HIERARCHICAL INDEXING FOR REGION BASED IMAGE RETRIEVAL

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
In partial fulfillment of the
Master of Science in Industrial Engineering

in

The Department of Industrial and Manufacturing Systems Engineering

by
Eka Aulia
B.S., Louisiana State University, 2001
May 2005

ACKNOWLEDGEMENTS

This thesis is a product of three semesters of learning and dialogues with my major advisor Dr. Gerald Knapp. I sincerely thank him for his numerous suggestions and commend his patience. It was an honor to have him as my advisor.

I am very great full for my years at Industrial engineering at Louisiana State University. I am especially indebted towards Dr. Xiaoyue Jiang and Dr. Charles McAllister for being a part of my thesis committee.

I would like to extend my gratitude to the department for awarding me a graduate assistantship (GA). I would like to thank my colleague Amin Shah-Hosseini for suggestions and discussions during my research. There are many people who have helped and supported me and I would like to take this opportunity to thank everyone.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Region-based image retrieval system has been an active research area. In this study we developed an improved region-based image retrieval system. The system applies image segmentation to divide an image into discrete regions, which if the segmentation is ideal, correspond to objects. The focus of this research is to improve the capture of regions so as to enhance indexing and retrieval performance and also to provide a better similarity distance computation.

During image segmentation, we developed a modified k-means clustering algorithm for image retrieval where hierarchical clustering algorithm is used to generate the initial number of clusters and the cluster centers. In addition, to during similarity distance computation we introduced object weight based on object's uniqueness. Therefore, objects that are not unique such as trees and skies will have less weight.

The experimental evaluation is based on the same 1000 COREL color image database with the FuzzyClub, IRM and Geometric Histogram and the performance is compared between them.

As compared with existing technique and systems, such as IRM, FuzzyClub, and Geometric Histogram, our study demonstrate the following unique advantages: (i) an improvement in image segmentation accuracy using the modified k-means algorithm (ii) an improvement in retrieval accuracy as a result of a better similarity distance computation that considers the importance and uniqueness of objects in an image.

Index Terms: Region based image retrieval, hierarchical clustering, image classification, image segmentation, region matching.

CHAPTER 1: INTRODUCTION

Digital photography, cheap storage, and high-capacity public networks have led to a rapid increase in the use of digital images in many application areas, such as publishing and the media, military, commerce, education, and the World Wide Web. The need to manage these images and locate target images in response to user queries has become a significant problem. One way to solve this problem would be describing the image by keywords. However, this method suffers from the need for manual classification of images, which is simply not practical in databases where thousands of new images may be added daily. In addition, subjectivity and ambiguity of the description by human perception, as well as incompleteness of a limited set of keyword descriptors, may significantly reduce query effectiveness. Using a Content Based Image Retrieval (CBIR), images can be analyzed and indexed automatically by automatic description which depends on their objective visual content.

1.1 Content Based Image Retrieval

Content Based Image Retrieval is a set of techniques for retrieving semantically-relevant images from an image database based on automatically-derived image features (Li and Wang 2000). The main goal of CBIR is efficiency during image indexing and retrieval, thereby reducing the need for human intervention in the indexing process. The computer must be able to retrieve images from a database without any human assumption on specific domain (such as texture vs. non texture or indoor vs. outdoor).

One of the main tasks for CBIR systems is similarity comparison, extracting feature signatures of every image based on its pixel values and defining rules for comparing images. These features become the image representation for measuring similarity with other images in the database. Images are compared by calculating the difference of its feature components to other image descriptors.

Early CBIR methods used global feature extraction to obtain the image descriptors. For example, QBIC (Flickner, 1995), developed at the IBM Almaden Research Center, extracts several features from each image, namely color, texture and shape features. These descriptors are obtained globally by extracting information on the means of color histograms for color features; global texture information on coarseness, contrast, and direction; and shape features about the curvature, moments invariants, circularity, and eccentricity. Similarly, the Photobook system (Pentland et al, 1996), VisualSeek (Smith and Chang 1997), and Virage (Gupta and Jain, 1997), use global features to represent image semantics.

These global approaches are not adequate to support queries looking for images where specific objects in an image having particular colors and/or texture are present, and shift/scale invariant queries, where the position and/or the dimension of the query objects may not relevant. For example, suppose in one image there are two flowers with different colors: red and yellow. The global features describe the image as the average of the global average color which is orange. This description is certainly not the representation of the semantic meaning of the image. Therefore, we can see that the weakness of global features is observable.

Region-based retrieval systems attempt to overcome previous method limitations of global based retrieval systems by representing images as collections of regions that may correspond to objects such as flowers, trees, skies, and mountains. A region based retrieval systems applies image segmentation (Shi and Malik, 1997) to decompose an image into regions, which correspond to physical objects (trees, people, cars, flowers) if the decomposition is ideal. The feature descriptors are extracted on each object instead of global image. Color, texture and shape features are extracted on each pixel that belongs to the object, and each object is described by the average value of these pixel features.

1.2 Feature Extraction

The three feature descriptors mainly used most frequently during feature extraction are color, texture and shape.

1.2.1 Color Features

Color is an important dimension of human visual perception that allows discrimination and recognition of visual information (Smith, 2002). Color features are relatively easy to extract and match, and have been found to be effective for indexing and searching of color images in image databases.

One of the main aspects of color feature extraction is the choice of a color space. A color space is a multidimensional space in which the different dimensions represent the different components of color. Most color spaces are three dimensional. An example of a color space is RGB, which assigns to each pixel a three element vector giving the color intensities of the three primary colors, red, green and blue. The space spanned by the R, G, and B values completely describes visible colors, which are represented as vectors in the 3D RGB color space. As a result, the RGB color space provides a useful starting point for representing color features of images. However, the RGB color space is not perceptually uniform. More specifically, equal distances in different intensity ranges and along different dimensions of the 3D RGB color space do not correspond to equal perception of color dissimilarity.

Alternative color spaces can be generated by transforming the RGB color space. The idea for color space transformation is to develop a model of color space that perceptually similar with human color vision. Color spaces such as HSV, CIE 1976 (LAB), and CIE 1976 (LUV) are generated by nonlinear transformation of the RGB space. The CIE color spaces represent, the three characteristics that best characterize color perceptually: hue, lightness, and saturation. However, the CIE color spaces are inconvenient because of the calculation complexities of the transformation to and from the RGB color space. HSV color space is also a nonlinear transformation of the RGB, but it is easily invertible (Smith, 2002). The HSV color space is approximately perceptually uniform. In this paper, we use HSV color space to extract color features.

1.2.2 Texture Features

Rao and Lohse (1993) identify three features as being important in human texture perception: *repetition*, *orientation*, and *complexity*. Repetition refers to periodic pattern and is often associated with regularity. Orientation refers to the presence or absence of directional textures. Complexity refers to the descriptional complexity of texture which is the combination of characterization of coarseness, contrast, directionality, line-likeness, regularity and roughness (Tamura, 1978).

According to Manjunath (2000) the existing texture descriptors can be classified into three categories:

1. Features that are computed in the spatial domain
2. Features that are computed using model based approach
3. Features that are computed in a transform domain

The texture categories are explained below.

- **Spatial Domain**

For texture features based on spatial-domain analysis, one way to describe the descriptor is using a second order statistics of pairs of intensity values of pixels in an image. This method, called co-occurrence matrices (Julezs, 1975), counts how often pairs of grey level of pixels, separated by a certain distance and lying along certain direction, occur in an image. Much work has been done on this feature descriptor; however it now appears that this characterization of texture is not very effective for classification and retrieval (Manjunath, 2000). In addition, these features are expensive to compute; for this reason, co-occurrence matrices are rarely used in modern image database applications (Manjunath, 2000).

- **Model Based Approaches**

Model-based texture methods try to capture the process that generated the texture. By using the model-based features some part of the image model is assumed and an estimation algorithm is used to set the parameters of the model to yield the best fit (Wu, 2003). To describe random field, assume the image is modelled as a function $f(r,\omega)$, where r is the position vector representing the pixel location in the 2D space and ω is a random parameter. For a given value of r, $f(r,\omega)$ is a random variable (because ω is a random variable). Once a specific texture ω is selected, $f(r,\omega)$ is an image, which is a function over the two-dimensional grid indexed by r. Function $f(r,\omega)$ is called as a random field (Rosenfeld, 1982).

There are currently three major model based methods: Markov Random Fields (MRF) by Dubes and Jain (Dubes, 1989), fractals by Pentland (Pentland, 1984), and the multi-resolution autoregressive (AR) features introduced by Mao and Jain (Mao, 1992).

Markov random fields define an efficient framework for specifying nonlinear interactions between features of the same nature or of a different one. They help to combine and organize spatial and temporal information by introducing strong generic knowledge about the features to be estimated. Fractal models, proposed by Mandelbrot (Mandelbrot, 1983), describe images with a set of self-similar functions characterized by fractal dimension, which correlated to perceived roughness of image texture (Pentland, 1984). The auto-regression model provides a way to use linear estimates of a pixel's grey level, given the grey levels in the neighbourhood containing it. The advantage of the auto-regression model is that it is easy to use the estimator in a mode that synthesises texture from any initially given linear estimator. However, it can only characterize textures that consist of micro textures (Wu, 2003).

- **Transform Domain Features**

The word transform refers to a mathematical representation of an image. There are several texture classifications using transform domain features in the past, such as discrete Fourier transform (DFT), and discrete wavelet transforms (DWT).

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. On the other hand, wavelet analysis breaks up of a signal into shifted and scaled versions of the original wavelet (mother wavelet) which refers to decomposition of a signal with a family of basis functions obtained through translation and dilation of a special function (Manjunath, 2000). Moments of wavelet coefficients in various frequency bands have been shown to be effective for representing texture (Unser, 1995). Therefore, in this paper we use wavelet transform to extract the texture features.

Wavelet transform computation involves recursive filtering and subsampling; and at each level, it decomposes a 2D signal into four subbands, which are often referred to as LL, LH, HL, and HH (L=Low, H=High) according to their frequency characteristics (Chang, 1993).

In this paper, to extract the texture features, we represent the features by the energy in the high frequency bands of the Haar wavelet transform (Daubechies, 1992). The reason for choosing Haar transform is that it has better reflection of texture properties (Unser, 1995) where the coefficient in different frequency bands signal variations in different directions, such as horizontal, vertical, and diagonal. In addition, Haar transform require less computation compared to other wavelet transform with longer filters (Wang, 1998).

1.2.3 Shape Features

Shape can be represented using a variety of descriptors such as moments, Fourier descriptors, geometric and algebraic invariants, polygons, polynomials, splines, strings, deformable templates, and skeletons (Kimia, 2002). The use of shape as a cue is less developed than the use of color or texture.

Several authors for image retrieval system have integrated the shape features with color and texture features to obtain overall similarity measures. Li and Wang (2000), the author of IRM system, integrates the shape features into a similarity distance calculation when the image is classified as a non-texture image. Zhang (2002), the author of Fuzzy Club, put less weight for the shape features compared to color and texture features. Moment inertia is a commonly used shape feature. This shape feature is defined as a vector containing three components for the normalized inertia (Gresho, 1979) of order 1 to 3 of a region, respectively. For a region H in k-dimensional Euclidean space $\Re^k$, the normalized inertia of order γ is

$$l(H,\gamma) = \frac{\int_H \left\| x - \bar{x} \right\|^{\gamma} dx}{[V(H)]^{1+\gamma/k}} \tag{1.1}$$

where *V(H)* is the number of pixels in region *H*. The normalized inertia is invariant with scaling and rotation. The minimum normalized inertia is achieved by spheres. Denoting the γ th order normalized inertia of spheres as L_γ, the shape features is defined as $l(H,\gamma)$ normalized by L_γ:

$$S_1 = l(H,1)/L_1, \qquad S_2 = l(H,2)/L_2, \qquad S_3 = l(H,3)/L_3 \tag{1.2}$$

1.3 Problem Statement

Region-based image retrieval has become an important research focus in the image database community. Several systems have been developed to improve the performance and efficiency during retrieval. Image segmentation is a crucial step for a region-based system to increase performance and accuracy during image similarity distance computation.

Our literature review has found that during image segmentation Li and Wang (2000) and Zhang (2002) used a k-means algorithm where the number of *k* is adaptively selected by gradually increasing *k* until the stopping criteria is met. However since the initial cluster assignment is random, different runs of the k-means clustering algorithm may not give the same final clustering solution. To deal with this we need to get good starting points for the initial cluster assignment. This leads us to modify the k-means clustering where hierarchical algorithm is used to provide the number of *k* and the initial cluster centers.

To increase the retrieval speed, Zhang (2002) grouped similar object to form classes using k-means algorithm. K-means has been known to work well when clusters are rather well separated from one another. However, when there are large differences in the size or geometries of different clusters, the k-means method may lead improper clustering and split large clusters to minimize the square error (Guha, 1998). The Hierarchical method may help overcome this weakness in k-means (Seo, 2003). In this study we compared the performance of precision during query using Zhang's object clustering result and using our hierarchical clustering algorithm.

Similarity distance computation is crucial to measure resemblance between two images. In the literature, to compare two images, the objects in an image are matched to other objects in another image and the distance between them are computed. Each object matching has a weight that corresponds to the importance of the object in the image. Larger objects maybe assumed to be the main object and therefore have a higher importance. The level of importance of an object is based on the percentage of the number of pixels in the object compared to the whole image. As a result, the objects that occupy larger areas indicate a higher importance. This may not be true for all images. In addition to object weight, in this paper we introduced a new weight for objects based on uniqueness. The calculation for uniqueness of an object is discussed later in Chapter 3.

In this research we address issues for improving image segmentation by modification of the clustering algorithm, increasing retrieval speed by pre-clustering the objects database, and improving accuracy for similarity distance computation. The performance of our approach is compared to existing region-based system.

1.4 Objectives

The objectives of this research:

1) *Develop an improved algorithm for image segmentation into objects using hierarchical and k-means clustering.* In some existing region-based retrieval system, a k-means algorithm has been successfully used to cluster similar pixels into objects/regions. However, the number of cluster centers needs to be pre-classified. In this study we use a hierarchical clustering result to determine the initial cluster centers. This method has been shown to produce better k-means clustering results compared to randomly generated initial clusters for other clustering problems, such as gene clustering (Seo, 2003).
2) *Develop an improved object clustering algorithm and an improved similarity distance computation.* To get faster retrieval speeds, we implemented a hierarchical clustering in the object database. We compared the performance between hierarchical algorithm and k-means algorithm during object clustering.
3) *Analyze query performance on a 1000 image COREL database*.
4) *Compare query performance with the well-known IRM and Fuzzy Club region-based image retrieval systems*.

CHAPTER 2: LITERATURE REVIEW

To better understand region-based image retrieval system, it is beneficial to examine some of the current literature on these topics.

2.1 Blobworld (Carson et al, 1999)

Blobworld is one of the earlier region based systems. Blobworld decomposes raw pixel data into a small set of image regions which are coherent in color and texture. Blobworld models an image as a set of regions (blobs) which are homogenous with respect to color and texture. Each blob is described by its color distribution and by its mean texture descriptors, using a 220-element feature vector (218 bin color histogram and 2 texture descriptors).

Blobworld first extracts pixel features, and then groups similar pixels together to form a region or a blob, and finally determines the feature vectors of the blobs. Each pixel is described by 8 dimensional space: three color descriptors in L*a*b color space, three texture descriptors (anisotropy, orientation, and contrast), and spatial position of the pixel (x and y axis). The pixel distribution is modeled using a mixture of two to five Gaussians distribution. To fit the mixture of Gaussian models to the pixel data, the Expectation Maximization (EM) algorithm is used (Dempster, 1977).

During a query, the user submits a query image and selects some of the blobs as regions of interest. Each blob in the query image is matched to all the blobs in the database image. A Euclidean distance function is used to calculate the distance function between the feature vectors of 2 blobs. Then the overall score is computed using weighted fuzzy-logic operators applied to the scores of matched blobs.

Finally, images are ranked according to their overall score and the *k* best matches are returned. In order to increase the speed during query, Blobworld uses the R-tree algorithm to index the color descriptors of the blob feature vectors (no texture features are taken into account during indexing).

During experiments, Blobworld was used to perform a variety of queries using a set of 10,000 images from the commercial Corel stock photo collection. Carson compared Blobworld to a global color histogram algorithm (Stricker, 1994). The result showed that Blobworld yields good results when querying for distinctive objects. Blobworld lose its performance when the image objects are not well distinguished from each other. However, Carson et al. argued that it has the potential to incorporate shape information in the region description, while global histograms do not encode the region information necessary to perform shape queries.

A disadvantage of BlobWorld (and region segmentation algorithms in general) is that the segmentation into regions may not be ideal . One object may be partitioned into several regions with none of them being representative of the object, especially for images without distinctive objects and scenes. Consequently it is often difficult for users to determine which regions and features should be used for retrieval. To resolve this problem, researchers like Li and Wang (2000) have developed similarity measures to combine and integrate information from all the regions. This way, the effect of inaccurate segmentation during similarity distance computation can be minimized.

2.2 IRM (Li and Wang, 2000)

Integrated Region Matching (IRM) allows for matching a region of one image to several regions of another image. That is, the region mapping between any two images is a many-to-many relationship. As a result the similarity between two images is defined as the weighted sum of distances is the feature space, between all regions from different images. Compared with retrieval systems based on individual regions, such as Blobworld, the IRM approach decreases the impact of inaccurate segmentation by smoothing over the imprecision in distances. IRM incorporates the properties of all the segmented regions so that information about an image can be fully used. Region-based matching is a difficult problem because of inaccurate segmentation.

To define the similarity measure, first regions in two images are matched. Being aware that segmentation cannot be perfect, the matching is "softened" by allowing one region of an image to be matched to several regions of another image. Here, a region-region match is obtained when the regions are relatively similar to each other in terms of the features extracted.

IRM first segments the image into blocks of 4x4 pixels and extracts a feature vector for each block. The k-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one region in the segmented image. Six features are used for segmentation. Three of them are color components (L*u*v color space), and the other three represent energy in high frequency bands of the wavelet transform (Daubechies-4 wavelet transform to the L component of the image).

To increase the robustness against segmentation errors, IRM allows a region to be matched to several regions in another image. Each of the matching is assigned with a significance credit which corresponds to the importance of the matching. There are several ways to assign the importance of a region. One can assume that every region is equally important. IRM views that important objects in an image tend to occupy larger areas, called an area percentage scheme. This scheme is less sensitive to inaccurate segmentation than the uniform scheme. If one object is partitioned into several regions, the uniform scheme raises its significance improperly, whereas the area percentage scheme retains its significance to the region.

The authors compared IRM with the WBIIS (Wavelet-Based Image Indexing and Searching) system (Wang et al, 1998) using the same image database. WBIIS forms image signatures using wavelet coefficients in the lower frequency bands and performs well with relatively smooth images, such as most landscape images. For images containing detail semantics, such as pictures containing people, the performance of WBIIS degrades. In general, the IRM system performed well both in smooth landscape images and images composed of fine details.

Image matching is performed on a COREL database containing 200,000 images. Precision was computed for both IRM and WBIIS. IRM system has been shown to be exceptionally robust to image alteration such as intensity variation, sharpness variation, intentional color distortion, intentional shape distortions, cropping, shifting and rotation.

2.3 Fuzzy Club (Zhang, 2002)

Fuzzy Club addresses the issue of effective and efficient content based image retrieval by presenting an indexing and retrieval system that integrates color, texture and shape information for the indexing and retrieval, and applies these features regions obtained through unsupervised segmentation, as opposed to applying them to the whole image domain.

Fuzzy Club emphasizes improving on a color feature “inaccuracy” problem in the region based literature – that color histogram bins are not independent. For instance, if the color spectrum is divided into 10 bins, these bins are not independent – some are closer or farther away from each other in the original color space. Fuzzy logic is applied to the traditional color histogram to solve this problem to some degree.

Fuzzy Club first segments an image is segmented into regions of 4x4 blocks and extract color and texture features on each block. The k-means algorithm is used to cluster similar pixels together to form a region. The LAB color space is used to extract color features and Haar wavelet transform is used to extract three texture features.

The k-means algorithm does not specify how many clusters to choose. The number of *k* is started with *k*=2 and stop increasing k if one of the following conditions is satisfied: the distortion of the distance between pixel to the cluster center is below a specified threshold or the number of k exceeds an upper bound. The average number of clusters for all images in the database varies in accordance with the adjustment of the stop constraint.

After segmentation, Fuzzy Club is ready for image indexing. Image indexing is based on the features defined in the regions obtained from the image segmentation. Within each region, three types of features are defined: color, texture, and shape, along with the conventional geometric information as the feature vector for image indexing. The distance between two regions is computed by applying Euclidean distance metric to fuzzy color histogram, texture vector, and shape vector, respectively.

A secondary clustering is performed to reduce query processing time. Regions with similar features are grouped together in the same class. This secondary clustering is performed offline, and each region's indexing data along with its associated class ID is recorded in the index files.

During the querying processes, the query image is segmented to obtain all the regions. The distances between each query region and all class centroids in the database is computed to determine which class of these query regions belong. The similar regions in the database are returned and all the member regions in the image set are retrieved. The query image is compared to the image sets where the query regions were retrieved during region retrieval. Finally, the global distance of the query image is compared with the images in the retrieved image set.

2.4 Geometric Histogram (Rao, et. al, 2000)

Geometric histogram generalized the color spatial distribution by computing the color histogram with specific geometric relationship between pixels of each color histogram bucket. The concept is a unification of some existing techniques such as color density maps, color correlogram and color tuples. A color density map is a set of pixels with the same color that considered as a geometric subset of the 2-D plane. The centroid and the radius of the subset is calculated and the number of pixels in each of the regions is computed to form a vector called the density map of color. The map is arranged of all colors into a matrix by making the density vector of each color as a row. On the other hand, color correlogram is a vector of three indices, say V_{ijk}, standing for the number of pixel pairs of distance k with colors i and j, respectively, where i and j run over all colors while k runs over the pre-defined possible choices of the distance between two pixels.

Geometric histogram is almost the same with region based algorithm. In Geometric histograms, the frequency of the arrangement of color subset and the list of geometric configurations is calculated. To simplify, let $G = \{g_1, g_2, \ldots, g_p\}$ be a list of regular geometric configurations, let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of colors after the quantization of images. For each $g_i \in G$, arrange a subset of C according to the configuration of g_i, then calculate the frequency of such an arrangement across the whole image (Rao, 2000). These frequencies formed a vector and called as geometric histogram of the image with respect to G. Here, G is the set of "rulers" used to measure the local arrangements of colors across the whole image.

The main disadvantage of geometric histogram is because there are too many choices and of possibilities of arranging colors in a given configuration. To solve this problem, the image database must be pre-categorized therefore by using specific geometric configurations for specific database can maximize the performance to arrange color with a given configuration.

2.4 Research Issues

There are two basic steps being taken to extract features from an image. The first step is feature selection. There are many different types of features that can be extracted from each image. For example, Li and Wang (2000) suggested using the LUV color space for color features and Daubechies-4 wavelet transform for texture features while Zhang (2002) suggested using the LAB color space for color features and Haar wavelet transform for texture features. The second step is pixel clustering to form objects/regions. Li and Wang (2000) and Zhang (2002) both proposed to use K-means clustering to group similar pixels. The number of cluster centers is gradually increased and stopped when a criterion is met.

Zhang (2002) proposed to perform a secondary cluster to the object database to increase speed during retrieval. On the contrary, Carson (1999) and Li's (2001) IRM doesn't use any object clustering in order to increase retrieval speed. Instead, the query images are compared to all images in the database. All proposed methods have their own disadvantages and advantages.

To compute the similarity distance between images, Wang (1998) proposed to add a significance matrix into the similarity distance algorithm. The significance matrix is determined by the importance of the object which means the area percentage on an object to the entire image. On the other hand, Zhang (2002) suggested only matching regions that have the smallest distance to other regions in another image during similarity distance computation. Additionally, Zhang also added the weight during the matching process to determine the importance of the object compared to the entire image.

CHAPTER 3: METHODOLOGY

The three main tasks of our region based image retrieval system are:

1) Image segmentation to obtain objects/regions. Images are segmented by grouping pixels with similar descriptions to form objects/regions.
2) Object clustering for faster retrieval.
3) Similarity distance computation between the image query and database.

Figure 3.1 provides an overview of the system architecture developed in this study. The steps in the process are detailed in the following sections.

Feature selection during image segmentation is a crucial step. A specific color space and texture analysis is selected to increase the performance during segmentation. Every pixel on the image is clustered using a modified k-means algorithm to group similar pixel together to form objects. To increase the retrieval speed during query, similar objects are clustered using a hierarchical clustering algorithm. The new similarity distance algorithm is introduced to minimize error obtained during image segmentation. Finally, the accuracy during retrieval is computed and compared against IRM, Geometric Histogram, and Fuzzy Club system.

3.1 Feature Extraction

In this study we follow an approach similar to Li and Wang (2000) and Zhang (2002). The image is partitioned into 4 by 4 blocks, a size that provides a compromise between texture granularity, segmentation coarseness, and computation time. As part of pre=processing, each 4x4 block is replaced by a single block containing the average value ever the 4 by 4 block. This way, we still have a good texture granularity while reducing the number of total pixels per image, therefore decreasing the computation time.

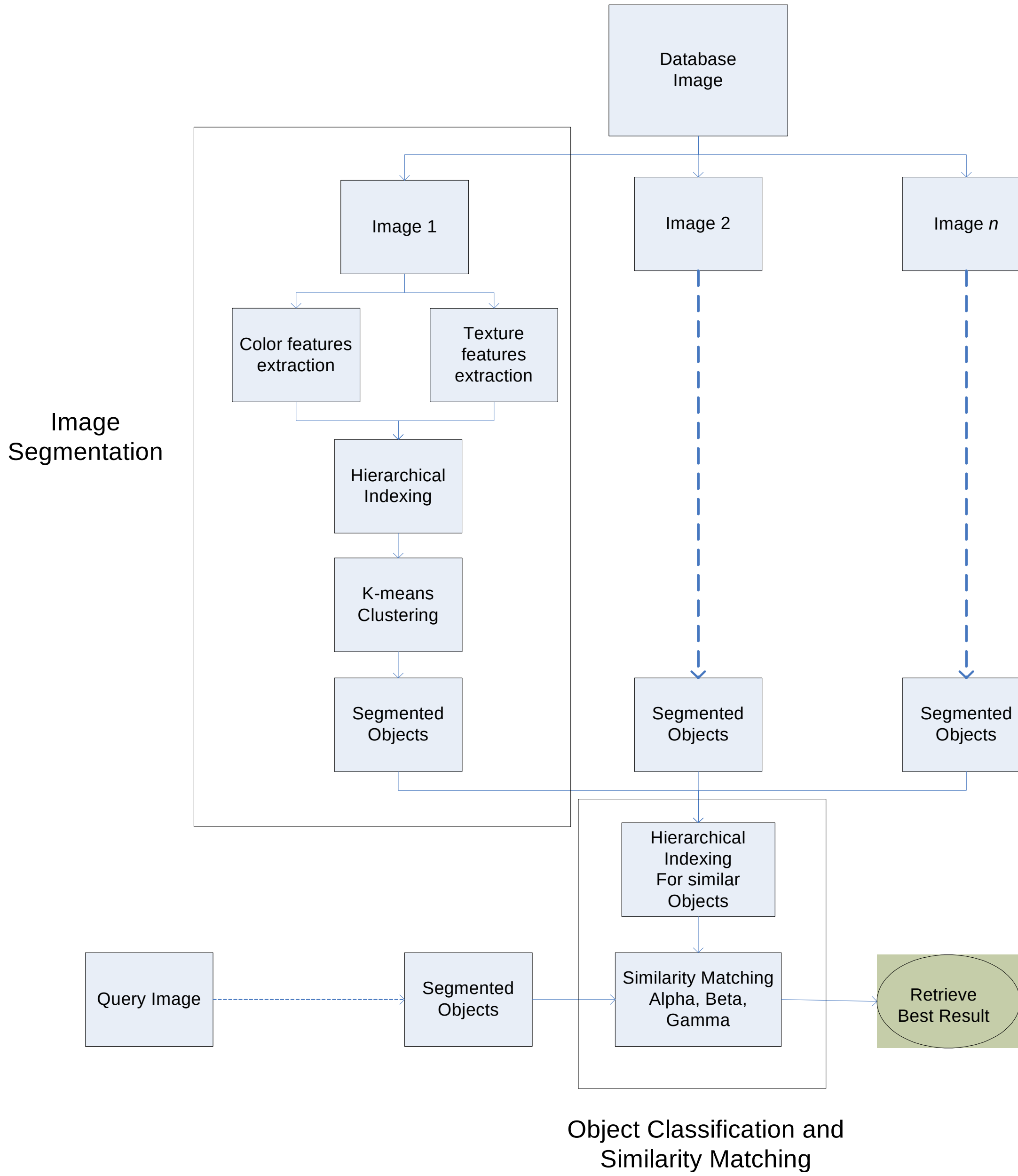


Figure 3.1: Block diagram of Image Retrieval System

To segment an image into objects, six features are extracted from each block (Figure 3.2). Three features are color features, and the other three are texture features. The HSV color space is selected during color feature extraction due to its ability for easy transformation from RGB to HSV and vice versa. Since HSV color space is natural and approximately perceptually uniform, the quantization of HSV can produce a collection of colors that is also compact and complete. These features are denoted as {F1, F2, and F3}.

To obtain the other three features, we apply the Haar wavelet transform to the L component of the image. The Haar wavelet is discontinuous and resembles a step function. It represents the energy in high frequency bands of the Haar wavelet transform (Daubechies, 1992). After a one-level wavelet transform, a 4 by 4 block is decomposed into four frequency bands, each band containing a 2 by 2 matrix of coefficients. Suppose the coefficients in the HL band are $\{c_{k+i}, c_{k,l+1}, c_{k+1,l}, c_{k+1,l+1}\}$. Then, the feature of the block in the HL band is computed as:

$$f = \left(\frac{1}{4} \sum_{i=0}^{1} \sum_{j=0}^{1} c_{k+i,l+j}^{2} \right)^{\frac{1}{2}} \tag{3.1}$$

The other two features are computed similarly in the LH and HH bands. These three features of the block are denoted as {F4, F5, and F6}.

In this paper, we did not consider shape features during similarity distance computation. Li and Wang (2000) considered shape features into IRM distance computation only for textured images, while non-textured images considered the shape features. A textured image defined as an image of a surface, a pattern of similarity shaped objects, or an essential element of an object. To do this, Li and Wang manually pre-

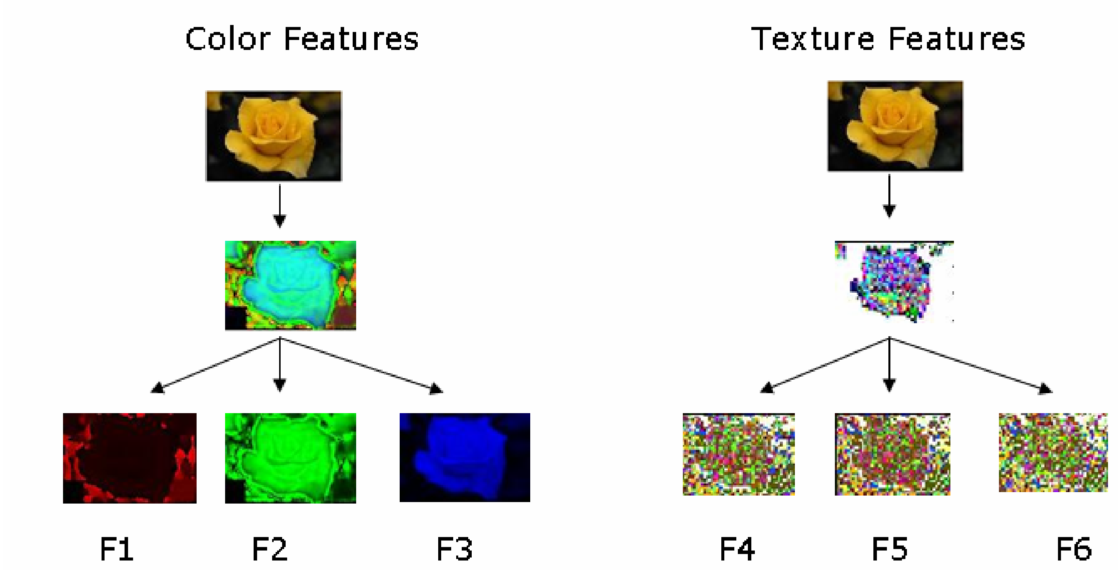


Figure 3.2: Color and texture feature extraction

classified images in the database into texture and non-texture images. The IRM distance computation is different between textured images and non-textured images. In this paper, to avoid manual pre-classification, we computed the similarity distance for texture and non-texture images using the same formula.

3.2 Pixel Segmentation

After obtaining these six features from all pixels on the image and store the information in an array (Table 3.1), we perform a modified k-means clustering to group similar pixel together and form objects.

Suppose the member pixels of an image are $\{x_i : i = 1,..., L\}$. The goal of k-means algorithm is to partition the observations into *k* groups with cluster centers $\bar{x}_1, \bar{x}_2,..., \bar{x}_k$ that minimize the square error ($\bar{x}_j$ is the mean of the cluster). The square error is defined below:

$$D(k) = \sum_{i=1}^{L} \min_{1 \le j \le k} (x_i - \bar{x}_j)^2 \tag{3.2}$$

The advantage of K-means algorithm is that it works well when clusters are not well separated from each other (Guha, 1998), which is frequently encountered in images. However, k-means requires the user to specify the initial cluster centers. In this paper, we perform hierarchical algorithm to the pixel image to obtain the initial cluster centers. This method is known to produce better clustering result compare to randomly generated initial clusters (Seo, 2003).

Hierarchical clustering first merges pair of the closest pixels to form cluster. These clusters are then merged to generate a new bigger cluster and finally form a single cluster that covers the whole image.

Table 3.1: An array table of feature information of all pixels in the image

Name	**F1**	**F2**	**F3**	**F4**	**F4**	**F6**
Pixel 1	H1	S1	V1	Ht1	D1	Vt1
Pixel 2	H2	S2	V2	Ht2	D2	Vt2
…	…	…	…	…	…	…
Pixel n	H_n	S_n	V_n	Ht_n	D_n	Vt_n

To calculate the distances between the new cluster and remaining clusters we use the average linkage method (Seo, 2003). Let C_n be a new cluster produced by merging of clusters C_i and C_j. Let C_k be a remaining cluster. The distances between the new cluster and the remaining clusters are:

$$DIST(C_n, C_k) = \frac{|C_i|}{|C_i| + |C_j|} DIST(C_i, C_k) + \frac{|C_i|}{|C_i| + |C_j|} DIST(C_j, C_k) \qquad (3.3)$$

The distance between each pixel is obtained through the Pearson correlation coefficient (Seo, 2003). In our image database, compared to Euclidean distance, the Pearson correlation coefficient has been shown to produce a better clustering result during image segmentation. Let p_i be the features of pixel 1 and q_i be pixel 2 (where i = 1,2,…,6). Using the Pearson correlation coefficient, the distance between pixel p and pixel q is:

$$Dist\,(p,q) = w_c \frac{\sum_{i=1}^{3} p_i q_i + - \frac{\sum_{i=1}^{3} p_i \sum_{i=1}^{3} q_i}{N}}{\sqrt{\left(\sum_{i=1}^{3} p_i^{\,2} - \frac{(\sum_{i=1}^{3} p_i)^2}{N}\right)\left(\sum_{i=1}^{3} q_i^{\,2} - \frac{(\sum_{i=1}^{3} q_i)^2}{N}\right)}} +$$

$$w_t \frac{\sum_{i=4}^{6} p_i q_i + - \frac{\sum_{i=4}^{6} p_i \sum_{i=4}^{6} q_i}{N}}{\sqrt{\left(\sum_{i=4}^{6} p_i^{\,2} - \frac{(\sum_{i=1}^{6} p_i)^2}{N}\right)\left(\sum_{i=4}^{6} q_i^{\,2} - \frac{(\sum_{i=1}^{6} q_i)^2}{N}\right)}} \qquad (3.4)$$

where N is the number of features. Since the number of features per pixel is 6, so N = 6. We put different weight between color (w_c) and texture (w_t). Zhang's FuzzyClub (Zhang,

2002) also put different weight between color and texture since it produced better clustering result.

3.2 Object Clustering and Similarity Distance Computation

In order to increase the retrieval speed during query, we cluster similar objects to form classes. Many existing retrieval system try to compare the query image to all images in the database. This results in a high computational cost, especially when the database is large. To solve the problem we perform hierarchical clustering to all the objects obtained in previous step. Each object described by 6 features which are the average features of all the member pixels. This information is stored in an array database (Table 3.2).

The query image goes through the same image segmentation algorithm with image database to obtain objects. These objects are compared to the cluster centers in the database above, and the overall similarity is calculated using L2 distance. The object in the database that has a minimum distance will be returned to perform global image distance computation between query image and database image (Figure 3.4).

Let say image 1 is the query image and the returned image database is image 2 (Figure 3.5). To compute the overall distance computation between the query image and the returned image, first we match all the objects from the two images.

Given two matched objects, the first object from image 1 (O_{11}) and the first object in image 2 (O_{21}), the object distance is defined as:

$$O_{11,21} = \sum_{i=1}^{6} (f_{11i} - f_{21i})^2 \tag{3.5}$$

where f_i is the color and texture features from each object. The similarity distance computation between image 1 (Im1) and image 2 (Im2) can be defined as:

Table 3.2: Feature information for each object in an image

Name	**F1**	**F2**	**F3**	**F4**	**F4**	**F6**
Image1Object1	H11	S11	V11	Ht11	D11	Vt11
Image1Object2	H12	S12	V12	Ht12	D12	Vt12
Image2 Object1	H21	S21	V21	Ht21	D21	Vt21
Image2Object2	H22	S22	V22	Ht22	D22	Vt22
…	…	…	…	…	…	…
Image$_n$Object$_i$	H_{ni}	S_{ni}	V_{ni}	Ht_{ni}	D_{ni}	Vt_{ni}

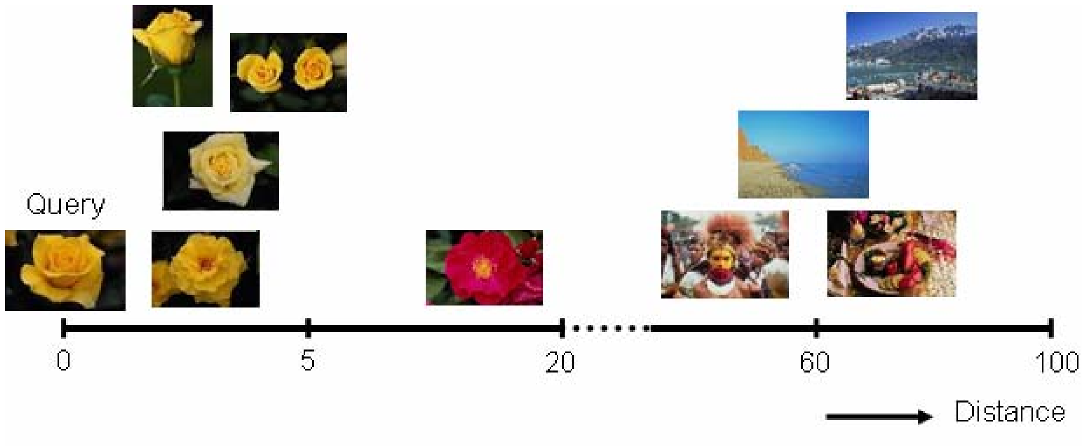


Figure 3.4: Similarity distance computation between image query and database

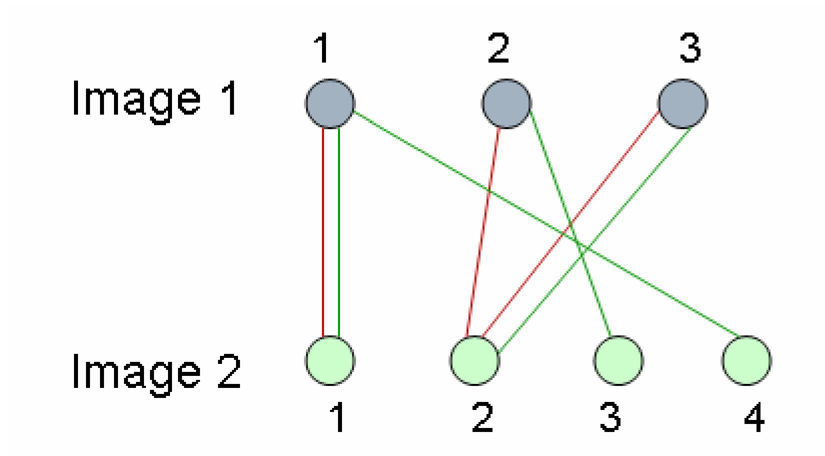


Figure 3.5: Overall similarity distance between image query and image database

$$D(Im1, Im2) = \sum_{r,t} w_r O_{r,t} + \sum_{r,t} w_r O_{r,t} \quad (3.6)$$

consequently, the distance between image 2 and image 1 is defined as:

$$D(Im2, Im1) = \sum_{r,t} w_t O_{t,r} + \sum_{r,t} w_t O_{t,r} \quad (3.7)$$

where w_r and w_t is the weight given for the object and $O_{r,t}$ is the object distance between object *r* in image 1 and object *t* in image2. The weight for each object is defined as:

$$w_p = \alpha_p \beta_p \gamma_p \quad (3.8)$$

where:

α_p = 1 when the two region being match is the closest distance

α_p = 0 when the two region being matched is not the closest distance

β_p = (# of pixel for object *p*) / (# of pixel in the whole image)

γ_p = Object uniqueness of the object based on the appearance of the object

The uniqueness of the object is based on the appearance of the object in the object cluster. When an object appear a more often in the database, it considers to be less unique and vice versa. To get the value of γ_p, first we perform hierarchical clustering to cluster similar object in the same group. The uniqueness of object *p* (γ_p) is obtained through the percentage of the number of objects that belong to the cluster compared to the total number of objects in the database. The similarity distance between D*(Im1, Im2)* and D*(Im2, Im1)* is not symmetric, therefore in order to make it symmetric, we take the average between them. The overall distance is defined as:

$$\text{Overall } D(Im1, Im2) = \frac{D(\mathrm{Im1}, \mathrm{Im2}) + D(\mathrm{Im2}, \mathrm{Im1})}{2} \quad (3.9)$$

Given this definition, it is straightforward to compute D(*ImP*,*ImQ*) for every images in the database. This definition of the overall similarity between two images captured by the overall distance between the images is a balanced scheme in similarity measure between regional and global matching.

3.3 Implementation Issues

The image retrieval system is implemented using MATLAB image processing tools and statistical tools. For hierarchical and k-means clustering we use Clustering Explorer 2.0 developed by the computer science department in University of Maryland (Seo, 2003). We use a general-purpose image database containing 1000 images from COREL. These images are pre-categorized into 10 groups: African people, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountains and glaciers, and food (Figure 3.6). All images have the size of 384x256 and 256x386.

During the implementation, we use a platform of Pentium 3.06 GHz CPU with 1G RAM. 1000 image database went through image segmentation algorithm to obtain more than 5800 objects. These images are manually divided into 10 classes such as African people, busses, building, and flowers. Feature extraction time for the whole database takes 20-25 minutes using MATLAB software corresponding to about 1.5 second for each image. The data from feature extraction is fed into a program called Hierarchical Clustering Explorer (HCE) to perform pixel clustering a modified k-means algorithm as mentioned in section 3.1. The pixel clustering time to obtain objects in each image takes 1 to 2 second using HCE.

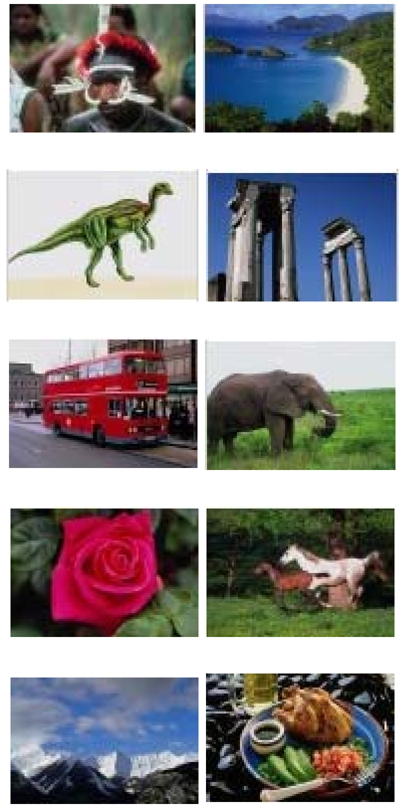


Figure 3.6: Images are pre-categorized into 10 groups

Once all images in the database are extracted, HCE will perform the second clustering to group similar objects. The object clustering time for the whole database takes 2-3 seconds. The data obtained from this clustering is stored in Microsoft SQL database.

To compute the overall similarity distance computation each object is matched to every other object in the database and calculates their distances. Since we have more than 5800 objects in the database, the object matching computation consists of more than 6.5 million combinations. The entire computation takes 15-17 minutes using Microsoft SQL Query Analyzer. The average query time for returning top 20 images per query is less than one second using Microsoft SQL Query Analyzer.

3.3 Experimental Plan

To implement our system, we store 1000 images from COREL on a computer. MATLAB image processing tools extract the pixel features on each image. Next, Hierarchical Clustering Explorer (HCE) clusters these pixels by grouping pixels that have similar features to form objects.

During experiments, to calculate the similarity distance between pixels we use Equation (3.4) and set different weight for color and texture features, where w_c=0.65 and w_t=0.35. We put this parameter between color and texture because this combination has been shown in FuzzyClub (Zhang, 2002) to produce a good clustering result.

3.3.1 Experiment #1 – Similarity Distance Measure

To set a proper number of objects per image during hierarchical clustering, we compare a similarity measure against a threshold value (Figure 3.7). This value compares the length

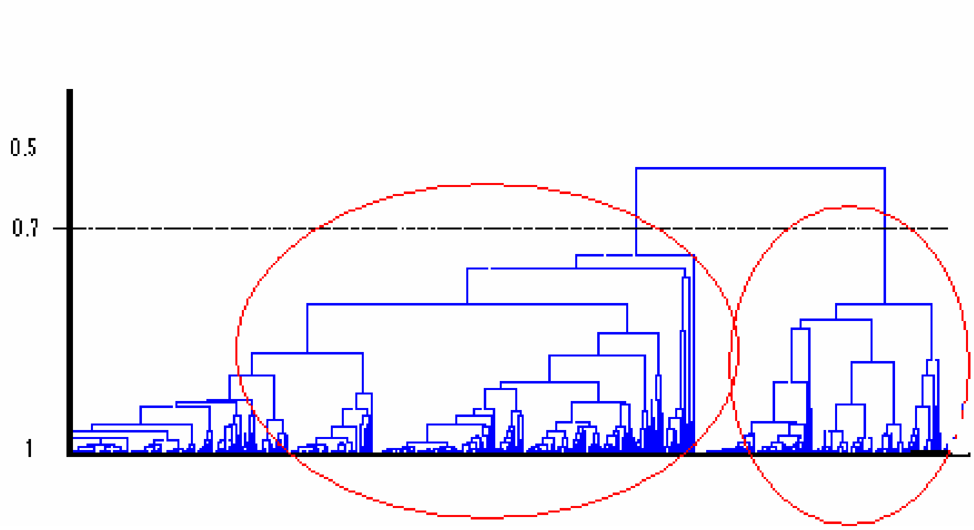


Figure 3.7: Similarity measure to obtain cluster during hierarchical clustering

of a link in a cluster hierarchy with the average length of neighboring links. Using the similarity measure value, pixels that are considered to be similar form a separate cluster.

During the experiment we analyze different values of similarity measure. We noticed that when similarity measure value is ≤ than 0.5, the number of clusters was very low, mostly one or two clusters. When the similarity value is ≥ 0.9 the number of clusters was very high typically greater than 15. Therefore, to analyze the similarity values, we choose three different measures at 0.6, 0.7 and 0.8. We randomly choose 10 images from each image class which corresponds to 100 images from the database. From each image, during the pixel clustering, we set the similarity measure to be 0.6, 0.7 and 0.8. The number of cluster based on these measures is recorded and the average number of cluster for each similarity measure is computed.

3.3.2 Experiment #2 – Object Uniqueness

After obtaining all object features for all images, object clustering is performed using Hierarchical clustering algorithm to obtain object groups. This cluster of object groups is necessary for two reasons: for faster image retrieval, and for determining the value of object uniqueness. The value of object uniqueness depends on the quantity of objects in a cluster because the larger the quantity in a cluster corresponds to a smaller value of object uniqueness. This way, object uniqueness is related to the result of object clustering. Therefore, to set the value of object uniqueness and get a better accuracy during retrieval, we need to find a suitable number of clusters during object clustering. To do this, we set four different numbers of object cluster for 1, 10, 25 and 35, and evaluate the accuracy during image retrieval.

3.3.2 Experiment #3 – Performance Comparison

The best result from the object cluster in Experiment #2 are then compared to three existing algorithm: IRM, FuzzyClub, and Geometric Histogram. Zhang (2002) already have the performance comparison between FuzzyClub, IRM and Geometric Histogram. In this paper, we use this performance comparison and compare it against our algorithm. In order to calculate the performance, we use the same approach as Zhang (2002). For each category in the 1000 database images, we randomly select 30 images as queries. Since we have 10 categories in the database, we have 300 query images. For each query, we examined the precision of the retrieval based on the relevance of the semantic meaning between the query and the retrieved images. Each of the 10 categories in the database portrays a distinct semantic topic, therefore this assumption is reasonable to calculate the precision. The average precisions for each groups based on the returned top 20 images were recorded. Since the number of relevant images in the database for each query image is the same, we do not calculate the recall explicitly since it's proportional to the precision in this case. The experiment ran once for each algorithm.

3.3.2 Experiment #4 – Query Return Size

To further determine our system's performance, we perform another evaluation where we took each of the 100 images on each image class as the image query, and return the top 10, 20, and 30 images from the database. The accuracy during retrieval is returned and we compare the accuracy using different return sizes. The result of this experiment is to show where the images that have similar semantic meaning to the query fall within the result.

CHAPTER 4: DEMONSTRATION

For the demonstration, we use 6 images to perform image retrieval system. The whole database contains 5 images and one query image. These images will undergo image segmentation, object clustering, and similarity distance. Three of the images are yellow flowers, one red flower, and one African people. The query image is a yellow flower (Figure 4.1).

Section 4.1 discusses the steps of image segmentation. First, six features are extracted on each pixel in the image using MATLAB image processing tools. The output of this feature extraction is stored on an array which then fed into Hierarchical Clustering Explorer (HCE) for pixel clustering to obtain objects. Each object is described as the average value of the member pixels that belong to the cluster.

Section 4.2 discusses the steps for image similarity distance computation. All objects are clustered to obtain object groups or object classes and the result is stored in an array. The array object cluster and all of the object features are stored in Microsoft SQL database to perform the distance computation.

4.1 Image Segmentation

To show an example of image segmentation, we use image *q* to show the algorithm. The following are the steps to segment the image:

1. Using MATLAB image processing tools, the image is partitioned into 4x4 block. The MATLAB code is provided Figure 4.2.
2. Color and texture features from each pixel are extracted using MATLAB image processing tools. The MATLAB code is provided in Figure 4.3.

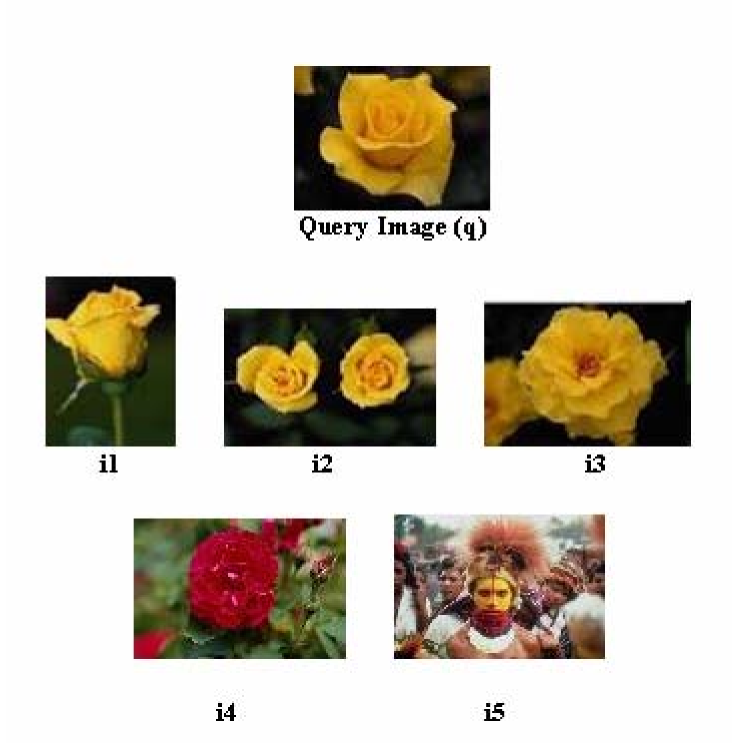


Figure 4.1: Images used in example

```
%Image partition
% read the image into MATLAB
image = imread('126.jpg'); %read the image of 50.jpg from database
image1 = image(:,:,1); %separate the array to have 2 dimensional matrix
image2 = image(:,:,2); %separate the array to have 2 dimensional matrix
image3 = image(:,:,3); %separate the array to have 2 dimensional matrix

%a function 'f' to calculate the mean of the block matrix operation
f = inline('uint8(round(mean2(x)))');

image1block=blkproc(image1, [4 4],f); %partition the image to 4-by-4
image2block=blkproc(image2, [4 4],f); %partition the image to 4-by-4
image3block=blkproc(image3, [4 4],f); %partition the image to 4-by-4

%concatenate the matrices to be a 3 dimensional again
imageblock = cat(3,image1block,image2block,image3block)
```

Figure 4.2: MATLAB code for image partition

```
% Color features extraction
% Convert to HSV color space
imageHSV = rgb2hsv (imageblock);

% Generating X,Y,Z plot for HSV color features
imageHSV=rgb2hsv(imageblock); % change image to HSV color space
X=imageHSV; % to simplify the name, we change the name of the image to be X
mX1 = X(:,:,1); % we take all of the first element
mY1 = X(:,:,2); % we take all of the second element
mZ1 = X(:,:,3); % we take all of the third element

X1 = mX1(1:end); % we make it into one vector matrix
Y1 = mY1(1:end);
Z1 = mZ1(1:end);

% now we need to concenate these X1, Y1, Z1 into a single 3 dimensional matrix
imX1=[X1;Y1;Z1];
imCX=imX1';

%% Texture features extraction
%change to indexed image using colorcube
imageColorCube=rgb2ind(imageblock,colorcube(256));
figure, imshow(imageColorCube,colorcube);

%transform using Haar wavelet transform
%perform one step decomposition (level one decomposition) using db1
[cA1,cH1,cV1,cD1] = dwt2(imageColorCube,'db1');

%transform back to display image coding
%(constructing the level-one approximation)
colormap(colorcube);
subplot(2,2,1); image(wcodemat(A1,256));
title('Approximation A1')
subplot(2,2,2); image(wcodemat(H1,256));
title('Horizontal Detail H1')
subplot(2,2,3); image(wcodemat(V1,256));
title('Vertical Detail V1')
subplot(2,2,4); image(wcodemat(D1,256));

%To construct the level-one approximation and
%details (A1, H1, V1, and D1) from the coefficients cA1, cH1,
% cV1, and cD1, type:
A1 = upcoef2('a',cA1,'db1',1);
H1 = upcoef2('h',cH1,'db1',1);
V1 = upcoef2('v',cV1,'db1',1);
D1 = upcoef2('d',cD1,'db1',1);

% resize coefficient matrices to original size
cA=imresize(cA1,2); cD=imresize(cD1,2);
cH=imresize(cH1,2); cV=imresize(cV1,2);

%XT image with texture features
%concatenate cD,cH,cV into one matrix from 96x64 each, become 96x64x3
XT=cat(3, cD, cH, cV);
tX1 = XT(:,:,1); % we take all of the first element
tY1 = XT(:,:,2); % we take all of the second element
tZ1 = XT(:,:,3); % we take all of the third element
X1 = tX1(1:end); % we make it into one vector matrix
Y1 = tY1(1:end);
Z1 = tZ1(1:end);

% now we need to concenate these X1, Y1, Z1 into a single 3 dimensional matrix
imTX1=[X1; Y1; Z1];
imTX=imTX1';
```

Figure 4.3: MATLAB code for features extraction

3. The output of MATLAB code in step two are saved in text file as an array containing 6 columns of color and texture features, and 6144 rows of the total number of pixel on each image.
4. These 6144 pixels are clustered using modified k-means to group similar features together and form regions/objects. During pixel similarity distance computation, color texture has a weight of 0.65 and texture feature 0.35. Clustering Explorer 2.0 is used to perform this clustering. (Table 4.1)
5. All of the images are segmented to form objects. Every object has 6 features which obtained from the average value of the member pixels. The "object importance" or the object weight (β_p) is computed based on the percentage of the number of pixels in the object compared to the total number of pixels in the image. The features of each object and the object weight (β_p) are stored in array Table 3.5.

4.2 Object Clustering and Similarity Distance Computation

In order to cluster all objects, including the query objects, hierarchical clustering algorithm is performed and the result is shown in Figure 4.4.

Objects that are similar to the entire query image (*i1,, i2,…,i5*) are retrieved. For the simplicity of this example, let's say every object are retrieved and considered similar to the query images. Now, the overall similarity distance of image *q* is compared to all the images (*i1, i2,…,i5*) using Equations (3.5) through (3.9).

To get the value of the uniqueness (γ_p) of the object, we perform hierarchical clustering to class similar objects in the same group. In this demonstration, for simplicity,

Table 4.1: Object weight and features for all images

Image/ Object	# of pixels	Weight	F1	F2	F3	F4	F5	F6
q/1	3024	0.492	0.1136	0.892	0.7122	-0.0091	-0.0124	0.0148
q/2	963	0.157	0.1246	0.5083	0.1321	-0.037	-0.0101	-0.1044
q/3	966	0.157	0.4355	0.247	0.0929	0.0404	-0.048	0.0176
q/4	345	0.056	0.3222	0.2119	0.0684	-0.0345	-0.0034	-0.0088
q/5	433	0.070	0.0487	0.1023	0.0557	-0.0478	0.0239	0.1649
q/6	227	0.037	0.0729	0.1848	0.0846	0.178	0.174	-0.0533
q/7	186	0.030	0.0072	0.0012	0.0261	0.0856	0.0519	-0.0191
i1/1	1131	0.184	0.1282	0.8436	0.8497	0.0059	0.0024	-0.0016
i1/2	624	0.102	0.1437	0.749	0.5514	-0.0166	0.0079	-0.0046
i1/3	433	0.070	0.236	0.6804	0.1917	-0.3852	-0.3918	-0.3932
i1/4	366	0.060	0.1978	0.5848	0.1145	-0.0016	-0.0101	-0.0174
i1/5	541	0.088	0.1944	0.5838	0.1432	-0.5509	0.0278	0
i1/6	1592	0.259	0.2331	0.6786	0.178	0.0958	0.1839	-0.0243
i1/7	186	0.030	0.2396	0.7094	0.1924	-0.0583	-0.8538	0.0548
i1/8	424	0.069	0.2346	0.6735	0.2045	0.4794	-0.0628	0.2672
i1/9	153	0.025	0.2344	0.6908	0.2523	0.5427	-0.1059	-0.1043
i1/10	964	0.157	0.3435	0.1561	0.0593	0.042	0.0218	0.0414
i2/1	989	0.161	0.2637	0.3827	0.0915	-0.0092	-0.0054	-0.0134
i2/2	2005	0.326	0.3481	0.4084	0.1365	-0.0367	-0.0323	-0.0297
i2/3	152	0.025	0.116	0.9158	0.8762	0.0824	-0.0162	-0.0173
i2/4	212	0.035	0.1134	0.9211	0.8581	-0.0377	-0.068	0.0638
i2/5	288	0.047	0.1144	0.9429	0.773	-0.0013	-0.0069	-0.0014
i2/6	150	0.024	0.1191	0.8683	0.8161	-0.0893	0.0154	0.0425
i2/7	120	0.020	0.1185	0.8536	0.804	-0.0424	0.0605	-0.0462
i2/8	1812	0.295	0.1227	0.5168	0.4211	-0.0094	0.0441	-0.0365
i2/9	416	0.068	0.1163	0.2774	0.1031	0.2052	-0.0877	0.2794
i3/1	3931	0.640	0.1165	0.8899	0.6244	0.0049	-0.0122	-0.0132
i3/2	790	0.129	0.1377	0.3356	0.0356	-0.0122	-0.0095	-0.0225
i3/3	504	0.082	0.4084	0.2542	0.0432	-0.0104	0.0557	0.0226
i3/4	322	0.052	0.196	0.1521	0.0251	0.036	-0.0458	0.0254
i3/5	310	0.050	0.0077	0.0292	0.0191	0.0858	0.0283	-0.0194
i3/6	287	0.047	0.0021	0.0061	0.0172	-0.0385	0.082	0.0377
i4/1	1906	0.310	0.1719	0.6276	0.3714	-0.0069	-0.0116	-0.0109
i4/2	758	0.123	0.7713	0.8015	0.6066	-0.0382	-0.0118	-0.0028
i4/3	2816	0.458	0.5367	0.6819	0.4424	-0.0943	-0.0231	-0.0108
i4/4	664	0.108	0.2045	0.5417	0.3473	0.5632	0.1453	0.1577
i5/1	1535	0.250	0.057	0.4776	0.6952	-0.0005	0.0184	0.0039
i5/2	2155	0.351	0.1514	0.3402	0.5939	-0.0599	-0.0009	-0.0739
i5/3	448	0.073	0.2155	0.139	0.6338	0.0067	-0.2032	0.0334
i5/4	436	0.071	0.1581	0.1781	0.6738	0.2312	0.1211	-0.0503
i5/5	371	0.060	0.0694	0.4106	0.3818	-0.0606	-0.2778	0.388
i5/6	310	0.050	0.1884	0.268	0.4076	-0.2427	0.4241	0.0763
i5/7	441	0.072	0.5299	0.4656	0.2802	0.2245	0.095	0.0318
i5/8	448	0.073	0.8866	0.3479	0.4004	0.0808	-0.0357	-0.0393

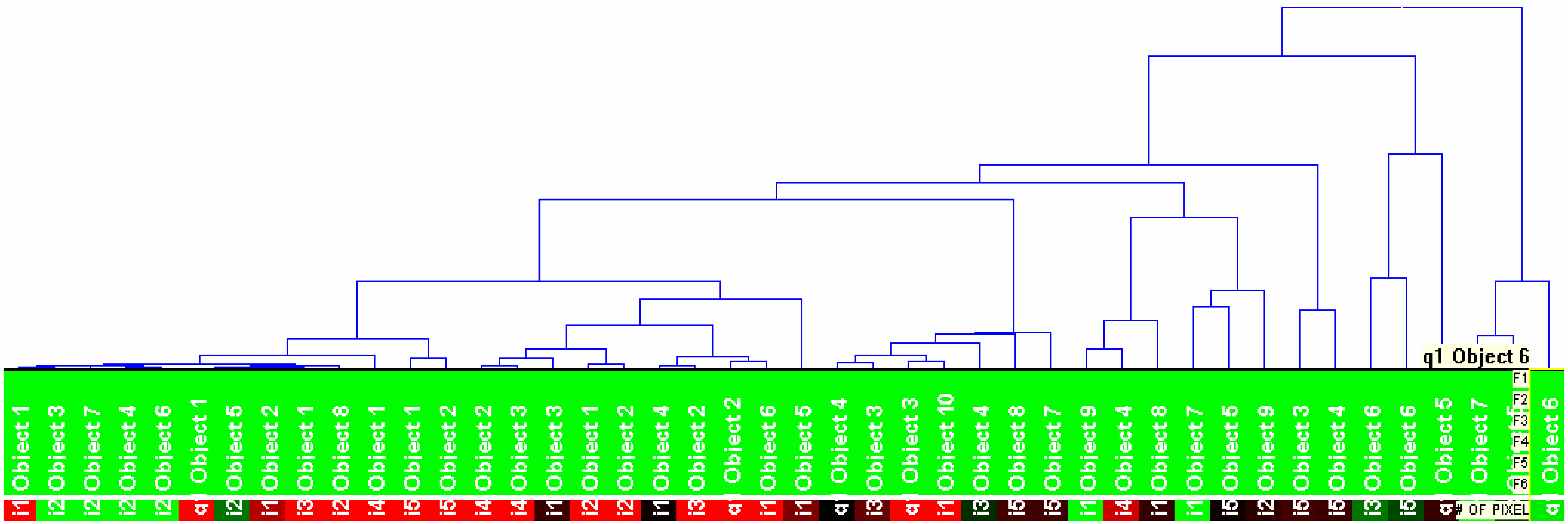


Figure 4.4: Object clustering using Hierarchical algorithm

we didn't optimize the number of clusters (such as experiment #2 in section 3.3.2) to get the best value of object uniqueness; we assume that the number of clusters is equal to five. This means we group all objects in to five different clusters. Objects that belong to a larger cluster, where the cluster has more objects compare to other clusters, are considered to be less unique and objects that belong to a smaller cluster, where the cluster has less objects compare to other clusters, are considered to be more unique.

Figure 4.5 and 4.6 shows the result of object clustering from image *q* and image *i2*. Each color represents an object. During image segmentation, image *q* is segmented into 7 objects and image *i2* into 9 objects. To calculate the similarity distance between object, each object on image *q* selects an object on image *i2* that has the minimum distance. Correspondingly, each object on image *i2* selects an object on image *q* that has the minimum distance (Figure 4.6). In Figure 4.7, the blue line corresponds to the minimum distance from an object in image *q* to object in image *i2*, whereas, the red line corresponds to the minimum distance from an object in image *i2* to object in image *q*. These distances are then added and divided by two to get the symmetric distance between image *q* and *i2*.

The object cluster and the uniqueness of the object are stored in array Table 4.2 and 4.3. The example of object cluster is shown in Figure 4.8. The similarity distances between images are computed and the result is shown in Figure 4.9 below.

The output of this example shows that the overall-similarity-distance between the query image and images that have similar semantic meaning have smaller distance compared to non similar semantic meaning.

Image *q (original)*

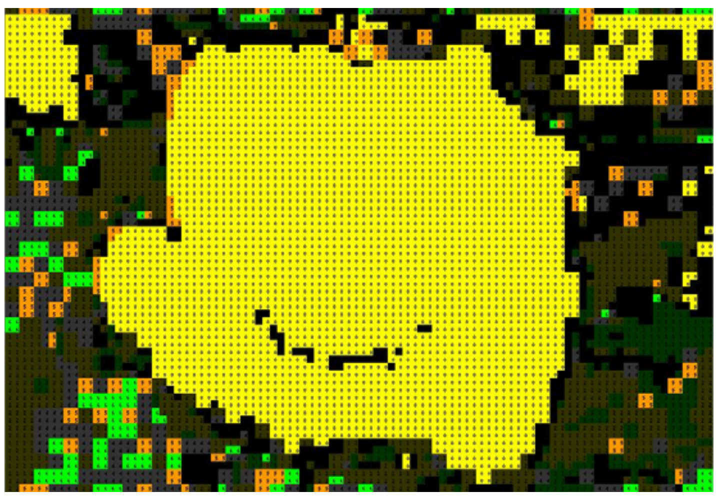


Image *q (segmented)*

Figure 4.5: Image segmentation of image *q*

Image *i2 (original)*

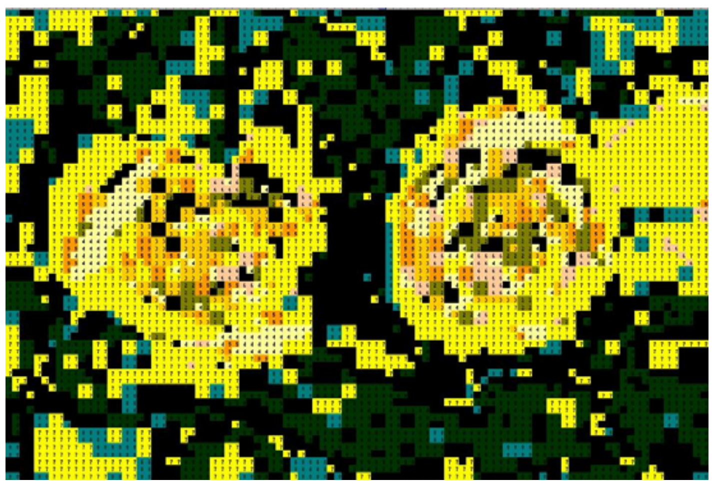


Image *i2(segmented)*

Object 1 Object 6
Object 2 Object 7
Object 3 Object 8
Object 4 Object 9
Object 5

Figure 4.6: Image segmentation of image *q* and image *i2*

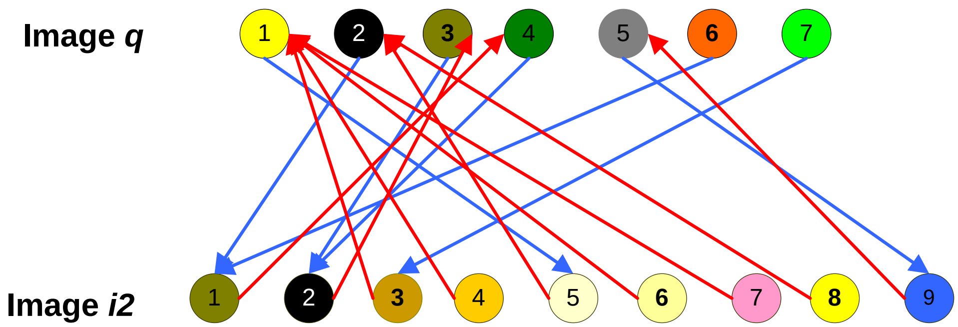


Figure 4.7: Minimum distance of objects from image *q* to image *i2* and vice versa

Table 4.2 Object uniqueness

Cluster	**# of objects**	**Object Uniqueness (γ_p)**
CLUSTER1	27	0.38636364
CLUSTER2	6	0.86363636
CLUSTER3	7	0.84090909
CLUSTER4	2	0.95454545
CLUSTER5	2	0.95454545

Table 4.3 Object cluster to form object group

Cluster 1	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
qObject 1	i1Object 8	i1Object 7	qObject 5	qObject 6
qObject 2	i1Object 9	i2Object 9	qObject 7	i3Object 5
qObject 3	i3Object 4	i3Object 6		
qObject 4	i4Object 4	i5Object 3		
i1Object 1	i5Object 7	i5Object 4		
i1Object 10	i5Object 8	i5Object 5		
i1Object 2		i5Object 6		
i1Object 3				
i1Object 4				
i1Object 5				
i1Object 6				
i2Object 1				
i2Object 2				
i2Object 3				
i2Object 4				
i2Object 5				
i2Object 6				
i2Object 7				
i2Object 8				
i3Object 1				
i3Object 2				
i3Object 3				
i4Object 1				
i4Object 2				
i4Object 3				
i5Object 1				
i5Object 2				

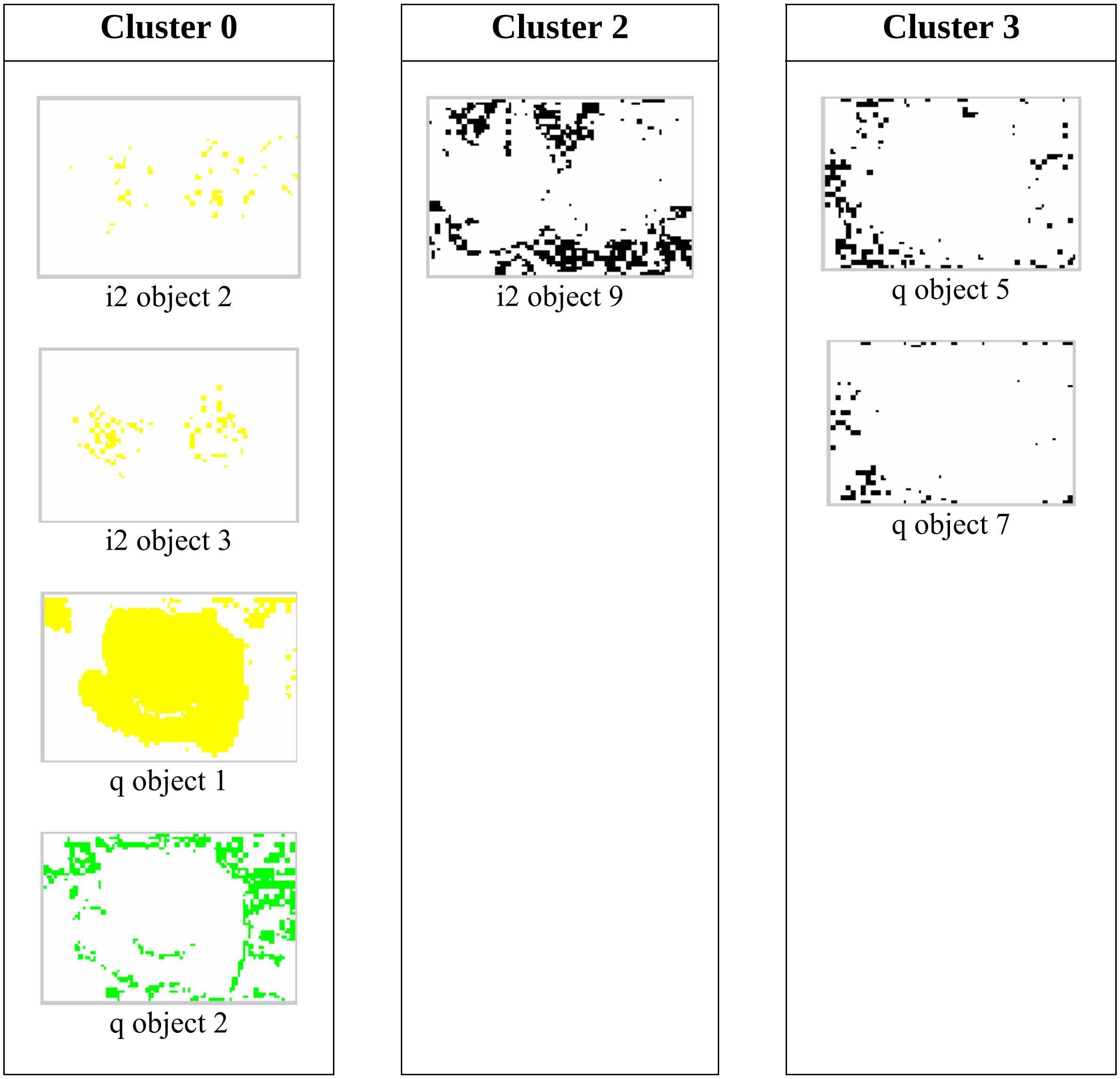
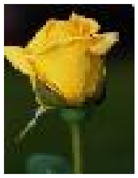


Figure 4.8 Example of object cluster to form object group

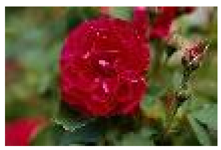
Query Image (q)



i1

i2

i3



i4

i5

Distance (q , i1) = 0.0373
Distance (q , i2) = 0.0190
Distance (q , i3) = 0.0090
Distance (q , i4) = 0.1560
Distance (q , i5) = 0.1315

Figure 4.9 Result of similarity distance computation

CHAPTER 5: RESULTS ANDANALYSIS OF PERFORMANCE

To analyze the performance of the retrieval process, we use three different approaches to evaluate our algorithm.

First, we need to get the optimum number of clusters during object clustering. The selection of optimum number of cluster is important since the result could influence the parameter for object uniqueness. Second, we randomly select 30 images from each class for query. Each query returns the top 20 images from database. The output of this precision is then compared to three existing algorithm: FuzzyClub, IRM, and Geometric Histogram. And finally, to determine the precision of the whole database, we select all 100 images from each image class for query. Each query returns the result of the top 10, 20, and 30 images from database. The output of this result is to evaluate the precision by different size of returns and to see where the similar images fall inside the returned result.

5.1 Experiment #1 - Similarity Distance Measure

During pixel clustering to obtain objects, to find a suitable number of objects per image, we need to set the similarity distance measure during hierarchical clustering. To do this, we randomly selected 10 images from each image class which corresponds to 100 images from the database. We ran this experiment for one time. From each image, we set the similarity measure at 0.6, 0.7 and 0.8. The number of clusters based on these measures was recorded and the average number of clusters for each similarity measure was computed (Table 5.1). We concluded to set a similarity measure equal to 0.7 because at similarity measure 0.7 the average number of clusters for each image is 5.8 which is a good number of objects. We argue that 5.8 is a good number of objects for this particular

Table 5.1 Correlation between similarity measures and average number of objects

Similarity Measures	**Avg number of objects**
0.6	2.3
0.7	5.8
0.8	9.6

database because when we randomly select 100 images and visually counted the number of objects per image, the average is 4.5. To see the performance of our algorithm we randomly select 4 images from different class, namely flower, dinosaur, bus, and elephant. Each query returns the top 10 images from database. The four query retrievals are shown in Figure 5.1.

5.2 Experiment #2 - Object Uniqueness

The idea of object uniqueness is to give more weight to objects that are more distinctive (such as bus, flower, and horse) compared to other objects that are more common (such as trees, and skies). To do this, we clustered all objects in database. Objects that belong to larger clusters will get less uniqueness weight compare to objects that belong to smaller clusters.

The selection of the number of clusters in object clustering affects the result of accuracy during image retrieval. The parameter of object uniqueness is obtained from the object clustering. Since object uniqueness is one of the variables to compute similarity distance computation, therefore, to select the number of cluster that results in the best performance. Therefore, during the experiment we choose four different number of cluster during object clustering: 1, 10, 25, and 35. All 100 images from each image class are selected for query, and each query returns the result of top 10 from the database. For each query we examine the precision of the returned images on their relevance of semantic meaning. Since similar semantic meaning belong to the same class, and each class has the same number of images, therefore it is straightforward to calculate the precision and recall. The precision and recall using different numbers of object clusters are then compared (Figure 5.2).

Figure 5.1a: Flower query, 9 matches from the top 10.

Figure 5.1b: Dinosaur query, 10 matches from the top 10.

Figure 5.1c: Bus query, 8 matches from the top 10.

Figure 5.1d: Elephant query, 5 matches from the top 10

From Figure 5.2 below, we examined that the precision and recall is higher in almost all image class when the number of object cluster is set to 25. The outcome of 35 object clusters is better compared with 1 object cluster, while the result for 10 object cluster is worse compare to the rest of the cluster number. From this result we conclude that the parameter of object uniqueness in similarity distance computation increase the performance of image retrieval. However, to get the best result, we need to find the accurate number of cluster (fine tuning) during object clustering.

5.3 Experiment #3 - Performance Comparison

In this section we evaluate the retrieval accuracy of our system and compare it with the existing region based algorithm. Since we are using the same comparison result in FuzzyClub (Zhang, 2000), we follow the same procedure as Zhang's FuzzyClub to obtain the performance result.

The result of this study is compared against the performance of IRM (Li, 2000) FuzzyClub (Zhang, 2002) and Geometric Histogram (Rao, 2000). We use the same 1000 images and the same technique to compute precision and recall. Figure 5.3 shows the comparison of average precision-recall of our algorithm against the existing algorithms.

This comparison shows that our algorithm (Hierarchical) performs significantly better than Geometric Histogram (Rao, 2000). Furthermore, compared to FuzzyClub and IRM, our algorithm performs slightly better in image group 4, 5, 6, and 7 which are bus, dinosaur, elephant, and flowers. On the other hand, image group 1, 2, 3, 8, 9, and 10 which are African people, buildings, horses, mountains, and food performs worse. In general, our system performs worse with relatively complex images such as landscape

ObjectCluster(1)	ObjectCluster(10)	ObjectCluster(25)	ObjectCluster(35)
0.27	0.32	0.29	0.27
0.32	0.29	0.34	0.31
0.38	0.36	0.394	0.394
0.74	0.612	0.73	0.73
0.92	0.89	0.99	0.95
0.4	0.36	0.416	0.402
0.59	0.55	0.71	0.68
0.495	0.43	0.506	0.48
0.31	0.28	0.32	0.32
0.41	0.39	0.423	0.41

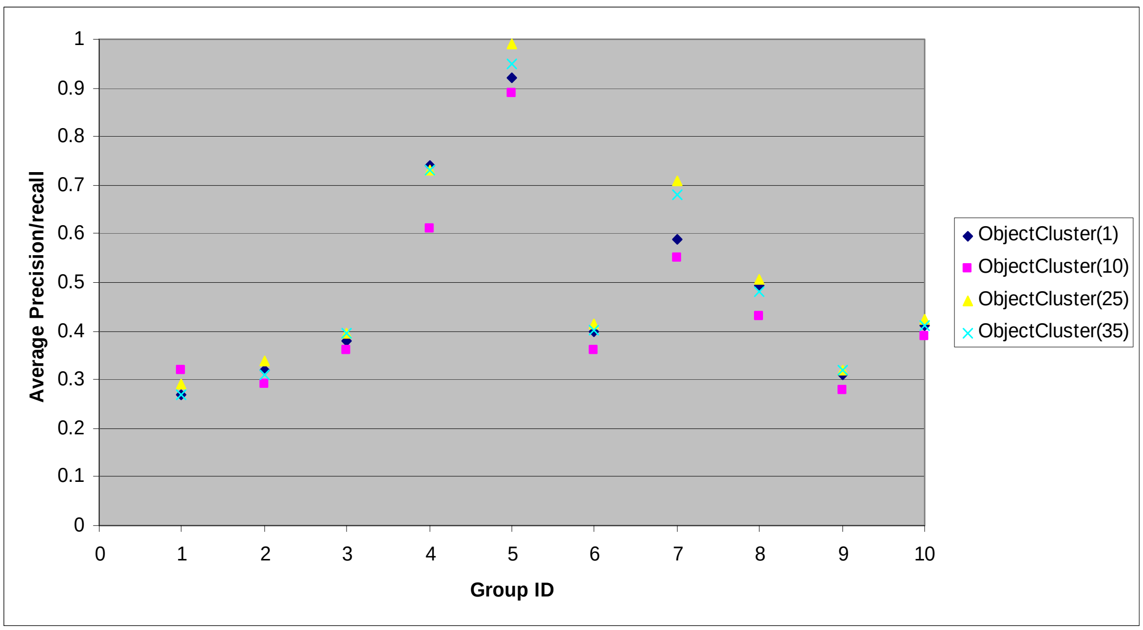


Figure 5.2: Average precision/recall using different number of cluster

Fuzzy Club	IRM	Geometric Histogram	Hierarchical
0.65	0.47	0.125	0.26
0.45	0.32	0.13	0.316
0.55	0.31	0.19	0.218
0.7	0.61	0.11	0.75
0.95	0.94	0.16	0.96
0.3	0.26	0.19	0.31
0.3	0.62	0.15	0.65438
0.85	0.61	0.11	0.502
0.35	0.23	0.22	0.29
0.49	0.49	0.15	0.38

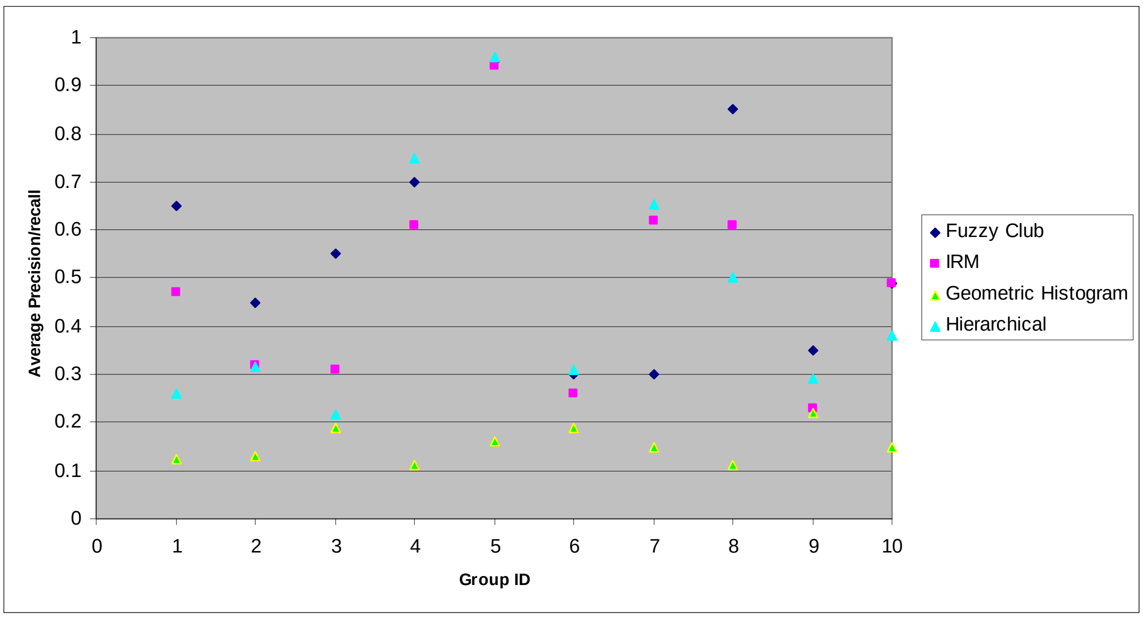


Figure 5.3 Average precision/recall comparisons

and buildings. The algorithm performs well with images that contain objects with color contrast from the background such as flowers and dinosaurs.

In our point of view, there are several possibilities for fixing this problem. First, during the pixel clustering, we set the weight for color features (0.65) to be greater than texture features (0.35). We selected this parameter because FuzzyClub (Zhang, 2002) concluded that this combination of color and texture weight gave the best result during image segmentation. Our system is not flexible enough to set different weights for object clustering; therefore we were not able to experiment with different weights during image segmentation. Second, shape features could have been considered during pixel clustering to obtain image segmentation.

5.4 Experiment #4 – Return Size

To further determine our system's performance, we perform another evaluation where we took all 100 images on each image class as image query and return the top 10, 20, and 30 images from the database. This result is to show where images that have similar semantic meaning to the query fall in the return images. Figure 5.4 shows the comparison of the precision-recall from Hierarchical system when the top 10, 20 and 30 images are returned. The result shows the accuracy is better when we use top 10 of the return result compared to 20 and 30. This shows that similar semantic images mostly fall in the beginning results during retrieval.

ObjectCluster(1)	ObjectCluster(10)	ObjectCluster(25)	ObjectCluster(35)
0.27	0.32	0.29	0.27
0.32	0.29	0.34	0.31
0.38	0.36	0.394	0.394
0.74	0.612	0.73	0.73
0.92	0.89	0.99	0.95
0.4	0.36	0.416	0.402
0.59	0.55	0.71	0.68
0.495	0.43	0.506	0.48
0.31	0.28	0.32	0.32
0.41	0.39	0.423	0.41

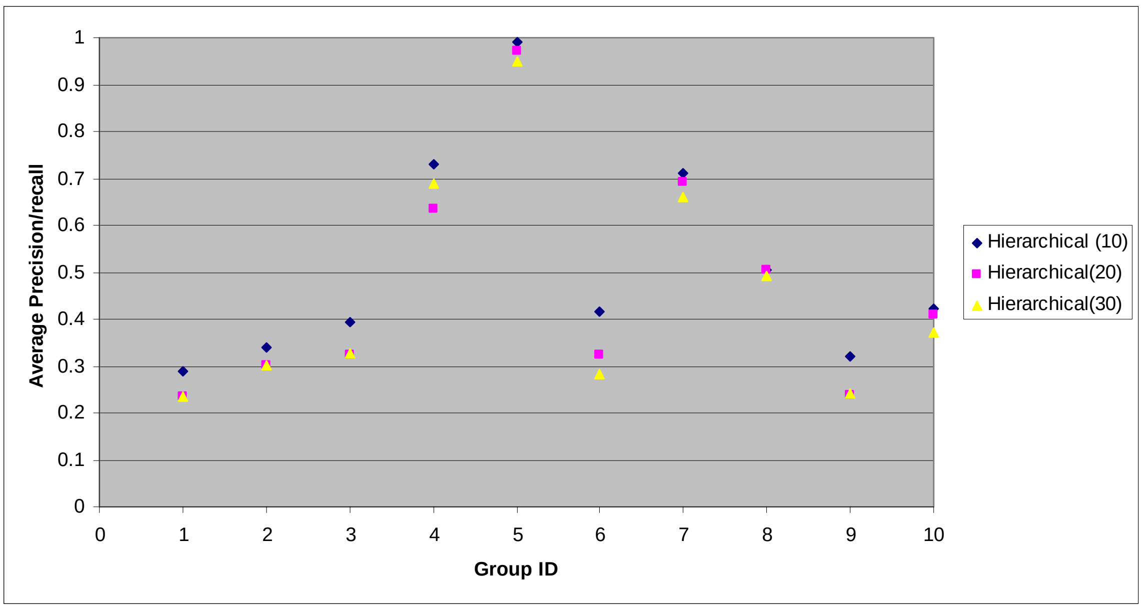


Figure 5.4: Average precision/recall comparison of top 10, 20, and 30 closest distances

CHAPTER 6: CONCLUSION AND FUTURE RESEARCH

6.1 Conclusions

We have developed an improved region-based image retrieval system. The system uses a modified k-means clustering algorithm to improve image segmentation, and uses a new similarity distance measure where object uniqueness is considered during computation. The algorithm has been implemented and tested using 1,000 COREL color image and the retrieval performance is compared to existing region-based algorithms (FuzzyClub, IRM, and Geometric Histogram).

The performance of our algorithm has been shown to perform better compared with Geometric Histogram. On the other hand, compared with FuzzyClub and IRM, our system performs better only in several image classes and perform worse in other image classes. Overall, our system performs better when the contrast between the main object and the background is visible in the image and performs worse when the image is complex and the objects have smooth edges. During the implementation, we also proved that by considering object uniqueness during similarity distance computation improve the accuracy during retrieval.

In conclusion, compared with the existing algorithm, our system demonstrates the following advantages:

1) An improvement in image segmentation accuracy, especially for simple images
2) An improvement during similarity distance computation by using the parameter of object uniqueness into consideration

6.2 Future Research

Following developments can be made in the future:

1) During pixel clustering to obtain objects, the system should be flexible to set different combination of weight for color and texture features; therefore we can maximize the performance by choosing the best combination between these weights.
2) To further improve the segmentation algorithm, the study of using the shape features into account during pixel clustering and similarity distance computation can be considered.
3) To get a better performance, the system can automatically pre-classified the database into different semantic images (such as outdoor vs. indoor, landscape vs. cityscape, texture vs. non texture images) and develop algorithm that are specific for particular semantic image class.

REFERENCES

Carson, C., Thomas, M., Belongie, S., Hellerstein, J. M., Malik, J., (1999), "Blobworld: A system for region-based image indexing and retrieval," *Third International Conference on Visual Information Systems, Springer*

Castelli, V. and Bergman, L. D., (2002), "Image Databases: Search and Retrieval of Digital Imagery", *John Wiley & Sons, Inc*. pp.285-311

Chang, T., and Jay Kuo, C.C., (1993), "Texture analysis and classification with tree-structured wavelet transform," IEEE Trans. Image Proc., **2(4)**, pp. 429-441

Dempster, A., Laird, N., and Rubin, D., (1977), "Maximum likelihood from incomplete data via the EM algorithm," *Journal Royal Statistical Society, Ser. B*, **39(1)**, pp. 1–38

Daubechies, I., (1992), "Ten Lectures on Wavelets," *Capital City Press, Montpelier, Vermont*

Dubes, R. and Jain, A.K., (1989), "Random field models in image analysis", *Journal Applied Statistic*, **16(2)**, pp.131-164

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. and Yanker, P., (1995), "Query by image and video content: The QBIC system," *IEEE Computer*, **28(9)**, pp.23-32

Gersho, A., (1979), "Asymptotically optimum block quantization," *IEEE Transaction Information Theory*, **25(4)**, pp.373-380

Guha, S., Rastogi, R., Shim, K., (1998), "CURE: An Efficient Clustering Algorithm for Large Databases," *Proc. of ACM SIGMOD International Conference on Management of Data*, pp.73-84

Gupta, A., and Jain, R., (1997), "Visual information retrieval," *Comm. Assoc. Comp. Mach.*, **40(5)**, pp. 70–79

Hartigan J.A., Wong, M.A, (1979), "Algorithm AS136: A k-means Clustering algorithm," *Applied Statistic*, **28**, pp. 100-108

Julezs, B., (1975), "Experiments in the visual perception of texture," *Scientific American*, **232(4)**, pp. 2-11

Kimia, B., (2001), "Shape Representation for Image Retrieval", *Image Databases: Search and Retrieval of Digital Imagery, John Wiley & Sons*, pp. 345-358

Li, J., Wang, J. Z. and Wiederhold, G., (2000), "Integrated Region Matching for Image Retrieval," *ACM Multimedia*, p. 147-156.

Mandelbrot, B.B., (1983),"The fractal geometry of nature", *W.H. Freeman, New York*

Manjunath, B.S., Ma, W.Y., (2002), "Texture Features for Image Retrieval," *Image Database: Search and Retrieval of Digital Imagery, John Wiley & Sons, New York*, pp. 313-344

Mao, J. and Jain, A.K., "Texture Classification and Segmentation using Multi-Resolution Simultaneous Autoregressive Models," *Pattern Recognition*, **25(2)**, pp. 173-188

Pentland, A.P., (1984), "Fractal-Based Description of Natural Scenes," *IEEE Transaction Pattern Analysis Machine Intelligence*, **6**, pp. 661-674

Pentland, A., Picard, R. and Sclaroff S.,(1996), "Photobook: Contentbased manipulation of image databases", *International Journal of Computer Vision*, **18(3)**, pp.233–254

Rao, A. R. and Lohse, G.L., (1993), "Identifying high level features of texture perception", *CVGIP: Graphical Models Image Process*, **55(3)**, pp.218-233

Rao, A., Srihari, R. K., Zhang, Z., (2000), "Geometric Histogram: A Distribution of Geometric Configuration of Color Substes", *Internet Imaging, Proceedings of SPIE*, pp.91-101

Rosenfeld, A., Kak, A.C., (1982) "Digital Picture Processing," A*cademic Press, New York*

Seo, J., Bakay, M., Zhao, P., Chen Y., Clarkson P., Shneiderman, B., Hoffman E. P., (2003), "Interactive Color Mosaic and Dendrogram Displays for Signal/Noise Optimization in Microarray Data Analysis," *IEEE International Conference on Multimedia and Expo*

Shi, J., and Malik, J., (1997), "Normalized Cuts and Image Segmentation," *Proceedings Computer Vision and Pattern Recognition*, June, pp. 731-737

Smith, J., (2001), "Color for Image Retrieval", *Image Databases: Search and Retrieval of Digital Imagery, John Wiley & Sons*, *New York,* pp.285-311

Smith, J.R., (1997), "Integrated Spatial and Feature Image Systems: Retrieval Analysis and Compression," *PhD thesis, Graduate School of Arts and Sciences, Columbia University, New York*

Smith, J.R., and Chang, S.F., (1997), "Single color extraction and image query," *In Proceeding IEEE International Conference on Image Processing*, pp. 528–531

Stricker, M., Swain, M., (1994), "The capacity of color histogram indexing," *Proceeding IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, pp. 704-708

Tamura, H., Mori, S., and Yamawaki, T., (1978), "Texture features corresponding to visual perception," *IEEE Transactions System, Man, and Cybernetics,* **8(6)**, pp. 460-473

Unser, M., (1995), "Texture classification and Segmentation Using Wavelet Frames," *IEEE Transaction Image Processing*, **4(11)**, pp.1549-1560

Wang, J.Z., Wiederhold, G., Firschein, O., Sha, X.W., (1998), "Content-based image indexing and searching using Daubechies's wavelets," *International Journal of Digital Libraries*, **1(4)**, pp. 311-328

Wu, J., (2003), "Rotation Invariant Classification of 3D Surface Texture Using Photometric Stereo", *PhD Thesis, Heriot-Watt University*

Zhang, R. and Zhang, Z., (2002), "A Clustering Based Approach to Efficient Image Retrieval," *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, pp. 339

VITA

Eka Aulia was born in Jakarta, Indonesia. He graduated from Indonesian high school in Bandung, Indonesia. In December of 2001 he received his Bachelor of Science degree in Industrial Engineering. In the spring 2002 he began a master's program in Industrial Engineering. He is a candidate for the degree of Master of Science in Industrial Engineering to be awarded in the commencement of May 2005.