2004

Automatic Annotation Of Database Images For Query-by-concept

Nualsawat Hiransakolwong
University of Central Florida

AUTOMATIC ANNOTATION OF DATABASE IMAGES FOR QUERY-BY-CONCEPT

by

NUALSAWAT HIRANSAKOLWONG
B.S. Srinakarinwirot University, Thailand, 1984
M.S. National Institute Development Administration (NIDA), Thailand, 1987
M.S Clemson University, USA, 1999

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
from the School of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2004

Major Professor: Kien A. Hua

ABSTRACT

As digital images become ubiquitous in many applications, the need for efficient and effective retrieval techniques is more demanding than ever. *Query by Example* (QBE) and *Query by Concept* (QBC) are among the most popular query models. The former model accepts example images as queries and searches for similar ones based on low-level features such as colors and textures. The latter model allows queries to be expressed in the form of high-level semantics or *concept* words, such as "boat" or "car," and finds images that match the specified concepts. Recent research has focused on the connections between these two models and attempts to close the *semantic-gap* between them. This research involves finding the best method that maps a set of low-level features into high-level concepts.

Automatic annotation techniques are investigated in this dissertation to facilitate QBC. In this approach, sets of training images are used to discover the relationship between low-level features and predetermined high-level concepts. The best mapping with respect to the training sets is proposed and used to analyze images, annotating them with the matched concept words. One principal difference between QBE and QBC is that, while similarity matching in QBE must be done at the query time, QBC performs concept exploration off-line. This difference allows QBC techniques to shift the time-consuming task of determining similarity away from the query time, thus facilitating the additional processing time required for increasingly accurate matching. Consequently, QBC's primary design objective is to achieve accurate annotation within a reasonable processing time. This objective is the guiding principle in the design of the following proposed methods which facilitate image annotation:

- A novel dynamic similarity function. This technique allows users to query with multiple examples: relevant, irrelevant or neutral. It uses the *range distance* in each group to automatically determine weights in the distance function. Among the advantages of this technique are higher precision and recall rates with fast matching time.

- Object recognition based on skeletal graphs. The topologies of objects' skeletal graphs are captured and compared at the node level. Such graph representation allows preservation of the skeletal graph's coherence without sacrificing the flexibility of matching similar portions of graphs across different levels. The technique is robust to translation, scaling, and rotation invariants at object level. This technique achieves high precision and recall rates with reasonable matching time and storage space.

- ASIA (*Automatic Sampling-based Image Annotation*) is a technique based on a new sampling-based matching framework allowing users to identify their area of interest. ASIA eliminates noise, or irrelevant areas of the image. ASIA is robust to translation, scaling, and rotation invariants at the object level. This technique also achieves high precision and recall rates.

While the above techniques may not be the fastest when contrasted with some other recent QBE techniques, they very effectively perform image annotation. The results of applying these processes are accurately annotated database images to which QBC may then be applied. The results of extensive experiments are presented to substantiate the performance advantages of the proposed techniques and allow them to be compared with other recent high-performance

techniques. Additionally, a discussion on merging the proposed techniques into a highly effective annotation system is also detailed.

To My Family and My Lovely Aunt

ACKNOWLEDGMENTS

I would like to take an opportunity to thank those whose unreserved support makes this day a reality. I am very much grateful to my advisor, Professor Kien A. Hua, and would like to express very sincere gratitude for his encouragement, support, guidance, and patience. This work would not have been possible without his fervent guidance and constant support over the past years.

I would like to express my gratitude to the committee members – Dr. Lang, Dr. Wu and Dr. Lin – for their valuable time and energy. Their helpful comments and wonderful recommendations have improved my work and strengthened my resume.

I would like to thank all those who have helped in giving advice, comments and suggestions, especially former and current Data System Group members. I would also like to thank the department of computer science at the University of Central Florida for the opportunity to work as a graduate teaching and research assistant.

I would like to thank Dr. Khanh Vu, Bernard Fudim, Lorie Munizzi, Alexander Aved and the writing center at the University of Central Florida for their help in proof reading this dissertation. I appreciate the cooperation of Yuqing Song for providing me the Monotonic Tree system used in my experimental study.

I would like to thank Dr. David Workman, Dr. Wallapak Tavanapong, Varunyu Vorachart, Sirikunya Nilpanich, Soontharee Koompairojn and the many Thai students that have helped and supported me. I would like to thank my family for their support, patience and love for me.

I would like to thank the Royal Thai Government and the King Mongkut's Institute of Technology Ladkrabang also that gave me a chance to continue my schooling.

Finally, I would like to thank those people who I failed to acknowledge. I am grateful for all those who have accompanied me on this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1 Motivation

Digital images have become more ubiquitous than ever before. Digital data is replacing traditional textual data and becoming a prominent medium in many application domains. This growth is due to the fact that image acquisition, manipulation and storage are becoming so affordable these days: inexpensive and high-resolution digital cameras and camera-equipped mobile phones are flooding the market, powerful image-processing software is freely available and the price of mass storage is decreasing every day ([46], [47] and [48]). Digital pictures are now taken by ordinary people and professionals alike, modifying, sharing, publishing and storing them in vast numbers. Meanwhile, other traditional applications of images continue to accumulate huge quantities of data in digitized formats. New applications such digital museums are also growing in number.

This phenomenon is the main drive for recent research in image manipulation such as archiving and retrieval. Efficient search techniques are critical for large image databases in image management systems [46]. Although there has been much research tackling this problem, it is still very challenging for an average user to search for particular images from a large collection. In the following section, recent query models for image retrieval are surveyed, the weaknesses of previous methods are highlighted, and the proposed approaches addressing these weaknesses are summarized.

1.2 Related Work

Many techniques have been proposed to search for images from a large image database ([1]-[4], [10], [12]-[18], [49]-[55], etc.) These techniques can be divided into Query-by-Example (content-based) and Query-by-Concept (keyword-based) searches depending on how queries are posed.

Query-by-Example (QBE) has been the focus of a lot of research (e.g., [1]-[3], [12]-[18]). In a typical QBE system, users present an example image and the system searches for similar images using low-level visual features from the example (such as color, texture and structure). The performance of QBE systems depend greatly on how close the example is to the desirable images. Given a bad example, even the best similarity measures would rank relevant images poorly. Even more challenging is the idea of searching for objects, which are often a small part of the example. With no mechanism in most QBE systems to isolate the desired object from the example's irrelevant regions, the result set contains many false hits and false dismissals. Consequently, it is difficult for users to find desirable images.

Query-by-Concept (QBC) is focused on keyword-based image retrieval ([4] and [10]) and has not been as widely studied as QBE. In these systems, users can retrieve images by entering keywords that describe the desired images. Manual keyword description of images has long been criticized as subjective, tedious, and limited. Recent keyword-based image retrieval, however, is designed around annotation, a process of naming (i.e., annotating) images with high-level concept keywords such as "car" or "tree". The annotation process is content-based, objective and fully automatic. Given an accurate annotation, relevant images can be retrieved

efficiently by employing high performance low-dimensional indexing techniques. Thus, in order to support QBC, it is critical to accurately annotate images based on their visual content.

One principal difference between the two query models is that while similarity matching in QBE must be done at the query time, QBC performs concept exploration off-line. As a result, QBC techniques are able to shift the time-consuming task of determining similarity away from the query time and thus affording additional processing time for more accurate matching. Consequently, QBC's primary design objective is to achieve accurate annotation with reasonable processing time. This objective is the guiding principle in the design of QBC methods.

1.3 Features of the Proposed Techniques

Some recent image retrieval techniques to facilitate image annotation are surveyed. In a typical image annotation system, sets of training images are used to discover the relationship between low-level features and predetermined high-level concepts. The best mapping with respect to the training sets is proposed and used to analyze images, annotating them with the matched concept words. This section highlights the major features of the proposed techniques which will be discussed in detail in subsequent chapters.

1.3.1 Noise in the Whole-Matching Image Retrieval

In most content-based image retrieval (CBIR) systems, images features are extracted from the entire image area. Noise greatly compromises the outcome of similarity computations [4]. This issue is discussed in more detail and the proposed technique [58] to address the problem

is presented in the next chapter. The manner in which this technique advances the quality of image annotation is addressed.

1.3.2 Handling Translation, Scaling, and Rotational Invariants

Robustness to translation, scaling, and rotation at the object level is desirable in many applications. Few retrieval systems, however, can handle the rotational invariant well. Chapters 2 and 4 address this problem and present techniques for handling the rotational invariant ([56], [57] and [58]).

1.3.3 Effectiveness of the Proposed Techniques

The most important measure of any image matching technique is how accurately the relevant images are identified. The manner in which the performances of the present techniques [56 – 59] measure up against competitors is described in Chapters 2, 3 and 4.

1.3.4 Time Complexity and Space Overhead

Although response time and space overheads are not the primary performance metrics of an annotation system, minimization of these overheads are always desirable measures, especially for very large databases. The range-distance and skeleton-matching techniques are able to reduce processing time and storage space for image features. The performance of the proposed techniques ([56], [57] and [59]) will be analyzed in Chapters 3 and 4.

1.4 Organization of the Dissertation

As stated above, this thesis focuses on accurate image annotation to support QBC. Towards this goal, several techniques are proposed to address different challenges of image matching. In chapter 2, ASIA, an automatic annotation technique for general images, is presented. ASIA employs a novel sampling-based matching framework to achieve robustness to rotation, scaling, and translation of interested objects. As noted, the first phase of QBC is in fact an efficient QBE used to annotate images at the build time. The focus of Chapters 3 and 4 is image matching given query examples. In Chapter 3, queries are formed by a set of examples. The set consists of three possible subgroups: positive, negative and neutral groups. The positive group is the group of images that possess the visual attributes that relevant images should have. The negative group consists of examples that are irrelevant. They and their resemblance should be filtered out from the returned set. The neutral group simply includes those images that should not belong to either of the clear-cut groups. They contain mixed features such that the user is unsure about their relevancy as a whole. Chapter 4 discusses another challenging matching problem: image matching based on shape. A skeleton-matching technique is proposed to address this problem. This technique achieves accuracy in identifying matched objects while reducing time complexity and storage space.

In Chapter 5, the prototype for each technique is described, and the advantages and disadvantages for each technique as compared to other recent approaches are highlighted. Finally, the conclusions and future work are discussed in Chapter 6.

CHAPTER TWO: ASIA

ASIA stands for an *Automatic Sampling-based Image Annotation* technique for high-level image retrieval. ASIA consists of two phases. In the first phase, an annotated image database with high-level image concepts using low-level image features, such as the colors and structure of sample objects, is created off-line. This annotation technique uses a new sampling-based matching framework which has been shown to be more robust than previous approaches to the scaling and translation invariants. This approach is also the first that is robust to the rotation invariant at the matching object level. In the second phase, the user can query the annotated image database using text-only concepts. Experimental results indicate that the ASIA method is significantly better than a recently proposed alternative the Monotonic Tree (MT) method. ASIA has the ability to identify more relevant images, while producing fewer false matches.

2.1 Introduction

Two popular query models for image retrieval systems are as follows:

- *Query by Example* (QBE): This approach accepts an image as a query and searches for similar images in the database.
- *Query by Concept* (QBC): This approach accepts concepts stated in text format as a query and searches for images in the database that match the specified concepts.

QBE requires the ability to automatically extract low-level image features such as color, texture, shape, etc to facilitate image matching. A limiting factor of this approach is finding an appropriate example query image. The required search time is too slow for a large image

database. In comparison, QBC accepts high-level concepts such as "boat" or "car" as queries, which are relatively easier to use. This faster QBC approach requires an efficient and reliable technique to annotate the database images. The ASIA (*Automatic Sampling-based Image Annotation*) technique, which enables us to exploit the power of low-level image features to support the convenience of QBC, is proposed in this chapter.

The ASIA process starts with a training set of sample images to train visual concepts for an image database. Each image in the training set represents a high-level concept such as "car," "sky," "flower," etc. Database images are annotated with high-level concepts based on the matched training images. The matching is done in a new sampling-based framework, in which each training image has a contour indicating the "area of interest." Thus, noise such as background and irrelevant object areas are excluded from the similarity computation to ensure the high quality of the annotation. The extracted features of the noise-free training images can be maintained in a *Concept Library* to support annotation of new images in the future.

In comparison with existing QBE techniques, the ASIA approach has the following advantages:

- Since image matching is done off-line, ASIA can afford substantially more time to achieve better quality matching.
- Noise in such a query can dilute the true intension of a query and may lead to unintended query results. Unlike the traditional QBE, which treats the entire image area as the query (even though a significant portion of the image may not be relevant) ASIA focuses on only the specific area of interest. Therefore, ASIA should give more accuracy.

Experimental studies indicate that ASIA significantly outperforms a recent QBC technique called the *Monotonic Tree* approach [10]. ASIA provides greater accuracy and is robust to rotation, scaling, and translation invariants at the matching object level.

The remainder of this chapter is organized as follows. In Section 2.2, some related previous work are reviewed. The ability to handle rotation, scaling, and translation invariants of objects in image matching for ASIA is discussed in Section 2.3. An indexing method for this environment is presented in Section 2.4. The ASIA annotation environment is introduced in Section 2.5. The experimental results are provided in Sections 2.6. Finally, concluding remarks are discussed in Section 2.7.

2.2 Related Work

There have been many proposals of QBE techniques (e.g., [1], [2], [3], [12], [13], [14], [15], [16], [17], [18], etc.); however, far fewer studies on automatic annotation techniques for the QBC approach. This section reviews some recent QBC approaches and discusses known problems for each approach.

The Monotonic Tree technique [10] represents each image as a hierarchical structure of the image elements similar to XML. The basic elements, called *structure elements,* correspond to a homogeneous region in the image and are used to store the low-level features of the region such as color, shape, harshness and spatial location. A knowledge base is used to organize such structure elements into higher-level concepts in the Monotonic Tree. For instance, a higher level concept “sky without cloud” is defined in the knowledge base as a smooth blue region in the

upper part of an image. Thus, images are annotated in this technique based on models of various high-level concepts. The drawbacks of this approach are as follows:

- The models are rudimentary and not specific enough to differentiate real-life objects.
- Building the knowledge base involves specifying many thresholds to define each concept. This process, done manually, is tedious and fairly subjective.

Another QBC technique, called CAMEL (*Concept Annotated iMagE Libraries*), is presented in [4]. This scheme addresses the second problem of the Monotonic Tree approach by using sample images to define visual concepts instead of characterizing them with a model. A content-based search method, called WALRUS (WAveLet-based Retrieval of User-specified Scenes) [3], is used to extract features from these sample images. To process a QBC query, CAMEL matches the feature vectors of the corresponding sample images with those of the images in the database. The drawbacks of this approach are as follows:

- The feature vectors are extracted from the entire image area. Including the noise areas in the similarity computation seriously affects the outcome of the image matching.
- The search technique is based on matching low-level features. This mechanism is relatively slow.
- This method does not support the rotation invariant at object level.

In the following section, a new QBC technique, called ASIA, that addresses all of the aforementioned shortcomings, is introduced.

2.3 The ASIA Approach

The image retrieval problem and image annotation problem are different in one subtle way. Short query response time is a key requirement of image retrieval systems. Unless a search technique is sufficiently fast, it is not practical for QBE systems. In contrast, since image matching techniques developed for image annotation are executed off-line, their primary design objective is to achieve highly accurate annotation even at the expense of a slow matching process. ASIA is designed following this perspective.

The ASIA environment is presented in this section, and the techniques for handling rotation, scaling, and translation of image objects are discussed.

2.3.1 Preprocessing of the Training Image

The ASIA approach allows the user to draw a contour on a training image to precisely outline the image area relevant to the visual concept being used for the annotation. The first step of the ASIA annotation procedure is to reduce the training image to the smallest square region that completely encloses the area of interest. This is done as follows:

- X_0 and X_1 coordinates are identified that fully cover the area of interest in the horizontal direction, and the width of the contour is computed as

$$\text{width} = \left| X_0 - X_1 \right|$$

- Similarly, Y_0 and Y_1 coordinates are identified that fully cover the area of interest in the vertical direction, and the height of the contour is computed as:

$$\text{height} = \left| Y_0 - Y_1 \right|$$

- *MIN(width, height)* and *MAX(width, height)* are defined as the minimum and maximum values, respectively, between the width and height of the area of interest. The reduced training image is a square region with dimension $c \times c$, where c = *MAX(width, height),* such that its center coincides with the centroid of the area of interest as illustrated in Figure 1.
- Finally, the dimensions of the reduced training image are rounded up to the nearest square selected from the following sizes: 96×96, 128×128, 160×160, 192×192, 224×224, and 256×256 pixels.

It will be clear later why one of the above six square sizes is used to reduce the original training image. For convenience sake, hereafter, the reduced training image will be referred as the training image.

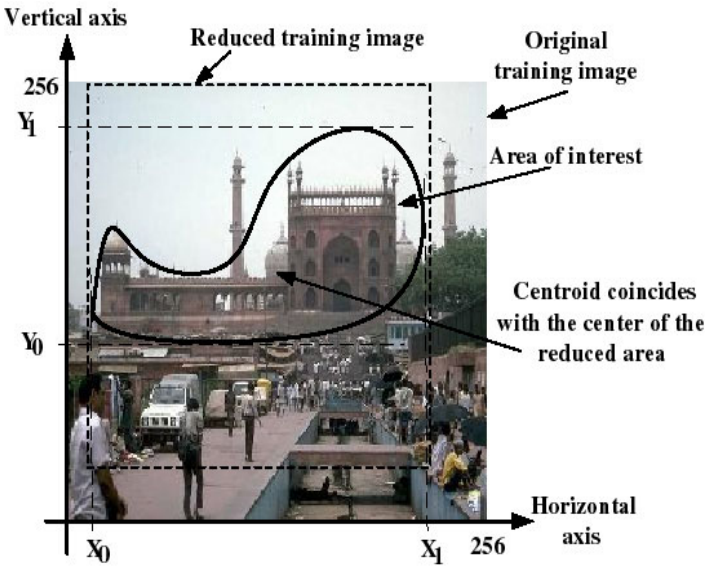


Figure 1: Finding the reduced area of a training image

2.3.2 Image Sampling

The ASIA matching technique is motivated by the process of digitizing audio waveforms. An audio waveform is approximated by taking samples of the waveform over time. This concept is adapted for images by taking samples over space (i.e., the image area). In ASIA, a sample is a circular block of image pixels. The diameter of this block is 16 pixels. The rationale is that the correlation between pixels tends to decrease after 15 to 20 pixels ([1], [5]). Using sample blocks with a 16-pixel diameter allows us to represent each block by its average color, called *local average*, to save storage and computation. The locations of the sample blocks are illustrated in Figure 2. They form eight concentric circles called *sample circles*. From a different perspective, these circular blocks form 16 *sampling rays* radiated from the center of the image. These sampling rays are uniformly spaced out at 22.5 degrees between adjacent sampling rays.

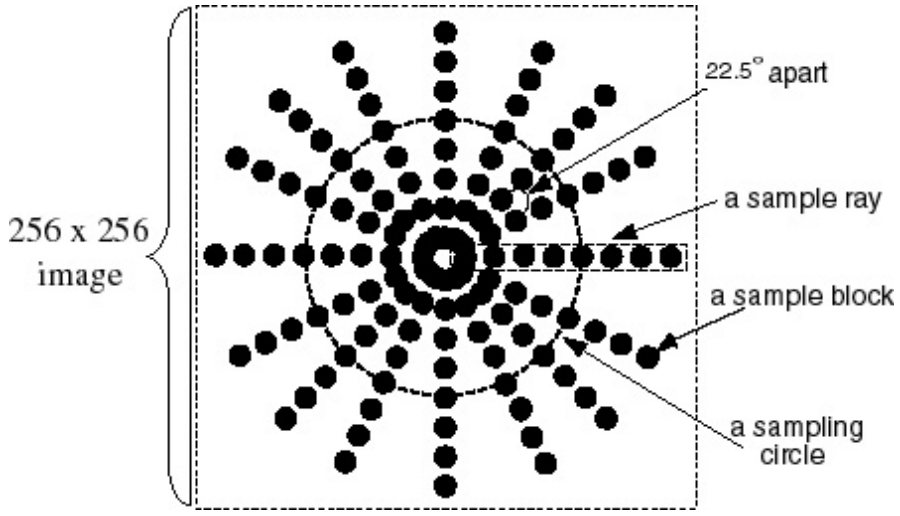


Figure 2: Sample blocks form eight concentric circles

Image matching is done in the ASIA framework by comparing the local averages of the corresponding sample blocks in the two images. To facilitate similarity computation, the local averages from RGB color is transformed and quantized them to 256 values in the Munsell (H, V, and C) system using the mathematical transformation presented in [19]. In this uniform color system, the dissimilarity of two colors is simply the distance between them.

2.3.3 Noise-Free Matching

To support noise-free matching, only the local averages spatially falling within the area of interest outlined by the contour are compared. For instance, to annotate relevant database images with the concept "car," a contour is drawn around a car in a sample (or training) image, and use it to match against the images in the database. For each database image that has the same contour area matching the one in the training image, this database image is annotated with the concept "car." Compared to CAMEL, the ASIA annotation technique is noise-free and therefore more reliable. ASIA also processes queries substantially faster because it compares the QBC query against the pre-computed annotations stored in the database instead of matching low-level features. In comparison with the Monotonic Tree approach, the ASIA technique is more reliable since it is based directly on the sample images, not thresholds subjectively defined by human users. ASIA is also more reliable due to the fact that its annotation process is based on individual sample images. Matching any one of these images (e.g., car pictures taken from different points of view) is sufficient to declare a positive match. In contrast, a model for a concept, as used in the Monotonic Tree, represents the average case derived from the experience of the human user. Using average data to look for matches is not specific enough and is prone to errors.

Let $\boldsymbol{fvTr_i}$ be the average color of a sample block Tr_i falling inside the contour of a training image. Similarly, $\boldsymbol{fvDB_i}$ denotes the average color of a sample block DB_i falling within the contour of a database image. The similarity of these two sample blocks is computed as follows:

$$\boldsymbol{Dist_i = (fvTr_i - fvDB_i)} \qquad (1)$$

To determine the similarity of the two contour areas, compute the matching score as follows:

$$\boldsymbol{Score} = \sum_{i=1}^{n} \frac{w_i}{1 + \boldsymbol{Dist_i}} \qquad (2)$$

Let n represent the number of sample blocks inside each contour. A '1' is added to the denominator to prevent division by zero. In the numerator, w_i is a weighting factor which can be set to indicate the significance of the match at sample block i. Since not all colors are equally likely in real-world images, matches on rare colors are more discriminating. The frequencies of the 256 possible average colors are determined from the images in the database. w_i is proportional to the inverse frequency of the average color $\boldsymbol{fvTr_i}$. If a similarity score is below a predefined threshold, it is not a match.

ASIA can be viewed as a structured-based or skeleton-based matching technique. This matching scheme implicitly takes into account the spatial correlation among the different meaningful parts of the matching object.

2.3.4 Handling Rotation in Matching Objects

Let A be a training image and B a database image to be annotated. The feature vector of A is denoted as $A^0 = \{A_1, A_2, ..., A_{16}\}$, where A_i represents the i^{th} sample ray of A. Similarly, the feature vector of B is $B^0 = \{B_1, B_2, ..., B_{16}\}$. The 16 different configurations of the feature vector of A are also defined as follows:

$A^0 = \{A_1, A_2, ..., A_{15}, A_{16}\}$

$A^1 = \{A_2, A_3, ..., A_{16}, A_1\}$

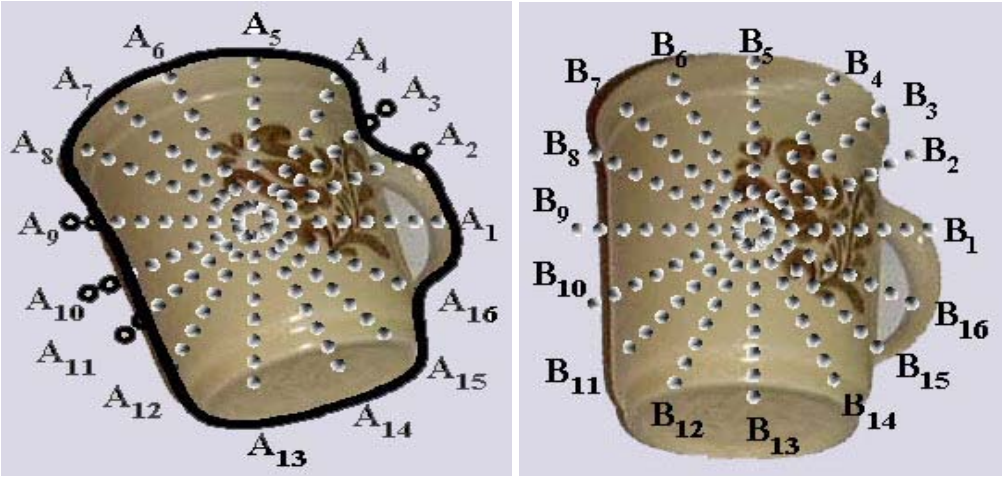
$A^2 = \{A_3, A_4, ..., A_1, A_2\}$

...

$A^{15} = \{A_{16}, A_1, ..., A_{14}, A_{15}\}$

That is, A^n is obtained by a rotational left shift of $A^{(n-1)}$. The similarity matching of images A and B is done by performing the following 16 comparisons: A^0 versus B^0, A^1 versus B^0, ... and A^{15} versus B^0. Each comparison is done as described in Section 2.3.2. If any one of these 16 comparisons results in a match, the whole matching process is considered as a match.

Informally, the matching process can be described as superimposing and rotating the training image over the database image for one complete circle at 22.5 degree increments. For each rotation position, the corresponding local averages falling inside the two contours are compared as discussed in Section 2.3.2. This matching scheme is illustrated in Figure 3. The best match occurs when A^{15} and B^0 are compared.



(A) **(B)**

Figure 3: Training image *A* and database image *B*

Training image *A* matches database image *B* when A^{15} and B^{0} are compared.

2.3.5 Handling Translation in Matching Objects

Handling translation requires comparing the training image against various local regions in the database image. In ASIA, subimages of six different sizes are considered as follows: 121 subimages of size 96×96 pixels, 81 of size 128×128, 49 of size 160×160, 25 of size 192×192, 9 of size 224×224 and 1 of size 256×256. The subimages of each size are evenly spaced out to cover as much of the whole image area as possible. Please note that subimages smaller than 96×96 pixels are not interesting for most image retrieval applications.

In total, there are 286 subimages for each database image. For matching purposes, each of these subimages is treated as an image and the same sampling rate is applied to all of them. As a result, the number of sample circles for each subimage size is as follows:

- a 96×96 subimage has 3 sample circles
- a 128×128 subimage has 4 sample circles
- a 160×160 subimage has 5 sample circles
- a 192×192 subimage has 6 sample circles
- a 224×224 subimage has 7 sample circles
- a 256×256 subimage has 8 sample circles

Processing a training image in ASIA is done as follows:

1) Six different sampling rates are applied to the training image. A higher sampling rate results in more sample blocks, and therefore, more sample circles over the same image area. This process creates six different training configurations with 3, 4, 5, 6, 7 and 8 sample circles, respectively.
2) Since the smallest dimensions of a training image are 96×96 pixels, a smaller sample block with a 6-pixel diameter is used in order to fit up to 8 sample circles in a 96×96 area. For each sampling rate, the sample blocks are evenly space out on their sample ray.
3) Each of the subimages is compared in the database against the training configuration with the same number of sample circles using Equations 1 and 2.
4) If a subimage matches any one of the training configurations, the database image containing the matching subimage is annotated with the corresponding concept.

2.3.6 Handling Scaling in Matching Objects

Since image comparison is done at various sampling rates, as described in Section 2.3.4, ASIA can match the same objects of different sizes as illustrated in Figure 4. It shows, at the top, the sample blocks of a database image. The three imaginary circles indicate the sample blocks inside each circle belong to the corresponding contain square subimage. The three images at the bottom represent three training images, each with the same star of a different size. Three different sampling rates are applied to these training images. If these three training images are compared with the appropriate subimage of the database image, they all match the star object in the database, although at different scales. The left training image, using a sampling rate higher than the sampling rate of the database image, is able to match a bigger star in the database image. The middle training image, using the same sampling rate used for the database image, can find a star of the same size. Similarly, the right training image, using a lower sampling rate, can match a smaller star.

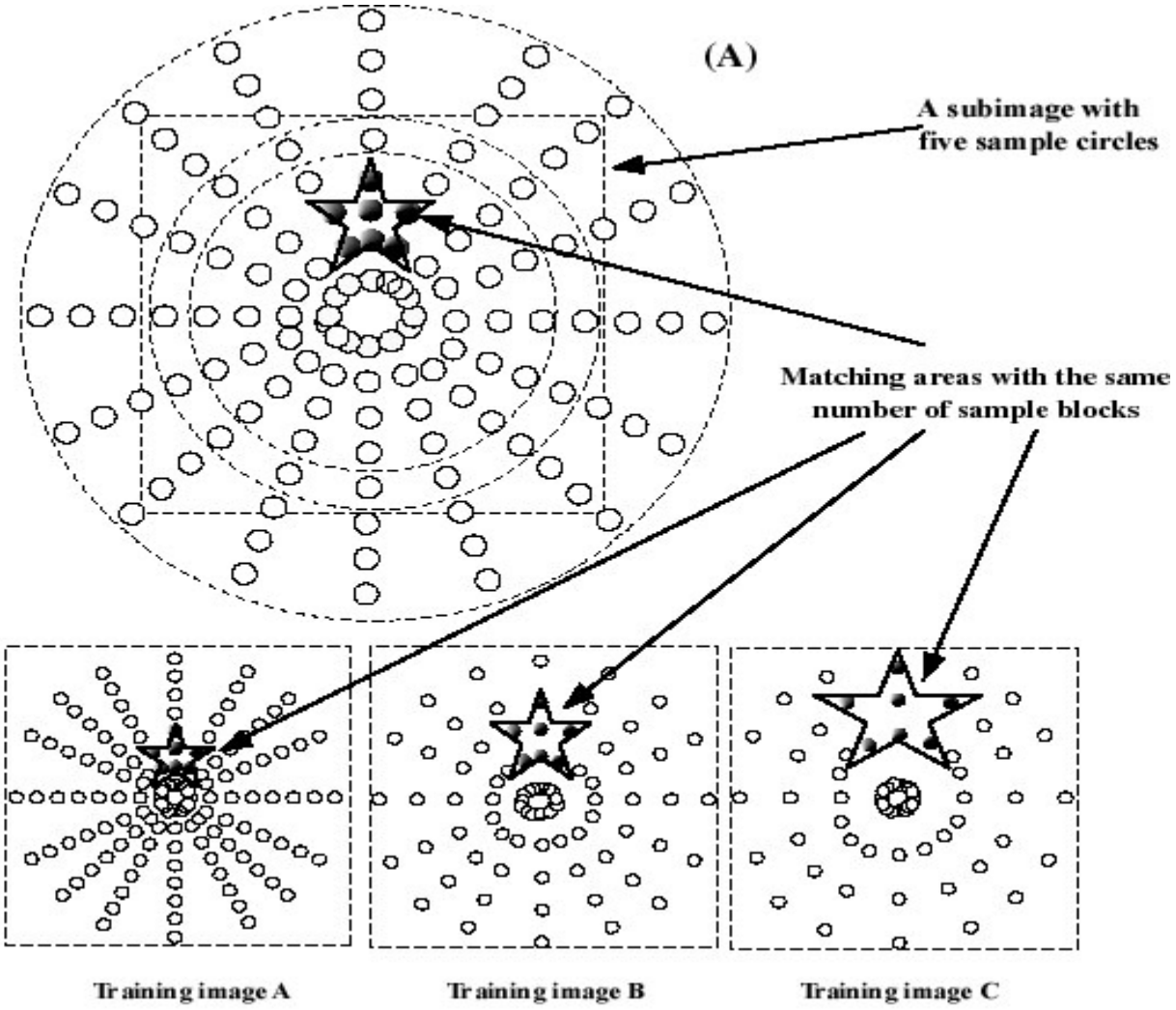


Figure 4: One example of scaling-invariant matching

This illustrates one fixed sampling rate for the database image in Figure 4 (A) and three sampling rates for the training images in Figure 4 (B, C and D) in order to match the database image with a bigger object, same-size object and smaller object, respectively.

2.4 Indexing

Although speed is not the primary performance metric for an annotation system, an indexing technique is presented in this section to help speed up the annotation process. This access structure is particularly important to applications that require frequent update to the Concept Library.

To leverage an indexing mechanism, the area of interest is approximated in a training image by a *core area* that maximizes the signal-to-noise ratio. This core area covers as many of the relevant local averages as possible without including too many irrelevant local averages (i.e., noise). An algorithm, presented in [2], can be used to compute this core area.

Internally, each database image is represented as five hyperrectangles in a multidimensional space. A multidimensional index such as R* tree is built on these database rectangles. To annotate the database images with a particular visual concept, the core area of the corresponding training image is used to probe the R* tree to search for potentially matching images in the database. Since the search results based on the core area is not noise-free, the candidate images returned in this phase are subject to a second-phase validation process in which the candidate images are compared against the training image using the technique described in Section 2.3.

To further explain this strategy, each of the 286 subimages of each database image is referred to as its *indexing subimage*. Internally, each indexing subimage is represented as a 14-element *signature*. To support the rotational invariant matching, the signature needs to be rotational invariant. It is computed as seven statistical average-variance pairs as follows:

1) Compute the color histogram to determine the three colors with the highest frequency, say C_1, C_2, and C_3, respectively.
2) The first statistical average-variance pair is computed from the colors of all pixels in the indexing subimage.
3) For each of the six color pairs (C_i, C_j), where i and j can be 1, 2, or 3, compute an average-variance pair for the signature as follows. The distance between any two pixels is defined at (x_1, y_1) and (x_2, y_2), respectively, as

$$d = \max(|x_1 - x_2|, |y_1 - y_2|).$$

For each pixel p_i of color C_i, compute the counts of the number of pixels of color C_j that are at distance 1, 3, 5, and 7, respectively, from p_i. Then compute the average and variance of these counts weighed by their respective distances. Thus, the signature has 14 total values.

Each signature in the database can be treated as a data point, also called *signature point*, in a 14-dimensional signature space. Since each database image has 286 indexing subimages, it has 286 signature points. To reduce the complexity, the signature points are clustered of each indexing subimage into five *minimum bounding rectangles* (MBRs). In other words, each database image is ultimately represented by its five MBRs in the 14-dimensional signature space.

After applying a certain sampling rate to a training image with an area of interest outlined by a contour, core area is determined and its signature point is computed as in the case of indexing subimages. This signature point can be treated as a tiny MBR, with essentially no dimension, in the 14-dimensional signature space. If this tiny MBR falls within any MBR of an indexing subimage, the corresponding database image is considered as a potential match for the training image. To speed up the search operation, a multidimensional access structure such as the R* tree can be used to index the MBRs in the database.

ASIA is still effective when the area of interest is slim and long as seen in the experimental results presented in Section 2.6. This is due to the fact that the core area is only used to eliminate database images that have no chance of matching the training image. A core area of a slim and longer area of interest is less specific. As a result, the index structure returns a longer list of image candidates. The irrelevant candidates will eventually be eliminated by the detailed block-to-block comparisons performed in the validation phase.

2.5 The ASIA Annotation Environment

The ASIA environment is illustrated in Figure 5. The numbers 1 through 7 label the steps involved in the annotation process. These steps are discussed as follows:

1) The *Fill-in Polygon* algorithm, presented in [9], is used to mark pixels in the area of interest. The reduced training image is then determined using the procedure given in Section 2.3.1.
2) For each sampling rate applied to the reduced training image, its feature vector is computed. This vector consists of the local averages as shown in Figure 2. These feature vectors are saved into the Concept Library to support annotation of new images in the future. The following steps are performed for each of the sampling rates.
3) Compute the core area of the area of interest using the algorithm given in [2].
4) Compute the signature of the core area.
5) The signature point of the core area is used as a dimension with less MBR to probe the R* tree to identify the potentially matching subimages in the database.

For each of these candidate indexing subimages, perform the following steps 6 and 7.

6) Compare the indexing subimage with the core area using the rotational matching technique presented in Section 2.3.4. For each rotation step, only the local averages within the area of interest are compared. The rotation stops as soon as a match is found.
7) If Step 6 results in a match, the database image corresponding to the matching index subimage is annotated with the visual concept being processed.

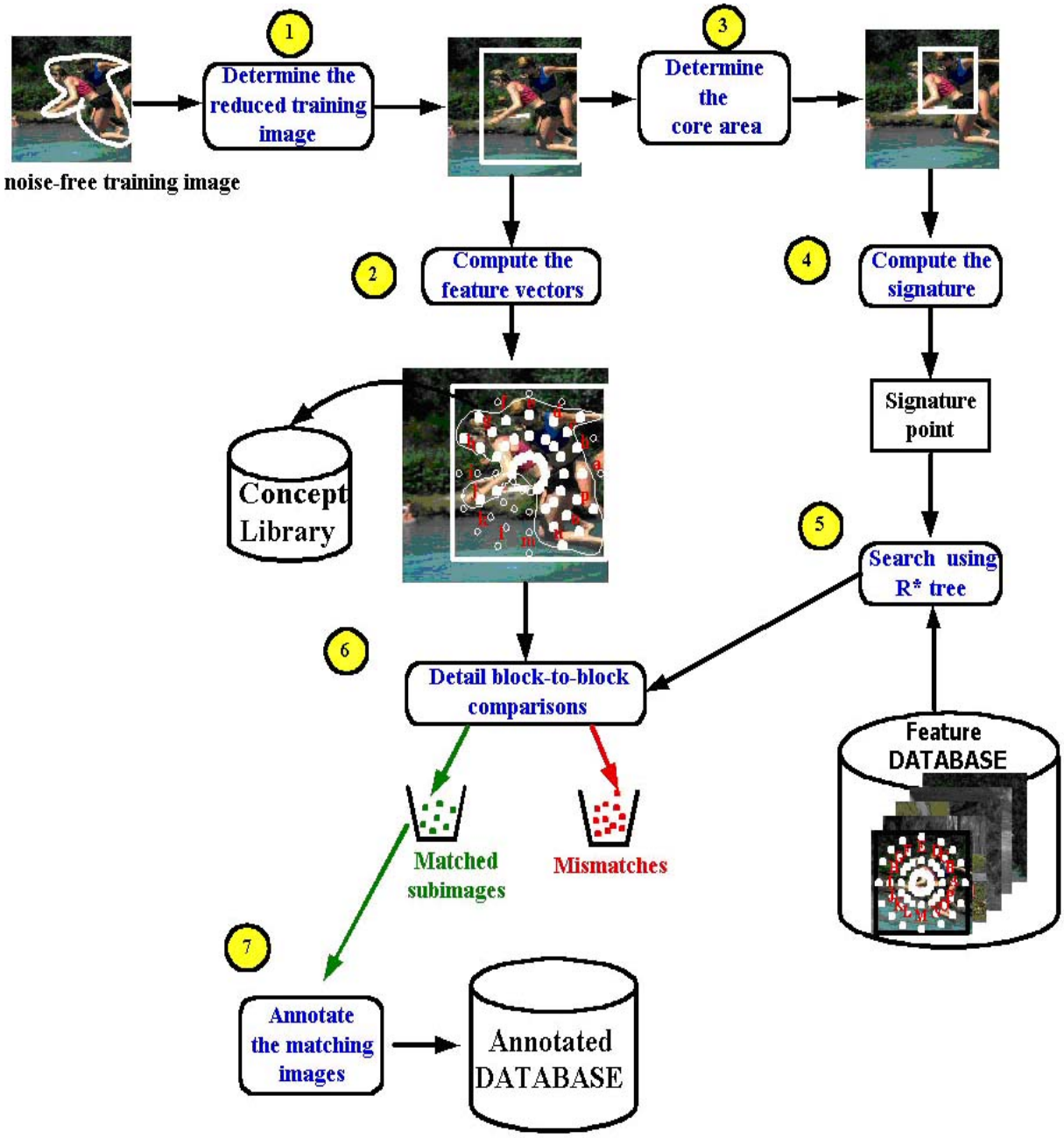


Figure 5: The ASIA algorithm

2.6 Experimental Study

Experimental studies are presented in this section. First, the database used for the experiments is discussed. Then the test environment is described. Finally, the experimental results are examined.

2.6.1 Image Database

A commercial image database titled *Art Explosion*, available from Nova Development is used. This database contains 15,000 square images. Each has 256 x 256 pixels. The images have been partitioned by content into 120 categories. 125 more images of cars, cups, plates, etc, are added to facilitate studies. Please note some of these pictures are not realistically oriented. However, they were designed to test and demonstrate the effectiveness of the proposed technique.

Sample training images, which would yield unfair advantages, were deleted from the database. In total, there were 70 training images from 25 main categories, including: fish, shell, coral, worm, starfish, sea lion, crab, building, art status, art picture, people, animal, balloon, bird, boat, sea, car, cup, bowl, plate, tree, water, sky and ground. For each training image, a contour was drawn around the desired object to indicate the area of interest. This area determines its category concept label.

2.6.2 Test Systems

Experiments were processed on a 1-GHz Pentium III processor running the Microsoft Windows 2000 operating system. The off-line annotation for the database of 15,125 images took ASIA approximately one day. To facilitate comparisons, the system based on the Monotonic Tree approach was processed with the same database on a Sun Ultra 5. This annotation process took approximately eight days.

2.6.3 Experiments

Some of the experimental results for the ASIA system are presented in Figures 6 through 10. For each test case, a sample noise-free training image is shown with its associated contour, followed by the top-20 matching image candidates selected by the ASIA system. Each of these images, indeed, contains the visual concept being annotated.

The results for the Monotonic Tree approach are summarized in Table 1. The results for the ASIA approach are summarized in Table 2. The images selected by Monotonic Tree approach is not shown because it does not rank the results. Instead, data can be only reported on the number of images annotated correctly, incorrectly or missed. Since there are too many database images, the amount of manually scanned database images have to be limited to 1,000. They are judged for accuracy with reference to visually correct interpretations. The results indicate that the Monotonic Tree approach is not as effective. The result of "sky" annotation is significantly better than the other three cases. This is due to the fact that a simple model is sufficient for the "sky" concept whose characteristics are relatively simple. On the contrary, one cannot rely on a simple model representing the average case to capture the characteristics of

water, which may reflect the surrounding environment. As a result, the accuracy for "water" annotation is very low. From table 2, the ASIA approach gives high percentage correct annotations for all these test cases.

Table 1: Experimental results for the Monotonic Tree approach (MT)

	Test cases for visual concepts			
	water	**sky**	**building**	**tree**
Visually correct	219	671	212	278
MT annotations	88	526	144	149
MT correct annotations	65	485	70	137
MT incorrect annotations	23	41	74	12
MT misses	154	186	142	141
Percentage correct annotations	29.7	72	33	49

Table 2: Experimental results for ASIA

	Test cases for visual concepts			
	water	**sky**	**building**	**tree**
Visually correct	219	671	212	278
ASIA annotations	313	630	283	350
ASIA correct annotations	217	558	201	232
ASIA incorrect annotations	96	72	82	118
ASIA misses	2	113	11	46
Percentage correct annotations	99.0	83.0	94.8	83.4

The visual concepts of "water," "sky," "building," and "tree" for the above studies are picked because the Monotonic Tree system, obtained from its authors, does have those same concepts in its relatively smaller Concept Library. To demonstrate that ASIA is robust with respect to translation, rotation and scaling of the matching objects, the experimental results are shown based on the visual concept "cup" in Figure 10. Again, it shows that ASIA annotated correctly the first 19 out of the 20 highest-ranked images regardless of the translation, rotation and scaling in the cup object. The result sets show robustness to translation, scaling, and rotation invariants at matching object level.

2.7 Conclusions

In this chapter, the ASIA approach is presented to support query-by-concept in image retrieval systems. Compared to traditional systems based on the query-by-example model, ASIA, based on textual queries, is easier to use. It is substantially faster on-line because extraction and

comparisons of low-level physical features are not necessary. ASIA is also more reliable since it has the luxury to spend substantially more time to compare the images off-line during the annotation process.

Few fully automatic query-by-concept systems have been reported. In comparison with a recent system based on the Monotonic Tree technique, my experimental results indicate that ASIA is significantly more accurate. This is due to the fact that the Monotonic Tree approach is based on comparing database images with models of various visual concepts. Their models represent the average cases and are not specific enough to accurately recognize all cases. Another drawback is due to the fact that building these models involves specifying many more thresholds to define various concepts. This process is tedious and fairly subjective.

Another recent query-by-concept system is CAMEL (*Concept Annotated iMagE Libraries*). CAMEL is not available to the public to access to this system. Therefore, CAMEL is not included in this comparative study. Nevertheless, CAMEL does not support noise-free training images and is not rotation invariant. These limitations would affect its performance significantly. Another drawback of this system is that query processing still relies on comparing low-level physical features. That search mechanism is relatively slow. ASIA does not have any of the above drawbacks.

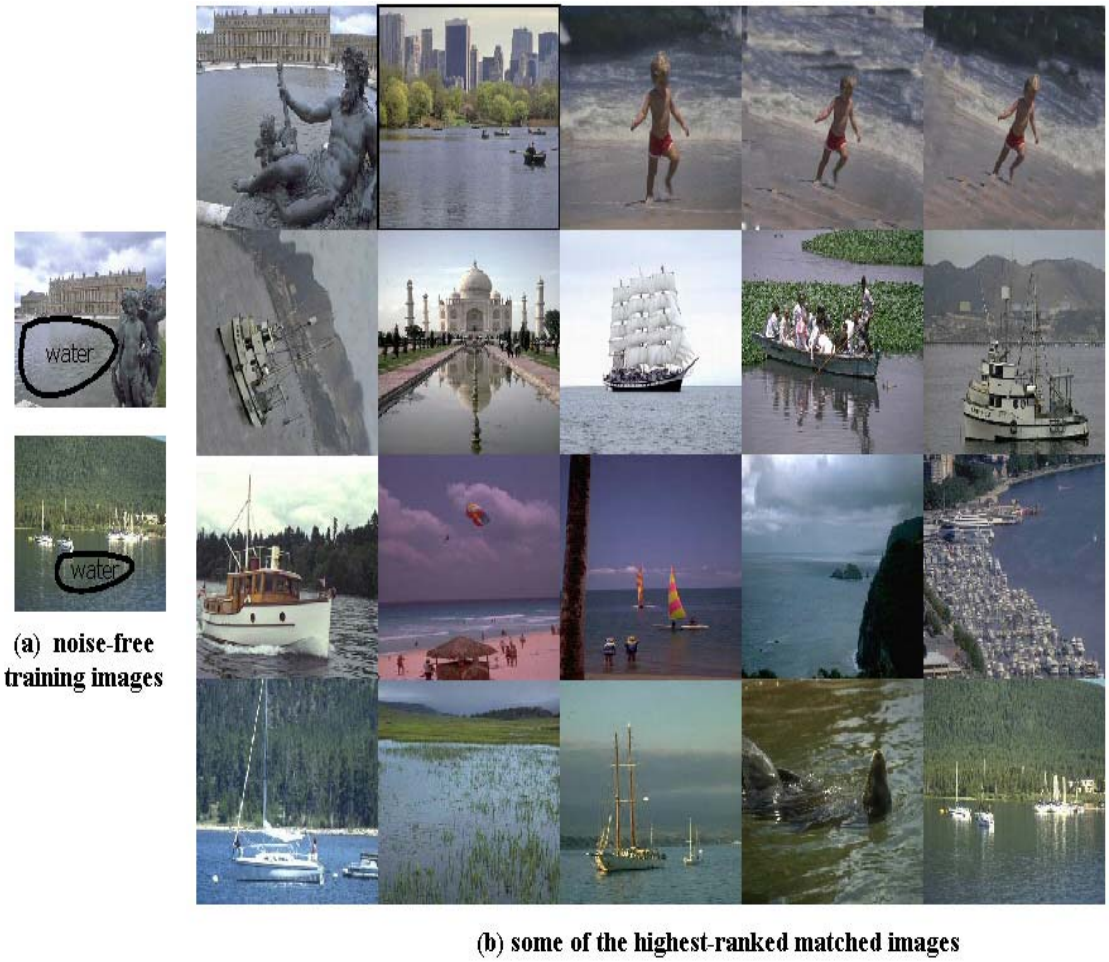


Figure 6: ASIA result set 1

This result set 1 (increasing rank: left-to-right) returned in response to the noise-free training images (left). All top-ranked annotated images contain “water” at any location.

Figure 7: ASIA result set 2

This result set 2 (increasing rank, left-to-right) returned in response to the noise-free training images (left). All top-ranked annotated images, except one, contain “sky.”

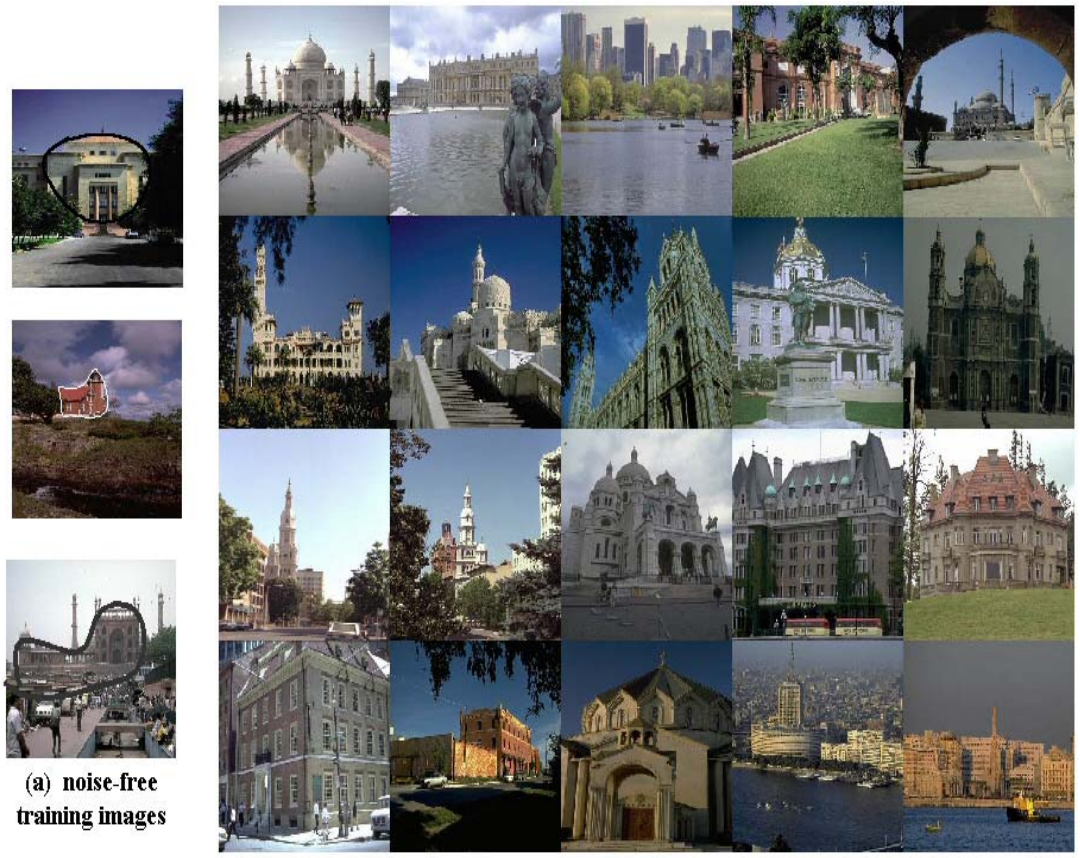


Figure 8: ASIA result set 3

This result set 3 (increasing rank, left-to-right) returned in response to the noise-free training images (left). All top-ranked annotated images contain “building,” which has many similarity matches.

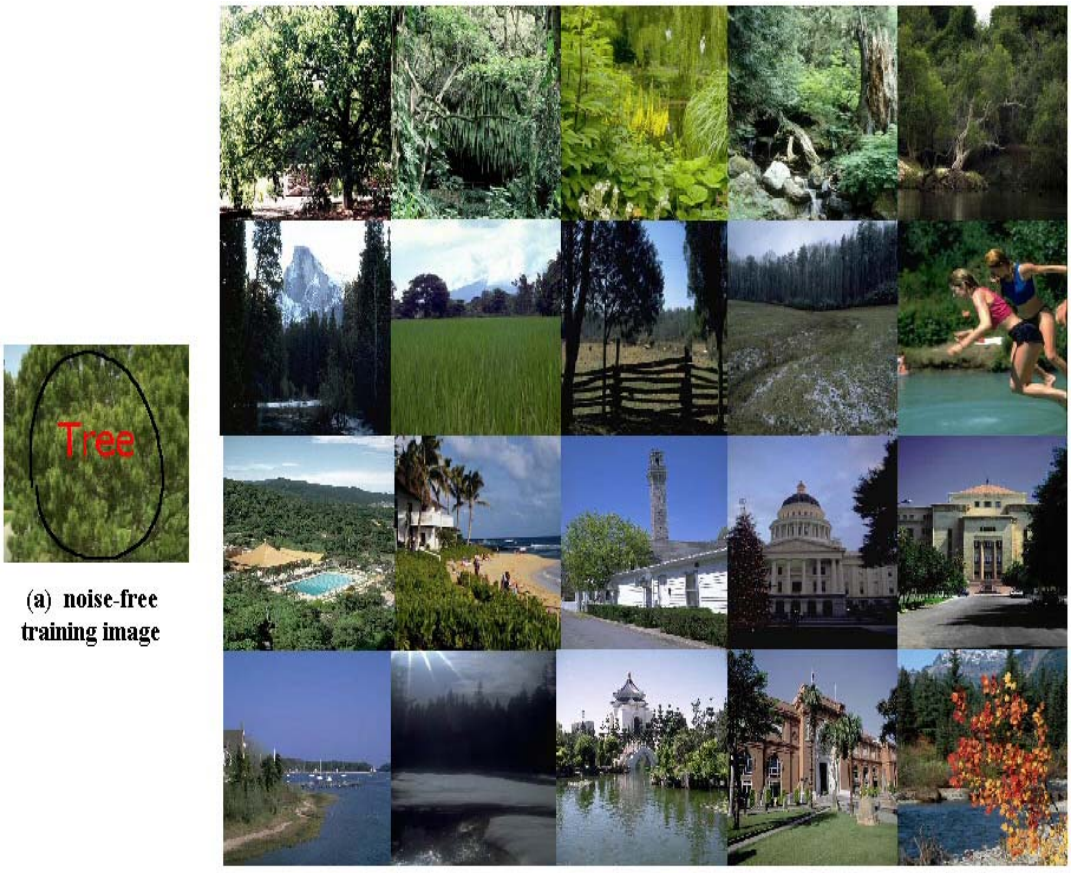


Figure 9: ASIA result set 4

This result set 4 (increasing rank, left-to-right) returned in response to the noise-free training image (left). All top-ranked animated images contain “tree” of any size and at any location.

(a) noise-free training image

(b) high-ranked matched images

Figure 10: ASIA result set 5

This result set 5 (increasing rank, left-to-right) returned in response to the noise-free training image (left). All the first 19 out of 20 top-ranked annotated images contain “cup” of any size, rotation and at any location.

CHAPTER THREE: RANGE-DISTANCE

The first phase of ASIA is an efficient new design of the QBE method. Example query images used as a training set are plugged into the designed QBE. Their result sets are annotated and retained for the annotated image database. This is an important phase for creating an accurately annotated image database. Although ASIA produces more accurate annotation than other techniques, ASIA uses many dimensions of low-level features and its matching time is quite higher than other QBE techniques. Chapters 3 and 4 focus on novel techniques which become efficient QBE techniques for solving the drawbacks of ASIA. Range-distance, the second technique, is the focus of this chapter.

For more than a decade query-by-one-example has been a popular query system for content-based image retrieval (CBIR). One query image is not sufficient to form its semantics or concept. For example, to build a concept "car," one needs many examples of car images in various colors. While the distance metric is used for measuring the similarity between images, all *N* dimensions of features (the static) have been used in the distance metric for other previous QBE techniques. For matching car images in any color, the color features should be discarded in the distance metric. Proposed is a novel approach called range-distance. In this approach, users can query by using one query image or by using multiple groups of query images. There are three possible groups: relevant (positive), irrelevant (negative) or neutral groups. For each feature within these groups of query images, their range distances are used for adjusting weights. As a result, some features may not be employed in the distance metric. This distance metric then becomes a dynamic distance metric for QBE. The range-distance approach is able to achieve a higher degree of precision and recall and, at the same time, significantly reduces the time

complexity of matching. This approach is tested against the *ImageGrouper* method. The results show that this range-distance approach is an effective and efficient technique for query-by-example(s). Similar to ASIA, the proposed range-distance approach is suitable for image database annotation.

3.1 Introduction

The most current QBE research [27-32] uses query-by-one-example for multimedia information. There exists previous research ([20] and [21]) that has used the combination of query images. From [21], this paper discussed analyzing feature-to-semantics mapping that such a query-by-one-example cannot realistically lead to scalable, satisfactory query performance. More specifically, they cluster a small image dataset based on the images' perceptual features and show that these image clusters are not coherent to the semantic categories of the images. Though some image categories are separated from the others in the input space formed by the perceptual features, most categories are co-located in more than one cluster. For a query concept that is mixed with others in a number of clusters, the query-by-one-example simply lacks information to clearly identify the target of the query concept and thus cannot achieve satisfactory query results.

ImageGrouper is described in [20]. It contains a *Graphical User Interface* (GUI) that allows users to interactively compare different combinations of query examples by dragging and grouping images on the workspace. The user can quickly review the query results as they are displayed in another portion of the workspace. Combining different queries is also easily done. Users can query with multiple positive groups and negative groups. Each group is composed of a

user's cluster of images; according to what the user thinks would be good representative images. This system uses the Euclidean distance of all of the static N dimensions of those features in a distance metric. They use the covariant between features contained within each group and between groups to adjust the weights in the distance metric. The covariant metric is minimized within the positive group and, concurrently, the covariant metric is maximized between the negative and positive groups.

An effective approach for QBE is still an open research problem. Range-distance, a novel QBE approach, is proposed in this chapter. In this approach users can query using a single query image or with multiple groups of query images. Users can specify three possible types of groups: positive, negative or neutral groups. For each feature within these groups of query images, their range distances are used for adjusting weights. As such, some features may not be employed in the distance metric. Therefore, the distance metric becomes a dynamic distance metric. In this chapter a novel dynamic distance metric is proposed. The number of features in a dynamic distance metric can be varied from 1 to N, where N is the dimensionality of the features. In other words, the dynamic distance metric is the distance metric with a various number of feature vectors. The range-distance approach is based on the matching of the dynamic distance metric. The matching algorithm is a fast computation and provides a high percentage of recall and precision. Experimental results show that this method offers a significant improvement over the *ImageGrouper* method, a recently proposed alternative. The range-distance technique is presented in the next section.

3.2 Feature Vectors

For each image, the extracted features of *ImageGrouper* are modified, because when used in this manner, both approaches get better results than the previous *ImageGrouper*'s extracted features. The visual image features in the range-distance approach consist of thirty-seven dimensional numerical values from three groups of features: *color*, *texture* and *edge structure*. Extracted features in the *texture* group are modified as follows. The standard deviation is a good representation if there are a large number of query images. Unfortunately, users do not normally have enough time to select many query images. For that reason, the average is a good representation when using just a few query images. Therefore, in the *texture* group, the average is used as a feature instead of the standard deviation. For the remainder (*color* and *edge structure*), extracted features are identical to the previous extracted features in the *ImageGrouper*. These features are extracted from the images and indexed in the meta-data database offline.

For color features, the HSV color space is used. The first, second and third moments for each of the H, S and V components are computed as features. Therefore, there are nine color features for each image.

For texture, *Wavelet-based Texture Features* [21][22][23] are used. First, the *Wavelet Filter Bank* [21][24] method is applied to each image, where the images are decomposed into ten de-correlated sub-bands (3 levels), as shown in Figure 11. The upper left image in the wavelets image is the low frequency sub-band, while the lower right image is the high frequency sub-band. For each sub-band, the average of the wavelet coefficients is extracted. Therefore, the total number of these texture features for each image is ten.

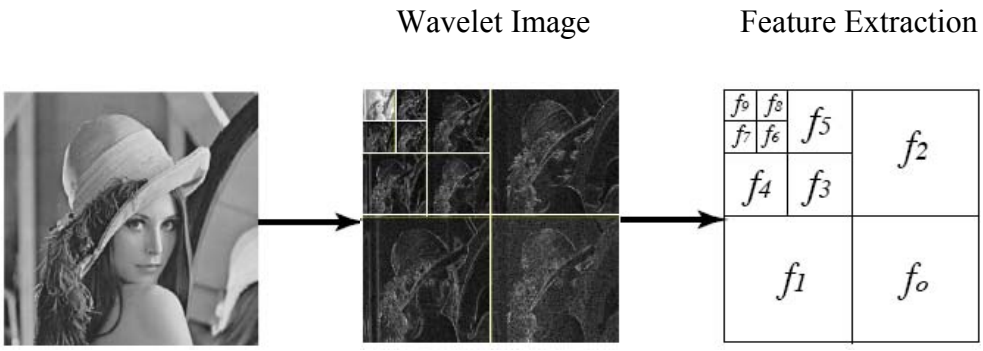


Figure 11: The wavelet texture features

For the edge structures, the *Water-Filling Algorithm* [25][21] is applied to the edge maps of the images. The Sobel filter is applied to each image and then followed by the thinning operation [26] to generate its corresponding edge maps. From the edge map, eighteen elements are extracted. These features include the longest edge length, the histogram of the edge length and the number of forks on the edges.

3.3 Query Sets

The query set consists of three possible groups: positive, negative or neutral. The positive group is the group of images that either have very high scores or are very close to a user's preference in deciding what most closely resembles his query. The negative group consists of the group of images that the user may want to filter from the result sets. The neutral group consists of that group of images about which the user is unsure.

3.4 Range Distance

Definition 1: For feature f_i, the lower bound (l_i) = $\min\{\forall f_i\}$ and the upper bound (u_i) = $\max\{\forall f_i\}$. The range distance (d_i) = $|u_i - l_i|$.

From Section 3.2, there are a total of 37 extracted features for each image. For each feature in each user's cluster (positive, negative and neutral groups), the corresponding lower and upper bounds are computed. The range distance is the distance between the lower bound and upper bound for each feature, as shown in Figure 12.

In Figure 12, only two dimensional features, f_i and f_j, are illustrated. Assume there are three images, called A, B and C. For feature f_i, $l_i = \min\{f_{iA}, f_{iB}, f_{iC}\}$ and $u_i = \max\{f_{iA}, f_{iB}, f_{iC}\}$. Then $d_i = |u_i - l_i|$. Similarly, for feature f_j, $l_j = \min\{f_{jA}, f_{jB}, f_{jC}\}$ and $u_j = \max\{f_{jA}, f_{jB}, f_{jC}\}$. Then $d_j = |u_j - l_j|$.

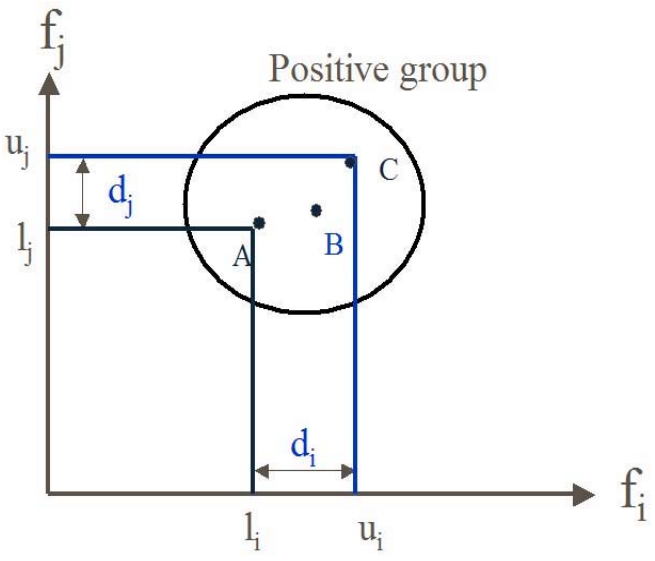


Figure 12: The range distances d_i and d_j

3.5 Dynamic Distance Metric

Definition 2: The distance metric $D(I,q)$ between any image (I) and query image (q) is defined as $D(I,q)=\Sigma w_i|I_i-q_i|$, where w_i is a weight in range [0,1] for the feature i. I_i is the feature i of database image I. The q_i is the feature i of query image q.

In the range-distance approach, users can query using a single query image or with multiple groups of query images. Users can specify positive, negative or neutral groups. Within each of these groups of query images, the range distances are utilized to adjust the weights in the distance metric. An algorithm describing how to adjust the weight is in the following section. Some weights will be zero and may not be used in the distance metric; this is called a dynamic distance metric.

3.6 The Range-Distance Algorithm

The matching algorithm is as follows:

1) For each positive, negative and neutral group
    - For each feature (f_i), compute the lower bound (l_i), the upper bound (u_i) and then compute its range distance (d_i) = u_i - l_i
    - Therefore, there are d_{ip}, d_{ine}, d_{inu} for the range distance in positive, negative and neutral groups, respectively.
2) For each feature (f_i)
    - compute $Min\_d_i = \min\{d_{ip}, d_{ine}, d_{inu}\}$

$flag_i = 0$;

if $Min\_d_i = d_{ine}$, then $w_i = 0$

else if $Min\_d_i = d_{inu}$, then $flag_i = 2$

else $flag_i = 1$

3) For each feature i and ($flag_i = 1$ or $flag_i = 2$)

3.1) Compute $D = \sum_i d_i$

where if $flag_i = 1$, then $d_i = d_{ip}$

else if $flag_i = 2$, then $d_i = d_{inu}$

3.2) Compute $d_i' = d_i / D$

where if $flag_i = 1$, then $d_i = d_{ip}$

else if $flag_i = 2$, then $d_i = d_{inu}$

3.3) Compute weight $w_i = 1- d_i'$

if $flag_i = 2$, then compute $w_i = w_i/num$

where num = the number of members in the neutral group

4) Calculate the distance metric

For each feature i in the positive group, compute the feature average (q_i) from all group members. This forms the query features (q). For each database image I, compute the distance metric $D(I,q)$ with the following equation.

$$D(I,q) = \Sigma (|I_i-q_i| \, . \, w_i)/(\Sigma w_i)$$

Where $w_i/(\Sigma w_i)$ is a normalization of a weight feature i into the range [0,1]. Then the results are sorted in the ascending order and then displayed with the top K nearest neighbor.

For each feature, its smaller range distance dominates the bigger ones. From step 2, if $Min\_d_i = d_{ine}$, then $w_i = 0$. On another hand, if a feature i in the negative group dominates the feature i in the positive group, it means that users do not want this feature to be included. Therefore, the feature i should be discarded from the distance metric. This makes the distance metric become dynamic. Steps 2 and 3 are an automatic approach to adjusting the weights from the user's information. The dominant features between the positive group and the neutral group are selected in the dynamic distance metric. The motivation for adjusting the weights comes from the fact that the smaller range distances should have more weight. Each group is the user's cluster of images. If user indicates more correct clusters, then this range-distance approach produces a higher percentage of the precision and recall results.

In the *ImageGrouper* approach, there is no meaning for the neutral groups. In the range-distance approach, however, these neutral groups are set with a lower weight when compared with the positive groups, as shown in step 3.3 of the above algorithm.

3.7 Time Complexity

Assume there are N dimensions of features and k is the number of members in all groups. From finding weights (in step 1-3 of the range-distance algorithm) in the last section, the time complexity is O(kN). The range-distance approach uses the simple absolute distance metric. In the *ImageGrouper approach*, covariants between features in each group are used to adjust the

features' weights. The time complexity for finding weights is $O(kN^2)$. They use the Euclidian distance metric that has a higher time complexity than the range-distance approach. Time requirements of both methods are reported in the experimental study section.

3.8 Experimental Study

The various datasets are described and the comparative study is discussed in the following sub-sections.

3.8.1 Datasets and Metrics

A dataset of about 3,000 images was selected from the COREL image database and the other images were added to test various features of the range-distance technique. To compose queries, images were selected from five categories: dishes, lions, views of the sky, flowers and cars. There are 18 images of 3-brown-dishes, 10 images of lions, 53 images of skies, 36 images of flowers and 165 images of cars in this dataset.

For each dataset image and query image, the 37 features were extracted in three groups (*color*, *texture* and *edge structure*) as described in section 3.2. Test sets were processed on a microcomputer with a 2.4 GHz Pentium IV processor. Programs are coded in C language and Java codes are used for their GUI.

The *ImageGrouper* approach used software that was downloaded from Image Formation and Processing Group at University of Illinois at Urbana-Champaign (http://www.ifp.uiuc.edu/~nakazato/grouper/).

3.8.2 Comparative Study

There were a total of 150 query sets. These query sets consisted of a variety of images including positive, negative and neutral groups. The same query sets were tested with both the range-distance approach and the *ImageGrouper* approach. The average time required for *ImageGrouper* was 0.456 seconds, while the average time required in the range-distance approach was 0.000667 seconds. In this experiment, the range-distance approach is more than 600 times faster than that of the *ImageGrouper* approach.

Some experimental result sets are shown in Figures 13-15. In Figures 13-15, (A) was the result set from the range distance approach and (B) was the result set from the *ImageGrouper* approach. The query set is on the right-hand side and the results on the left-hand side. The results are ordered from left to right and top to bottom. In Figure 13, this query tried to query images of dishes that have a 3-dish structure, excluding images of only one dish. From the results, the range-distance approach is more efficient and more accurate than the *ImageGrouper* approach. In Figure 14, this query tried to retrieve images of the red or pink flowers. People wearing red and black clothes were selected in the negative query group. The red car was selected as a neutral query set. Again, in this example, the range-distance approach is more effective than the *ImageGrouper* approach. In Figure 15, this query tried to retrieve images of brown lions, but not a white tiger. Again, in this experiment, the range-distance approach produces a better quality set of matching images than the *ImageGrouper* approach. The reasons that the range-distance is a faster and more accurate approach than the *ImageGrouper* approach are as follows:

- The range distance can be used to identify the dominant features and discard the less important features.

- Only the dominant features in the distance metric make the range-distance approach more effective. The *ImageGrouper* uses all of the same N-features in the distance metric. The less important features that are employed in the distance metric can not filter out the irrelevant images.
- The range-distance approach computes only the absolute distance features in the dynamic distance metric. In comparing the time complexity between the two approaches to adjust the weights, the range-distance approach takes O(kN) time while the *ImageGrouper* spends O(kN^2) time. The *ImageGrouper* computes the Euclidian distance of all the same N-features in the static distance metric. Thus the range-distance approach is faster than the *ImageGrouper* approach.

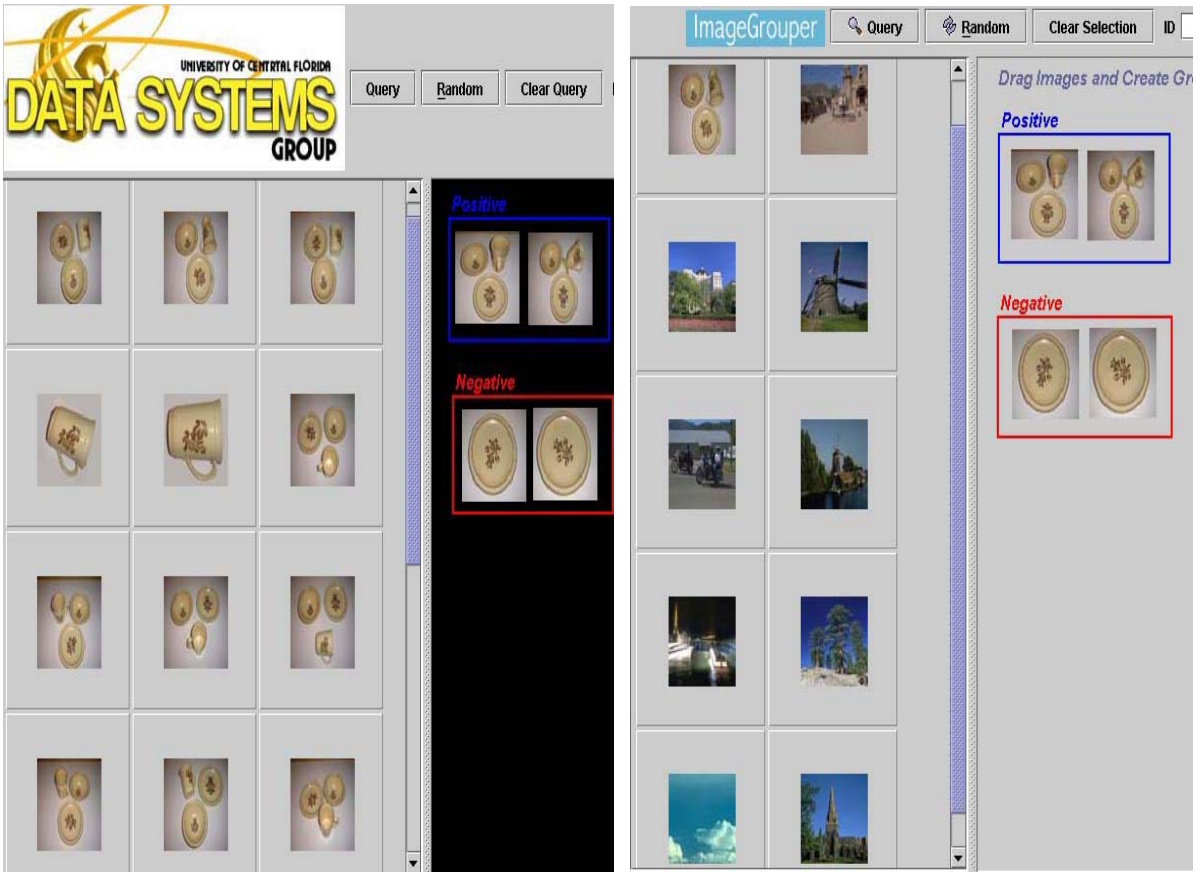
A summary of the recall and the precision rates between the range-distance approach and the *ImageGrouper* approach is shown in Table 3. The approach that had higher rates was more precise. The range-distance approach gave better recall and precision rates than the *ImageGrouper* approach for all categories of the query sets. Therefore, the range-distance approach is a more efficient approach than the *ImageGrouper* approach.

3.9 Conclusions

An efficient and effective approach for QBE is still an important open research problem. An algorithm is proposed to automatically adjust all the features' weights by using their range distances. These weights are varied, not static, and are also used in the dynamic distance metric. The presented technique achieves a high recognition rate and, at the same time, significantly reduces time complexity of matching. The results show that the range-distance approach is an effective and efficient technique for QBE.

Table 3: Summary of the recall and precision rates in example categories

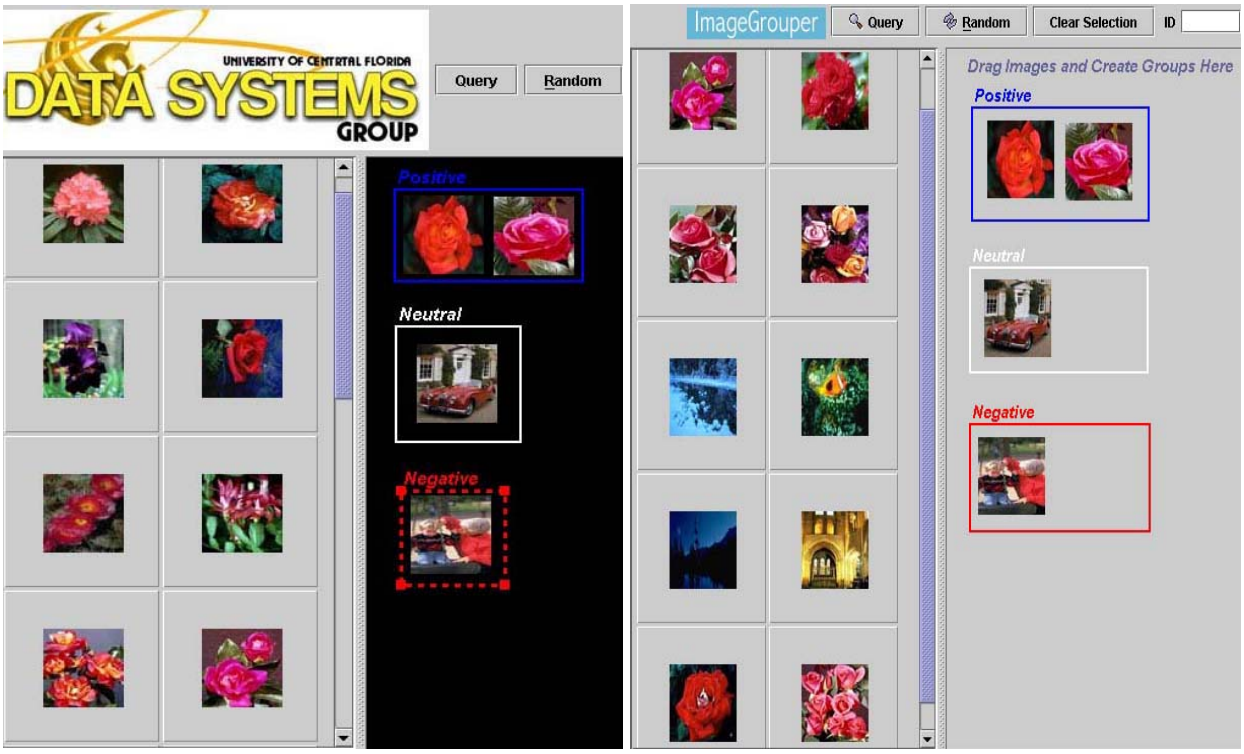
Categories	**Range-distance approach**		***ImageGrouper* approach**	
	Recall	**Precision**	**Recall**	**Precision**
Dishes	1.0	0.8	0.56	0.45
Lions	0.94	0.9	0.75	0.75
Sky	0.87	0.97	0.72	0.83
Flowers	0.83	0.94	0.72	0.86
Cars	0.84	0.88	0.54	0.65



(A) **(B)**

Figure 13**:** Result set 1 (the range-distance approach & the *ImageGrouper* approach)

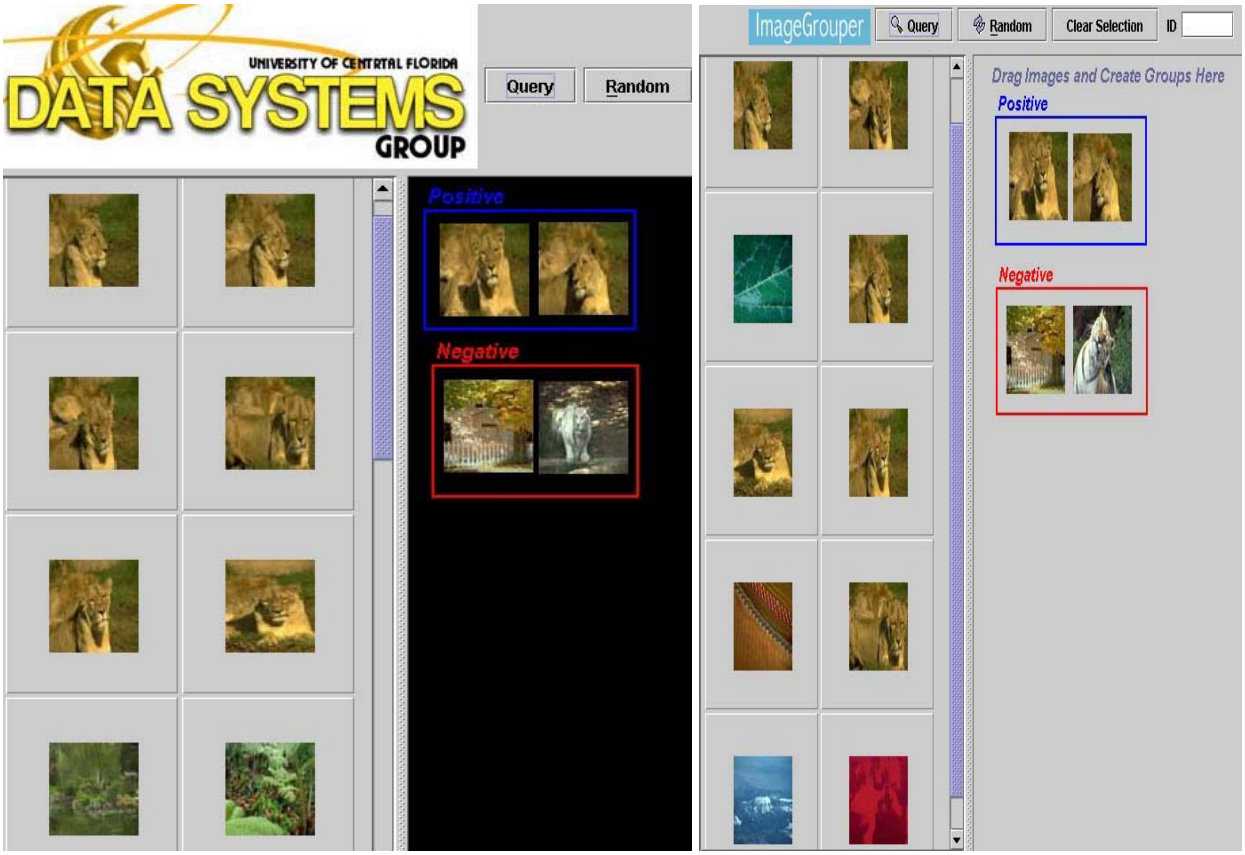
Figure 13 (A) is an example of the range-distance approach, while Figure 13 (B) provides an example of the *ImageGrouper* approach. For each approach, there is a query set containing two groups of images on the right-hand side, while the top ranked result set is on the left-hand side. The results are ordered from left to right and top to bottom. This query set tries to query images of dishes containing a 3-dish structure, while excluding images of only one dish. From the results, the range-distance approach is more efficient and more accurate than the *ImageGrouper* approach.



(A) **(B)**

Figure 14**:** Result set 2 (the range-distance approach & the *ImageGrouper* approach)

Figure 14 (A) is another example of the range-distance approach. Figure 14 (B) displays another example of the *ImageGrouper* approach. For each approach, the query set contains three groups of images on the right-hand side, while the top ranked result set is on the left-hand side. This query tries to query images of red or pink flowers, but not people who wear red clothes. Again, in this experiment, the range-distance approach is more effective than the *ImageGrouper* approach.



(A) **(B)**

Figure 15**:** Result set 3 (the range-distance approach & the *ImageGrouper* approach)

Figure 15 (A) provides another example of the range-distance approach. Figure 15 (B) presents an example of the *ImageGrouper* approach. For each approach, the query set consists of two groups of images on the right-hand side, while the top ranked result set is on the left-hand side. This query tries to query images of lions that are brown in color, but should not find a white tiger. Again, this experimental result shows that the range-distance approach produces a better quality set of matching images.

CHAPTER FOUR: SKELETON-MATCHING

Skeleton-matching, the third technique is addressed in this chapter. A skeletal graph is a shape representation recently developed for shape recognition. In this chapter, a novel many-to-many matching algorithm, called skeleton-matching, is proposed, for matching skeletal graphs. The topology of skeletal graphs is captured and compared at the node level. Such graph representation allows preservation of the skeletal graph's coherence without sacrificing the flexibility of matching similar portions of graphs across different levels. By using appropriate sampling resolution, the skeleton-matching approach is able to achieve a high recognition rate and, at the same time, significantly reduces space and time complexity of matching. The skeleton-matching approach was tested against the Directed Acyclic Graph (DAG) method on noisy graphs and occluded or cluttered scenes. The results show that the skeleton-matching approach is an effective and efficient technique for shape recognition.

4.1 Introduction

Although there has been extensive work in the area of shape representation and matching, an effective approach to shape recognition is still an open research problem. It is well known that the underlying representation of the shape of objects can have a significant impact on the effectiveness of a recognition strategy. Shapes have been represented by their outline curves [35, 36], point sets, feature sets [37, 38] and by their medial axis [39, 40], among others. Each has its own strength and weaknesses, but few have not been extensively tested on noisy, occluded or

cluttered scenes. High space, time complexity and moderate recognition rate are still the limiting factors for their acceptance.

In this chapter, a technique called skeleton-matching is proposed. The skeleton-matching significantly reduces space and time requirements while improving recognition rate. The skeleton-matching approach is based on many-to-many graph matching of skeletal graphs constructed from shapes. Skeletal graphs are encoded at the node level: the topology of nodes (length to parent node, children and their relative positions) is captured in a table. The metric matches nodes at their children level not only to preserve coherence of shapes, but also the flexibility of matching across levels. The matching algorithm is shown robust to scaling, rotation and translation. It is less sensitive to noise and occlusion. Experimental results show that the skeleton-matching method is significantly better than a recently proposed alternative shock graph represented in the Directed Acyclic Graph (DAG) method.

Section 4.2 reviews related work in the area of shape recognition. Section 4.3 discusses the skeleton-matching technique in detail and its properties are proved. Section 4.4 presents the results of the experimental study. Concluding remarks are given in Section 4.5.

4.2 Related Work

Below, some recent work on shape matching are reviewed. The distinguishing features of a work are shape representation and matching, based on the extracted shape features.

In curve-based representations of shapes, the matching is typically based on either aligning feature points by an optimal similarity transformation or by finding a mapping from one curve to the other that minimizes an “elastic” performance functional, penalizing the “stretching”

and “bending” energies of deformation [35, 36]. The curve outline-based matching methods often suffer from one or more of the following drawbacks: asymmetric treatment of the two curves, sensitivity to sampling, lack of rotation, scaling invariants and sensitivity to articulations and deformations of parts.

Another type of shape representation models the shape outline as a point set and matches the point set using an assignment algorithm. Gold et al. [38] use *graduated assignment* to match image boundary features. In a more recent approach, Belongie et al [37] use the *Hungarian* method to match unordered boundary points, using a coarse histogram of the relative location of the remaining points as features. These methods have the advantage of not requiring ordered boundary points; however, the match does not necessarily preserve the coherence of shapes in that the relationship among portions of shape in the process of matching may not be preserved.

Shapes have also been represented by their medial axis, which can then be used for matching. Zhu and Yuille [41] have proposed a framework (FORMS) for matching animate shapes by comparing their skeletal graphs, using a branch and bound strategy. The inherent instabilities of their skeletal representation are accounted for by using a priori defined model graph. The applicability of their approach to inanimate objects is limited due to the choice of primitives used in modeling.

A variant of the medial axis is the shock structure, which is obtained by viewing the medial axis as the locus of singularities (shocks), formed in the course of wave propagation (grass-fire) from boundaries [42]. Many approaches to shock graph matching have emerged, including [34, 39, 40 and 44]. However, these approaches have not been extensively tested on noisy graphs, occluded scenes or cluttered scenes.

4.3 Many-to-Many Skeletal Graphs Matching

A skeleton is an undirected tree captured from a silhouette of a binary image using the medial-axis method [43], see Figure 16. A skeleton tree consists of nodes and edges. There is a length associated with each edge. Each node has a maximum of eight connected neighbors. The angle between edges is a multiple of 45 degrees. The level of a node is defined as how close it is to the center of the tree, where a center is a vertex u such that the maximum value of the distances between u and all other vertices is a minimum. It is a well-known fact a tree has either a single center, or two centers connected via an edge [45]. A skeletal tree will be represented from the leaf nodes toward the center(s). The "root" of this representation corresponds to a center node. In the case of two centers, selecting either as root node will determine the level of all nodes in the skeleton.

In order to represent a skeletal graph, the topology of its individual nodes, i.e., the nodes' connected neighbors, the relationship with each neighbor and the length of the edges is captured. A signature table is constructed to record information regarding the topology of nodes in the graph. There is only one table to store the nodes' information for the entire dataset.

The signature table consists of three attributes: signature, ID and "node detail". Thus, a row in the table is a 3-tuple. ID is a positive integer assigned to each signature. It serves as an identification of a node's topology. A signature consists of eight fields, corresponding to eight possible connected neighbors. Each field identifies the existence of a connected neighbor, the topology via an ID and the relationship with the node. A field containing 0 signifies there is no connected neighbor at that angle. A field of 3, for example, indicates there is a connected neighbor whose topological ID is 3, while a field of "P" implies the connected neighbor is its

parent node. The fields are filled starting with P (the parent) or the highest ID for root nodes and going in the clockwise direction.

Node detail captures the fact that there could be many nodes with the same signature and ID. It is a linked list of nodes, each in the form *node/length/parent_node/image_number (list of children nodes or parent)*, and *length* is the length of the edge connecting the *node* with its *parent_node*.

Table 4 shows the signature table filled with node information of the two examples of skeleton trees. The first row encodes the information for all leaf nodes. They have signature (P,0,0,0,0,0,0,0), which is assigned ID = 1. Node detail is a linked list of all leaves in both trees in the order of processing.

The next type of node, whose ID = 2, has signature (1,1,1,0,1,0,0,0). There is only one node, node 2 of Figure 16, is of this type. Therefore, node detail contains 2/0/0/1(3,4,5,1). The signature and the node detail indicate that node 2 of Figure 16 is the root node (length=0, and no parent), whose children nodes are nodes 3, 4, 5 and 1, counting clockwise.

The last row of Table 4 describes node 3 of Figure 17. This node is the root node and has three children nodes: 5, 2 and 4. Its signature indicates that its children node 5 is a node of ID=4, node 2 of ID= 3 and node 4 of ID=1, see Figure 17.

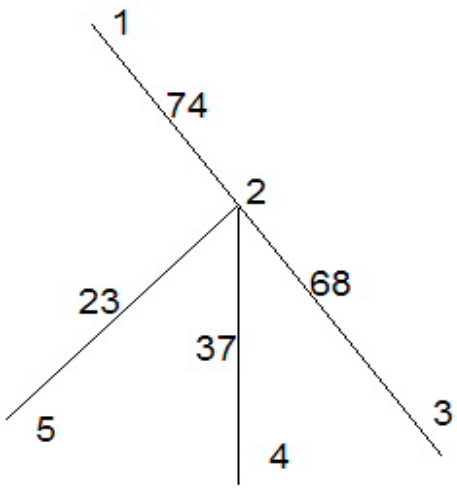


Figure 16**:** A skeleton tree in database

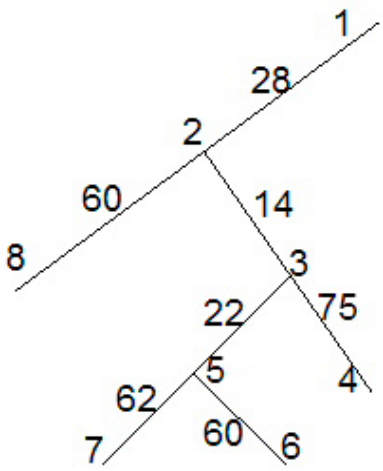


Figure 17: Another skeleton tree in database

Table 4**:** The signature table for Figure 16 and Figure 17

Signature	**ID**	**Node detail** Node/length/parent/figure number (nodes of children or parent)
(P,0,0,0,0,0,0,0)	1	1/74/2/16 (2), 3/68/2/16 (2), 4/37/2/16 (2), 5/23/2/16 (2), 1/28/2/17 (2), 4/75/3/17 (3), 6/60/5/17 (5), 7/62/5/17 (5), 8/60/2/17 (2)
(1,1,1,0,1,0,0,0)	2	2/0/0/16 (3,4,5,1)
(P,0,1,0,0,0,1,0)	3	2/14/3/17 (3,8,1)
(P,0,1,0,1,0,0,0)	4	5/22/3/17 (3,6,7)
(4,0,3,0,0,0,1,0)	5	3/0/0/17 (5,2,4)

4.3.1 Construction of the Signature Table

Node's signature is coded from leaves to root. The procedure to construct the signature table is given below. For each skeletal graph:

1) While the skeleton tree is not empty,
2) Code all leaf nodes.
3) Temporarily store the configuration of the parent nodes of the leaves (e.g., positions).
4) Delete all leaves from the skeleton tree.
5) Go to step 1.

Figure 18 shows a query skeleton. This is a 2-root skeleton, (i.e., there are 2 centers, nodes 2 or 3); therefore, two different signature tables are possible, one corresponding to each root node, see Tables 5 and 6. Note the assigned IDs in the tables; new IDs are created because new topologies were detected.

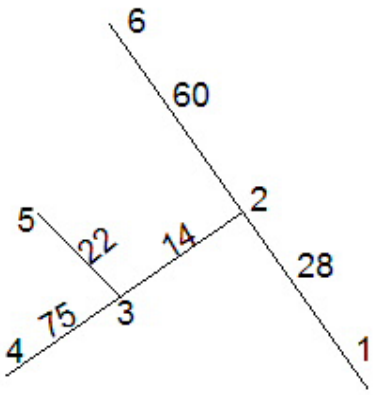


Figure 18**:** A query skeleton tree

Table 5**:** The signature table for Figure 18 with node 2 as root

Signature	**ID**	**Node detail**
(P,0,0,0,0,0,0,0)	1	1/28/2/18 (2), 4/75/3/18 (3), 5/22/3/18 (3), 6/60/2/18 (2)
(P,0,0,0,1,0,1,0)	6	3/14/2/18 (2,4,5)
(6,0,1,0,0,0,1,0)	7	2/0/0/18 (3,6,1)

Table 6**:** The signature table for Figure 18 with node 3 as root

Signature	ID	Node detail
(P,0,0,0,0,0,0,0)	1	1/28/2/18 (2), 4/75/3/18 (3), 5/22/3/18 (3), 6/60/2/18 (2)
(P,0,1,0,0,0,1,0)	3	2/14/3/18 (3,6,1)
(3,0,0,0,1,0,1,0)	6	3/0/0/18 (2,4,5)

4.3.2 Matching Shapes

The idea behind skeleton-matching for shape recognition is as follows. First, note that signatures are designed to ensure translation and rotation invariance. For non-root nodes, "P" is fixed at the beginning of the signature. Since there is no parent for the root node, circular-left-shift the query signature of the root node is done to identify all matches with DB nodes. In the similarity measure, this method also takes into consideration matching similar topologies in different scales.

When a skeleton is a 2-root type, there are two possible configurations (Figure 18). To ensure matching, e.g., between a 2-root query and a 2-root skeleton, both configurations of either need to be compared. One configuration for each database skeleton is chosen to code, either 1-root or 2-root. 2-root queries will be coded in both configurations.

Signatures are used to identify two types of matches: an exact match for two nodes of the same topology and at the same level, and a close match for two nodes of similar topologies at any levels. Exact matches can be declared when the two signatures are the same.

When neither of the DB node nor the query node is a root node, it is a close match if query node's children match some children of the DB node. For example, query signature (P,0,0,0,1,0,1,0) is a close match with the DB signature (P,0,1,0,1,0,1,0).

When the query node is a root node, the DB signature is compared with all circular-left-siftings of the query signature. For example, query's signature (3,0,0,0,1,0,1,0) is shifted to (1,0,1,0,3,0,0,0) and then to (1,0,3,0,0,0,1,0), where a close match with DB signature (4,0,3,0,0,0,1,0) is declared. For a close match, the order of the children in node detail is updated to reflect the change in its signature (shown in Table 7).

Table 7: An example for matching signatures (query signature: Table 6, DB signature: Table 4)

Query		**Database**	
Signature	**Node detail**	**Signature**	**Node detail**
(P,0,0,0,0,0,0,0)	1/28/2/18 (2), 4/75/3/18 (3), 5/22/3/18 (3), 6/60/2/18 (2)	(P,0,0,0,0,0,0,0)	1/74/2/16 (2), 3/68/2/16 (2), 4/37/2/16 (2), 5/23/2/16 (2) 1/28/2/17 (2), 4/75/3/17 (3), 6/60/5/17 (5), 7/62/5/17 (5), 8/60/2/17 (2)
(P,0,1,0,0,0,1,0)	2/14/3/18 (3,6,1)	(P,0,1,0,0,0,1,0) (exact match)	2/14/3/17 (3,8,1)
(3,0,0,0,1,0,1,0)	3/0/0/18 (2,4,5)	No match	
(1,0,1,0,3,0,0,0)	3/0/0/18 (4,_,5,2)	(1,1,1,0,1,0,0,0) (close match)	2/0/0/16 (3,4,5,1)
(1,0,3,0,0,0,1,0)	3/0/0/18 (5,2,4)	(4,0,3,0,0,0,1,0) (closest match)	3/0/0/17 (5,2,4)

The matching algorithm is summarized as follows:

1) Clear all marks on the DB signatures.
2) While there are query signatures:
    i. Select one query signature,
    ii. Identify all DB exact and close matched signatures,
    iii. Mark all DB matches accordingly,
    iv. Remove the query signature.

The procedure produces a set of marked signatures. From their node details, gather a set of matched nodes, and for each candidate match, their corresponding node details are paired. Since query images might be occluded, there can be fewer children for the query node. In that case, the list of children of the query node is expanded by using "_" to fill the positions. For example, a close match between (1,0,1,0,3,0,0,0) and (1,1,1,0,1,0,0,0), see row 4 of Table 7, also indicates a missing child in node 3/0/0/18(4,5,2), which is extended to 3/0/0/18(4,_,5,2) to compare against node 2/0/0/16(3,4,5,1). The similarity of the nodes' topology can be tested by comparing nodes 3,4,_,5,2 of the query skeleton with nodes 2,3,4,5,1, respectively, see Table 7.

To filter out unlikely (close) matches, (DB *node_length*)/(query *node_length*) ratios are used and computed their median. Matches whose ratio is outside the range [*median ratio – threshold, median ratio + threshold*] are eliminated, as shown in Tables 8 and 9. For each of the remaining matching pairs, exact or close match, the matching score is computed:

$$\text{score} = w\left(\frac{CN(T_1,T_2)}{\max(|T_1|,|T_2|)}\right) + (1\text{-}w)\left(\frac{CP(T_1,T_2)+1}{\max(|T_1|,|T_2|)}\right)$$

Where *CN* is the number of matched nodes, *CP* the number of matched nodes whose ratios in the range [*median ratio – threshold, median ratio + threshold*]. $|T_1|$ is the size of query

skeleton tree T_1, and $|T_2|$ is the size of DB skeleton tree T_2. w is the weight in range [0,1], which is set to adjust the relative significance of the two terms. In experiments, weight and threshold are set: $w = 0.5$ and $threshold = 0.25$.

With respect to Figure 18, the formula produces Figure16's score = 0.5(4/6) + 0.5((2 + 1)/6) = 0.58 and Figure17's score = 0.5(6/8) + 0.5((5 + 1)/8) = 0.75. This indicates Figure 17 is more similar to Figure 18 than Figure 16.

Table 8**:** An example for matching nodes from Table 7 between query image and Figure 16

Query nodes$_{length}$	**DB nodes$_{length}$**	**ratio**	**CN**	**CP**
1_{28}	-	-	0	0
2_{14}	1_{74}	5.28	1	0
3_{0}	2_{0}	-	1	0
4_{75}	3_{68}	0.9	1	1
5_{22}	5_{23}	1.04	1	1
6_{60}	-	-	0	0
		Median =1.04	Total CN = 4	Total CP = 2

Table 9**:** An example for matching nodes from Table 7 between query image and Figure 17

Query nodes$_{length}$	**DB nodes$_{length}$**		**Ratio**	**CN**	**CP**
	Begin	**Final**			
1_{28}	1_{28}	1_{28}	1	1	1
2_{14}	2_{14}	2_{14}	1	1	1
3_{0}	3_{0}	3_{0}	-	1	0
4_{75}	4_{75}	4_{75}	1	1	1
5_{22}	5_{22}	5_{22}	1	1	1
6_{60}	8_{60}	8_{60}	1	1	1
			Median =1	Total CN = 6	Total CP = 5

4.3.3 Properties of the Algorithm

To prove the properties of the skeleton-matching algorithm, a distance measure is defined that it is equivalent to the scoring function:

$$d(T_1, T_2) = 1\text{- score} = 1 - w \frac{CN(T_1, T_2)}{\max(|T_1|, |T_2|)} - (1-w) \frac{CP(T_1, T_2) + 1}{\max(|T_1|, |T_2|)}$$

Theorem 1: For any skeleton trees T_1, T_2 and T_3 the following properties hold true. (See proof at Appendix.)

$0 \le d(T_1, T_2) \le 1$

$d(T_1, T_2) = 0 \Leftrightarrow T_1$ and T_2 are isomorphic to each other

$d(T_1, T_2) = d(T_2, T_1)$

$d(T_1, T_3) \le d(T_1, T_2) + d(T_2, T_3)$

Theorem 1 indicates that the distance measure in skeleton-matching is a distance metric. It is also clear that the matching procedure based on node topology is translation and rotation invariant, and robust to scaling.

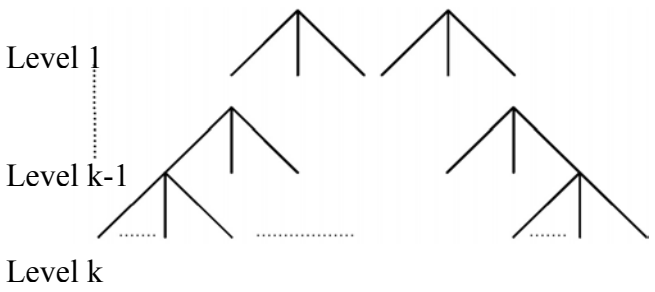


Figure 19: The skeleton graph is translated into levels of graph

4.3.4 Complexity Analysis

It can be shown that the number of rows, *m*, in the signature table is O(n), when the table contain a single graph, where n is the number of nodes of the graph. To see this, let n_1, n_2, ….., n_{k-1}, and n_k be the number of nodes at level 1, 2, …, k-1 and k, respectively. The total number of nodes therefore is $n_1 + n_2 + \ldots + n_{k-1} + n_k = n$. Since all nodes at level k are leaf nodes, they are encoded in the first row. Since there are n_{k-1} nodes at level (k-1), there are no more than n_{k-1} rows for this level. Generally, there are no more than n_{k-i} rows for nodes at level (k-i), for $1 < i < k$. *m* is therefore bounded by $n_{k-1} + n_{k-2} + \ldots + n_2 + n_1 \leq n$ or O(n). Finally, the number of entries in the node detail of all records is precisely *n*, for each node is encoded exactly once. It follows that the procedure requires O(*n*) space. Note that the number of nodes in a skeleton can be reduced by down-sampling the image, i.e., changing its resolution.

Similarly, the time complexity of the algorithm in the worst case is O(nm), where n is the number of query's nodes. This is true because of, for each query node, finding its exact and close matches in m rows.

Compared with the DAG method, the skeleton-matching method is a significant improvement. The DAG method takes $O(n^3M)$ time for matching, where M is the number of database images. In experiments, there were about m = 1,700 signatures for a total of M=4,000 database images. In terms of storage space, DAG uses the eigenspaces of the adjacent matrix, costing $O(n^2)$ for each image. The results of analysis are summarized in the following theorem.

Theorem 2: The space complexity of the skeleton-matching technique is O(*n*) for each query, and the time complexity is O(*mn*) for executing a query against the entire database, where *m* = the number of rows in the signature table, and *n* = the number of nodes in the query's skeletal graph.

4.4 Experiment Study

This section is divided into three sub-sections. The datasets are described in Section 4.4.1. The configuration of the datasets is explained in Section 4.4.2. In this section, the best resolution for the datasets is defined. The comparative study is discussed in Section 4.4.3.

4.4.1 Datasets

The dataset was from the Computer Science department at University of Toronto (http://www.cs.toronto.edu/~dmac/download.html) and some images were added to test various

features of the skeleton-matching technique. Images are views computed from 3-D graphics models. Each is centered in a uniformly tessellated view-sphere and a silhouette is generated for each vertex in the tessellation. A skeletal graph is computed for each silhouette with a particular block size. The depth first search algorithm is used to traverse the graph, and then mark it as a node where the graph changes direction. A total of 147 queries (occluded or unoccluded images) are executed. Recognition rates are used as the main performance metric, as they are for the DAG method.

4.4.2 Configuration

Since the block size greatly affects both processing time and recognition rate, first the best block size is determined to construct skeleton trees. Five different block sizes were experimented as follows: 1x1, 2x2, 4x4, 8x8 and 16x16. As expected, the processing time reduces as the block size increases, see Table 10.

Table 10: Processing times for various resolutions

Block size (pixels)	**Processing time (mins.) for 4,000 images**
1x1	28
2x2	17
4x4	11.29
8x8	8.46
16x16	8

The relationship between recognition rates and block size is more interesting; see Figure 20. When block size is increased from the pixel level of 2x2 to 4x4, the recognition rate improves. This indicates that block size 4x4 helps remove noise and smooth out the skeletal graphs. Recognition rates, however, deteriorate rapidly for larger block sizes. This is attributed to the fact that the skeletal graphs now become too coarse to capture shapes in sufficient detail. Since block size 4x4 yields the highest recognition rate at reasonable processing time, this block size is used for the skeleton-matching technique in subsequent experiments.

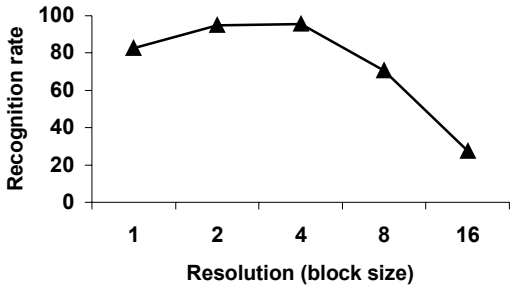


Figure 20: Recognition performance as a function of sampling resolution

4.4.3 Comparative Study

In this experiment, the performance of the skeleton-matching technique and the DAG method are evaluated on a set of unoccluded queries. The same dataset were used as the DAG's, consisting of up to 1,408 views of 11 objects with each object having 128 views. The recognition rates were measured as the database size increased. As seen from Figure 21, the recognition rate of the skeleton-matching approach is consistently better than the DAG's.

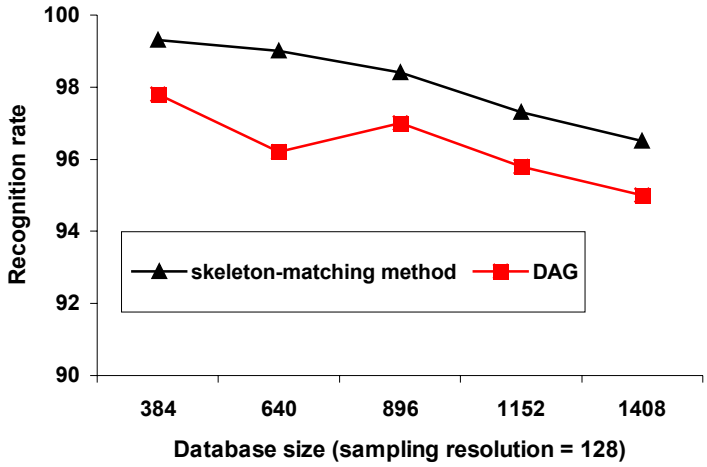


Figure 21: Recognition performance as a function of object database size

Figure 22 compares the performance of the skeleton-matching technique and the DAG on a set of unoccluded queries as the number of views varies. The results indicate that as the number of views per object increases, the skeleton-matching method achieves higher recognition rate faster than the DAG does.

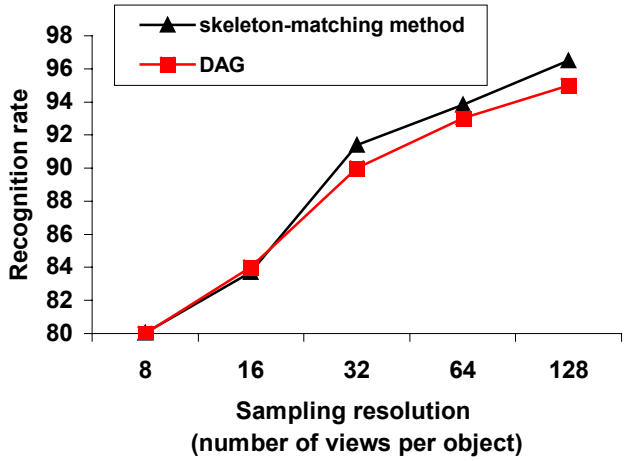


Figure 22: Recognition performance as a function of sampling resolution

Figure 23 shows the performance of both methods on a set of queries with various degrees of occlusion. The recognition rates comparably reduce as the percentage of occlusion grows. Both methods yield recognition of greater than 70, even for 50% occluded queries.

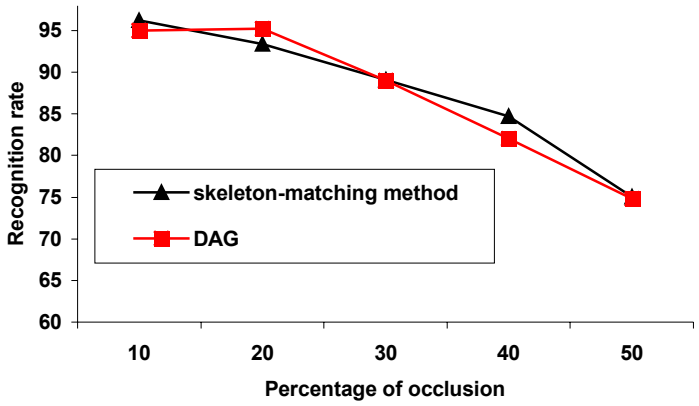


Figure 23: Recognition performance as a function of degree of occlusion

4.4.4 Example Queries

Figures 24 to 27 show the results of some example queries against the database of about 4,000 images. For each query, top-ranked images are displayed along with their scores. The higher the score is, the closer the match. Table 11 summarizes the results of these queries.

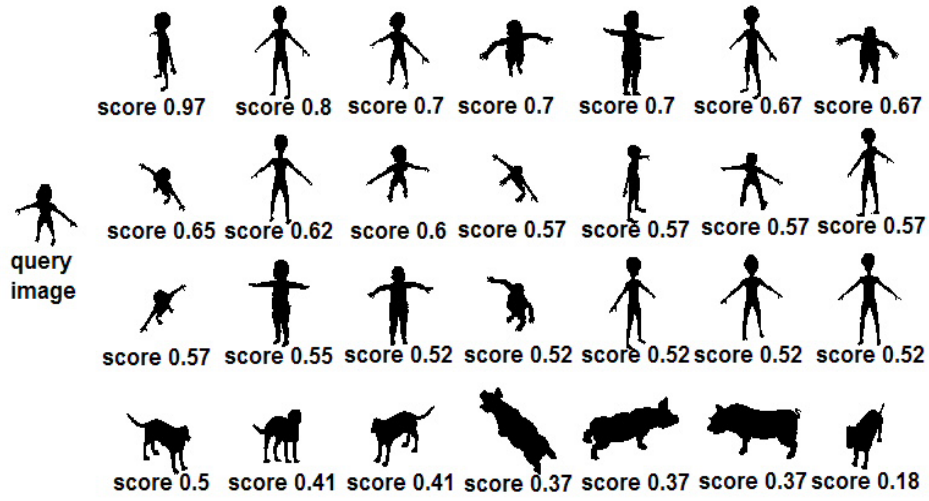


Figure 24: Result set 1: query image and top-ranked images

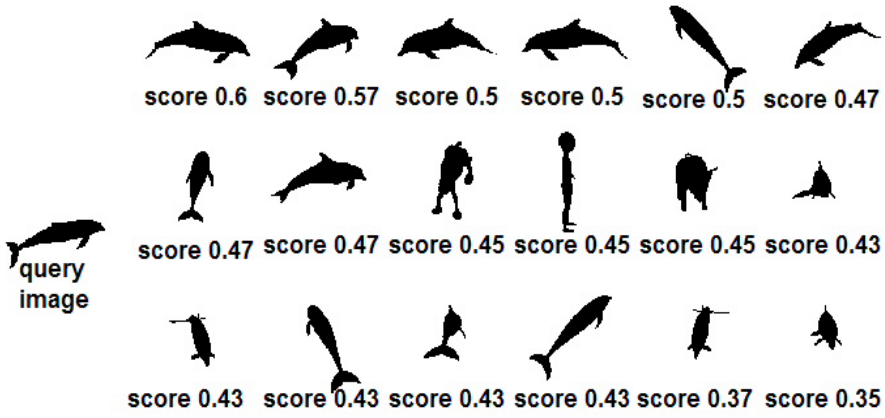


Figure 25: Result set 2: query image and top-ranked images

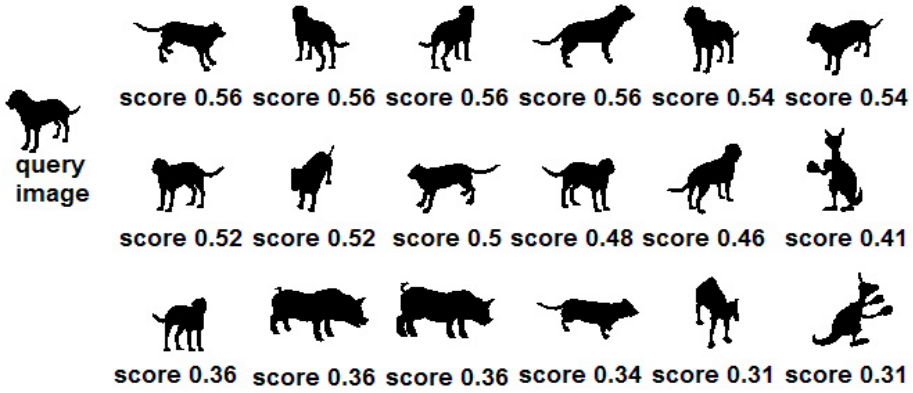


Figure 26: Result set 3: query image "unoccluded dog" and top-ranked images

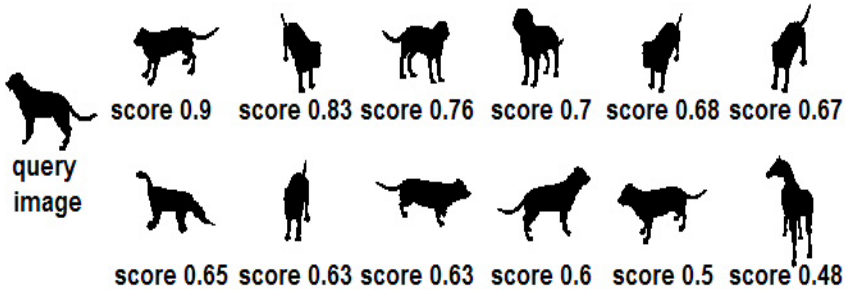


Figure 27: Result set 4: query image "occluded dog" and top-ranked images

Table 11: summary of the results of the queries

Figure	Query	Relevant	Retrieved	Recall
24	"human"	63	58	92%
25	"dolphin"	52	46	88%
26	"dog"	28	26	92.8%
27	"dog"	28	26	92.8%

Observe that the ranks of relevant images are very high, and the images are views at various angles and scaling of the queried object. In the last query, see Figure 27, for heavily occluded queried objects (dog's legs are occluded), the skeleton-matching approach is still effective to achieve high recall (compared with the recall in the previous query: "unoccluded dog", see Table 26).

4.5 Conclusion

Shape matching is an important yet open problem. Skeleton graphs are used to represent shapes. A matching algorithm for skeletal graphs is proposed that the topology of skeletal graphs is captured and compared at the node level. Such graph representation allows preservation of the skeletal graph's coherence without sacrificing the flexibility of matching similar portions of graphs across different levels. The present technique achieves a high recognition rate, and at the same time, significantly reduces space and time complexity of matching. The results show that the skeleton-matching approach is an effective and efficient technique for shape recognition.

CHAPTER FIVE: THE GRAPHICAL USER INTERFACE

This chapter focuses on the graphical user interface (GUI) for the software that implements the proposed algorithms. Sections 5.1, 5.2 and 5.3 describe the respective GUIs for the ASIA, range-distance, and skeleton-matching techniques. Section 5.4 then illustrates the advantages and disadvantages for each technique.

5.1 The ASIA Search Engine

The ASIA algorithm is illustrated in Figure 5 of Chapter 2. To construct a query with ASIA, a user uses the GUI to select a training image from the image database. The user then identifies and marks an area of interest in the image which becomes the noise-free training image. This noise-free training image is the input to ASIA algorithm as described in Figure 5 of Chapter 2. The GUI implemented for ASIA is described as follows:

1) The main window is shown in Figure 28. Users first select a directory of database images. In this example, a user has chosen the database images located in the directory "e:\db." Having selected an appropriate directory, there are three ways to browse for images: by image category, image ID or randomly.

The image ID and random browsing features are self-explanatory. The image category allows users to browse images in the category of their interest. In the Figure 28 example, "dsc," or the dish category, has been selected. The upper area of the main window shows images in the specified category, "dsc." The number of pages required to show the images in the category is displayed in the lower portion of the

window. The example indicates “p.4/5” meaning there are a total of five pages of images in this category and the current page is number 4. Users can navigate between the pages of images by clicking the “<<<” and “>>>” buttons in the lower portion of the window. Users choose a query image by clicking on the desired image and then clicking the “Select image” button. The image file name will be shown in the text box.

Figure 28: The main window for ASIA

2) After highlighting the image name and clicking on the “View image” button shown in Figure 28, The “area of interest” window pops up, as shown in Figure 29.

Figure 29: “The area of interest” window

3) As shown in Figure 30, users can then mark the boundary points in the image by using the mouse to click around the boundary. One of three contour options may be selected. The “None” button, illustrated below; means there are no connected lines between any two consecutive points.

Figure 30: Contour points in “The area of interest” window

4) Selecting the “Outlined” contour button, causes connected lines to be drawn between consecutive points, from the beginning point to the ending point, as shown in Figure 31.

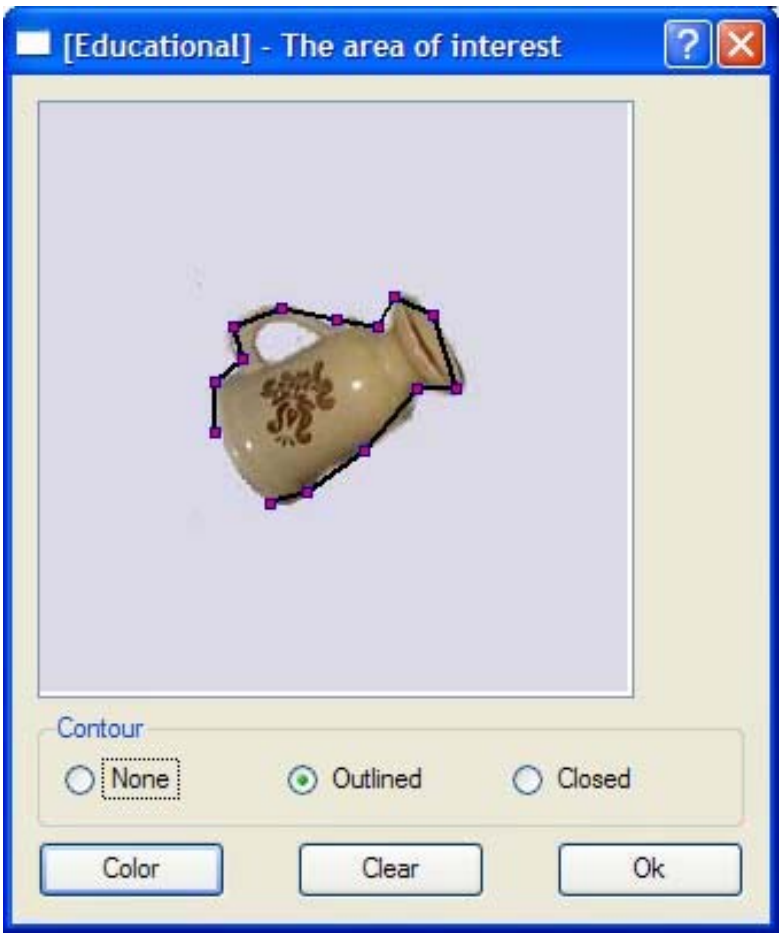


Figure 31: The contour points and their connected lines

5) In Figure 31, the contour is not closed. Selecting the “Closed” contour button closes the contour along the contour points by connecting the beginning and ending points, as shown in Figure 32.

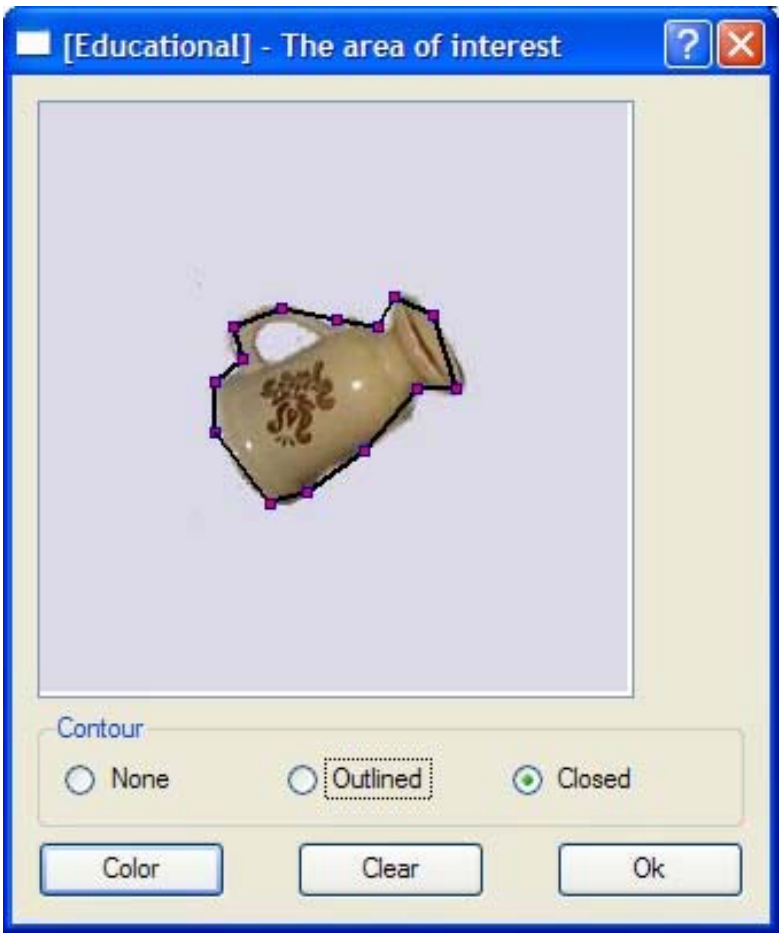


Figure 32: The closed contour along the contour points

6) The default color of the contour lines is black. Because some images may be to dark for a black contour line to be easily visible, the GUI allows a user to select the color of the contour line. Figure 33 illustrates how a contour can be changed into a selected color. In this example, the color green is selected in place of black.

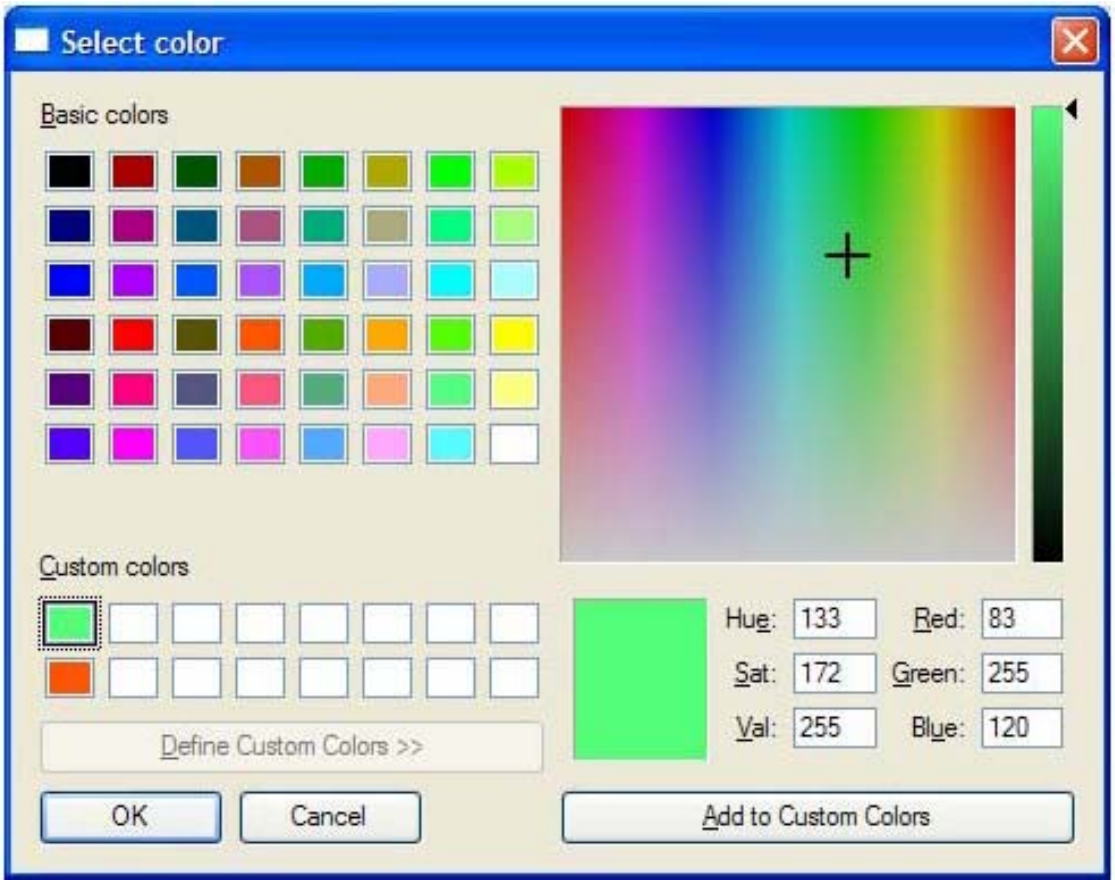


Figure 33: The window for color choices

7) Figure 34 shows the image with the new green contour that was selected in step 6. The “clear” button may be clicked a any time to remove all of the contour points and curves.

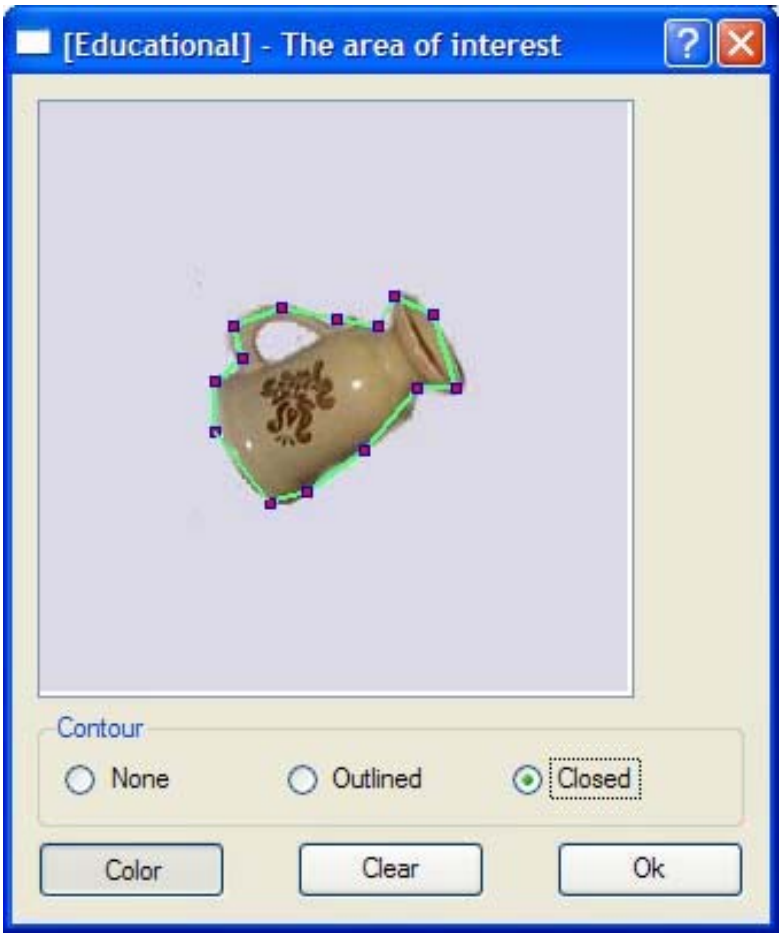


Figure 34: A green contour selected from Figure 33

8) The area of interest is inside the contour. Users can adjust the curve of the contour by clicking one of the contour points and dragging it and ending with a right mouse click to lock in the new position of the curve of the contour. Users can click the "Ok" button at any time in Figures 30, 31, 32 or 34. Users have to click the "Show result" button in the main window as shown in Figure 28. The ASIA search engine calculates the query's features and matches with the database's features as described in Figure 5 in Chapter 2. The results are shown in Figure 35.

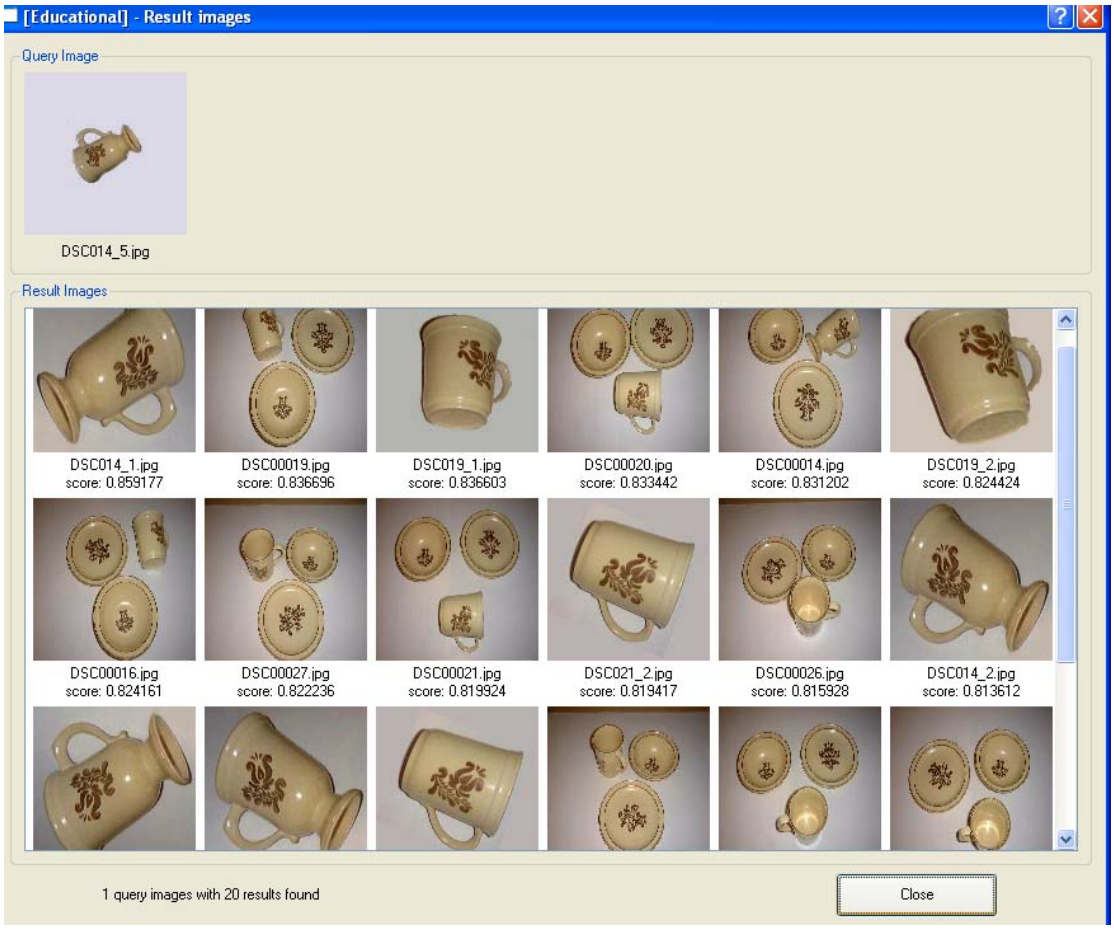


Figure 35: The result set

5.2 The Range-Distance Search Engine

The range-distance algorithm is explained in Section 3.6 of Chapter 3. Figure 37 illustrates the algorithm in Section 3.6. In the range-distance method, the GUI is used to collect information from the user in order to group images into the three possible groups: positive, negative or neutral. After these groups of images are identified, four steps will be processed, as explained in Section 3.6 of Chapter 3. The GUI for the range-distance method is shown in Figure 36.

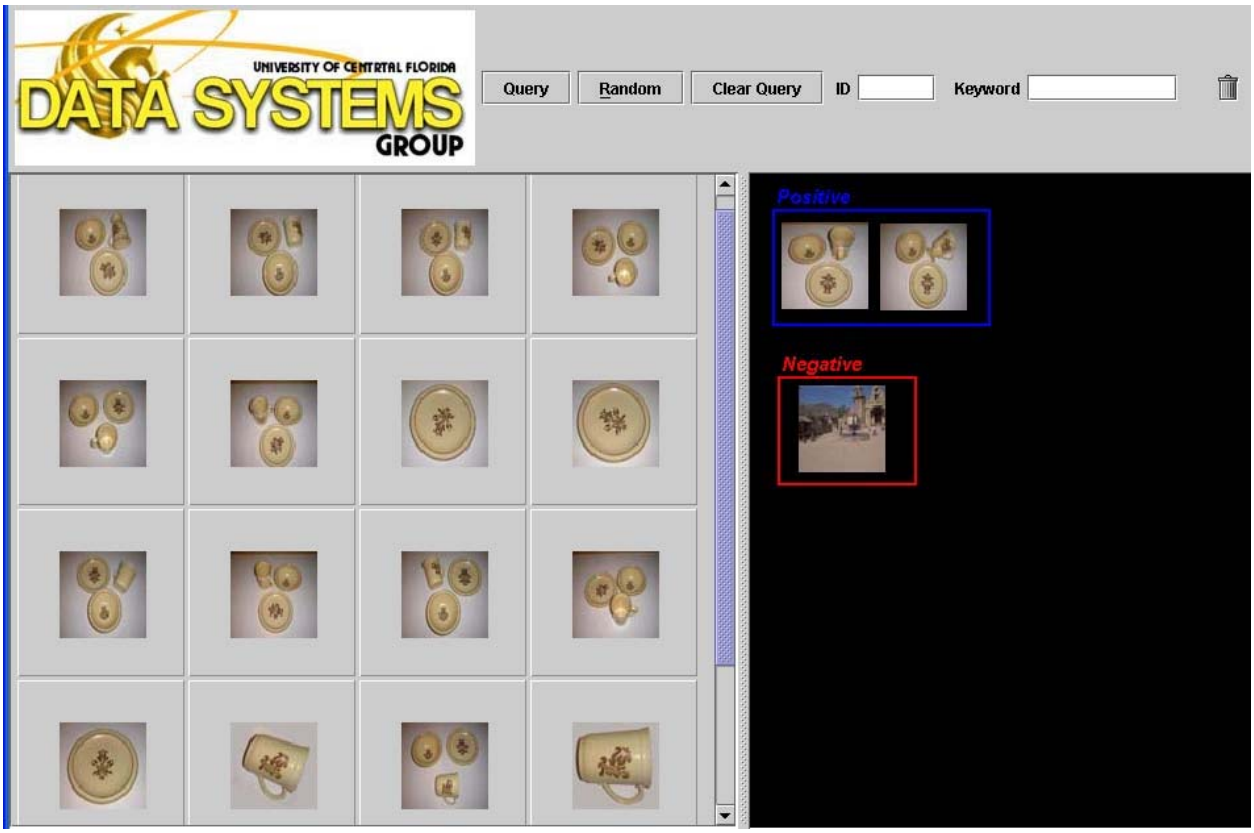


Figure 36: GUI for the range-distance method

In Figure 36, images are initially shown in a random ordering. Users can click the "Random" button to access other random images. Alternatively, users can also browse images by image ID or by image category. The images are shown at the left panel and the user can drag an image from the left panel and drop it on the right panel. A box can then be drawn around the images and then a right mouse click to identify the group type (positive, negative or neutral). Here, for example, there are two groups: the positive group and the negative group. There is one image for each group type. Upon clicking the "Query" button, the range-distance algorithm is processed (as shown in Figure 37) and the image results are shown on the left panel of the window shown in Figure 36.

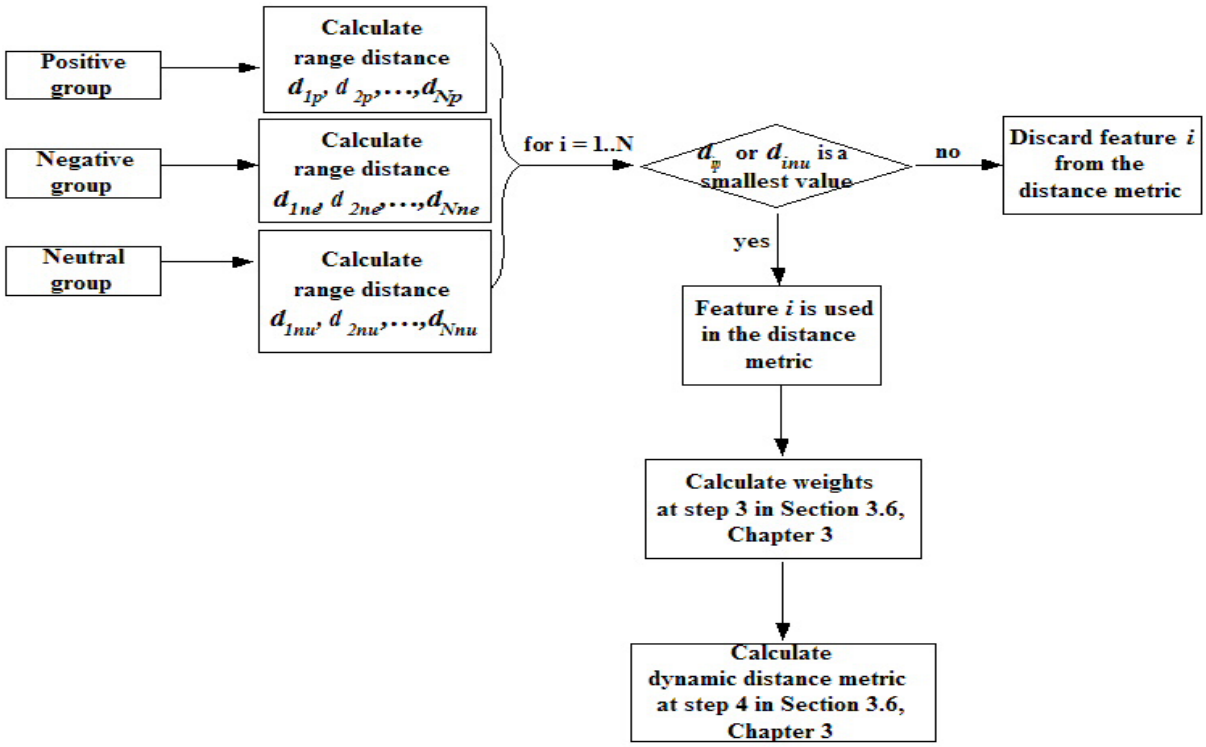


Figure 37: Skeleton-matching algorithm

5.3 The Skeleton-Matching Search Engine

The skeleton-matching algorithm is explained in Sections 4.3.1 and 4.3.2 of Chapter 4 and as illustrated in Figure 38. In the skeleton-matching method, the GUI is used to collect the users' information about the query image. After the query image is identified, all of the steps on the right-hand side of Figure 38 will be processed. All steps on the left-hand side, in the dashed box, in Figure 38, are processed off-line. The GUI for the skeleton-matching method is shown in Figures 39-42. Note that although the GUIs for the ASIA and skeleton-matching algorithms are similar, the algorithms they execute are different.

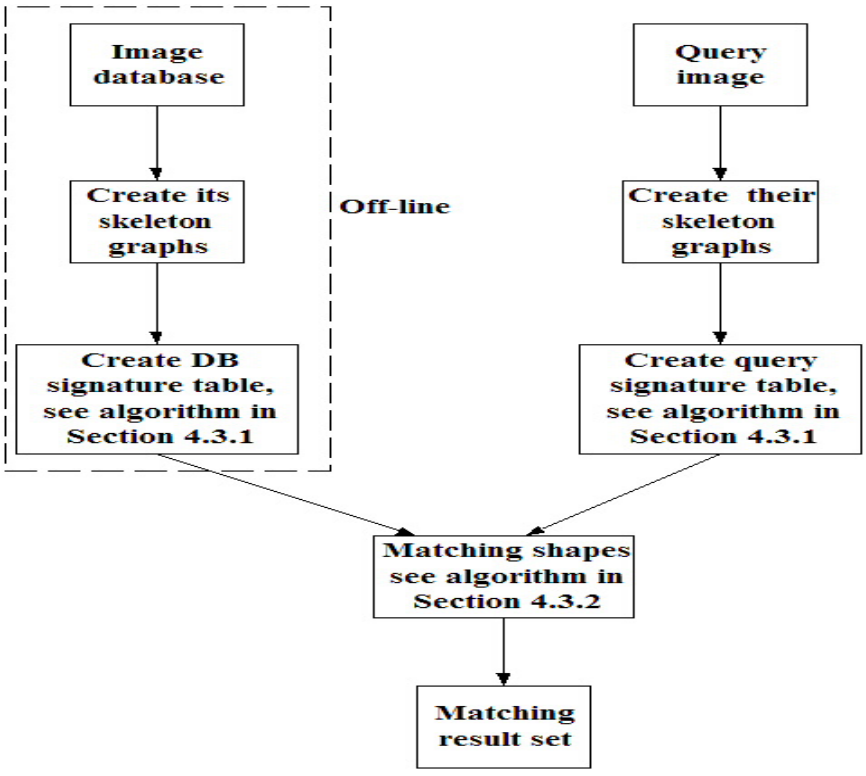


Figure 38: The skeleton-matching algorithm

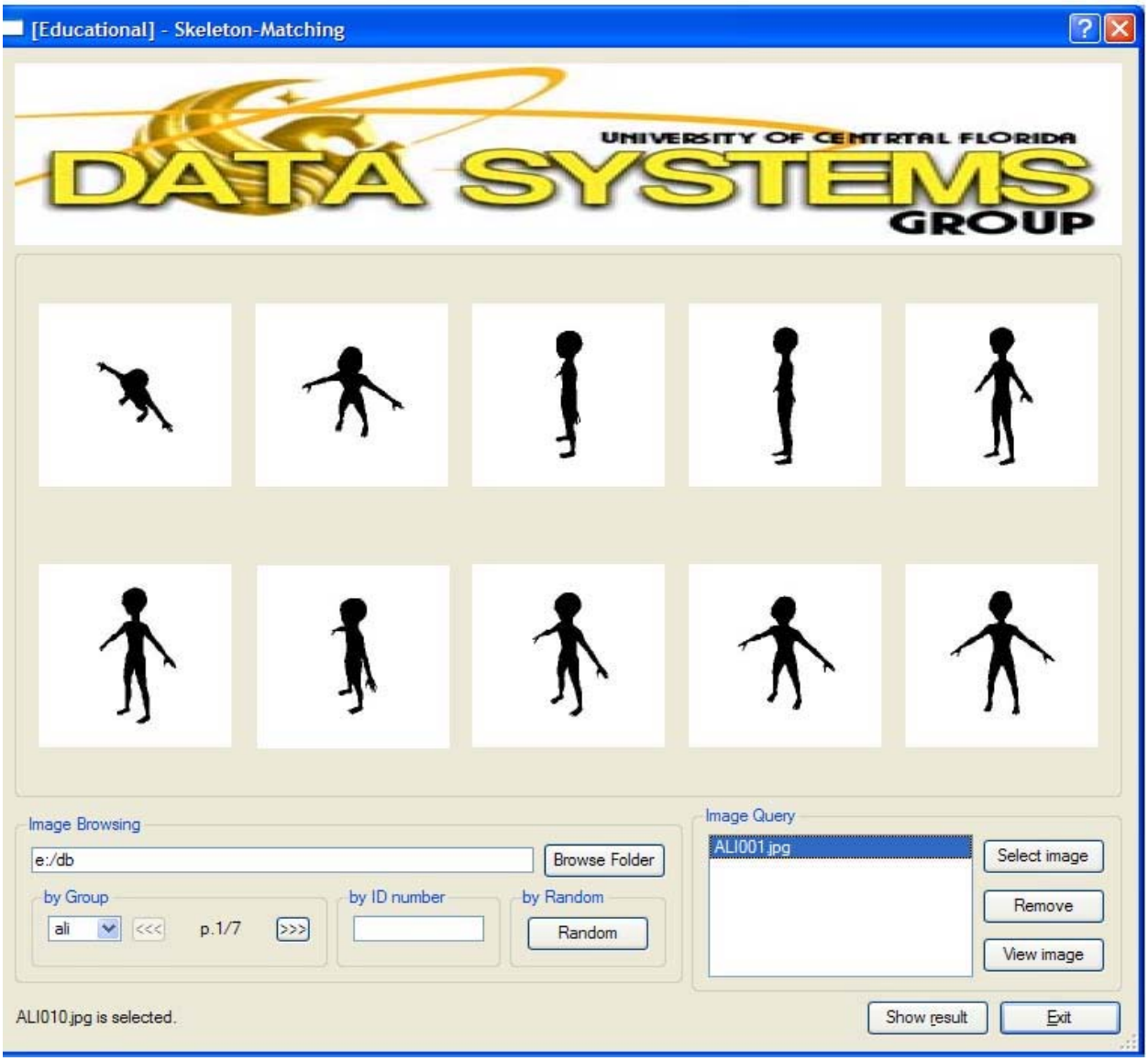


Figure 39: The main window for the skeleton-matching

In Figure 39, users browse images by the category type “ali” or “alien.” In this example, the image on the top row at the extreme left is selected as a query image, as shown in Figure 40. After users click the “Ok” button in Figure 40 and clicks the “Show result” button in Figure 39,

the result set is shown in Figure 41. The new window pops up to show the compared skeletons between the first and the last matches as shown in Figure 42.

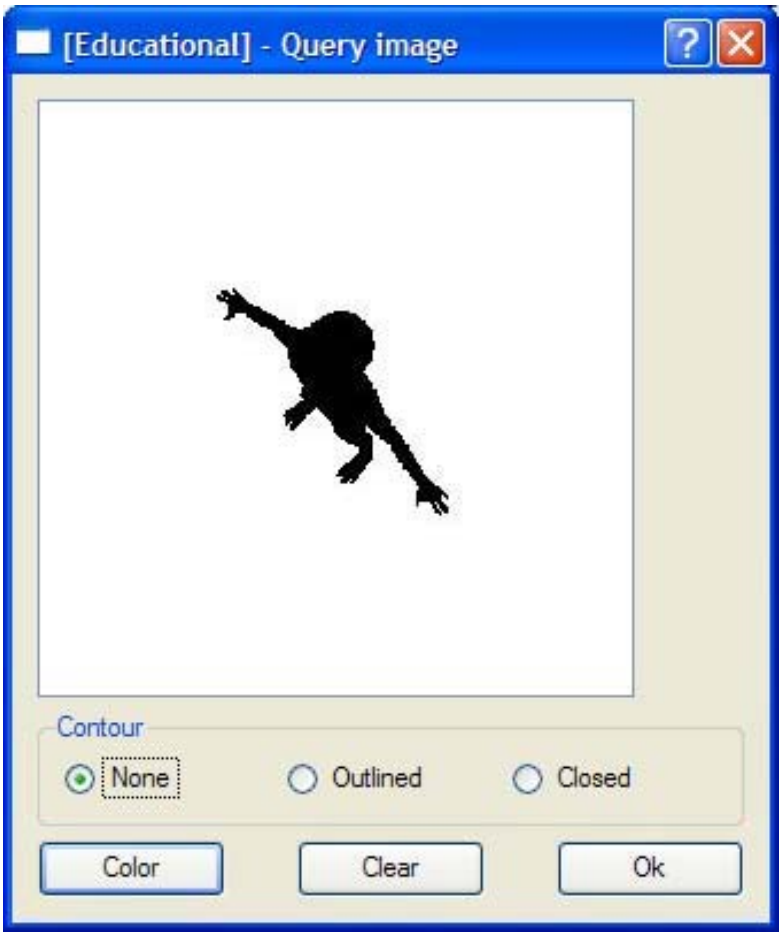


Figure 40: The query image in skeleton-matching approach

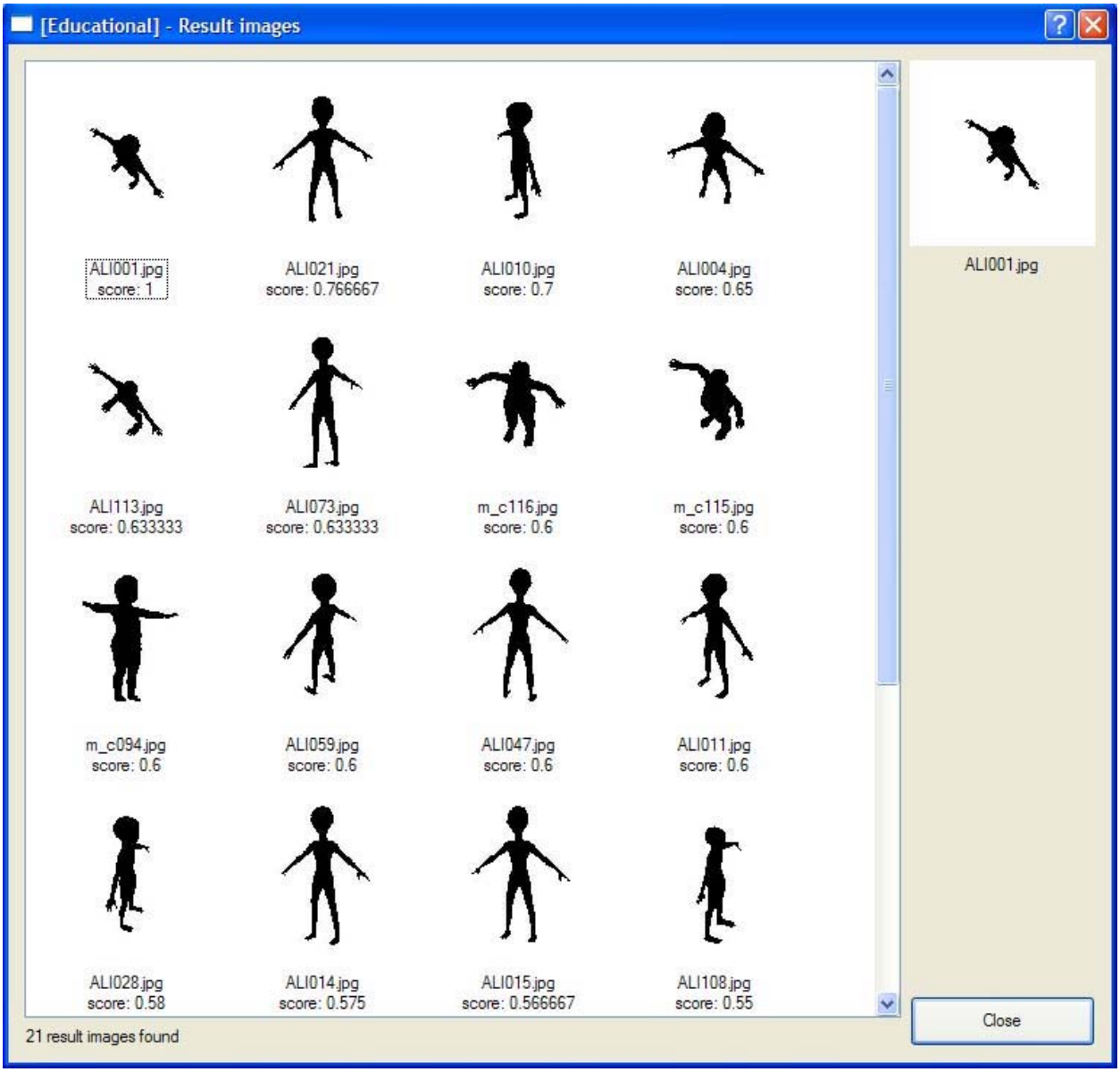


Figure 41: The result set after clicking the "Show result" button from Figure 39

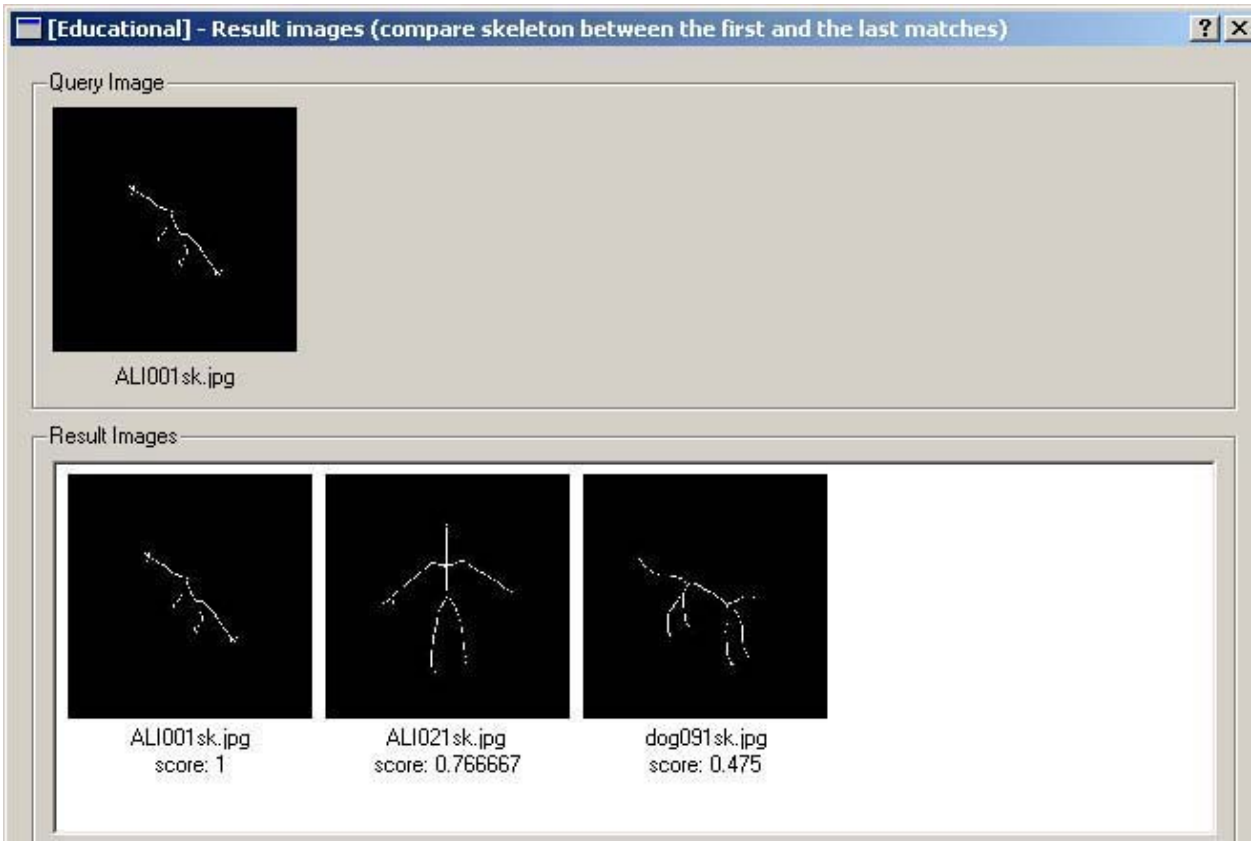


Figure 42: The compared skeletons between the first and the last matches

5.4 Discussions

The advantages and disadvantages for each proposed technique are discussed in this section.

5.4.1 ASIA

- Advantages of the ASIA method:
  - ASIA returns matching result sets with higher accuracy as compared to the Monotonic Tree approach.
  - ASIA is a matching technique that is robust to translation, scaling, and rotation invariants at matching object level.
- Disadvantages of the ASIA method:
  - For handling translation and scaling invariants, ASIA slides six different sizes of windows along the horizontal and vertical accesses of each image. Each window is called a subimage. There are 286 subimages for each image. ASIA retains features in 128 sampling circles for each subimage. ASIA retains more feature elements, resulting in higher accuracy; however, ASIA requires excessive storage space for all of the features, per image, that are stored.

5.4.2 Range-Distance Method

- Advantages of the range-distance method:
  - The range-distance method returns matching result sets with higher accuracy as compared to the *ImageGrouper* approach.
  - The range-distance method requires very little storage space for each image's features as compared with the other methods. It needs space for only 37 feature elements per image.
  - The range-distance method automatically adjusts weights in the distance metric, resulting in a variety of weights. Some of the weights may be zero in value. Therefore, the distance metric becomes a dynamic distance metric.
  - With a single query image or with multiple groups of query images, the range-distance method learns the user preferences more quickly than most traditional QBE approaches.
  - The range-distance method has a lower matching time complexity than the *ImageGrouper* approach.
- Disadvantages of the range-distance method:
  - The range-distance method handles translation, scaling, and rotation invariants among elements in the whole image; however, the range–distance method does not handle translation, scaling, and rotation invariants at the matching object level.

- The range-distance method does not provide for a noise-free training image.

These two disadvantages for the range-distance method can be resolved by using a design similar to that used in the ASIA method.

5.4.3 Skeleton-Matching Method

- Advantages of the skeleton-matching method:
  - The skeleton-matching method returns matching result sets with higher accuracy as compared to the DAG approach.
  - The skeleton-matching method is a shape matching technique that is robust to translation, scaling, and rotation invariants at the matching object level.
  - The skeleton-matching method has a lower matching time complexity than the DAG approach.
  - The skeleton-matching method requires less storage space for each image's features than does the DAG approach.
  - The skeleton graph can be used as feature vectors. Because the skeleton-matching method supports matching graphs with high accuracy, the skeleton graph is a good feature vector.
- Disadvantages of the skeleton-matching method:
  - The skeleton-matching method is tested only for binary images. The skeleton-matching method requires a good object segmentation approach.

CHAPTER SIX: CONCLUSION

6.1 Remarks

ASIA is a technique that uses query-by-example to annotate image databases. These query examples are used to create annotated databases that support query-by-concept. Since ASIA does not need to compare low-level features, the processing done during the query-by-concept phase will be fast. ASIA eliminates noise and irrelevant image areas by allowing users to identify the area of interest. It is designed as a new sampling-based framework which is robust to translation, scaling, and rotational invariants at the object level. Without the noisy areas and with this new framework, ASIA produces higher accuracy rates than a recently proposed alternative, the Monotonic Tree method.

The first phase of ASIA is QBE. The more effective QBE results in a more accurate annotated image database. Therefore, modifying the QBE to become more effective is essential. Different query groups may focus on different features. For example, query-by-example with car images specifies a car of any color. The color features should not factor in the similarity distance metric. At the same time, weights in the distance metric should be varied, rather than static. The range-distance technique is proposed to solve all of these defects. The range-distance technique allows users to identify positive, negative and neutral groups of query images. It then uses this information to automatically adjust the weights in the distance metric. When compared with the *ImageGrouper* approach, range-distance achieves better performance in terms of both precision and recall with less matching time complexity.

The matching time complexity and storage space for image features are important factors for an efficient QBE. The skeleton-matching technique is proposed to reduce matching time complexity and require less storage space for image features. A skeleton graph is used to represent shape. Skeleton-matching is a method for matching skeleton graphs that are robust to translation, scaling, and rotational invariants at object level. It results in higher precision and recall rates, while requiring less matching time complexity and less storage space for image features when compared with the DAG approach.

6.2 Future Work

Three efficient techniques for QBE have been proposed. They are the fundamental work to support an efficient annotation database system. This work can be extended as follows.

6.2.1 Combine the Techniques for the Image Annotation System

There are advantages and disadvantages for each technique as mentioned in Section 5.4 of Chapter 5. There are means to combine ASIA with the range-distance method because the disadvantage of the ASIA approach is the advantage of the range-distance method, and vice versa.

Efficient skeleton-matching requires an efficient object segmentation technique. Efficient object segmentation is also an open research topic. In other words, skeleton graphs for each image are represented as features. Skeleton-matching supports matching these features, thus it may be possible to combine these three techniques. Combining the effectiveness of these three

techniques, while eliminating some of the disadvantages, is the next step in creating an efficient system for the automatic annotation of database images for query-by-concept.

6.2.2 Leveraging Relevant Feedback for Image Annotation

From the query set, the results may include some irrelevant images. These results reduce the precision rate. Interactive image retrieval is a technique that learns the users' preferences from the relevant feedback. This technique increases the precision and recall rates by eliminating irrelevant images and making it possible to achieve a higher percentage of relevant images. The range-distance technique supports relevant feedback because it is a technique that automatically adjusts the weights in the dynamic distance metric. Therefore, the relevant feedback with the combined system in Section 6.2.1 is an efficient system for automatic annotation database images for query-by-concept.

APPENDIX

PROOF OF THEOREM 1

From Chapter 4, *ratio* = (DB *node_length*)/(query *node_length*) is defined and their median is computed, where *CN* is the number of matched nodes, *CP* is the number of matched nodes whose ratios in the range [*median ratio – threshold, median ratio + threshold*], $|T_i|$ is the size of query skeleton tree T_i, and w is the weight in range [0,1].

Definition 1: A distance measure is defined that it is equivalent to the scoring function:

$$d(T_1, T_2) = 1\text{- score} = 1 - w \frac{CN(T_1, T_2)}{\max(|T_1|, |T_2|)} - (1-w) \frac{CP(T_1, T_2) + 1}{\max(|T_1|, |T_2|)}$$

Theorem 1: For any trees T_1, T_2 and T_3, the following properties hold true:

(a) $0 \leq d(T_1, T_2) \leq 1$

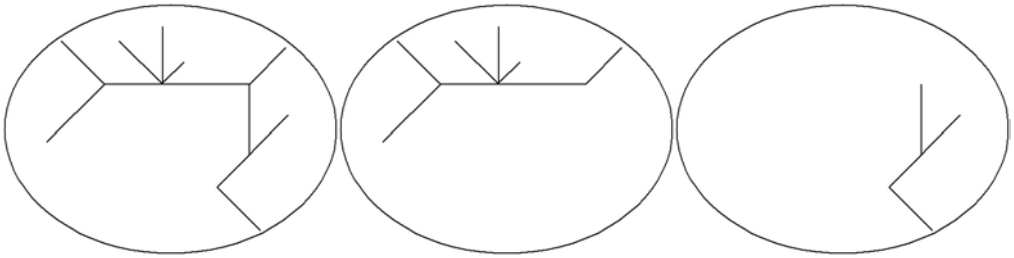
(b) $d(T_1, T_2) = 0 \Leftrightarrow T_1$ and T_2 are isomorphic to each other

(c) $d(T_1, T_2) = d(T_2, T_1)$

(d) $d(T_1, T_3) \leq d(T_1, T_2) + d(T_2, T_3)$

Proof Properties (a)-(c) following directly from Definition 1. The triangle in equality of property (d) will be proofed. There are two cases for proofing the property (d).

Case I: disjoint between $CN(T_1, T_2)$ and $CN(T_1, T_3)$



T_1 T_2 T_3

This disjoint case $\Rightarrow CN(T_1, T_2) + CN(T_2, T_3) \leq |T_2|$

And $(CP(T_1, T_2)+1) + (CP(T_2, T_3)+1) \leq |T_2|$

$$(w\, CN(T_1, T_2) + (1-w)\, (CP(T_1, T_2)+1) + w\, CN(T_2, T_3) + (1-w)\, (CP(T_2, T_3)+1)) \leq |T_2| \qquad (1)$$

From the property (d),

$$1 - w \frac{CN(T_1, T_2)}{\max(|T_1|, |T_2|)} - (1-w) \frac{CP(T_1, T_2)+1}{\max(|T_1|, |T_2|)}$$

$$+ 1 - w \frac{CN(T_2, T_3)}{\max(|T_2|, |T_3|)} - (1-w) \frac{CP(T_2, T_3)+1}{\max(|T_2|, |T_3|)}$$

$$\geq 1 - w \frac{CN(T_1, T_3)}{\max(|T_1|, |T_3|)} - (1-w) \frac{CP(T_1, T_3)+1}{\max(|T_1|, |T_3|)} \qquad (2)$$

If LHS ≥ 1 is proofed, then (2) is true.

LHS $\geq 1 \Leftrightarrow \max(|T_1|,|T_2|) \max(|T_2|,|T_3|)$

$$\geq (w\ CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))\ max(|T_2|,|T_3|)$$

$$+ (w\ CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\ max(|T_1|,|T_2|) \quad (3)$$

Case 1: $|T_1| \geq |T_2| \geq |T_3|$

Then (3) is $|T_1|\,|T_2| \geq (w\ CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))\ |T_2|$

$$+ (w\ CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\ |T_1| \quad (4)$$

From (1), $|T_1|\,|T_2| \geq (w\ CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))\ |T_1|$

$$+ (w\ CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\ |T_1|$$

Since $|T_1| \geq |T_2|$, thus, (4) is true.

Case 2: $|T_1| \geq |T_3| \geq |T_2|$

Then (3) is $|T_1|\,|T_3| \geq (w\ CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))\ |T_3|$

$$+ (w\ CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\ |T_1| \quad (5)$$

From $|T_3| \geq |T_2|$, $|T_1|\,|T_3| \geq |T_1|\,|T_2|$

From (1), $|T_1|\,|T_3| \geq |T_1|(w\ CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))$

$$+ |T_1|(w\ CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))$$

Since $|T_1| \geq |T_3|$, therefore, (5) is true.

Other cases: Performed in similar manner to previous cases

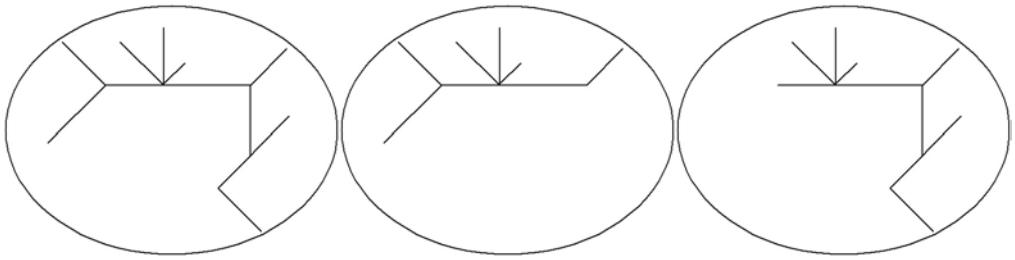
$$|T_2| \geq |T_1| \geq |T_3|$$

$$|T_2| \geq |T_3| \geq |T_1|$$

$$|T_3| \geq |T_1| \geq |T_2|$$

$$|T_3| \geq |T_2| \geq |T_1|$$

Case II: There is some common between $CN(T_1, T_2)$ and $CN(T_1, T_3)$



T_1 T_2 T_3

Let $N = CN(CN(T_1, T_2), CN(T_2, T_3)) > 0$

Each of these $CN(T_1, T_2), CN(T_1, T_3), CN(T_2, T_3) \geq N$

Each of these $(CP(T_1, T_2)+1), (CP(T_1, T_3)+1), (CP(T_2, T_3)+1) \geq N$

$w\,(CN(T_1, T_2) + CN(T_2, T_3) - N) \leq w\,|T_2|$

$(1\text{-}w)((CP(T_1, T_2)+1) + (CP(T_2, T_3)+1) - N) \leq (1\text{-}w)\,|T_2|$

$w\,CN(T_1, T_2) + w\,CN(T_2, T_3) + (1\text{-}w)\,(CP(T_1, T_2)+1) + (1\text{-}w)(CP(T_2, T_3)+1) - N \leq |T_2|$ (6)

From (2), If LHS $\geq 1 - \dfrac{N}{\max(T_1, T_3)}$ is proofed, then (2) is true.

$$\text{LHS} \geq 1 - \frac{N}{\max(T_1, T_3)} \Leftrightarrow 1 - w\,\frac{CN(T_1, T_2)}{\max(|T_1|, |T_2|)} - (1\text{-}w)\,\frac{CP(T_1, T_2)+1}{\max(|T_1|, |T_2|)}$$

$$+ 1 - w\,\frac{CN(T_2, T_3)}{\max(|T_2|, |T_3|)} - (1\text{-}w)\,\frac{CP(T_2, T_3)+1}{\max(|T_2|, |T_3|)}$$

$$\geq 1 - \frac{N}{\max(T_1, T_3)}$$

$$\Leftrightarrow \max(|T_1|,|T_3|)\max(|T_1|,|T_2|)\max(|T_2|,|T_3|)$$

$$\geq (w\,CN(T_1, T_2)+(1-w)(CP(T_1, T_2)+1))\max(|T_2|,|T_3|)\max(|T_1|,|T_3|)$$

$$+ (w\,CN(T_2, T_3)+(1-w)(CP(T_2, T_3)+1))\max(|T_1|,|T_3|)\max(|T_1|,|T_2|)$$

$$- N\max(|T_1|,|T_2|)\max(|T_2|,|T_3|) \quad (7)$$

Case 1: $|T_1| \geq |T_2| \geq |T_3|$

Then (7) is $|T_1|\,|T_1|\,|T_2| \geq (w\,CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))|T_2||T_1|$

$$+ (w\,CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\,|T_1||T_1| - N\,|T_1||T_2|$$

$$|T_1|\,|T_2| \geq (w\,CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))|T_2|$$

$$+ (w\,CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\,|T_1| - N\,|T_2| \quad (8)$$

From (6) $|T_1|\,|T_2| \geq (w\,CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))|T_1|$

$$+ (w\,CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\,|T_1| - N\,|T_1|$$

Since $(w\,CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1)) - N > 0)$ *and* $|T_1| \geq |T_2|$. Thus, (8) is true.

Case 2: $|T_1| \geq |T_3| \geq |T_2|$

Then (7) is $|T_1|\,|T_1|\,|T_3| \geq (w\,CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))|T_3||T_1|$

$$+ (w\,CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\,|T_1||T_1| - N\,|T_1||T_3|$$

$$|T_1|\,|T_3| \geq (w\,CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))|T_3|$$

$$+ (w\,CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\,|T_1| - N\,|T_3| \quad (9)$$

From (6) $|T_1|\,|T_3| \geq |T_1|\,|T_2| \geq (w\,CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1))|T_1|$

$+ (w\,CN(T_2, T_3) + (1-w)(CP(T_2, T_3)+1))\,|T_1| - N\,|T_1|$

Since $(w\,CN(T_1, T_2) + (1-w)(CP(T_1, T_2)+1) - N > 0)$ and $|T_1| \geq |T_3|$, therefore (9) is true.

Other cases: Performed in similar manner to previous cases

$|T_2| \geq |T_1| \geq |T_3|$

$|T_2| \geq |T_3| \geq |T_1|$

$|T_3| \geq |T_1| \geq |T_2|$

$|T_3| \geq |T_2| \geq |T_1|$

REFERENCES

[1] Khanh Vu , Kien A. Hua , JungHwan Oh. "Indexing for efficient processing of noise-free queries". *In Proceedings of the ninth ACM international conference on Multimedia* pages 509-511, October 2001.

[2] Khanh Vu , Kien A. Hua , Duc A. Tran. "An Efficient Core-Area Detection Algorithm for Fast Noise-Free Image Query Processing". *In Proceedings of the 2001 ACM symposium on Applied computing,* pages 258-263, March 2001.

[3] A. Natsev, R. Rastogi, and K. Shim. "Walrus: A similarity retrieval algorithm for image databases". *In Proc. of the* 1999 *ACM SIGMOD int. conf. On Management of Data*, pages 395 – 406, 1999.

[4] A. Natsev, A. Chadha, B. Soetarman and J. Vitter. "CAMEL: Concept Annotated iMagE Libraries". *Proc. SPIE Vol. 4315, p. 62 – 73, Storage and Retrieval for Media Databases*, 2001.

[5] P.K. Andleigh and K. Thakar. *Multimedia Systems Design*. Prentice hall, New York, 1996.

[6] N. Beckman, H. P. Kriegal, R. Scheider, and B. Seeger. "The r*-tree: an efficient and robust access method for points and rectangles". In *ACM SIGMOD*, pages 322 – 331, May 1990.

[7] N. Katayama and S. Satoh. "The r-tree: An index structure for high-dimensional nearest neighbor queries". In *Proc. Of the 1997 ACM SIGMOD Int. Conf. On Management of Data*, pages 369 – 380, May 1997.

[8] T. Zhang, R. Ramakrishnan, and M. Livny. "Birch: An efficient data clustering method for very large databases". *In Proc. Of the 1996 ACM SIGMOD Int. Conf. On Management of Data*, Pages 103 – 114, June 1996.

[9] Foley, Van Dam, Fiener, and Hughes. *Computer Graphics: Principles and Practice 2nd Ed*. Addison Wesley, 1997.

[10] Y. Song, W. Wang, A. Zhang. "Automatic Annotation and retrieval of Images". In *Proc. Of the IFIP Conference on Visual Database Systems* (VDB – 6), special session on Multimedia Information Management and Retrieval, May 2002.

[11] L. Zhu, A. Zhang, A. Rao and R. Srihari. "Keyblock: An Approach for Content-based Image Retrieval". *In Proc. Of ACM Multimedia*, 2000.

[12] W. Niblack, R. Barber, W. Equitz, M. Flicker, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos. "The qbic project: Query images by content using color, texture and shape". In *SPIE V1908*, 1993.

[13] M.E.J. Wood, N.W. Campbell, and B.T. Thomas. "Iterative refinement by relevance feedback in content-based digital image retrieval". In *The Sixth ACM International Multimedia Conference*, pages 13-20, September 1998.

[14] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, and R. Barber. "Efficient and effective querying by image content". *Journal of Intelligent Information Systems*, 3(3): 231-262, 1994.

[15] Y. Gong, H. Chua, and X. Guo. "Image indexing and retrieval based on color histogram". *In Proc. Of the 2nd Int. Conf. On Multimedia Modeling*, pages 115-126, November 1995.

[16] M. Swain. "Interactive indexing into image database". In *SPIE V1908*, 1993.

[17] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih. "Image indexing using color correlograms". In *Proc. Computer Vision and Pattern Recongnition*, pages 762-768, 1997.

[18] D. Chetverikov. "Detecting regular structures for invariant retrieval". In *Third International Conference on Visual Information Systems*, pages 459-466, 1999.

[19] M. Hiyahara and Y. Yoshida. Mathematical transform of (r, g, b) color data to munsell (h, v, c) color data. In *SPIE Visual Communications and Image Processing*, pages 650-657, 1988.

[20] Munehiro Nakazato and Thomas S. Huang, "Extending Image Retrieval with Group-Oriented Interface," In Proceedings of *IEEE ICME2002*, 2002.

[21] Thomas E. Bjoerge and Edward Y. Chang, "Why one example is not enough for an image query," In Proceedings of *IEEE ICME 2004*, 2004.

[22] J. R. Smith and S-F Chang, Transform features for texture classification and discrimination in large image databases. In *Proceedings of IEEE Intl. Conf. on Image Processing,* 1994.

[23] J. R. Smith and S-F Chang, "Quad-Tree Segmentation for Texture-based Image Query." In *Proceedings of ACM 2nd International Conference on Multimedia,* 1994.

[24] G. Strang and T. Nguyen, "Wavelet and Filter Banks," Wellesley-Cambridge Press, 1997.

[25] X. S. Zhou and T. S. Huang, "Edge-based structural feature for content-base image retrieval," *Pattern Recognition Letters, Special issue on Image and Video Indexing*, 2000.

[26] R. C. Gonzales and R. E. Woods, "Digital Image Processing," Addison-Wesley, 1992.

[27] A.W.M. Smeulders et al, Content-based image retrieval at the end of the early years. *IEEE Trans. On PAMI*, 22(12): 1349-1380, 2000.

[28] Ying Dai and Dawei Cai, "Imagery – based digital collection retrieval using eigen SGLD matrices," In Proceedings of *IEEE ICME 2004*, 2004.

[29] Joo-Hwee Lim and Jesse S. Jin, "Image retrieval using spatial icons," In Proceedings of *IEEE ICME 2004*, 2004.

[30] S. Berretti, G.D. Amico and A. Del Bimbo, "Shape representation by spatial partitioning for content based retrieval applications," In Proceedings of *IEEE ICME 2004*, 2004.

[31] B. Ko, H. S. Lee, and H. Byun, "Region-based image retrieval system using efficient feature description," In Proceedings of *IEEE Int. Conf. on Pattern Recognition*, Barcelona, Spain, 2000, pp. 283-286.

[32] F, Liu, X. Xiong, and K. L. Chan, "Natural image retrieval based on features of homogeneous color regions," In Proceedings of *IEEE on Image Analysis and Interpretation*, Austin, Texas, Apr. 2000, pp. 73-77.

[33] D. Macrini, A. Shokoufandeh, S. Dickinson, K. Siddiqi and S. Zucker. View-Based 3-D Object Recognition using Shock Graphs. *ICPR*, 2002.

[34] T. B. Sebastian, P. N. Klein and B. B. Kimia. Recognition of Shapes by Editing Shock Graphs. *ICCV*, pages 755-762, 2001.

[35] T. B. Sebastian, P. N. Klein and B. B. Kimia. Alignment based recognition of shape outlines. *IWVF*, pages 606-618, 2001.

[36] L. Younes. Computable elastic distance between shapes. *SIAM J. Appl. Math*., 1996.

[37] S. Belongie and J. Malik. Matching with shape contexts. *CBAIVL*, 2000.

[38] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *PAMI*, 18(4): 377-388, 1996.

[39] M. Pelillo, K. Siddiqi and S. Zucker. Matching hierarchical structures using association graphs. *PAMI*, 21(11): 1105-1120, 1999.

[40] K. Siddiqi, A. Shokoufandeh, S. Dickinson and S. Zucker. Shock graphs and shape matching. *IJCV*, 35(1): 13-32, 1999.

[41] S. C. Zhu and A. L. Yuille. FORMS: A flexible object recognition and modeling system. *IJCV*, 20(3), 1996.

[42] H. Tek and B. B. Kimia. Symmetry maps of free-form curve segments via wave propagation. *ICCV*, pages 362-369, 1999.

[43] H. Blum. A Transformation for Extracting New Descriptors of shape, pages 362-380, MIT Press, 1967.

[44] A. Shokoufandeh, S. Dickinson, C. Jonsson, L. Bretzner and T. Lindeberg. On the Representation and Matching of Qualitative Shape at Multiple Scales. *ECCV*, Copenhagen, May 2002.

[45] F. Harary. *Graph Theory*, Addison-Wesley, Reading, Massachusetts, page 35, 1969.

[46] M. Nakazato, L Manola, and T. S. Huang, "ImageGrouper: A Group-Oriented User Interface for Content-based Image Retrieval and Digital Image Arrangement," In *Journal of Visual Language and Computing*, 14/4 pp. 363-386, 2003.

[47] D. Gonzales, Digital Cameras are no longer just for the Digerati. *New York Times*, November 25, 2001.

[48] S. Miles, Camera buyers increasingly focus on digital. *CNET NEWS.com,* September 26, 2000.

[49] C. Carson and et. al., "Region-based Image Querying," In *Proceedings of IEEE Workshop on CBAIVL,* June, 1997.

[50] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas and P. N. Yianilos, "The Bayesian Image Retrieval System, *PicHunter*: Theory, Implementation, and Psychophysical Experiments," In *IEEE Transactions on Image Processing*, Vol. 9, No. 1, January 2000.

[51] M. Flickner and et al., "Query by image and video content: The QBIC system," *IEEE Computers*, 1995.

[52] J. Laaksonen, M. Koskela, and E. Oja, "Content-based image retrieval using self-organization maps," In *Proceedings of 3rd International Conference in Visual Information and Information Systems,* 1999.

[53] G. Lu, "Multimedia Database Management Systems," Artech House, Inc., 1999.

[54] M. T. Maybury (editor) "Intelligent Multimedia Information Retrieval," MIT Press, 1997.

[55] J. R. Smith and S. F. Chang, "VisualSEEk: a fully automated content-based image query system," In *ACM Multimedia'96*, 1996.

[56] Nualsawat Hiransakolwong, Khanh Vu, Kien A. Hua and Sheau-Dong Lang, "Shape Recognition Based on the Medial Axis Approach." In *the Proc. of Int'l Conf. on Multimedia and Expo. (ICME 2004)*, June 27-30, 2004, Taipei.

[57] Nualsawat Hiransakolwong, Khanh Vu, Kien A. Hua and Sheau-Dong Lang, "Many-to-Many skeletal-Graphs Matching Approach to Shape Recognition." In *SETIT'04 – Sciences of Electronic Technologies of Information and Telecommunication*, March 15-20, 2004, Tunis.

[58] Nualsawat Hiransakolwong, Kien A. Hua, Khanh Vu and Yao H. Ho, "ASIA: An Automatic Annotation Technique for Query-by-concept in Image Retrieval Systems." In *MMM'03 - The 9th International Conference on Multi-Media Modeling*, January 8-10, 2003, Taiwan, pages 70-89.

[59] Soontharee Koompairojn, Kien A. Hua, Nualsawat Hiransakolwong and Khanh Vu, "Range Distance in Query-by-Example(s): An Application." To appear in *the IASTED International Conference on Advances in Computer Science and Technology (ACST 2004)*, Nov 22-24, 2004, St. Thomas, US Virgin Islands.