image Retrieval (CBIR) has therefore evolved into a necessity. It is a challenging task to design an effective and efficient CBIR system. Current research works attempt to obtain the semantics or meaning of the image to perform better retrieval. Segmentation of an image into regions may reveal the true objects in the image. The local properties of regions can help matching objects between images and thereby contribute towards a more meaningful CBIR.

The main contribution of this thesis is a CBIR algorithm, called SNL, that utilizes the regional properties of the images. Each image is segmented and features including the colour, shape, size and spatial position of the region are extracted. Regions are matched by comparing the region content, shape and spatial position and the Integrated Region Matching (IRM) distance measure between the whole images is calculated. The relative importance of the above features is investigated. SNL outperforms the Global Colour Histograms (GCH) and Colour Based Clustering (CBC) in terms of precision-recall.

A more efficient version of SNL, SNL$^+$, is designed using the *Omni* filtering technique recently proposed, along with the IRM distance measure. Our experiments have shown that SNL$^+$ can significantly reduce the query time without losing effectiveness.

Acknowledgements

I would first of all like to thank my family including my father Mr. A.P. Sridhar, my mother Mrs. Girija Sridhar and my sister S. Vijayalakshmi whose guidance, prayers and confidence in me, have been a key driving force throughout my life. I would like to thank Ganesh for encouraging me, being so patient with me and for proof reading my thesis. I would like to thank my supervisors Dr. Nascimento and Dr. Li for their able guidance and support without which this thesis work would not have been possible. I also wish to thank Renato Stehling for providing me the code for the segmentation algorithm and for the constructive discussions. My thanks to my colleagues Stanley and Jeffrey who were always working with me and helping me out in various discussions. Special thanks to Haseeb and Vishal for their help and guidance during my course work. Finally I would like to thank my my cousins Kiran, Kripa, Karpoor, Poorna, Jaishree, Guru, Lavanya, Bharath and Raghavendran for their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and motivation

1.1 Background

Image database management and retrieval has been an active research area since the 1970s [28]. With the rapid increase in computer speed and decrease in memory cost, image databases containing thousands or even millions of images are used in many application areas such as medicine, satellite imaging, and biometric databases, where it is important to maintain a high degree of precision. With the growth in the number of images, manual annotation becomes infeasible both time and cost-wise. Content-Based Image Retrieval (CBIR) is a powerful tool since it searches the image database by utilizing visual cues alone. CBIR systems extract features from the raw images themselves and calculate an association measure (*similarity* or *dissimilarity/distance*) between the query image and database images based on these features. CBIR is becoming very popular because of the high demand for searching image databases of ever-growing size. Since speed and precision are important, we need to develop a system for retrieving images that is both efficient and effective.

1.2 Applications of CBIR

The potential applications of image retrieval systems are increasing day by day. So far the most popular use of CBIR is for web searching. There are a number of retrieval engines such as QBIC [1], Netra [2], Simplicity [3], Yahoo! Picture Gallery [4], Google Image Search [5] etc., that facilitate searching images from the web. Some of these systems use captions or the text surrounding the image, while others use a combination of various features such as, colour, texture, spatial position etc. Another application of CBIR which has recently become very popular is in the area of crime prevention [14]. Databases that contain photographs [6] [7], fingerprints [8] and shoe prints [9] can be used in criminal investigations. Another important application is in the field of medical diagnosis [10]. CBIR is used in a number of diagnostic techniques [39] such as mammography, tomography and histopathology [14]. Image Retrieval can be very useful in identifying similar cases that have been treated in the past to assess the type of treatment to be given to a case in hand. CBIR is also used in the Geographical Information Systems (GIS) and remote sensing [56]. CBIR can be used for retrieving parts of videos such as movies and games. Other applications include online museums, advertising agencies and fashion designing.

1.3 Research areas in CBIR

CBIR involves a number of research areas. In what follows, we try to highlight some of them and the issues involved.

Understanding human-understanding of images: This research area requires knowledge in cognitive science since it completely relies on human judgment. The fundamental problem that a retrieval system faces is to try and understand what humans would perceive as similar and what impression an image can leave on most people's mind. This can in turn help define a more acceptable set of relevant im-

[1] http://wwwqbic.almaden.ibm.com/
[2] http://maya.ece.ucsb.edu/Netra/netra.html
[3] http://wang14.ist.psu.edu/cgi-bin/zwang/regionsearch_show.cgi
[4] http://gallery.yahoo.com/
[5] http://images.google.com/
[6] http://www.viisage.com/
[7] http://www.faceit.com
[8] http://www.east-shore.com/
[9] http://www.fosterfreeman.co.uk/sicar.html
[10] http://www.brisbio.ac.uk/

ages corresponding to a given query image. A benchmark in this case is a standard set of query images along with their set of relevant images, which would provide a "target" for different CBIR systems to implement and evaluate their systems. Benchathalon [11] is the first major effort to try and come up with such a benchmark. Benchathalon is a series of contests conducted every year to exercise one or more CBIR benchmarks. In 2001, a Benchmark called Birds-1 [21] was designed as a distributed computing architecture using a client server model to test both the effectiveness and efficiency of retrieval systems. This is still a work in progress. Some of the problems involved in forming benchmarks are:

- To find a fairly large number of randomly chosen people to mark the relevant and non-relevant images corresponding to a given query image.
- The set of relevant images obtained depends on the database in hand. Therefore each time the database is changed, the relevant image set has to be changed too.

Due to these problems, no benchmark has been universally accepted.

Building human friendly user interfaces: This is the problem of how to let the user query a database. There are two areas that could be explored. One is specification through query languages and the other is to develop a more natural interface. Systems that use captions or some text describing the image have interfaces, where users are asked to enter keywords to describe the image/object they are looking for. Some languages have been developed to describe the query image.

An effort was made by Henning *et al.* [44] to develop MRML (Multimedia Retrieval Mark-up Language). It is an open-communication protocol for content-based image retrieval system and is similar to SQL. Another work is the EXQUISI (Expressive Query Interface for Similar Images) by Faulus and Ng [16]. This work describes an expressive query language and an interface for querying, that allows the user to incorporate some ambiguities and imprecisions during the query specification process. The system allows the user to specify colour range queries as well

[11] `http://www.benchathlon.net/`

as texture range queries. Specification using an SQL-like language is very accurate, however, it is not very popular because the user is forced to learn the database schema and the query language. In addition, these query languages do not recreate the user's query visually for him to make sure that the specification, in fact, did render the image that the user had in mind. In other words, such language-based descriptions never manage to capture the visual content sufficiently [58]. A wrong description can therefore result in a disastrous mismatch of information described by the user and the image that he is presented with by the system. IFQ [37] is a visual query interface which attempts to combine the best of both the above mentioned areas. It allows the user to specify query objects with a fine granularity and also to describe the spatial relationship between them. IFQ then translates the query into CSQL (Cognition and Semantics-based Query language) [34] because CSQL queries precisely address the database system.

Some systems require the user to draw a rough diagram describing what he/she is looking for. But the most popular technique for querying images is Query By Example (QBE), where the user gives the system a sample image and expects the system to retrieve images similar to the given image. Given a sample query image, there are two possible queries one can come up with: finding the most similar image and to find images whose similarity to the query image is greater than or equal to a given threshold S.

- Range query: Given a query object and a range called the query radius, find all objects in the database, whose distance to the query object is less than the query radius S.

- Nearest neighbour query: Given a query object, find the object in the database, whose distance to the query object is the least. A popular variation of this type of query is the k-nearest neighbour query.

There are several other issues involved in constructing user-friendly interfaces such as whether or not the user should be able to assign weights to some features of interest and if the user should be burdened with the task of picking out the relevant and non-relevant images which can then be used as a relevance feedback to enhance retrieval performance.

Extracting and representing image features and developing a similarity measure that reflects the human perception of similarity: Direct comparison of two images, pixel by pixel, is infeasible. Therefore, each image must be represented by a set of features and a similarity/distance measure must be defined based on these features. The quality of the features, the effectiveness of the feature extraction and the discriminative power of the similarity measure are critical for a CBIR system. Therefore, these research issues (feature extraction and similarity measure design) are the focus of many CBIR systems. It is also important that the similarity measure agrees with human perception of similarity. Humans tend to decide similarity based on the objects in the image. This cognitive side of image retrieval requires some knowledge about the objects in the image and how these objects relate to each other.

Compact storage structures for images: The storage space required for images and their representations directly relates to the efficiency of the system. Providing compact storage space depends on the representation of the image. The features extracted from the image should be simple and condensed. The system should be prudent enough to retain only some important features of the image that are crucial in distinguishing between images. One can also use some storage structures such as binary signatures [7] to decrease the storage requirements.

Query processing time: People usually prefer reasonable results obtained in a short time to good results that can be obtained after a long wait. Hence query time (efficiency) is as important an issue as the quality of results for CBIR systems. The aim is to decrease the query processing time as much as possible by making online calculations simple and fast by using index structures or filtering techniques.

1.4 Research goal

In this thesis, we have developed a technique to extract and represent features from images and also defined a good distance measure. The motivation behind this representation is based on the fact that recent approaches to represent images require the image to be segmented into a number of regions (a group of connected pixels which share some common properties). This is done with the aim of extracting the objects in the image. However, there is no unsupervised segmentation algorithm that is capable of always partitioning an image into its constituent objects, especially when considering a database containing a collection of heterogeneous images. Therefore, an inaccurate segmentation may result in an inaccurate representation and hence in poor retrieval performance. The thesis concentrates on a robust colour representation and a composite colour-size distance between regions of images. In the proposed technique we also take into account some other features such as shape and spatial location of the regions in the image. The last research issue, related to the query processing time, is also accounted for when we optimize the proposed technique.

1.5 Outline of the thesis

The reminder of this thesis is organized as follows. Chapter 2, discusses some related work in the field of content based image retrieval using visual attributes like colour, shape, spatial position and also some works related to region-based image retrieval. Chapter 3 presents our new CBIR approach, SNL, which focuses on a colour representation that is not very sensitive to segmentation inaccuracies and also accounts for other features of regions. Chapter 3 further describes some of the graphs that depict the performance of the SNL technique. Chapter 4 describes an efficient SNL-based technique called SNL$^+$ and the set of experiments that were performed to evaluate this technique. Finally in Chapter 5, we present the conclusions and state some directions for future work.

Chapter 2

Related work

2.1 Colour and spatial features

Several features have been used to represent images in CBIR systems. The most commonly used feature is colour. Global Colour Histogram (GCH) is a simple and effective way of utilizing the colour features. An example image and its GCH are shown in Figure 2.1. The GCH is an n-dimensional vector $(h_1, h_2, ..., h_n)$, where each element h_j represents the percentage of pixels of colour j in an image. GCH is invariant to scaling and rotation and very simple to compute. However, GCH suffers from the fundamental disadvantage of being too general. In other words, GCH takes into account only the distribution of colours but disregards the inherent relation between the bins. That is, "light green" is no similar than "red" to "dark green" if they are in different bins. Therefore, bin definition or colour quantization is a critical issue. As mentioned above, perceptually similar colours may be quantized into different bins and vice versa. Another drawback of GCH is that it does not consider the spatial location of the colours present in an image.

To avoid some of the problems stated above, local colour histograms (Figure

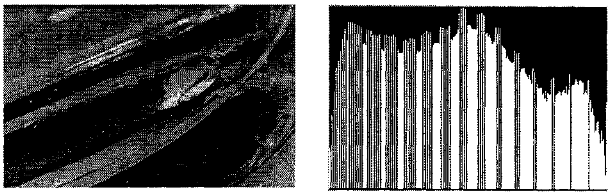


Figure 2.1: Example image and its Global Colour Histogram

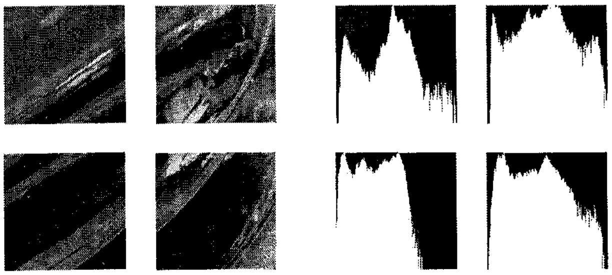


Figure 2.2: Example image and its Local Colour Histogram (shown in gray scale)

2.2) were proposed. An image is partitioned into equal sized sub-images/blocks and the similarity between two images is based on the histogram distances between corresponding blocks. This method is not capable of handling geometric transformations like rotation and translation and it suffers from problems like cell-cross talk [62] and variance to absolute spatial location. Some solutions, such as [1], [55] and [69] have been proposed to make the grid-based approach invariant to rotation and translation, but they are computationally expensive.

In a paper by Natsev *et al.* [45], image retrieval is performed based on the colour-layout property. Each image is divided into several sub-images by sliding windows of various sizes and for each sub-image, a colour layout signature is extracted. The similarity between images is then computed by comparing the signatures of these sub-images. The advantage of this system is that it is able to reduce the shifting and scaling sensitivity, the disadvantage is that the computation complexity increases and the system does not consider other features such as texture and shape.

Smith and Chang proposed the colour sets [57], which approximates the colour histogram in order to speed up the retrieval process in the case of very large databases. A colour set represents a set of colours chosen from a quantized colour space and since features are represented as a bit-string, a binary tree is used to speed up the search process.

Another colour based approach was proposed in [65], where an image was represented with the help of the first three moments namely the colour average, variance and skewness. The technique has the advantage of low space overhead and is also simple to compute. The similarity between two images is calculated as the weighted sum of the absolute differences between moments in the query image and the moments of all the images in the database. Even though colour moments were able to avert the quantization effects unlike the colour histograms, they still lacked spatial information.

Gathering spatial information of objects in an image is an essential process for GIS systems. This involves representing the absolute spatial position and also the relative spatial position of objects. Operations such as encapsulation, intersection and overlapping are used. Colour layout combines spatial information with the colour information present in the image and forms an important feature during the retrieval process.

Pass *et al.* [48] proposed a new method using the colour coherence vectors (CCV). They proposed a histogram based approach that incorporated some spatial information as well. The image is initially blurred to remove small differences between pixels and then the colour space is discretized to n-colours. Pixels within a bucket were classified as either coherent or incoherent depending on whether they were part of a large similar-pixeled region.

Stehling *et al.* proposed the cell/colour histograms (CCH) [63] for image retrieval. It was an effort to elegantly combine the information represented by local histograms in a partition based approach and global colour histograms. The representation makes use of the fact that a low number of distinct quantized colours are usually present in images to lower its space overhead.

An approach using colour-spatial information was proposed by Hsu *et al.* [25]. Their aim was to include some spatial information that was absent in the histograms. They made two assumptions. The first was that two images can be called similar if they have patches of a similar colour at approximately the same positions in the image and secondly, they also assumed that human eyes were attracted to the center of the image rather than the corners. They then constructed the global colour histogram and the histogram for a central window and two colours were selected from each histogram. The first two colours formed the background colours and the second two formed the object colours. The spatial properties of the chosen colours were obtained using a technique called maximum entropy discretization that determined clusters of these representative colours in the form of rectangular regions. In order to calculate the similarity between two images, two different types of measures were proposed. The direct measure computes the intersection between all possible pairs of rectangular regions between two images for all the representative colours in them. The indirect measure uses different possible configurations between any two rectangles to decide whether two regions overlap and also their type of overlap. Based on this fact, if two regions have the same configuration, the intersection between the two regions is found.

In a paper by Ooi *et al.* [47], the authors propose to solve both the qualitative matching problem and the indexing problem in image retrieval. They propose an image retrieval system that is based on both colour and spatial information of images. To extract the colour-spatial information, they use a three phase heuristics. In the first phase, they extract the dominant colours from the colour histogram. In the second phase, a set of clusters (rectangular shaped) for each dominant colour is obtained based on the maximum entropy discretization. In phase three, the clusters are ranked in the descending order of their sizes. But here again too many clusters can hamper the performance in terms of quality and speed. So they define a parameter $Cl_{dominant}$, which is the number of dominant clusters that will be selected in the second phase. They determine a value for this $Cl_{dominant}$ after conducting several experiments. They also propose a multi-tier indexing mechanism called the Sequence Multi-Attribute Tree (SMAT). They implement a two layer SMAT, where the first layer is used to prune away clusters that are of different colours, while the second layer discriminates clusters of different spatial locality. SMAT essentially consists of multiple trees integrated together in a hierarchical manner and only the leaf nodes of the lowest level contain a pointer to the image data. The search algorithm in SMAT is simple since it is derived from the R-tree [22] search algorithm. There are several issues related to the construction and maintenance of the SMAT tree such as initial loading, insertion, deletion and height balancing. They also proposed some methods to maintain the height balance of the SMAT structure.

2.2 Shape features

Shape, next to colours, is considered an important characteristic in describing the salient objects in images and can help discriminate between two images and therefore in retrieval.

Shape extraction involves several steps. The first step is to use a suitable segmentation method to divide the image into regions. Segmentation techniques can be classified into three broad categories: region-based, boundary-based and pixel-based. Region-based segmentation methods include region growing by pixel aggregation, region splitting and merging techniques. Edge detection technique is a common boundary-based method and thresholding is a popular pixel-based segmentation method.

Once the image is segmented and regions are obtained, features belonging to the obtained regions should be recorded. Any segmentation technique mentioned above can use any of the representation schemes. Chain codes [19] use the 8-connectivity or the 4-connectivity to represent the line segments that constitute the boundary of a region. Signatures, shape numbers and polygonal approximation are other representation schemes.

The next step is to use appropriate descriptors for these regions so that they can be used while matching regions of different images. Shape descriptors are classified into three types. Boundary based descriptors define the properties of the boundary (2D closed curve) or the exterior of a region. Boundary based techniques mainly use the outline of the region to calculate shape. Fourier descriptor is one of the well-known methods belonging to this category (e.g., [54]). In this technique, the boundary of a given region is obtained and Fourier transformed [19]. The dominant Fourier coefficients are used as the shape descriptors. Other descriptors in this category are shape numbers and moments [19].

Regional descriptors on the other hand, describe the content or the interior of the region. Moment invariants [28] is the most commonly used descriptor. Hu [26] proposed seven such moments and there were several papers, e.g., [71], [30] that improved upon his idea. Area, calculated as the total number of pixels in a region, minimum/maximum bounding rectangle/circle/ellipse and the ratio between the sides, radii, and length of the radius are other regional descriptors. Compactness, measured as the ratio between the squared perimeter and area, Elongatedness, which is the ratio between the length of the longest chord in the region and the chord perpendicular to it, are also examples of descriptors belonging to this category.

If a region has a complex shape, it can be further broken down into simpler shapes such as rectangles or circles and some properties of these simple shapes and their relationships can be used for shape descriptors. Other regional descriptors include colour and texture. Some characteristics of regions such as the center of gravity are not specific to the boundary or content of the region. They are a separate class by themselves.

Shape matching can be done in the following ways: by representing shape with a feature vector and using some notion of distance between these vectors; by measuring the effort required to transform one shape to another. Some of the shape based image retrieval systems are described in the following paragraphs.

Jagadish [29] proposed a shape based retrieval technique which could handle different notions of similarity including changes in scale, position and even relative sizes of objects. In addition, his technique was robust to the presence of noise. The aim of this technique was to not only output shapes that matched the given query shape exactly, but also to render shapes that were similar to the given shape by using "area difference". This technique was restricted to images/objects that consisted of rectilinear shapes. The technique proposed two rectangular covers, an additive cover and a general cover and the shape representation was the relative positions of these rectangles. Kupeev and Wolfson [33] proposed a shape representation technique that worked for irregular contours. Shapes were represented as weighted graphs called G-graphs, which characterized the quantitative characteristics of a contour in a given orientation. A similarity measure to match the contours was also proposed and modified to allow matching of perceptually convex objects.

Grosky and Mehrotra [20] have approximated shapes by using polygonal approximations. Each vertex v is represented by the X, Y coordinates, the internal angle of the vertex and the distance from the adjacent vertex in the clockwise direction. Thus, a shape is represented as a string and similarity between two shapes is computed using a string edit distance.

In [42], the authors proposed a similar shape retrieval technique that was general enough to handle rigid, articulated objects and flexible enough to handle simple and complex queries. For each shape, certain boundary points or interest points are found. A pair of interest points are chosen to form the basis vector. The basis vector is assumed to be a unit vector along the x-axis and all the other interest points are transformed to this coordinate system. Feature vector of a shape is represented as a sequence of normalized interest points. The transformation parameter vector consists of the scale of the basis vector, translation or the location of the tail of the basis vector and angle of the basis vector with the x-axis. Articulated shapes were represented by its rigid components and the articulation points. The similarity between two shapes is computed as the Euclidean distance between the feature vectors. A K-D-B-tree structure [53] is used for indexing. This technique is quite slow and the object recognition is done manually.

Fan proposed a shape based retrieval technique in [15] based on distance histograms. A distance histogram contains the distribution of the radii lengths from sample boundary points to the centroid of a given object. The distance between two objects is calculated as the Euclidean distance between their corresponding distance histograms.The main drawback of this technique is that, it is capable of comparing shapes of single objects only.

The disadvantage of most of the shape based retrieval systems is that boundary based techniques are applicable only to "sketch-databases" i.e. databases with images that contain the sketch of a single object only. For using region based descriptors, obtaining a region is a major problem. So due to this inaccuracy of the region itself, the descriptors may become ineffective.

2.3 Texture features

Texture is a powerful regional descriptor that helps in the retrieval process. Texture, on its own does not have the capability of finding similar images, but it can be used to classify textured images from non-textured ones and then be combined with another visual attribute like colour to make the retrieval more effective. One of the popular representations of texture feature is the co-occurrence matrix proposed by Haralick *et al.* in [24] . The matrix is based on pixel orientation and inter-pixel distance. Meaningful statistics from the co-occurrence matrix are extracted and represented as texture information. Tamura *et al.* [66] proposed a method to extract six visual texture properties namely coarseness, contrast, directionality, likeliness, regularity and roughness. Since we do not use them, they are not reviewed any further.

2.4 Towards semantic features

Obtaining the semantics or the meaning of an image is one of the most current research topics in the area of image retrieval. Visual features alone are not enough to distinguish between images. For example, there might be two images - that of a blue sky and the other of a blue sea. Using colour, texture and other attributes they might be deemed similar, but semantically they are completely different. Of course, it cannot be denied that without the help of visual features, it is impossible to derive the semantics of an image, unless they are annotated manually. One of the most important factors in a semantic based retrieval system is to not just look at the image on the whole, but in fact, to look at the objects in the image and to try and find relationships between these objects. Partitioning or segmenting the image into regions may reveal the "true" objects within an image. Local properties of regions could help in understanding these objects, thus contributing to more meaningful image retrieval. For this purpose, it is important to partition the image into its constituent objects. There are several image retrieval systems that adopt a region-based approach.

Chua *et al.* [9] attempted to capture the semantics of images by using domain knowledge. They developed a model for knowledge based image retrieval where each query was modeled as a hierarchy of concepts and the leaf of this hierarchy was defined in terms of multiple image-content features such as text, colours and textures. Each concept comprised a name, its relationship with other concepts and rules for identification within the images' contents. They also developed a concept-based query language which included three operators namely COMPOSE, AND and OR.

In [8], an image is segmented to eliminate the unwanted background details. From each of the objects obtained, a set of colour-pairs is derived. The authors also define a similarity measure, which is normalized with respect to the size of the object.

Some papers such as [13] have proposed to represent colour-induced emotions or sensations such as warm-cold, light-dark, impressive-expressive using Itten's formalism. The images are segmented into regions with homogeneous colours and the intra-region properties e.g. colour, hue, luminance, saturation, warmth etc and inter-region properties such as contrast and temperature are represented using fuzzy sets.

Chunhui et al proposed a framework [10], *ifind*, that made use of the relevance feedback technique for retrieval of images. A semantic network is constructed from the image database by associating the keywords and images, with weights that describe the degree of relevance between them. They propose two new methods to find these keywords. One way is to ask the crawler that finds the images to also find the tag associated with the image e.g. name, keyword tag etc. Another way is when the user types keywords, retrieves images and gives a positive feedback, then these keywords are associated with the images.

In Blobworld [5], [6], objects are recognized by segmenting the image into regions that have roughly the same colour and texture. Each pixel is then associated with a vector that consists of colour, texture and spatial features. A model of the distribution of pixels is developed in an 8-D space. The distance between two images is calculated as the distance between the blobs in terms of colour and texture.

Netra [40] is another image retrieval system which segments images into regions of homogeneous colour and then uses the colour, texture, shape and spatial properties for measuring similarity. Both Blobworld and Netra require the user to select the region of interest from the segmented image and only this region is then compared with regions in other images in the database thus avoiding noise during the matching process. There are however some disadvantages of this method. The user is burdened with the task of selecting his object of interest, when in fact the segmentation may not have yielded regions close to the human perception of an object. The other problem is that humans often tend to associate objects with the background and other surrounding information to give it some meaning. So depending on the background where a particular object is present, users may perceive the same object differently.

An attempt towards capturing the semantics to help find similar images was made by Wang *et al.* in [68] and Stehling *et al.* in [61]. In Simplicity [68], the authors make use of semantics to classify images into the following categories: Textured vs Non-textured using the well known X^2 measure and Graphs vs Photographs using the probability density of wavelet coefficients in high frequency bands. They first segment the images by dividing the image into 4x4 blocks and then extract a feature vector consisting of six features (three of which are the average colour components and the other three indicate the energy in high frequency bands of wavelet transforms). Then a K-means algorithm [23] is used to cluster these feature vectors. While [68] makes use of the colour of each region to find similar images within a category of images, in [61] the colour and the spatial position of each region is used. The distance used by both [68] and [61] to compute the similarity between the images is the IRM measure proposed in [36]. The advantage of the IRM distance is that it is not affected by over or under segmentation because it considers all the regions in an image.

2.5 Image retrieval systems

Several image retrieval systems have been developed based on basic cues such as colour, spatial position, shape and texture. Systems that were developed until about five years ago, depended on captions assigned to images apart from the features mentioned above. Short descriptions about some of these systems are given below.

QBIC (Query by Image Content) [1] is a popular image retrieval system which uses two approaches: a localized approach to represent the colour features and a region-based approach. In the first approach, the image is divided into grids of size 6×8 or 9×12 cells. Each grid is represented by its average colour in Munsell colour space and the five most dominant colours and their frequencies. In the region-based approach, pixels are clustered based on their colour and spatial proximity and the total number of colours in the segmented image is reduced to a very small number. For each colour thus obtained, a minimum bounding rectangle is computed with all the pixels of the same colour. Each cluster has a rank with respect to all the other clusters in the image depending on how close a given cluster is to the other clusters. If two clusters have their mutual ranks less than a given threshold then they are merged. The disadvantage of this approach is that regions are restricted to equal sized rectangles. This approach also suffers from variances to rotation, scaling and translation.

Virage [2] makes use of four basic properties namely - global colour, local colour, structure and texture. The user is presented with four different layers of image abstraction in order to increase the flexibility of viewing the same image from different view points. They are domain objects and relations, domain events and relations, image objects and relations, image representations and relations. The user is also given the freedom to manipulate the weights of the visual features. This system depends heavily on relevance feedback.

PicSOM [35] developed by Laaksonen *et al.* uses the tree structured Self-Organizing Maps (SOMs) for retrieving images similar to a given query image. The user first selects a database of interest and the system then displays a random set of images on the browser. The user then selects a set of images that are relevant to what he is looking for. The system now assigns positive weights to the selected set of images and a negative weight to the others. With this additional information , the system recomputes distances and displays another set of images to the user. This process is iterated till the user finds the best set of relevant images. PicSOM uses several features such as the R, G, B values of five separate regions in the image and the texture properties of these regions.

Photobook [50], [51] is an interactive image retrieval system. The interactiveness of the image browsing is achieved by using a Motif interface. Text annotations are initially used to select a category e.g. cloth samples for curtains, mechanical tools etc. Photobook then improves the quality of retrieval by using some visual attributes to sort the images in a particular category by making use of shape and texture properties or a combination of them. Version six of Photobook allows for dynamic code loading that makes use of a user-defined matching code to compare two images. The system also uses relevance feedback with the help of a distinct interactive agent (Four Eyes) that has the ability to learn from user's selection.

Visualseek [58] and Webseek [59] are image retrieval systems that were built at Columbia University and were meant for purely educational purposes. Visualseek makes use of features such as colour, texture and spatial layout. Using a technique called binary feature set back-propagation on colour and texture, significant regions are extracted and an arbitrary spatial layout including the absolute location and the relative location is assigned to these regions. Spatial issues such as adjacency, overlap and encapsulation are handled by Visualseek. Queries are sketched by the user instead of using QBE. This system can handle complex queries owing to its efficient indexing structure which is based on binary trees. Webseek is a catalog-based engine and it was built for the World Wide Web. Webseek supports queries based on the catalog as well as queries using visual features. Initially a web spider collects many images from the web and then automatically indexes them based on the surrounding HTML tags.

Excalibur [17] is a commercial application development tool and is used by popular systems such as Yahoo! Image surfer that facilitates content-based image retrieval from the World Wide Web. Many image indexing and matching techniques are made use of by Excalibur Technologies.

MARS [27] is an image retrieval system that was developed at University of Illinois. This system uses a variety of features and several different similarity measures to compare images. The weights of these features are assigned by the user because the authors feel that this is important to reflect human similarity perceptions. Hence relevance feedback plays a major role in this system.

CBIRD (Content-Based Image Retrieval from Digital libraries) [38] is an image retrieval system that combines automatically generated keywords and several visual features such as color, texture, shape and feature localization to index both images and videos in the web. This system uses color channel normalization for finding similar images present in different illumination conditions. They also present a technique to search by object model.

Informedia [67] is a very interesting system developed by Wactlar *et al.* to perform video retrieval by using speech and image processing. The features that are extracted from each video scene include colour histograms, motion vectors, speech and audio tracks. They are then indexed based on several of these features. Some examples include objects in images, significant words from audio tracks, captions etc. Since the system uses so many features from videos, it is ideally suited for applications such as retrieval from news and movie archives.

2.6 Conclusion

In this chapter, we discussed several image retrieval systems focusing on some issues like image abstraction, querying and similarity measures. Research work based on primitive features such as colour, shape, spatial position and texture was presented. Some important image retrieval systems such as QBIC, Virage, Webseek etc. were also highlighted. We then detailed a number of image retrieval systems that attempted to include some kind of semantics into their retrieval system, especially focusing on Bolbworld, Netra, Simplicity and CBC thus, providing an insight into the problem of region-based image retrieval. Each retrieval technique discussed in this chapter was tested by the respective authors on a different database with a different query image (and relevant) set. Therefore, unless they are all implemented and tested on a common database with the same query image set, the precision-recall curves corresponding to these techniques cannot be compared directly.

Chapter 3

The proposed segmentation-based CBIR algorithm: SNL

3.1 Motivation

The proposed algorithm, SNL [1] is a segmentation-based CBIR technique that utilizes a more effective representation of image regions and a more accurate image similarity/distance calculation. SNL attempts to capture the properties of objects within an image in a way that is closer to our perception, thus generating a more meaningful association (distance) measure between images. As a result, SNL provides a more effective CBIR.

With all the techniques mentioned in Section 2.4, the segmentation results obtained using a single set of parameters on thousands of images may not always correspond to human perception of objects. This is illustrated with an example shown in Figure 3.1, where we see a query image A and two database images, B and C. On careful inspection of the segmented images, we notice that all three contain "something" at the center, surrounded by a green background. Also we notice that the "things" presented in images A and C have a lot of colours in common including black, white and some gray patches. Based on this information, most retrieval techniques would deem images A and C to be more similar than images A and B. However, when one looks at the actual images shown in Figure 3.2, we see that A and B are both images of tigers and are certainly more similar than A and C.

[1] The name SNL stands for the last names of the inventors of the technique, i.e., Sridhar, Nascimento and Li

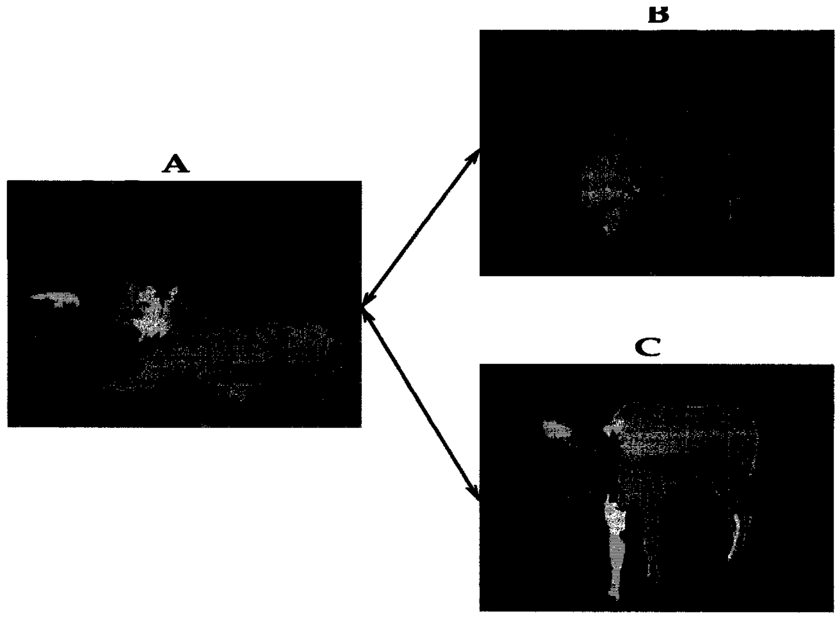


Figure 3.1: Motivation for SNL (segmented images)

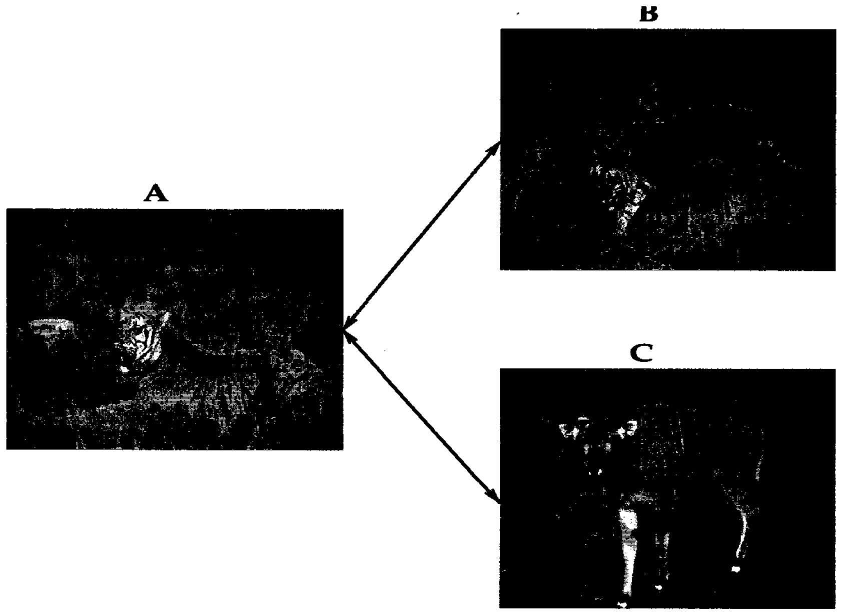


Figure 3.2: Motivation for SNL (original images)

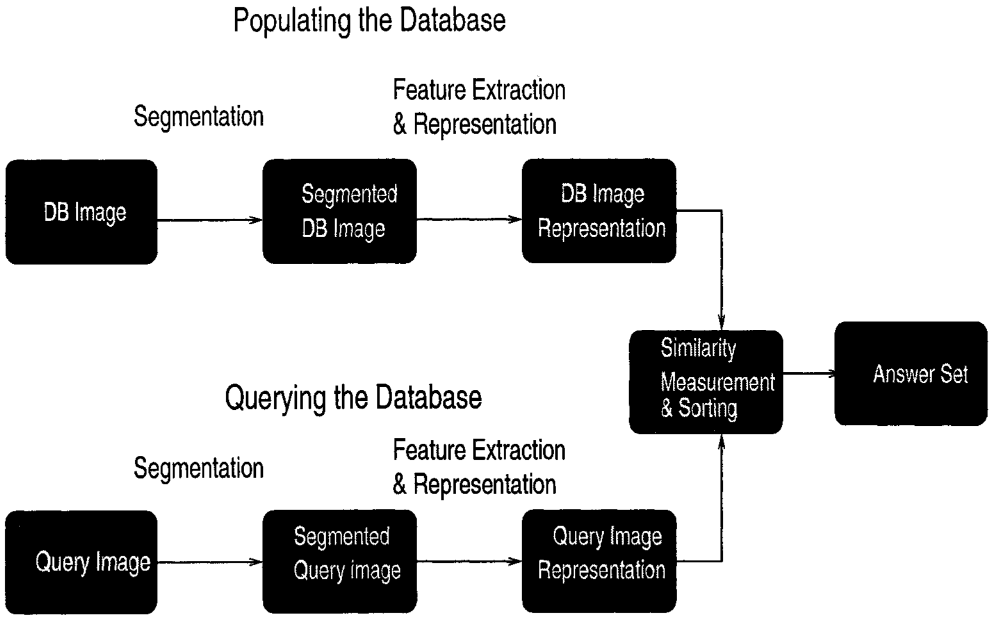


Figure 3.3: Architecture of the SNL technique

It is clear that a correct image segmentation is not enough. An accurate representation of the image regions is important, even critical, in measuring image similarity. To improve along this line, the proposed SNL technique contains three parts: image segmentation, feature extraction and similarity calculation.

3.2 The SNL architecture

The architecture of the SNL technique is shown in Figure 3.3. Any CBIR system consists of two phases. In the first phase, the database is populated and this process is done offline. Images in the database are initially segmented. From these segmented images, features are extracted and stored in a meta-data database. Querying the database is the second phase. Whenever a query image comes in, it is segmented and its features are extracted and stored. The features of the query image are compared with features of all the images in the database using a similarity measure. Thus, the three important processes involved in our retrieval system are segmentation, feature extraction and representation and calculation of the similarity measure. In the next few sections, we shall step through each of these processes for the SNL technique.

3.3 Segmentation

The first step in our retrieval technique is to segment the image into regions that (ideally) would correspond to the objects present in the image. For this purpose, we need a segmentation algorithm that is effective in rendering homogeneous regions in a short time. We investigated three different segmentation algorithms namely K-means [41], a segmentation method proposed by Comaniciu *et al.* [12] and a clustering technique proposed recently by Stehling *et al.* [61].

K-means is one of the most popular partitional clustering methods and its implementation is very simple and straightforward. It works by randomly initializing the mean value of K clusters and then calculating the difference between each pixel and the mean of each cluster. This calculation decides the cluster to which a particular pixel belongs to. Then the means are re-calculated and this process is iterated. Despite the fact that K-means is computationally simple and takes little time, the number of segments is an input parameter to the algorithm (which is a clear disadvantage). We wanted an automatic clustering algorithm that could decide on the number of clusters based on the content of the image. Hence, K-means did not suit our requirements.

Another segmentation approach was proposed by Comaniciu *et al.* [12] based on the mean-shift algorithm [2]. For each pixel in the image, a feature space is constructed based on its neighbouring eight pixels. Then a feature pallet is constructed with the most significant colours in the image and based on these colours, homogeneous regions are determined. Post-processing is done by eliminating very small sized regions. The segmentation process is completely automatic, however it is time consuming (ten seconds for a 512 x 512 image using a Celeron 533MHz processor).

Recently a paper was published by Stehling *et al.* [61] which presents a single-link region growing algorithm used along with a minimum spanning tree. The algorithm can be described as follows. The image is first converted into a graph whose vertices are the pixels in the image and whose edges are neighbourhoods of four pixels. The weight of each edge is the Euclidean distance between the colours

[2] `http://www.caip.rutgers.edu/riul/research/code.html`

of the four-pixel neighbourhoods. The pixels are clustered using two thresholds: colour threshold and size threshold. A set of connected pixels whose colour similarity is greater than or equal to the colour threshold forms a region. Then, regions less than the given size threshold are considered to be noise and hence merged with the nearest neighbour having the greatest similarity in terms of colour. The clustering algorithm proposed here is not only automatic, but also uses spatial and colour features and takes less time (four seconds for a 512 x 512 image). Hence we decided to use this clustering algorithm to obtain regions in the image.

3.4 Feature extraction and representation

The next phase is the regional feature extraction phase, wherein the segmented images are analyzed and a feature vector is constructed for each region. Spatial position, shape, colour and size of the regions are included in the feature vector.

3.4.1 Extracting colour

One of the most effective features that helps in distinguishing one image from another is colour. As mentioned before, the problem with any segmentation/clustering algorithm is that a single set of parameters cannot be applied to all the images in the database, especially when considering a heterogeneous collection. Even within an image, it would make more sense if some objects had a more detailed representation than others. The segmentation algorithms mentioned before, cluster pixels on the basis of the most significant colours present in the image and tend to ignore or merge smaller segments with the larger ones closest to them, either in terms of colour or spatial location or some other property. It is definitely true that significant colours help in identifying similarity between images, but they also lead to a lot of false positives. For instance, a yellow sunflower, yellow sun and a yellow ball (of the same size) would all be segmented into roughly circular regions with the dominant colour which is yellow. In terms of the mean colour of the region, size and shape they would all be deemed very similar. But semantically they are not similar at all. In fact, the subtle difference between them can be brought out by the less dominant colours in the region, e.g. the black center in the sunflower and the orange tinge in the sun. Thus, from the above discussion we can infer that while the dominant colours help in finding regions that are similar to each other, less dominant colours help in eliminating false positives. For this reason, we decided to represent the colour feature of each segment with its histogram which gives us the distribution of colours in that region. Thus, for each region i in the image I, we have a colour histogram representation, $C(i)$.

Colours are represented using a specific colour model. Colour models help in expressing the colours in some standard, accepted format [19], [4]. In this work, we observe that there is a change in retrieval performance with change in the colour models. We considered the two most popular colour models namely, RGB and HSV [4].

The RGB colour model [4] can be represented in the form of a cube with the primary colours red, blue and green occupying the three corners of the cube. The gray-scale values from black to white are present along the diagonal of the cube from the origin. Any other colour is expressed as a combination of the primary colours. The RGB model is used in colour CRT monitors, thus this is a hardware-oriented model. The disadvantage of the RGB colour model is that the space is not perceptually uniform and equal importance has to be given to all the three components during quantization.

The HSV colour model [4] is more intuitive and can be visualized as a hexacone or a six-sided pyramid. Value (V) is 1 at the top of the hexacone and corresponds to the relatively bright colours. Hue (H) is represented along the perimeter of the base of the hexacone with red at 0 degree, green at 120 degrees and so on. It ranges from 0 to 360 degrees. In the HSV colour model, complementary colours are 180 degrees apart from one another. S represents the saturation. It is measured from the center of the hexacone to the corners and ranges from 0 to 1. The main advantage with this colour space is that it can be quantized easily [31] and can also be directly translated from the RGB colour space. Algorithm 1 shows how to convert the RGB colour space into the HSV colour space. Details about the quantization schemes used in the RGB and the HSV colour spaces are discussed later.

```
Input  : $r(0,1)$, $g(0,1)$ and $b(0,1)$
Output : $h(0,360)$, $s(0,1)$ and $v(0,1)$
$min = MIN(r,g,b)$;
$max = MAX(r,g,b)$;
$v = max$ // v;
$delta = max - min$;
if $max \neq 0$ then
    $s = delta/max$ // s;
endif
else
    $s = 0$ // s = 0, v is undefined;
    $h = -1$;
endif
if $s \neq 0$ then
    if $r = max$ then
        $h = (b - g)/delta$;
    endif
    else
        if $g = max$ then
            $h = 2 + (b - r)/delta$;
        endif
        else
            $h = 4 + (r - g)/delta$;
        endif
    endif
endif
$h* = 60$ // h in degrees;
if $h < 0$ then
    $h+ = 360$;
endif
Return $h$, $s$ and $v$;
```

Algorithm 1: Conversion of RGB to HSV colour space

3.4.2 Extracting other features

Apart from colour, we also extract other features from regions in the image. In order to extract the shape of each region, we compute the ratio between the height and the width of the Minimum Bounding Rectangle (MBR) of each region. Eccentricity has been used before e.g. [1], [43] and is easy to calculate. The shape representation in SNL is similar to eccentricity but is sensitive to changes in rotation.

The spatial position of each region is denoted by the center of gravity of each region. The x and the y coordinates ($X(i)$ and $Y(i)$) of the center position are normalized by the image coordinates.

The size of each region is the total number of pixels in the region. This is then normalized by the original image size. The size of each region i in the image I, is thus, a value between 0 and 1 and is expressed as $A(i)$.

3.5 The similarity/distance measure

Next to image representation, similarity measure is one of the key items in the process of image retrieval that decides the effectiveness and the efficiency of the retrieval technique.

There are basically two kinds of similarity measures: pre-semantic and semantic [4]. Pre-semantic does not involve any kind of interpretation and is simply based on the features obtained from the images and is computed at an earlier stage in the visual pathways. Semantic similarity measures are features that belong to images that have been interpreted and hence are more meaningful. Though we do not necessarily interpret the image precisely, we do attempt to analyze the objects inside the image. Our similarity measure therefore lies between the two.

In the case of retrieval using regions of an image, it is important to choose a similarity measure that is robust to segmentation inaccuracies. It is also important that the measure agrees with the human perception of similarity and is easily computable. Since images have been decomposed into their respective segments, the similarity between two images is in fact the similarity between their constituent segments. As mentioned in the previous section, each region is represented by its spatial location, shape, colour and size. Hence, to compare two regions their respective features should be compared.

The distance between the spatial positions of two regions, i of image $I1$ and j of image $I2$, is calculated as the Euclidean distance between the centers of the two regions. This is shown below:

$$D_S(i,j) = \sqrt{(X(i) - X(j))^2 + (Y(i) - Y(j))^2}$$

where $X(i)$ and $Y(i)$ are the x and y coordinates of the centers of the regions.

The shape distance is the distance between the height/width ratios of the MBRs enclosing two regions i and j of images $I1$ and $I2$ respectively. It can be computed as:

$$D_E(i,j) = |e(i) - e_{(j)})|$$

where $e(i)$, $e(j)$ are the height/width ratios.

So far, we measured the distance between two regions in terms of shape which is a boundary feature and in terms of the spatial position. The distance between two regions in terms of their content i.e., colour and size, can be calculated using the following equation.

$$D_C(i,j) = \frac{\sum_{k=0}^{k=N} |C(i)[k] - C(j)[k]|}{\sum_{k=0}^{k=N} C(i)[k] + \sum_{k=0}^{k=N} C(j)[k]}$$

where $C(i)$, $C(j)$ are the colour histograms of regions i of $I1$ and j of $I2$ containing N bins each.

The rationale behind using the above measure is explained using an example. Consider the two sets A and B shown in Figure 3.4. Individual shapes present within these regions are matched, squares with squares, circles with circles etc. In A and B a total of 8 objects have been matched. The number of unmatched objects can be used as the distance between sets A and B. This value is then normalized by the total number of objects present in both these regions. Thus, the distance between A and B can be expressed as:

$$D_C(A,B) = \frac{\#(unmatched)}{\#(\mathrm{A}) + \#(\mathrm{B})} = \frac{|2-5| + |3-1| + |1-4|}{6+10} \qquad (3.1)$$

where $\#(x)$ is the cardinality of x and $unmatched$ is the set of objects that are not present in both A and B. We can rewrite the numerator as $\#(unmatched) =$

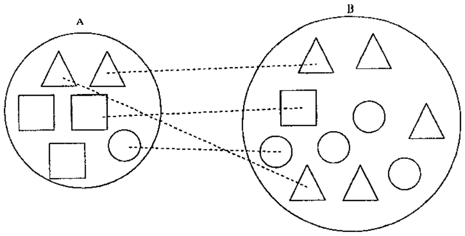


Figure 3.4: An example to demonstrate the distance measure

$\#(A \bigcup B) - \#(A \bigcap B) = \#(A) + \#(B) - 2 * \#(A \bigcap B)$. Therefore, Equation 3.1 can be written as

$$D_C(A, B) = \frac{\#(unmatched)}{\#(\text{A}) + \#(\text{B})} = \frac{\#(\text{A}) + \#(\text{B}) - 2 * \#(\text{A} \bigcap \text{B})}{\#(\text{A}) + \#(\text{B})} = 1 - \frac{2 * \#(A \bigcap B)}{\#(A) + \#(B)} \tag{3.2}$$

In Equation 3.2, $\frac{2*\#(A \bigcap B)}{\#(A)+\#(B)}$ happens to be the Dice's coefficient [52]. Also this distance that we are measuring is a metric since it follows the three metric rules namely non-negativity, symmetry and triangle inequality [52].

Thus, we have separately measured the distance between the spatial position, shape and the content of regions. But to differentiate between the regions, we need a single overall measure, which can be obtained by combining these three distances. Distance between two regions i and j of images $I1$ and $I2$ is defined as:

$$D(i, j) = \alpha \times D_C(i, j) + \beta \times D_S(i, j) + \gamma \times D_E(i, j)$$

where D_C is the distance between the region content and α is the weight assigned to it, D_S is the spatial distance with its corresponding weight β and D_E is the shape distance between two regions with weight γ.

In order to measure the similarity between two images, the IRM proposed in [68] is used. The IRM measure to calculate the distance $D_I(I1, I2)$, between two images $I1$ with m regions and $I2$ with n regions is calculated as shown in Algorithm 2. The main idea behind the IRM measure is to match images completely. The inter-region distances between all pairs of regions in the two images are computed. The two most similar regions (least inter-region distance) are completely matched, if the regions have the same size, otherwise a partial match occurs and the unmatched portion is matched with some other region. This process is repeated until all the regions are matched completely.

```
Input   : Region features of $I1$ and $I2$
Output  : Distance $D_I(I1, I2)$
for each pair of regions, $i$ in $I1$ and $j$ in $I2$ do
    Calculated the distance $D(i, j)$;
endfor
Sort all distances in the ascending order;
Mark all regions as "not-done";
for all $m \times n$ pairs of regions in $I1$ and $I2$ do
    Pick the pair of regions $i, j$ with the lowest distance between them;
    if both are marked "not-done" then
        $D_I(I1, I2) += D(i, j) \times min(A(i), A(j))$ ;
        $A(i) -= min(A(i), A(j))$ ;
        $A(j) -= min(A(i), A(j))$ ;
        Mark the region with the minimum size as "done";
    endif
endfor
Return $D_I(I1, I2)$;
```

Algorithm 2: IRM: Calculating the similarity between two images $I1$ and $I2$

The process of calculating the IRM measure requires quadratic time since we need to compare all segments of image $I1$ with all segments of image $I2$. In our case, however, due to our configuration, we obtain only a few regions (5 regions on an average, for colour threshold = 3, size threshold = 1 in the segmentation algorithm) as opposed to CBC (40 regions on an average, for colour threshold = 3, size threshold = 0.1 in the segmentation algorithm) [3]. We therefore can afford to use this measure.

For every query image, the extracted regional features are compared with the meta-data of all the images in the database using the distance formulae and then using the IRM measure, the image similarities are computed. After obtaining the similarities, the database images are re-ranked in the order of decreasing similarity (or increasing distance).

[3] Details about the segmentation parameters are given in Section 3.7.5.

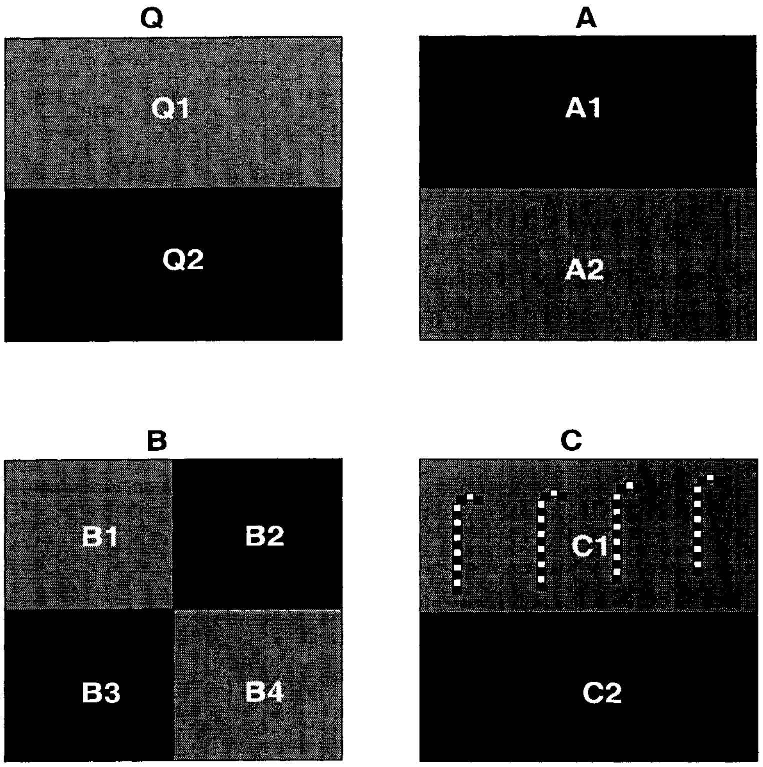


Figure 3.5: Sample image set

3.6 Discussion with an illustration

In this section, we differentiate our approach from two other approaches namely GCH (Global Colour Histograms) and CBC proposed by Stehling *et al.* [61] using four example images. For simplicity let us assume that our colour palette consists of only three colours: black, gray and white.

In the first case, we illustrate the fact that SNL is capable of perceiving changes such as rotation and in the second case, we point out the importance of using a histogram representation for the colour property of a region. Consider Figure 3.5. In this example, we compare image Q with images A, B and C. We know that image A is a rotated version of image Q and is assumed to be more similar to Q than B. It is also clear that image C is not the same as image Q because C contains some "candy canes". Therefore, if the human perception of distance between two images i, j is termed as $H(i, j)$, then the assumptions we made earlier are $H(Q, A) < H(Q, B)$ and $H(Q, C) \sim 0$ but $\neq 0$.

The distance between image Q and the other three images, A, B and C are

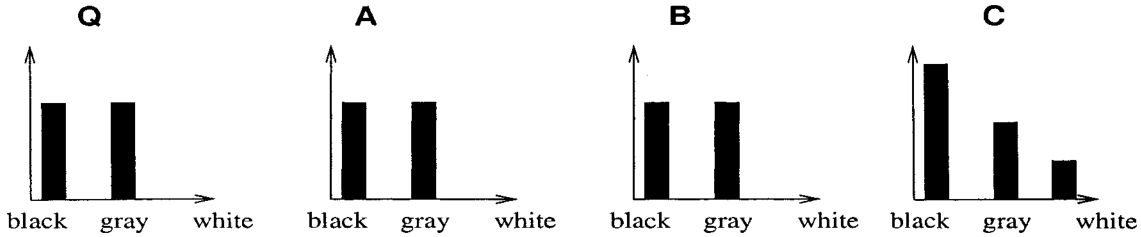


Figure 3.6: Image distance computed using the GCH technique

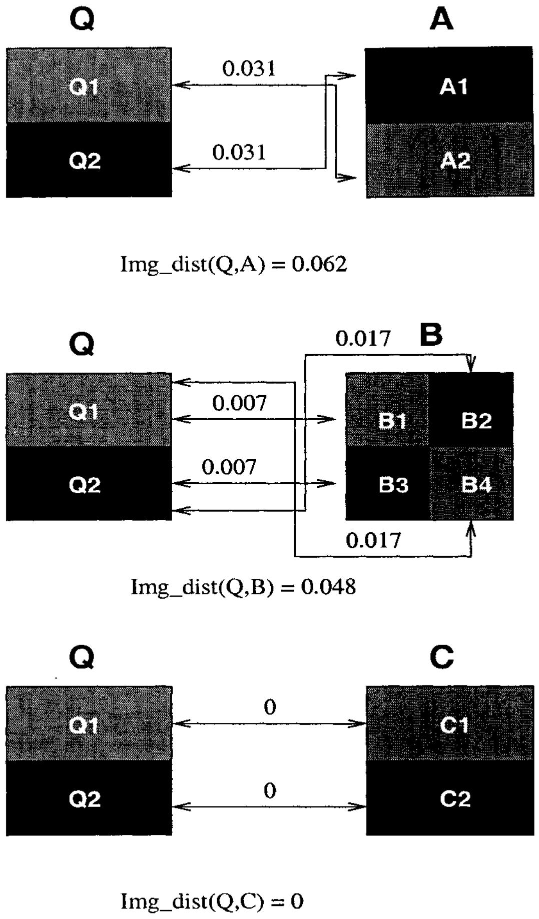
calculated using the above mentioned techniques and are shown in Table 3.1.

Table 3.1: Distance calculation using three techniques ($\alpha = 0.7, \beta = 0.15, \gamma = 0.15$)

Techniques	$D_I(Q,A)$	$D_I(Q,B)$	$D_I(Q,C)$
GCH	0	0	0.2
CBC	0.062	0.048	0
SNL	0.075	0.308	0.07

When GCH is applied, $D_I(Q,A) = D_I(Q,B) = 0$ because the colour composition of Q, A and B are the same as shown in Figure 3.6. Due to difference in colour composition, the distance between Q and C determined by the GCH technique is much greater than 0. Thus, GCH does not agree with both our assumptions on human perception of similarity. From this particular case, we can deduce that colour composition is important, but it is not enough to differentiate between images where the spatial distribution of colours is different.

For applying SNL and CBC, images need to be segmented. Q and A are segmented into two regions each, $Q1$, $Q2$ and $A1$, $A2$ and B is segmented into four regions $B1$, $B2$, $B3$ and $B4$. In C, the smaller regions constituting the "candy canes" are merged with region $C1$ to form a single region with the average colour, gray. The second region is $C2$. When CBC technique is applied, $D_I(Q,A) \neq 0$ and $D_I(Q,B) \neq 0$ as can be seen in Figure 3.7. The reason is that the matching technique takes into account the colour and also the spatial location of the region as mentioned before in Section 2.4. However, since they do not consider the shape properties of the regions, $D_I(Q,A) > D_I(Q,B)$ as seen from Table 3.1. Also, $D_I(Q,C) = 0$. This is because, during segmentation the small regions inside $C1$ were merged with it and the average colour was represented. It is quite contrary to



Img_dist(Q,A) = 0.062

Img_dist(Q,B) = 0.048

Img_dist(Q,C) = 0

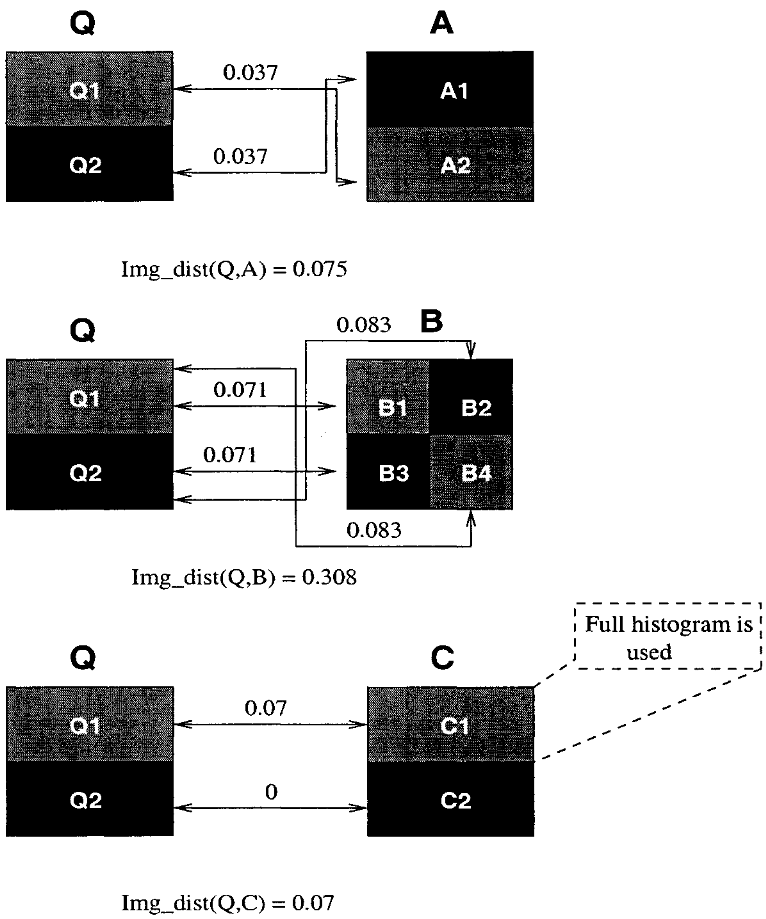
Figure 3.7: Distance between images using CBC

what human beings would likely perceive.

SNL determines the distance between Q and A to be smaller than the distance between Q and B (see Table 3.1) since SNL uses the colour, size, spatial location and the shape of each region. SNL is also capable of distinguishing Q from C despite the disadvantage of the segmentation process, as shown in Figure 3.8. SNL satisfies both the assumptions made earlier and is therefore better suited to represent human perception of similarity. Thus, using some example figures we have illustrated that we combine the advantages of GCH and CBC to make our technique more similar to our perception.

In Figure 3.9 [4], we give three examples of real life images. Images A and B are more related (they are both pictures of tigers) than A and C. Figure 3.9 also

[4] http://jxw.stanford.edu/cgi-bin/zwang/regionsearch_show.cgi/



Img_dist(Q,A) = 0.075

Img_dist(Q,B) = 0.308

Img_dist(Q,C) = 0.07

Figure 3.8: Distance between images using SNL

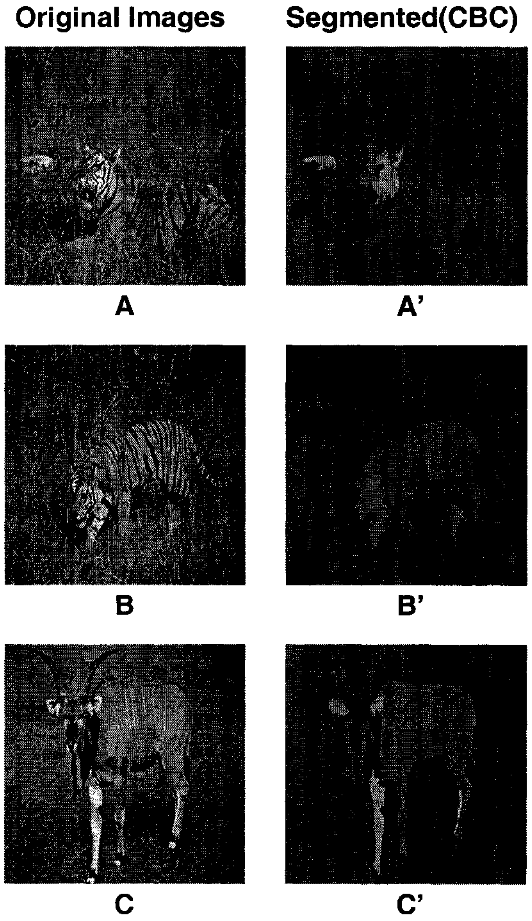


Figure 3.9: Examples of segmentation inaccuracies

shows the segmented images A', B' and C'. In this example, we shall compare the distance between image A and the other two images calculated using GCH, CBC and SNL as shown in Table 3.2. Here again, we know that human beings would perceive images A and B to be more similar than images A and C, since A and B are images of tigers in a similar background. The assumption made here can be stated as $H(A,C) > H(A,B)$.

Table 3.2: Distance calculation using three techniques ($\alpha = 0.7$, $\beta = 0.15$, $\gamma = 0.15$)

Techniques	$D_I(A,B)$	$D_I(A,C)$
GCH	0.316	0.220
CBC	0.043	0.038
SNL	0.137	0.150

While GCH and CBC determine images A and C to be more similar than A and B, SNL deems B to be more similar to A than C, thus agreeing with our assumption on human notion of similarity. Thus, we see how false positives can be avoided by SNL.

3.7 Experiments and results

3.7.1 Experimental setup

All the experiments in this thesis were performed on a large heterogeneous database containing $50,000$ images obtained by combining two commercially available collections: *PrintArtist Platinum* by Sierra Home, and *Master Photos 50,000 Premium Photo Collection* by COREL. For all the experiments, we have used about 15 query images [5]. The query images along with their relevant set of images were obtained from the *COREL Gallery Magic 65,000*. The relevant set size varied from 6 to 24. The relevant images resembled each other in colour distribution and semantics [7]. Query images constituted of a set of Bonzai Trees, Halloween Pumpkins, Beach Scenes etc.

In this section, we discuss about the evaluation measures used and the experiments performed. Three sets of experiments were conducted to observe and mea-

[5] `http://www.cs.ualberta.ca/~mn/CBIRone/`

sure the performance of the proposed retrieval technique. The first experiment relates to the quantization scheme to be applied to the RGB and HSV colour space. In the second set of experiments, weights to be assigned to the content, spatial and shape features of each region are determined. The third set of experiments, presents the performance of SNL technique in comparison with the GCH and the CBC technique proposed by Stehling *et al.* [61].

3.7.2 Evaluation measure

The most popular way to evaluate the performance of a retrieval system is to calculate the percentage of relevant documents retrieved and also their relative order. Ideally, a system should retrieve all the relevant documents first, keeping the number of non-relevant documents that are retrieved before the relevant ones as minimum as possible. Recall [70] is the percentage of the total relevant documents retrieved and is defined as:

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}} = \frac{|\text{Ra}|}{|\text{R}|}$$

Precision refers to the capability of the system to retrieve only the relevant documents. Precision can be expressed as:

$$\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}} = \frac{|\text{Ra}|}{|\text{A}|}$$

Figure 3.10 shows the diagrammatic representation of precision and recall.

The precision and recall values are normalized to lie between $[0, 1]$. Ideally both precision and recall should be closer to 1. A set of recall and precision curves are joined together and this curve is called the precision-recall curve. Since there are 15 query images, we generate the average of the precision-recall curves from them. In this curve, recall is inversely proportional to precision. For the sake of comparison, we prefer a non-increasing precision-recall curve and hence we use an interpolated precision-recall curve as our retrieval measure.

3.7.3 Quantization schemes

The first experiment was done to select a good quantization scheme for both the RGB and the HSV colour space. We used about 10,000 images to test the perfor-

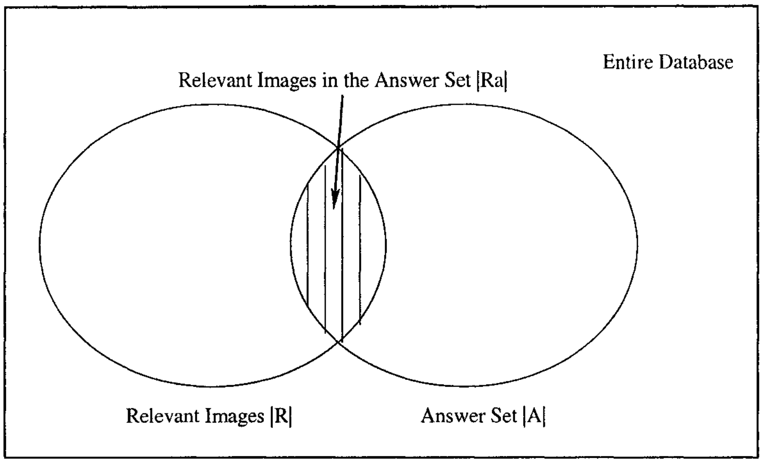


Figure 3.10: Precision an recall (adapted from [72])

mance of the colour spaces, RGB and HSV, for various quantization levels . The colour property of each region in an image is represented with a histogram in the above mentioned colour spaces. Uniform quantization is applied to each region's histogram consisting of 27 (3 × 3 × 3), 64 (4 × 4 × 4) and 125 (5 × 5 × 5) bins for RGB and 81 (9 × 3 × 3), 135 (15 × 3 × 3) and 162 (18 × 3 × 3) bins for HSV respectively and their performance was observed.

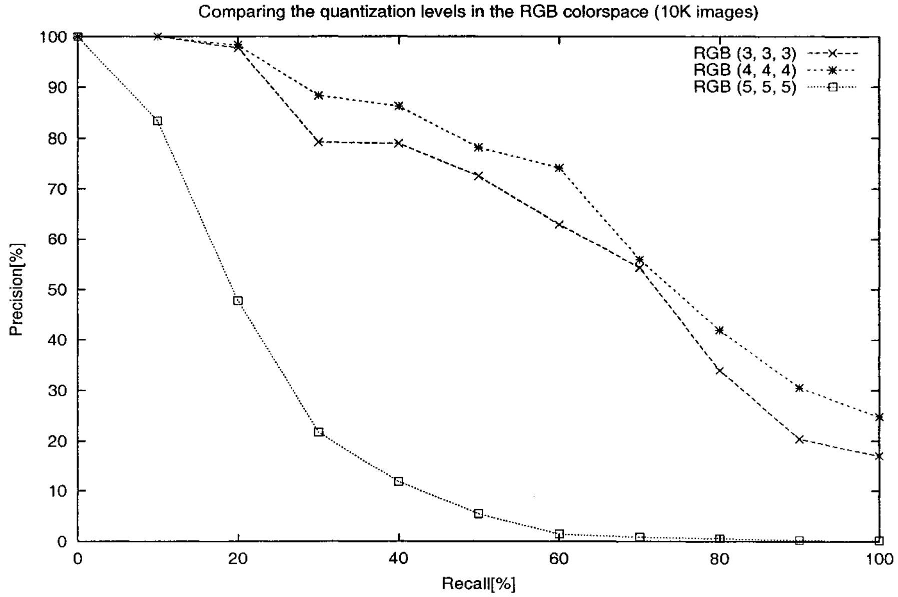


Figure 3.11: Performance variation with various quantization levels in the RGB space

In Figure 3.11, it is seen that the performance of the 64 colour quantization is

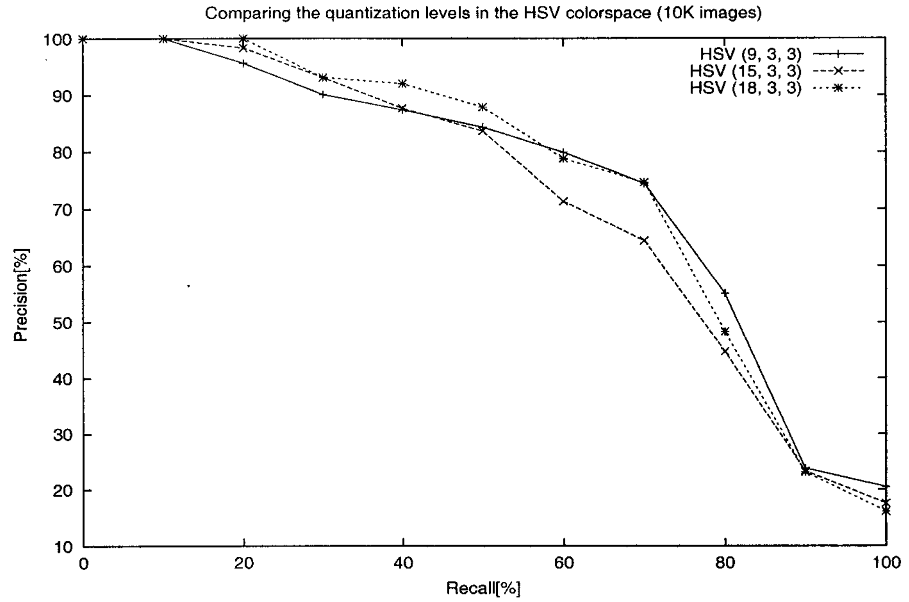


Figure 3.12: Performance variation with various quantization levels in the HSV space

the best and the curve is drastically pulled down by an increase in the quantization space. This is because two colours which are very similar to each other can be classified into two different bins and since only a one-to-one difference between the bins is calculated, the distance between two similar colours is increased. Decreasing the number of bins also affects the performance because with just 27 bins the separability between colours is reduced. The performance is not affected as much due to the fact that the regions obtained from segmentation are homogeneous in colour to some extent and 27 colours are sufficient to represent the colours within such a homogeneous region. Since the 64 colour quantization scheme in the RGB colour space was the best, we adopt the same for all our future experiments. We also observe in Figure 3.12 that the performance of the HSV colour space does not vary as much with change in quantization levels. We selected the 81 colour quantization scheme not only because it performed well but also because the storage overhead was considerably low when compared to the other two schemes using 135 and 162 colours.

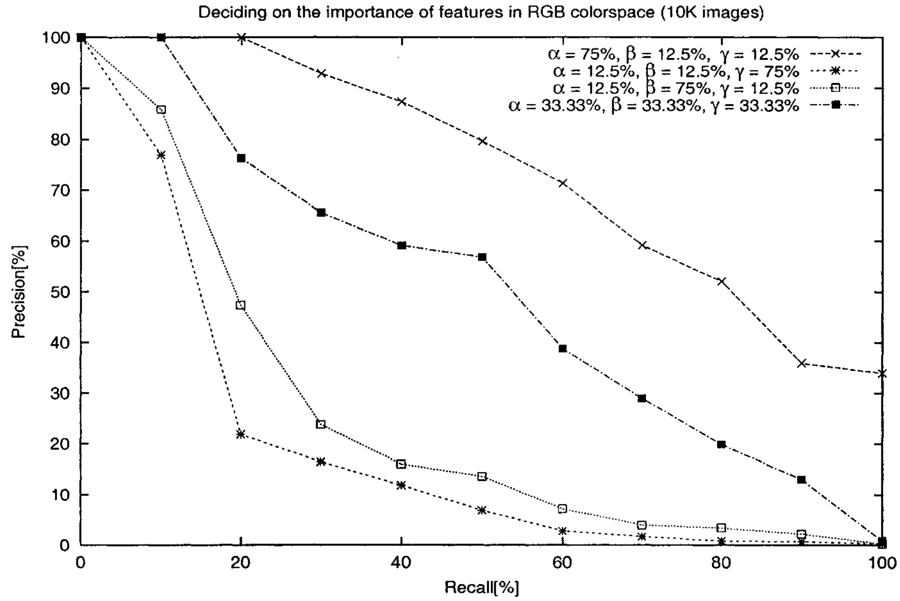


Figure 3.13: Performance variation with varying importance to different features in the RGB space

3.7.4 Assigning weights to regional features

In Section 3.5, we discussed about calculating the distance between two regions. This distance is a weighted sum of the region content distance, shape distance and the spatial distance between any two regions. The second experiment was done to decide on the values to be assigned to α, β and γ. Again a set of 10,000 images was considered and the importance of each of these three features was studied by assigning different values for α, β and γ. In Figures 3.13 and 3.14, we observe that colour is clearly the most important feature that affects the retrieval performance. Shape and spatial properties do not account for the performance very much. Thus, we know that the value of α has to be higher than both β and γ. To further refine these weights, we decided to consider a few sample points to calculate the average precision for all recall values in a database of 10,000 images. The graph corresponding to this experiment (Figure 3.15) indicated that an α value of 0.7 yielded a very good result in terms of effectiveness. Since both the spatial feature distance and shape feature distance were almost equally unimportant (Figures 3.13 and 3.14) β and γ were assigned a value of 0.15 each. An important thing to note here is that, in both the HSV and the RGB colour space, consistent observations related to the

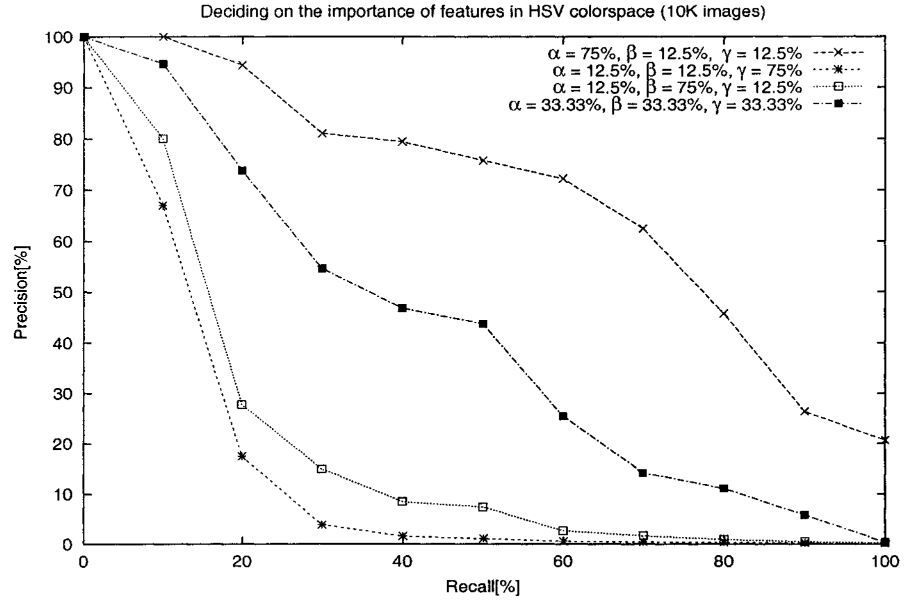


Figure 3.14: Performance variation with varying importance to different features in the HSV space

weights to be assigned to each feature were made.

3.7.5 Comparison with existing approaches

The third experiment compared SNL with the CBC technique [61] and GCH. CBC was proposed recently and claimed to perform better than CCV [49] and Colour Moments [65].

The colour and size thresholds in the segmentation step were set to be 3 and 0.1 respectively. The colour threshold determines the similarity of pixels within a given cluster or in other words the homogeneity of the cluster in terms of colour. The size threshold decides how small the cluster can be (minimum size of a cluster). In [61], the authors suggest that this set of parameters (3 and 0.1) result in a good compromise between the number of regions, effectiveness and robustness. The small size threshold in CBC results in many details in the form of small regions. As the size threshold is increased, fewer regions are obtained. We felt that for a region to be more meaningful, its size needed to be at least 1% of the total image size. Hence in the case of SNL, we set the colour and the size thresholds to be 3 and 1 respectively. Moreover, since SNL is robust to segmentation inaccuracies a

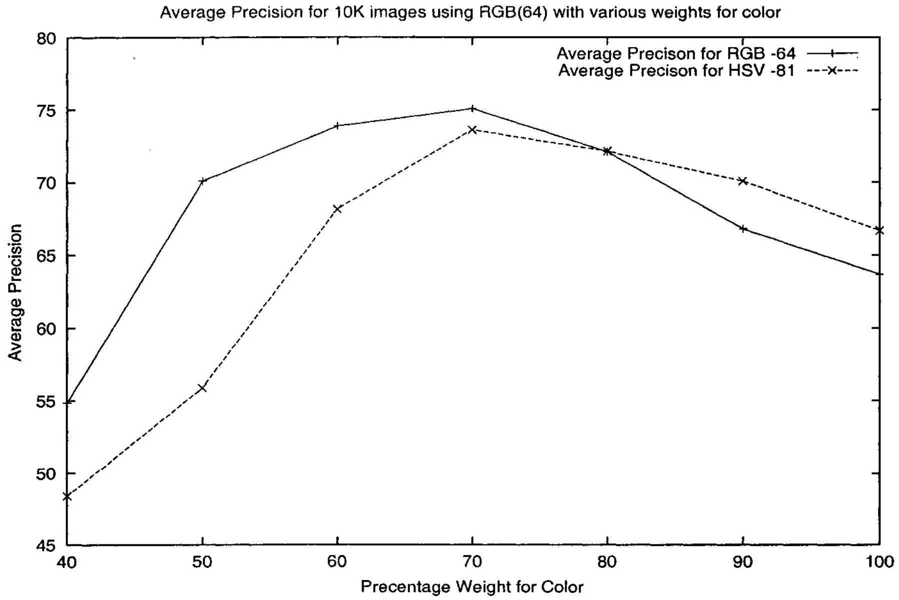


Figure 3.15: The average precision at varying colour weights for 10K images

high size threshold does not affect the results and in fact leads to a smaller number of regions that need to be compared during query time. We also compared our technique with the GCH using the HSV colour space. Since it is important to see how well our technique scales up, we experimented with sets of 10,000, 20,000 and 50,000 images.

Figure 3.16 indicates the performance of the three techniques in a database containing 10,000 images. The SNL technique performs better than both the GCH and CBC. When scaled up to a database containing 20,000 images, we can clearly observe from Figure 3.17 that while the curves of CBC and GCH drop, the curve corresponding to the SNL technique is quite stable.

In the experiments using 50,000 images, the robustness of SNL becomes more evident as shown in Figure 3.18. The curves of GCH and CBC drop down even further as compared to a small drop in the curve corresponding to our technique. We also notice that the SNL technique, in both the RGB and the HSV colour space, starts dropping after 60% recall. The reason for this drop is the fact that the SNL technique is currently tuned with a large emphasis on the colour of each region. For example, consider a query image of a blue car (one of the 15 query images). The relevant set corresponding to this image contains some cars that are red in colour.

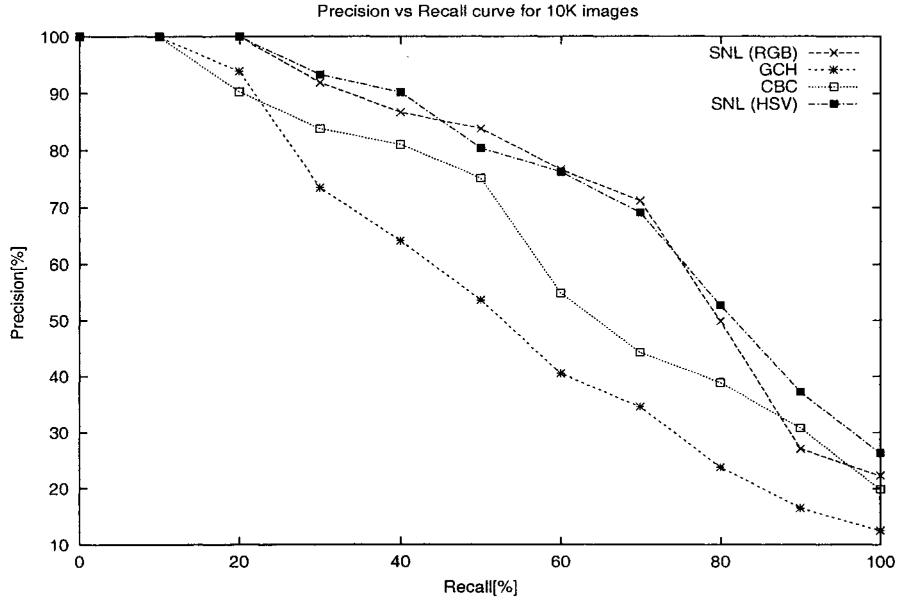


Figure 3.16: Comparing different techniques using a database of 10K images

SNL retrieves the blue cars very quickly but retrieves these red cars at the very end thereby decreasing the precision.

Thus we infer that the SNL technique in the RGB as well as the HSV colour space scales up well. The performance of the SNL technique in the previous three graphs also indicates that it is able to handle false positives well. As the database size increases, the number of false positives also increase proportionally. Since the performance of the SNL technique did not decrease with the increase in false positives, SNL is a better technique when compared to CBC and GCH.

3.8 Storage requirements

Storage space is an important measure for the efficiency of a technique. Even though it is no longer as critical an issue as it used to be about ten years ago, it is nevertheless not negligible. The storage requirements of GCH, CBC and SNL are listed in Table 3.3. In GCH, each image uses about 81 integers (assuming a 81 bin uniform quantization in the HSV colour space). Thus it requires only about 162 bytes (assuming two bytes per integer). In the case of CBC, an average of 40 regions are obtained and each region stores three float values for colour, two float values for the spatial position and one float value for the size of the region and

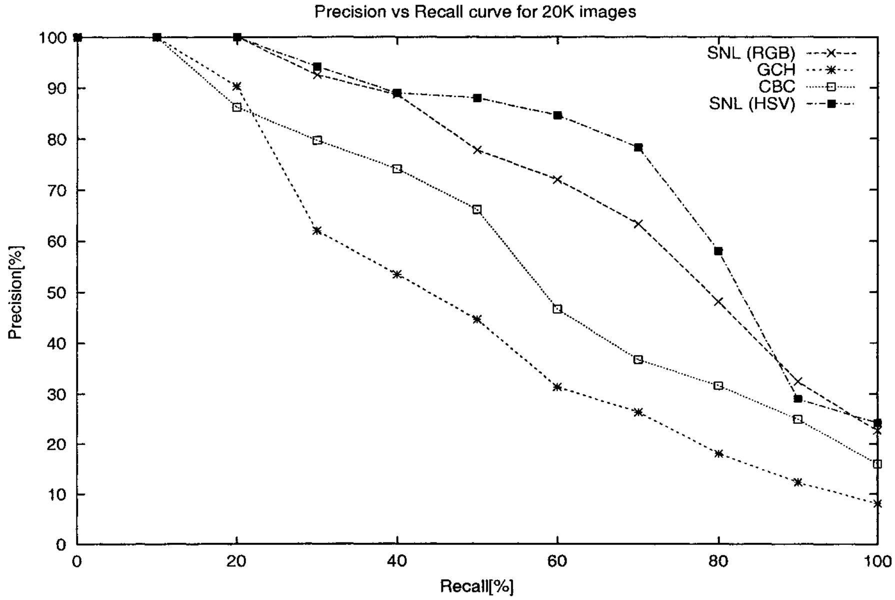


Figure 3.17: Comparing different techniques using a database of 20K images

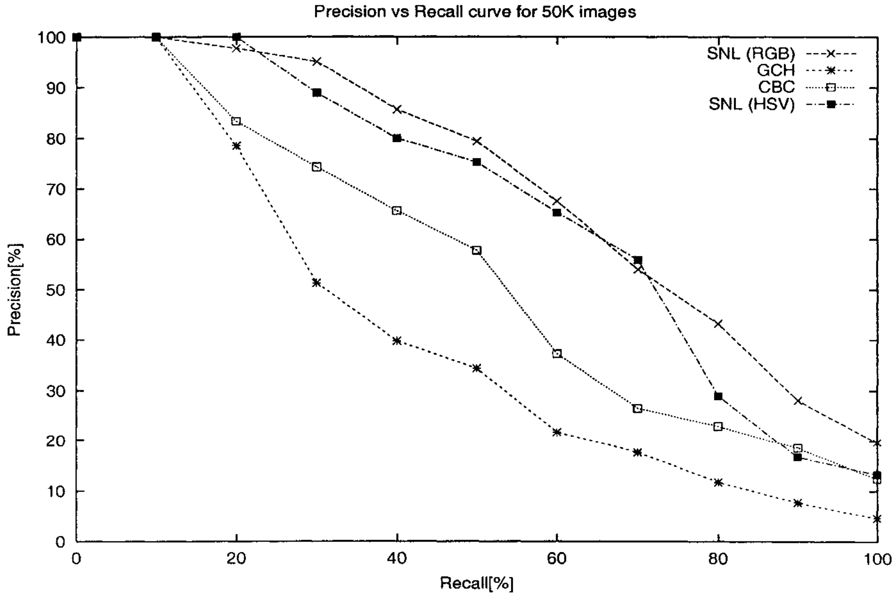


Figure 3.18: Comparing different techniques using a database of 50K images

therefore requires about 960 bytes (assuming four bytes per float value). In SNL, on an average, we obtain about five regions and each region requires about 81 integers for the histogram, one float value for the size of each region, one float value for the shape of each region and two float values for the position of each region. Therefore it uses a total of 890 bytes (720 bytes in the RGB colour space) i.e., it is not only more effective than CBC but also more economical in terms of storage requirement. As one can see, SNL is not nearly as economical as GCH but it is conceptually more elaborate and much more effective.

Table 3.3: Storage space for the techniques

Techniques	Storage (in bytes)
GCH	162
CBC	960
SNL (HSV)	890
SNL (RGB)	720

3.9 Conclusions

We present a content based image retrieval technique, SNL, that is based on the histogram representation of the content of a region. This SNL technique is effective and deals with segmentation inaccuracies better. We used the segmentation algorithm proposed in [61] to divide the image into regions. Each region was given a representation capturing the colour, shape, spatial position and size. The features extracted, in particular the colour representation, was robust to segmentation inaccuracies. In order to maintain this robustness in the similarity measure as well, we used the IRM measure proposed in [68].

Experiments were conducted to choose a good colour space quantization scheme in the RGB and HSV colour space. We also conducted several experiments to decide on the weights α, β and γ that had to be assigned to the region content, shape and size component of each region. An α value of 0.7, β value of 0.15 and γ value of 0.15 yielded very good retrieval effectiveness. We compared SNL technique with CBC and GCH. For this purpose we used three different databases of size 10,000, 20,000 and 50,000 images and in all three cases, we observed that SNL performed better than the other two. SNL also scaled up well with changes in database size. The storage requirements of SNL, CBC and GCH are compared and SNL occupied less space when compared to CBC.

Chapter 4

SNL$^+$

4.1 Motivation and background

The proposed SNL technique is effective from the precision-recall point of view. However, efficiency related issues such as query processing time need to be addressed in the SNL technique. This chapter presents an improved version of SNL called SNL$^+$, which gives almost the same performance in terms of effectiveness but is much more efficient.

4.2 Query processing time

Given below is a table that indicates the time taken by the SNL, CBC and the GCH techniques to process a single query image using the sequential scan algorithm. Although sequential scan is a simple algorithm, when the database size grows larger, this technique does not scale up. The query processing time is directly proportional to the number of images in the database as indicated by the query time for SNL, CBC and GCH in Table 4.1. Since the load on the machine where the experiments were conducted varied all the time, it was very difficult to accurately measure the actual time taken. Hence for all experiments, time was measured relative to the time taken by the GCH technique.

To improve the query processing time, several access methods have been proposed. Access methods use complex data structures and techniques to organize visual features and manage the search process so that visual features relevant to a query can be located easily. The aim of the access methods is to divide the search

Table 4.1: Query processing time for the techniques relative to the GCH technique

Techniques	10K	20K	50K
GCH	1.00	1.00	1.00
CBC	4.56	4.61	4.72
SNL	2.55	2.72	3.05

space into several subspaces such that it would be possible to completely eliminate searching in some of the subspaces and thereby save time [60].

There are basically two types of access methods, the Spatial Access Method (SAM) and the Metric Access Method (MAM). Spatial access methods (SAMs) group points in space by making use of the spatial coordinates. SAMs are limited by two assumptions [11]. Objects are mapped on to feature values in a multidimensional space for indexing purposes. The dissimilarity between two objects should not introduce any correlation between feature values. They are all smitten by the dimensionality curse, since they are affected by the number of dimensions in the vector space. Some existing SAMs include R-tree [22] and R*-tree [3]

An alternative solution to the above mentioned problem is to use Metric Access Methods (MAMs). While SAMs use the absolute spatial location of objects to partition and search the space, MAMs use the relative distance between objects to reduce the search space. M-tree [11] is a well-known MAM representative. M-tree is a dynamic MAM. It is a height-balanced tree structure and is capable of efficient query processing. This is because it can index objects using features that are compared by distance functions which may neither belong to the vector space nor use an L_p metric. The M-tree structure optimizes the I/Os as well as the distance computations. It combines the advantages of dynamic balanced SAMs and the capabilities of static MAMs to index objects.

A third alternative is the use of filtering techniques. Filtering techniques are based on simpler distances that are lower-bounds of the original complex distances, to find the similarity between images. This simple distance is used to filter off a large portion of the irrelevant images in the database. Filtering techniques make extensive use of the triangle inequality property to eliminate the non-relevant images. In [32], the authors have proposed a hierarchical clustering algorithm to cluster images and filter some of them based on the colour content. For each of these clusters, a representative which is the cluster center is found. Thus, whenever the query image comes in, it is initially compared with all the cluster centers instead of all the images in the database. The SNL$^+$ technique makes use of a filtering technique (detailed next) to reduce the query processing time.

4.3 The *Omni* approach

We adopted an approach proposed in [18] to decrease the query processing time. In [18], the authors have proposed a technique to reduce the number of distance calculations required to answer similarity queries. This technique assumes that the distance function used to calculate the similarity between two images is a metric. For a set of objects O = $\{o_1, o_2, ..., o_n\}$, if the distance function *d* is a metric, then it has the following properties

- Symmetry: $d(o_1, o_2) = d(o_2, o_1)$
- Non-Negativity: $0 < d(o_1, o_2) < \infty, o_1 \neq o_2$
- Triangle Inequality: $d(o_1, o_3) \leq d(o_1, o_2) + d(o_2, o_3)$

The triangle inequality property is extensively used in the *Omni* approach to prune the search space and thereby reduce the number of complex distance calculations. This property guarantees that during space reduction, the pruning process will not filter out any of the relevant images. The space reduction, however, introduces some false positives which have to be eliminated by calculating the actual complex distance for a smaller subset, thereby saving a lot of query-processing time. Some techniques, such as [46], [32] use the triangle inequality property on non-metric distances to reduce the search space. Some people [60] argue that it is just yet another approximation step brought into a process where there are several approximations such as:

- The visual features that are used to represent and compare images is an approximation of the visual content of the images;

- The distance measure devised to calculate the similarity between images is an approximation of the human perception of similarity;

- The weights assigned to the features extracted are also an approximation of what would be perceived as most important;

- The retrieval threshold that is used in the query processing phase is an approximate estimation of the similarity between relevant images.

Therefore, it is acceptable to lose a small number of relevant images. The IRM distance which is used to measure the similarity between images is a very good heuristic to obtain the optimum solution in most cases. However, it is not a metric. Though the symmetry and non-negativity properties hold true all the time, it does not comply with the triangle inequality property all the time. A simple example is shown in Figure 4.1, where the three images A, B and C have regions whose property is defined by a single number normalized between 0 and 1 (for illustration purposes). The IRM distance between images A and B is 0.5625. The distance between images B, C and A, C are 0.2 and 0.2625 respectively. In this case, $D_I(A, B) = 0.5625$ is greater than $D_I(B, C) + D_I(A, C)$ which is 0.4625, clearly demonstrating a violation of the triangle inequality property. In this chapter, the technique, SNL$^+$, explores the possibility of using a non-metric distance with the *Omni* approach. Since the pruning criteria in the *Omni* approach is based on the triangle inequality property, using a non-metric distance may result in pruning off some of the relevant objects. In other words, one of the potential problems could be a large number of false dismissals.

In the *Omni* approach, a set of global representative points are initially chosen from the database. These representative points are static in nature, i.e., they are only selected once. These set of points are called the foci and the set of foci of a database forms the *Omni-foci* base. Each object (image representation) is mapped on to a lower dimension space and in this space they are represented by the *Omni-coordinates*. Dimensions of objects in the *Omni-coordinate* system are actually the original distance between objects and the foci. Whenever a new object is inserted, the *Omni-coordinates* of this object are calculated and stored. It is important to

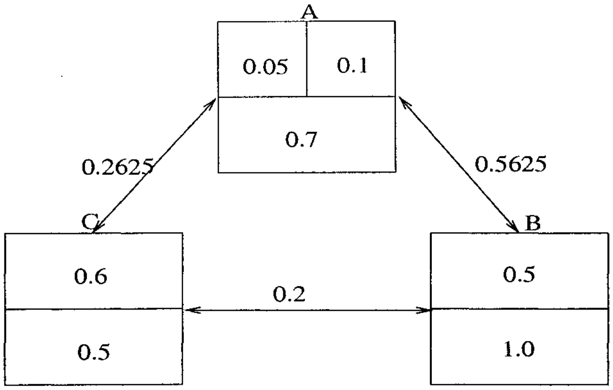


Figure 4.1: An example where IRM measure does not follow the triangle inequality

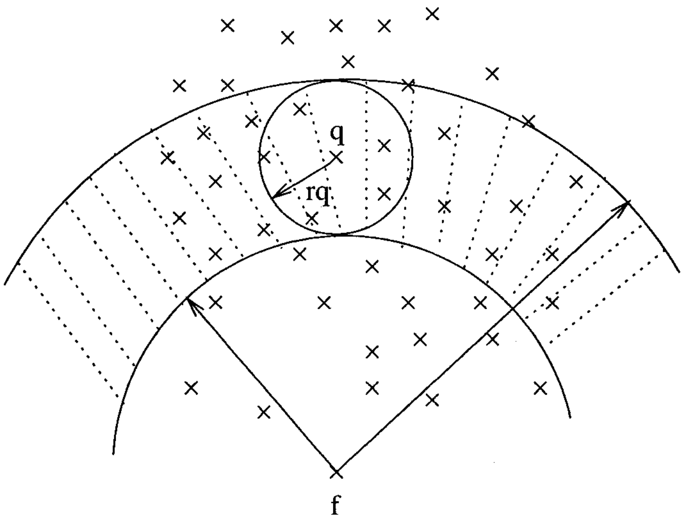


Figure 4.2: Pruning using a single focus (adapted from [18])

note that both the calculation of the foci and the *Omni-coordinates* of the objects in the database are done offline. While querying, the *Omni-coordinates* of the query image are calculated, i.e the query image is also mapped onto a lower dimension space. In this space, the triangle inequality property is extensively used to prune many distance calculations. Figure 4.2 illustrates the *Omni* approach with a single focus point f, q being the query image and r_q being the query radius. Each focus point defines a metric sub-space ring called the *mbOr* as indicated by the area between the two rings in Figure 4.2. An *mbOr* includes all the objects that the *Omni-coordinates* identify as part of the answer set. All points outside the ring are filtered. Points inside the ring cannot be pruned by the focus and hence for this subset of images, the original distance needs to be calculated.

Calculation of the *Omni-foci* and *Omni-coordinates* involves some cost in terms of storage space and time. The authors argue that multimedia objects and their features occupy a lot of storage space and a few more bytes would be insignificant. They also argue that by selecting a reasonably small number of foci, both the storage space and the number of extra calculations required can be kept low. Hence, this technique works well if it is possible to maximize the gain using a small number of well selected foci. The authors have proposed the HF-algorithm to select the foci, as shown in Algorithm 3.

```
Input  : Dataset $S$ and the number of foci $n$
Output : Foci set $F$
Randomly choose object $s_i$ in S;
Find the farthest object $f_1$ from $s_i$;
Insert $f_1$ in $F$;
Find the farthest object $f_2$ from $f_1$;
Insert $f_2$ in $F$;
Set $edge = d(f_1, f_2)$;
Set $minerror = -\infty$;
while $|F| <= n$ do
    for each $s_i$ in $S$ do
        for each $f_k$ in $F$ do
            $Error(i) += |edge - d(f_k, s_i)|$;
        endfor
        if $minerror > Error(i)$ then
            $selected = s_i$;
            $minerror = Error(i)$;
        endif
    endfor
    Insert $selected$ in $F$
endw
Return $F$;
```

Algorithm 3: The HF-algorithm to find foci of a dataset

The logic behind the HF-algorithm is illustrated in Figure 4.3. The objects are represented as colour circles and the distance between these objects is the difference between their colours in the RGB space. In this algorithm, a random object, say A, is chosen initially. The object farthest to this random object A, becomes the first

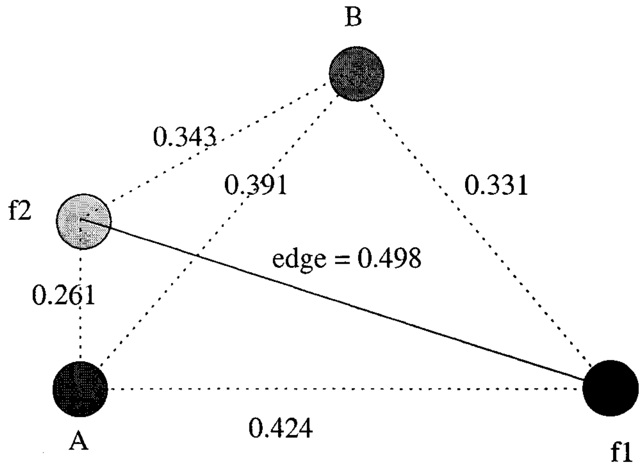


Figure 4.3: Illustration for HF and HF' algorithms

focus f_1 and the object farthest to the first focus f_1 becomes the second focus f_2 as shown in Figure 4.3. Then the next step is to calculate the edge which is the original distance between objects f_1 and f_2 as shown in Figure 4.3. Using f_1, f_2 and the edge, the next focus is calculated by computing the cumulative error as shown in Algorithm 3. This makes A the next focus. However, the foci should be as far apart as possible, so that each one significantly contributes in the pruning process. In this example, object B is a better candidate for a focus than object A. Hence we modified the HF-algorithm to find foci that are spread far apart in the object space. The modified algorithm, HF', is shown in Algorithm 4.

In the HF'-algorithm, the first two foci are determined exactly as in the HF-algorithm. For all other foci, instead of calculating the cumulative difference in the distance between each candidate object and the set of foci found so far, the minimum distance between a candidate object and the set of foci found so far, is calculated. Then, the maximum of the minimum distances is found and the corresponding candidate object becomes the next focus. In the illustration shown in Figure 4.3, objects f_1 and f_2 are the first two foci similar to the HF-algorithm. For the third focus point, there are two candidate objects namely A and B. The minimum of the distances between A and the two foci f_1 and f_2 is 0.261. For object B, the minimum distance is 0.331. Since the distance from object B is greater, it becomes the next focus point. Thus, HF'-algorithm finds foci that are spread further

```
Input   : Dataset $S$ and the number of foci $n$
Output  : Foci set $F$
Randomly choose object $s_i$ in S;
Find the farthest object $f_1$ from $s_i$;
Insert $f_1$ in $F$;
Find the farthest object $f_2$ from $f_1$;
Insert $f_2$ in $F$;
Set $edge = d(f_1, f_2)$;
while $|F| <= n$ do
    for each $s_i$ in $S$ do
        Set $Error(i) = \infty$;
        Set $maxerror = -\infty$;
        Set $e = 0$;
        for each $f_k$ in $F$ do
            $e = |edge - d(f_k, s_i)|$;
            if $Error(i) > e$ then
                $Error(i) = e$;
            endif
        endfor
        if $maxerror < Error(i)$ then
            $selected = s_i$;
            $maxerror = Error(i)$;
        endif
    endfor
    Insert $selected$ in $F$
endw
Return $F$;
```

Algorithm 4: The HF'-algorithm to find foci of a dataset

apart in space. In our preliminary experiments, we observed that the HF'-algorithm indeed resulted in focus images that were visually more different when compared to the ones HF-algorithm.

The authors of [18] have shown that the *Omni* approach can be used with the sequential scan algorithm, the R-trees and the B-trees. Our aim was to investigate the effect of adopting the *Omni* approach to the SNL technique which used the IRM distance (non-metric). Hence, the *Omni-sequential* algorithm has been used in the SNL$^+$ technique. To execute a range query with radius r_q using this algorithm, first of all the original distance between the query object o_q and each foci $f_k \in F$, $df_k(o_q)$ is calculated, thereby creating omni-coordinates for the query object. Then, for each object o_j in the database, if $|df_k(o_j) - df_k(o_q)| > r_q$, for each focus $f_k \in F$ then the original distance calculation between objects o_q and o_j is skipped. Otherwise, the original distance is computed to check if the object o_j lies within the radius r_q of the query object o_q. Thus, we have seen how the *Omni-sequential* algorithm is used in SNL$^+$ to filter out images.

4.4 Experiments

This section talks about the experiments that were conducted to effectively adopt the *Omni* approach in the SNL$^+$ technique. The database images and the 15 query images that were used for experiments in this section are described in Chapter 3. Also all experiments in this chapter are limited to the RGB colour space, since in Chapter 3 we saw that performance of both the colour spaces were almost similar.

The experiments discussed in this section can be categorized into two types. The first set of experiments were conducted to select a suitable number of foci and also to decide on the query radius of our range queries. The second set of experiments were performed to compare the SNL technique with the SNL$^+$ technique in terms of precision-recall and query processing time.

4.4.1 Selection of foci cardinality and query radius

In order to select the foci, the HF' algorithm is used. The number of foci is critical because for every focus, there is an extra dimension added which involves some computation. Unless this additional focus is good enough to filter a fairly large fraction of images and thereby save us some query processing time, the overhead of the space and computation time is not worth it. Hence it is very important to choose a good number of focus points. An experiment was performed wherein the number of foci was varied from 1 to 10 and the percentage of images filtered was noted down. In Figure 4.4, we see that there is a sharp rise in the percentage of images filtered as the number of foci is varied from 1 to 6 and from then on, the curve does not show too much variation. Also it should be noticed that, when the radius is 0.025 and the number of foci is for example, 10, only about 2% of the database is read, whereas for the same number of foci when the radius is 0.1, about 50% of the database is read. Thus, the number of images retrieved can be controlled by the radius.

The aim of proposing the SNL^+ technique is to reduce the query processing time. The number of foci is a factor that directly affects the query processing time. Hence before selecting the number of foci, the behavior of the query processing time with respect to change in the number of foci was noted as shown in Figure 4.5. In this graph, we see that there is a decrease in query processing time when the number of foci is between 1 and 6. The time steadies down between 6 and 7 and then starts increasing gradually.

The Euclidean distance calculation between the omni-coordinates is assumed to be a trivial operation. But as the number of dimensions increase, this is not so trivial anymore. Also, the number of images filtered was not large enough to compensate for this time loss (Figure 4.4). Therefore, from the above two graphs, it would be reasonable to select the number of foci to be six.

The query radius is also another factor that affects the processing time. There is a trade off between the number of relevant images retrieved and the total number of images retrieved (the more the number of images retrieved, the greater the processing time). In Figure 4.6, we can see that when the query radius is 0.1, all the

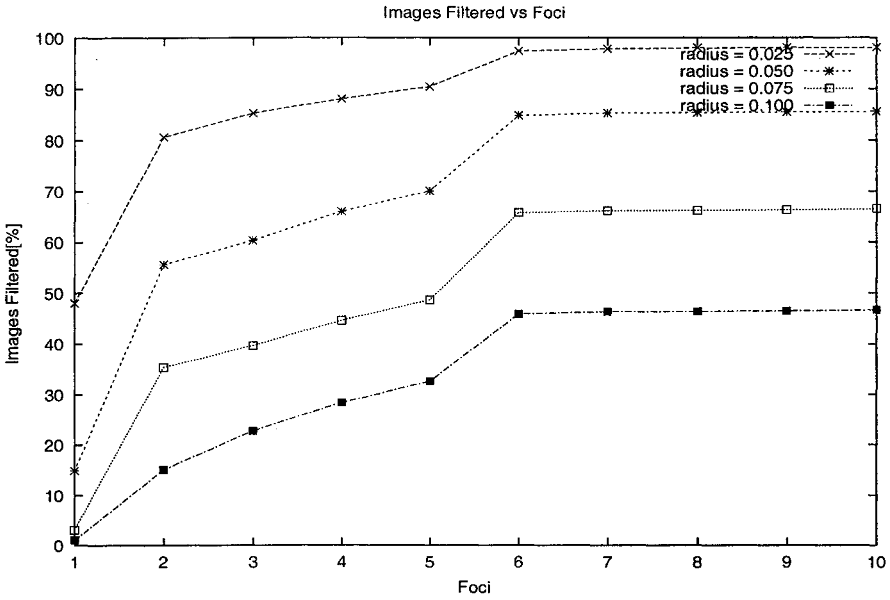


Figure 4.4: Variation of images filtered with foci

relevant images are retrieved. But to achieve this, we need to process about 60% of the database. Whereas for a query radius of 0.075, about 96% of the relevant images are retrieved and only 35% of the database needs to be processed. Therefore, a radius of 0.075 was selected for our range query, since it was sufficient to retrieve almost all the relevant images corresponding to the query images. This is of course tunable, depending on the number of relevant images one is willing to lose.

4.4.2 Comparing SNL$^+$ with SNL, CBC and GCH

As mentioned before, the IRM measure used to calculate the similarity between images is not a metric and despite this fact, SNL$^+$ uses it with the *Omni-sequential* approach, which by default works on metric distances. Using non-metric distances makes the SNL$^+$ approach liable to losing some of the relevant images. In order to determine the amount of relevant images lost, the precision and recall values for 10K, 20K and 50K images using the SNL$^+$ approach are measured and compared with the original SNL approach. The number of foci was fixed as 6 and a radius of 0.075 was used. The results of this experiment are shown in Figures 4.7, 4.8 and 4.9. The graphs show that the curves are very close to each other indicating that the loss of relevant images that occurs by approximating the *Omni-sequential*

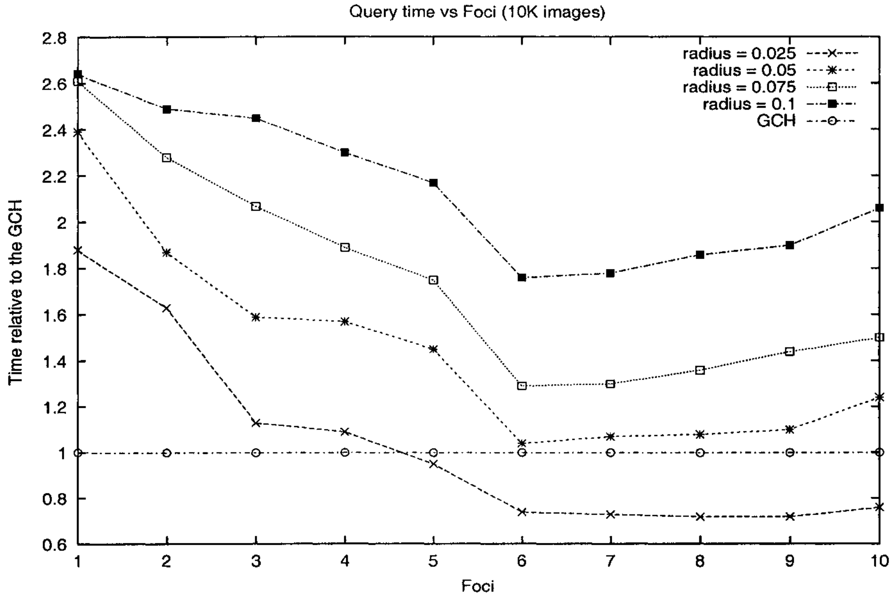


Figure 4.5: Variation of query processing time with foci

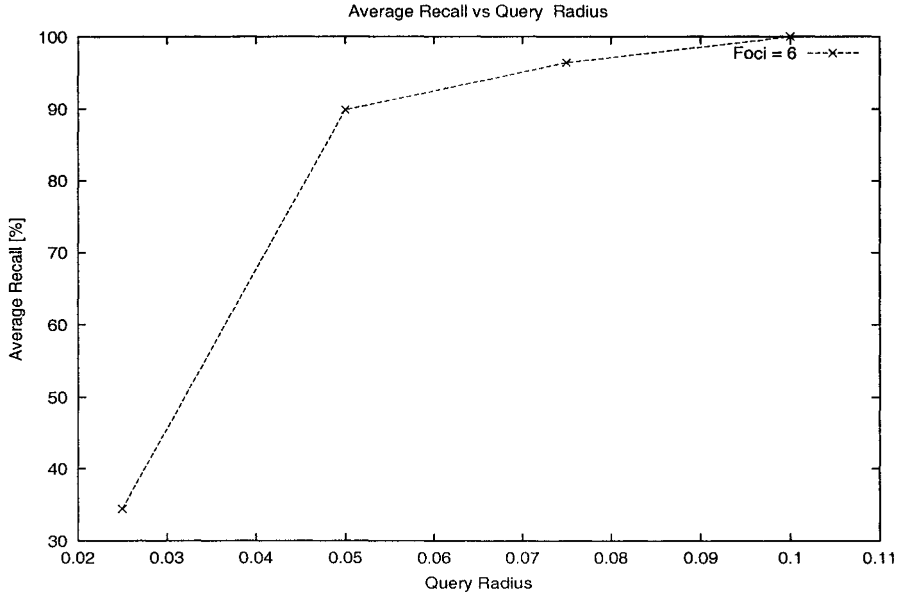


Figure 4.6: Variation of recall with query radius

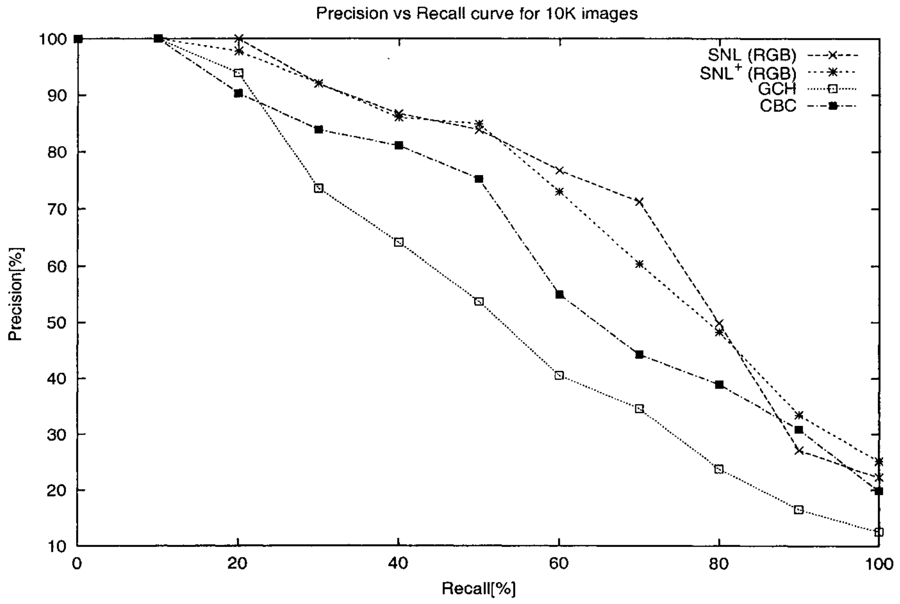


Figure 4.7: Comparing SNL and SNL$^+$ with 10K images

algorithm with a non-metric distance is less. At certain retrieval points, the SNL$^+$ has a higher precision compared to the SNL technique. This is due to the fact that in SNL$^+$, the original distance is calculated only for a very small fraction of the database thus further decreasing the room for false positives.

Finally, Table 4.2 compares the query processing time taken by the SNL technique and the SNL$^+$ technique. The actual query processing time of the SNL$^+$ technique in a database of 10K images using six foci and 0.075 as the query radius is only about six seconds. It can be seen that the processing time has reduced by almost 50% using the SNL$^+$ technique. Even when the database size is changed from 10K to 50K, the query processing time for SNL$^+$ is still small and in fact comparable with GCH's processing time (see Table 4.2). Thus, the *Omni* approach has considerably helped in reducing the query time. Despite of using a non-metric distance, there is very little loss in terms of effectiveness.

Table 4.2: Query processing time relative to GCH for SNL and SNL$^+$

Techniques	10K	20K	50K
SNL$^+$	1.28	1.36	1.45
SNL	2.55	2.72	3.05

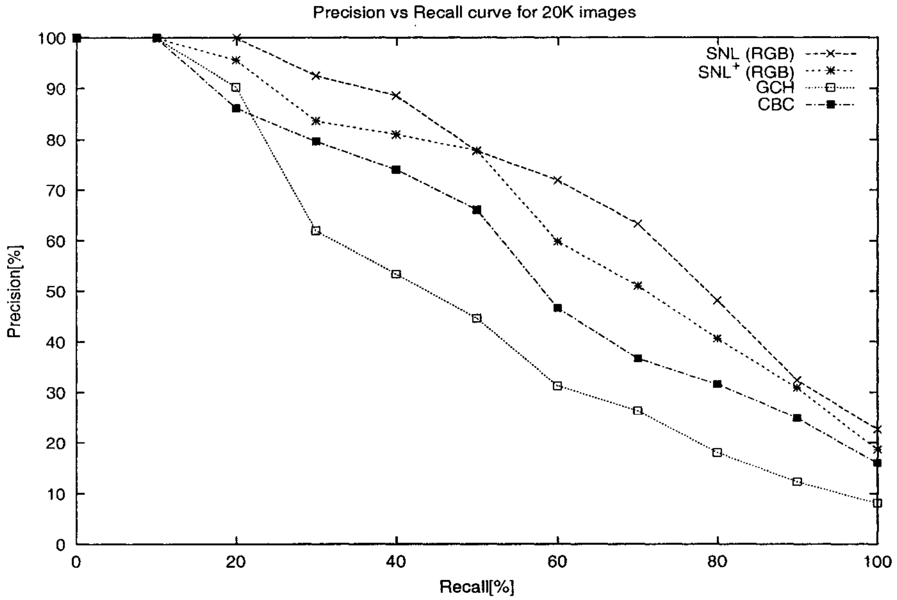


Figure 4.8: Comparing SNL and SNL$^+$ with 20K images

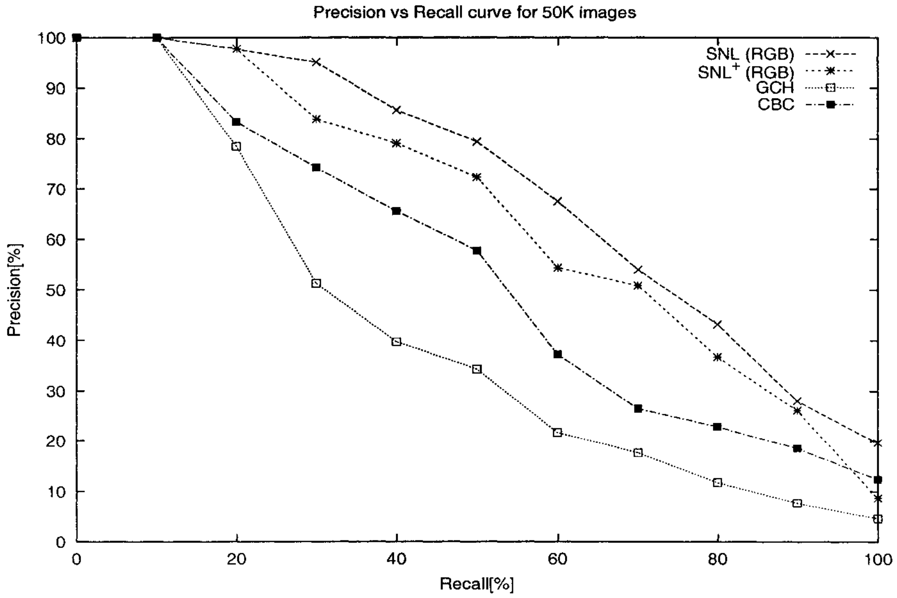


Figure 4.9: Comparing SNL and SNL$^+$ with 50K images

4.5 Conclusion

In this chapter, an efficient version of the SNL technique namely the SNL$^+$ technique was proposed. The *Omni-sequential* algorithm put forth by Santos Filho *et al.* in [18] was adopted. The HF-algorithm to determine the foci set in a database was improved. The effect of using a non-metric distance with the *Omni-sequential* approach was investigated and found to be as efficient as a metric distance. Experiments were conducted to select a suitable set of focus points. A query radius was also selected to obtain sufficient number of relevant images. The SNL and the SNL$^+$ techniques were compared for both effectiveness and efficiency. It is shown that the SNL$^+$ technique performs as well as the SNL technique in terms of precision and recall and at the same time requires only half the amount of query processing time.

Chapter 5

Conclusions and future work

5.1 Conclusions

In this thesis, three important research issues of content-based image retrieval have been addressed. An effective image retrieval technique, the SNL, has been proposed. This technique makes use of a representation technique (the first issue), which is robust to segmentation inaccuracies. The SNL technique also makes use of a distance measure (the second issue), IRM [36], which is capable of reflecting on the human perception of visual similarity. The retrieval behavior of the SNL technique was studied by varying the colour space from RGB to HSV. After several experiments, suitable weights were assigned to the content of a region, its shape and spatial features. The SNL technique was also proven to be more effective than GCH and CBC by measuring the precision-recall curves. All the experiments were performed in databases whose size ranged from 10K to 50K images.

The third issue addressed in this thesis is related to the efficiency of a retrieval technique. The SNL$^+$ technique was proposed with the aim of reducing the query processing time. For this purpose, the *Omni* approach [18] was adopted to filter out some irrelevant images during the query time. The HF-algorithm proposed in this technique was modified to find foci that were spread far apart in space. The IRM distance (not a metric) was used with the *Omni* approach, which was proposed to work for metric distances. Using this approach, the query processing time was reduced by half, when compared to the SNL technique. The effectiveness of the SNL$^+$ and the SNL technique was compared and found to be very similar indicating very little loss of information due to the usage of a non-metric IRM distance in the *Omni* approach.

5.2 Future work

Future work for the thesis presented can be categorized into two parts. The first one is related to the SNL technique, i.e. improving upon the effectiveness and the second one is related to SNL$^+$ i.e. improving on the efficiency of the technique. In SNL, currently there exists a robust representation for the colour of each region. It would be interesting to come up with such representations for the spatial position as well as the shape of the region. The IRM measure currently being used is not a metric. There is an alternate distance measure [64] that is a metric and can be easily indexed. In this thesis, only two colour spaces have been investigated. There are other spaces such as the Lab which might be studied. Also currently for histograms of regions, a uniform quantization is used. When looking at the cumulative colour histograms of the dataset used, one can see that there is no uniform distribution of colour probabilities. Hence one might want to explore non-uniform quantization schemes in dealing with the colour information of images.

So far, we have only applied a filtering technique to improve the query processing time. In order to efficiently manage querying in huge databases real time, it is important to use an index structure. There are some variations of the *Omni* approach such as the *Omni B-tree* and the *Omni R-tree* proposed in [18] that could be applied to SNL$^+$ to build indexes. Another important issue which was not addressed in this thesis, was reducing storage requirements of SNL and SNL$^+$. It is quite obvious though that we do require a lot of space to store the colour histograms, spatial position, size and shape of each region. In order to optimize the storage space, one could perhaps use binary signatures [7] or similar structures.

Bibliography

[1] J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee W. Niblack, and D. Petkovic. Automatic and semi-automatic methods for image annotation and retrieval in QBIC. In *Proc. of Storage and Retrieval for Image and Video Databases (SPIE) III, volume 2420*, pages 24–35, 1995.

[2] J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. Shu. Virage image search engine: An open framework for image management. In *Proc. of the Storage and Retrieval for Image and Video Databases (SPIE) IV, volume 2670*, pages 76–87, 1996.

[3] N. Beckmann, H.P Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proc. of ACM SIGMOD Intl. Conf. on Management of Data*, pages 323–331, 1990.

[4] A. D. Bimbo. *Visual Information Retrieval.* Morgan Kaufmann Ed, 1999.

[5] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Region-based image querying. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 42–49, 1997.

[6] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Proc. of the 3^{rd} Intl. Conf. on Visual Information Systems*, pages 509–516, 1999.

[7] V . Chitkara. Color-based image retrieval using binary signatures. Master's thesis, University of Alberta, 2001.

[8] T. Chua, S.-K. Lim, and H.K Pung. Content-based retrieval of segmented images. In *Proc. of the Second Intl. ACM Multimedia Conf.*, pages 211–218, 1994.

[9] T. Chua, K. Teo, B. Ooi, and K. Tan. Using domain knowledge in querying image databases. In *Proc. of Intl. Conf. on Multimedia Modeling*, pages 12–15, 1996.

[10] Y. Chunhui. A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *Proc. of 8^{th} ACM Multimedia Intl. Conf.*, pages 31–38, 2000.

[11] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proc. of the 23rd Intl. Conf. on Very Large Data Bases (VLDB'97)*, pages 426–435, 1997.

[12] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 750–755, 1997.

[13] J.M. Corridoni, A. Del Bimbo, and P. Pala. Image retrieval by color semantics. *ACM Multimedia System*, 7(7):175–183, 1999.

[14] J.P. Eakins and M.E. Graham. Content-based image retrieval. Technical report, JISC Technology Applications Programme, Institute for Image Data Research, University of Northumbria at Newcastle, UK, 1999.

[15] S. Fan. Indexing and retrieving shapes via distance histograms. Master's thesis, University of Alberta, 2001.

[16] D.S. Faulus and R.T. Ng. An expressive language and interface for image querying. *Machine Vision and Applications*, 10(2):74–85, 1997.

[17] J. Feder. Towards imaging content-based retrieval for the world-wide web. *Advanced imaging*, 11(1):26–28, 1996.

[18] R.F. Santos Filho, A. Traina, C. Traina Jr., and C. Faloutsos. Similarity search without tears: The omni family of all-purpose access methods. In *Proc. of the 17th Intl. Conf. on Data Engineering (ICDE 2001)*, pages 623–630, 2001.

[19] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison-Wesley, third edition, 1992.

[20] W. I. Grosky and R. Mehrotra. Index-based object recognition in pictorial data management. *Computer Vision, Graphics, and Image Processing*, 52(3):416–436, 1990.

[21] N.J. Gunther and G. Beretta. A benchmark for image retrieval using distributed systems over the internet: Birds-1. Technical Report HPL-2000-162, HP Labs, 2000.

[22] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. of ACM SIGMOD Intl. Conf. on Management of Data*, pages 47–57, 1984.

[23] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[24] R. Haralick, K. Shanmugam, and I. Dinstein. Texture feature for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[25] W. Hsu, T.S. Chua, and H.K. Pung. An integrated color-spatial approach to content-based image retrieval. In *Proc. of* 3^{rd} *ACM Multimedia Conf.*, pages 305–313, 1995.

[26] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–187, 1962.

[27] T. Huang, S. Mehratra, and K. Ramchandran. Multimedia analysis and retrieval system. In *In Proc. of the 3rd Intl. Workshop on Information Retrieval Systems*, 1996.

[28] T. Huang and Y. Rui. Image retrieval: Past, present, and future. In *Proc. of the Intl. Symposium on Multimedia Information Processing*, pages 1–23, 1997.

[29] H.V. Jagadish. A retrieval technique for similar shapes. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pages 208–217, 1991.

[30] D. Kapur, T. Saxena, and Y.N. Lakshman. Computing invariants using elimination methods. In *Proc. of IEEE Intl. Symp. on Computer Vision*, pages 97–102, 1995.

[31] P. Kerminen and M. Gabbouj. Image retrieval based on color matching. In *Proc. of the Finnish Signal Processing Symp.*, pages 89–93, 1999.

[32] S. Krishnamachari and M. Abdel-Mottaleb. Hierarchical clustering algorithm for fast image retrieval. In *IS&T/SPIE Conf. on Storage and Retrieval for Image and Video Databases VII, volume 3656*, pages 427–435, 1999.

[33] K.Y. Kupeev and H.J. Wolfson. A new method of estimating shape similarity. *Pattern Recognition Letters*, 17:873–887, 1996.

[34] G. Kuper, S. Ramaswamy, K. Shim, and J. Su. A constraint-based spatial extension to SQL. In *Proc. of ACM Symposium on GIS*, pages 112–117, 1998.

[35] J. Laaksonen, M. Koskela, and E. Oja. Content-based image retrieval using self-organizing maps. In *Proc. of 3rd Intl. Conf. in Visual Information and Information Systems*, pages 541–548, 1999.

[36] J. Li, J. Z. Wang, and G. Wiederhold. IRM: Integrated Region Matching for image retrieval. In *Proc. of the 8^{th} ACM Multimedia Conf.*, pages 147–156, 2000.

[37] W.S. Li, K.S. Candan, K. Hirata, and Y. Hara. IFQ: A visual query interface for object-based image retrieval. In *Proc. of the IEEE Multimedia Computing and Systems Conf.*, pages 353–361, 1997.

[38] Z.-N. Li, O. R. Zaïane, and B. Yan. C-BIRD: content-based image retrieval from image repositories using chromaticity and recognition kernel. In *Proc. Intl. Workshop on Storage and retrieval Issues in Image and Multimedia Databases, in conjunction with the 9th International Conference on Database and Expert Systems (DEXA'98)*, pages 361–366, 1998.

[39] Y. Liu, F. Dellaert, and W.E. Rothfus. Classification driven semantic based medical image indexing and retrieval. Technical Report CMU-R1-TR-98-25, Carnegie Mellon University, 1998.

[40] W.Y. Ma and B.S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[41] J. MacQueen. Some methods of classification and analysis of multivariate observations. In *in Proceedings of the Fifth Berkeley Symposium on Mathemtical Statistics and Probability*, pages 281–297, 1967.

[42] R. Mehrotra and J.E. Gary. Similar-shape retrieval in shape data management. *IEEE COMPUTER*, 28(9):57–62, 1995.

[43] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *Proc. of* 1^{st} *Intl. Workshop on Image Databases and Multimedia Search*, pages 35–42, 1996.

[44] W. Muller, Z. Pecenovic, H. Mller, S. Marchand-Maillet, T. Pun, D. Mc. G. Squire, A. P. De Vries, and C. Giess. MRML: An extensible communication protocol for interoperability and benchmarking of multimedia information retrieval systems. In *SPIE Photonics East - Voice, Video, and Data Communications*, pages 5–8, 2000.

[45] A. Natsev, R. Rastogi, and K. Shim. WALRUS: A similarity retrieval algorithm for image databases. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pages 395–406, 1999.

[46] W. Niblack, X. Zhu, J.L. Hafner, T. Breuel, D. Ponceleón, D. Petkovic, M. Flickner, E. Upfal, S.I. Nin, S. Sull, B. Dom, B.-L. Yeo, S. Srinivasan, D. Zivkovic, and M. Penner. Updates to the QBIC system. In *Proc. of SPIE – Storage and Retrieval for Image and Video Databases VI, volume 3312*, pages 150–161, 1998.

[47] B.C. Ooi, K.-L. Tan, T.S. Chua, and W. Hsu. Fast image retrieval using color-spatial information. *VLDB journal*, 7(2):115–128, 1998.

[48] G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. In *Proc. of IEEE Workshop on Applications of Computer Vision*, pages 96–102, 1996.

[49] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proc. of the* 4^{th} *ACM Multimedia Intl. Conf.*, pages 65–73, 1996.

[50] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *Intl. Journal of Computer Vision*, 18(3):233–254, 1996.

[51] A. Pentland, R.W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *Proc. of Storage and Retrieval for Image and Video Databases (SPIE) II, volume 2185*, pages 34–47, 1994.

[52] C.J. Van Rijsbergen. *Information Retrieval.* Butterworths, second edition, 1979.

[53] J.T. Robinson. The K-D-B-Tree: a search structure for large multidimensional dynamic indexes. In *Proc. ACM-SIGMOD Intl. Conf. on Management of Data*, pages 10–18, 1981.

[54] Y. Rui, A. She, and T. Huang. Modified fourier descriptors for shape representation – a practical approach. In *Proc. of the* 1^{st} *Intl. Workshop on Image Databases and Multimedia Search.*, pages 22–23, 1996.

[55] E.D. Sciascio, G. Mingolla, and M. Mongiello. Content-based image retrieval over the web using query by sketch and relevance feedback. In *Proc. of the* $4^{th)}$ *Intl. Conf. on Visual Information Systems*, pages 123–130, 1999.

[56] G. Sheikholeslami, A. Zhang, and L. Bian. A multi-resolution content-based retrieval approach for geographic images. *Geo-Informatica*, 3(2):109–139, 1999.

[57] J. Smith and S. Chang. Single color extraction and image query. In *Proc. of IEEE Intl. Conf. on Image Processing*, pages 528–531, 1995.

[58] J.R. Smith and S. Chang. VisualSEEk: A fully automated content-based image query system. In *Proc. of the Fourth ACM Intl. Multimedia Conf.*, pages 87–98, 1996.

[59] J.R. Smith and S. Chang. Image and video search engine for the world wide web. In *Proc. of the Conf. on Storage and Retrieval for Image and Video Databases (SPIE) V, volume 3022*, pages 84–95, 1997.

[60] R.O. Stehling, M.A. Nascimento, and A.X. Falcão. Techniques for Color-Based Image Retrieval", Chapter 3 of "Book of Intelligent Multimedia Document", Kluwer - 2002 (To Appear).

[61] R.O. Stehling, M.A. Nascimento, and A.X Falcão. An adaptive and efficient clustering-based approach for content based image retrieval in image databases. In *Proc. of the Intl. Data Engineering and Application Symposium*, pages 356–365, 2001.

[62] R.O. Stehling, M.A. Nascimento, and A.X. Falcão. Techniques for color-based image retrieval. Technical Report 16, University of Alberta, 2001.

[63] R.O Stehling, M.A. Nascimento, and A.X. Falcão. Cell histograms versus color histograms for image representation and retrieval. *Knowledge and Information Systems (KAIS) Journal (To Appear)*, 2002.

[64] R.O. Stehling, M.A. Nascimento, and A.X Falcão. MiCRoM: a metric distance to compare segmented images. In *Proc. of the 2002 Visual Information Systems Conf. (VISUAL'02)*, pages 12–23, 2002.

[65] M.A. Stricker and M. Orengo. Similarity of color images. In *Proc. of Storage and Retrieval for Image and Video Databases (SPIE)-III, volume 2420*, pages 381–392, 1995.

[66] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems,Man, and Cybernetics*, SMC-8(6):460–473, 1978.

[67] H.D. Wactlar, T. Kanade, M.A. Smith, and S.M. Stevens. Intelligent access to digital video: Informedia project. *IEEE Computer*, 29(5):46–52, 1996.

[68] J.Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.

[69] S. Wang. A robust CBIR approach using local color histograms. Master's thesis, University of Alberta, 2001.

[70] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes*. Morgan Kaufmann, Second edition, 1999.

[71] L. Yang and F. Albregtsen. Fast computation of invariant geometric moments: A new method giving correct results. In *Proc. of the* 12^{th} *Intl. Conf. on Pattern Recognition*, pages 201–204, 1994.

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