Techniques for boosting the performance in Content-Based Image Retrieval Systems

by

NING YU
B.S. Tsinghua University, 2000
M.S. University of Central Florida, 2005

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

Fall Term
2011

Major Professor: Kien A. Hua

ABSTRACT

Content-Based Image Retrieval has been an active research area for decades. In a CBIR system, one or more images are used as query to search for similar images. The similarity is measured on the low level features, such as color, shape, edge, texture. First, each image is processed and visual features are extract. Therefore each image becomes a point in the feature space. Then, if two images are close to each other in the feature space, they are considered similar. That is, the k nearest neighbors are considered the most similar images to the query image. In this K-Nearest Neighbor (k-NN) model, semantically similar images are assumed to be clustered together in a single neighborhood in the high-dimensional feature space. Unfortunately semantically similar images with different appearances are often clustered into distinct neighborhoods, which might scatter in the feature space. Hence, confinement of the search results to a single neighborhood is the latent reason of the low recall rate of typical nearest neighbor techniques. In this dissertation, a new image retrieval technique - the Query Decomposition (QD) model is introduced. QD facilitates retrieval of semantically similar images from multiple neighborhoods in the feature space and hence bridges the semantic gap between the images' low-level feature and the high-level semantic meaning. In the QD model, a query may be decomposed into multiple subqueries based on the user's relevance feedback to cover multiple image clusters which contain semantically similar images. The retrieval results are the k most similar images from multiple discontinuous relevant clusters.

To apply the benifit from QD study, a mobile client-side relevance feedback study was conducted. With the proliferation of handheld devices, the demand of multimedia information retrieval on mobile devices has attracted more attention. A relevance feedback information retrieval process usually includes several rounds of query refinement. Each round incurs exchange of tens of images between the mobile device and the server. With limited wireless bandwidth, this process can incur substantial delay making the system unfriendly

to use. The Relevance Feedback Support (RFS) structure that was designed in QD technique was adopted for Client-side Relevance Feedback (CRF). Since relevance feedback is done on client side, system response is instantaneous significantly enhancing system usability. Furthermore, since the server is not involved in relevance feedback processing, it is able to support thousands more users simultaneously.

As the QD technique improves on the accuracy of CBIR systems, another study, which is called In-Memory relevance feedback is studied in this dissertation. In the study, we improved the efficiency of the CBIR systems. Current methods rely on searching the database, stored on disks, in each round of relevance feedback. This strategy incurs long delay making relevance feedback less friendly to the user, especially for very large databases. Thus, scalability is a limitation of existing solutions. The proposed in-memory relevance feedback technique substantially reduce the delay associated with feedback processing, and therefore improve system usability. A data-independent dimensionality-reduction technique is used to compress the metadata to build a small in-memory database to support relevance feedback operations with minimal disk accesses. The performance of this approach is compared with conventional relevance feedback techniques in terms of computation efficiency and retrieval accuracy. The results indicate that the new technique substantially reduces response time for user feedback while maintaining the quality of the retrieval.

In the previous studies, the QD technique relies on a pre-defined Relevance Support Support structure. As the result and user experience indicated that the structure might confine the search range and affect the result. In this dissertation, a novel *Multiple Direction Search* framework for semi-automatic annotation propagation is studied. In this system, the user interacts with the system to provide example images and the corresponding annotations during the annotation propagation process. In each iteration, the example images are dynamically clustered and the corresponding annotations are propagated separately to each cluster: images in the local neighborhood are annotated. Furthermore, some of those images are returned to the user for further annotation. As the user marks more images,

the annotation process goes into multiple directions in the feature space. The query movements can be treated as multiple path navigation. Each path could be further split based on the user's input. In this manner, the system provides accurate annotation assistance to the user - images with the same semantic meaning but different visual characteristics can be handled effectively. From comprehensive experiments on Corel and U. of Washington image databases, the proposed technique shows accuracy and efficiency on annotating image databases.

ACKNOWLEDGMENTS

Getting a PhD degree is one of the most significant accomplishments in my life. I am grateful to have many wonderful people to help me through this journey. I take this opportunity to thank those who help me during the process of achieving this accomplishment.

As my advisor, Prof. Kien A. Hua receives my utmost gratitude. He provided me many opportunities from which I learned much about and discovered my passion for computer science. Through working on projects on research assistantships, I gained valuable experience working in group projects. Through lab sessions in teaching assistantships, I discovered my passion for sharing my knowledge. Through research, I discovered many interesting fields of computer science and have a greater appreciation of computer science. I am fortunate to have him as my advisor. Also, I like to thank all members of my committee for their helpful comments regard to my research and for being accommodating in schedule.

I thank members of the Data System Group. I learned much about web development working on software development projects with them. Their warm smiles and humorous words brighten my days and alleviate my frustration and confusion with administrative tasks. I enjoy discussions about challenges of pursuing the degree and matters beyond academic. Not being alone makes facing challenges throughout the journey all that much joyful.

Many thanks go to my parents Xiulan Zhang and Yongfa Yu, and my husband Yuxiao Hu who supported me financially and with their never ending love and care.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CHAPTER 1: INTRODUCTION

1.1 Query Decomposition

Content-based Image retrieval (CBIR) is a core technology for many applications ranging from image search engines, electronic commerce, and medical diagnosis to computer aided drug design. Nearest neighbor search [68] is the typical technique for most of today's CBIR systems. In this framework, each image is represented by a large number of visual features (e.g., color layouts, texture, etc.) and managed as points in this multidimensional feature space. Images are considered similar if they are located "close" to each other in this high-dimensional space, according to some distance measure [81]. Query processing is performed by finding k nearest images to a given example image in this feature space. Because it is generally impossible for humans to define a query in terms of these feature vectors, example images and user's relevance feedback (RF) are usually used as an remedy. Most of the CBIR techniques focused on finding the optimal result set in a single neighborhood [4, 39–41, 65, 70, 81, 90, 96] with the aid of user's relevance feedback. A limitation of this approach is due to the weak correlation between the target objects/concepts and their appearance in images. A given object type can have various color patterns and diverse shapes. Consequently, the best-matching images are not necessarily located near each other in any single neighborhood in the feature space. A better technique should be able to discover all the scattered clusters which contain relevant images and combine the top ranked images from all these semantically relevant clusters in order to enhance the overall precision and recall. This motivates the Query Decomposition (QD) approach described and examined in this chapter. A QD search is not based on the traditional k nearest neighbors in a single neighborhood, but rather finding the k best-matching images wherever they might be in the feature space.

1.1.1 Limitation of Traditional k-NN Model

In CBIR systems, images are represented by their low-level features. Image searching is then conducted in this feature space. Since it is inconvenient and inherently difficult to express queries in terms of low-level visual features, Query by Example (QBE) is used in essentially all CBIR systems today. The popular query processing strategy in today's CBIR techniques is known as the k-NN model. It aims to retrieve the images corresponding to the k data points closest to the query point. This k-NN model confines the search result to a single neighborhood in the feature space. In reality, however, the k nearest images in this single neighbor may not include all the semantically similar images. Cars, for example, come in many different shapes and colors; even the same car may look very different from different viewpoints. Hence, these images would belong to different clusters in the feature space; and a simple retrieval of the k nearest images to one query point is not able to capture all the "car" images. The inherent weak association between the high-level semantic concepts human perceives in images and the low-level visual features used to characterize images causes poor recall in today's CBIR systems. To further illustrate the limitation of the k-NN model, consider the images of "a white sedan" in a real database. This database is organized according to a 37-dimensional feature space (see Section 4 for additional details). Principal Component Analysis (PCA) is applied to project the data set to a 3-dimensional orthogonal space, as shown in Figure 1.1. In this 3-dimensional space, it can be observed that four distinct clusters of the "white sedan" images, namely "side-view", "front-view", "back-view", and "angle-view". Among the four clusters, there are many irrelevant images (represented as the triangles). Under this circumstance, retrieval techniques based on the k-NN model would not be able to find all these semantically identical images because they are not in any single k-NN neighborhood. This problem causes poor recall. Although this situation can be improved by using a very large k value to include more "Sedan" images in

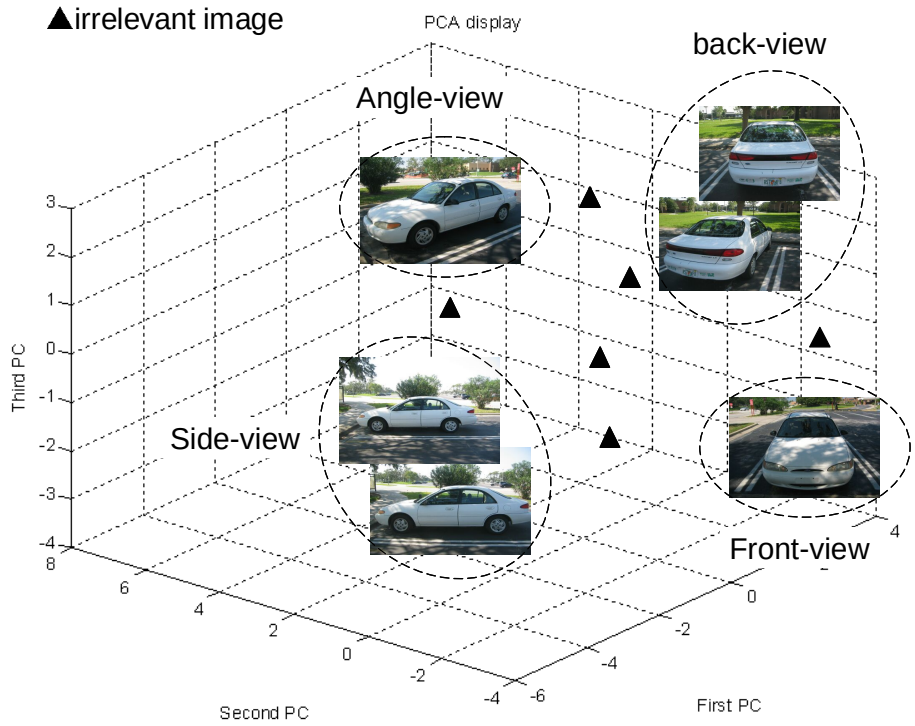


Figure 1.1: Four distinct "white sedan" clusters in a 3-dimensional feature space of an image database.

the database, it would inevitably also cover many semantically irrelevant images resulting in poor precision.

1.1.2 Multiple Neighborhoods Approach

To address the limitation of the k-NN model with the single neighborhood confinement, Query Decomposition (QD) technique is introduced. The idea is to hierarchically decompose an initial query into localized subqueries based on user's relevance feedback. These subqueries are processed independently, each to discover a distinct cluster of relevant images. Their local results are merged to form the final result. This strategy ensures that all semantically relevant data clusters are considered, even if they are located far apart in the feature space. A Relevance Feedback Support (RFS) structure was proposed to support the QD technique. The RFS is constructed by constructing a tree structure on the whole

database. The RFS structure stores the image relations of the database. The system decomposes the query depending on the RFS structure. The user provides relevance feedback with the help of the RFS structure.

The RFS structure is relatively small compared to the wholde database size so that it can be stored in a limited size memory. These properties make the QD technique and RFS structure perfect for a mobile image retrieval environment. Mobile devices with high resolution built-in camera are becoming ubiquitous. It is desirable to support multimedia information retrieval using the images taken from the camera. As examples, a student can use pictures of a plant to search for information of similar species in a remote digital library. Providing this capability calls for efficient techniques to facilitate Content-Based Image Retrieval (CBIR) in mobile environment. In a CBIR system, images are characterized by their low level features such as color, texture, and shape. Since it is difficult to "describe" a query through those features, RF is widely used and plays an important role [10, 19, 24, 27, 28, 35, 39]. It helps the system to understand the user's intention. In such a system, the user interacts with the system as follows: In each round, he helps by identifying the relevant images within the returned set; the system then utilizes this feedback to modify the query and thus to improve its retrieval results in the next round. This process can be repeated until the user is satisfied with the results.

Lots of information retrieval systems in mobile environment have been introduced in recent years. In those systems, the user's query is send back to a remote server, then the result is sent back to the user for refinement. However, this computation model is not suitable for RF because the wireless bandwidth is limited for exchanging images in each round of user RF. The long delay in feedback would affect the usability of the mobile system. In this dissertation, the QD approach was utilized to process the user's relevance feedback on the client-side. Only the final query is sent back to the remote server for result.

During the study of relevance feedback in image retrieval, it is noticed that, typically, multiple rounds of feedback are involved. In each round, the refined query must be processed

over the database stored on disks. Considering the size of the databases it can be, this is a lengthy process and may lead to high reneging user rates reducing the advantages of the RF system. Hence, scalability is a limitation of today's RF systems that manage large image collections. In this work, an in-memory RF technique is investigated. The high dimensionality of image feature vectors residing on disk are reduced to much lower one so that the compressed database fits in memory for fast RF processing. The main goal is to minimize the number of accesses to disk, executing similarity computation on the full feature set only in the final round of the retrieval. Moreover, short turn-around time for RF processing improves system usability and therefore encourages the user to try more rounds for higher-quality results.

In the previous studies, the RFS structure was used to process the RF. It is noticed that this pre-defined structure might confine the search range. So a Multiple Direction Search with dynamic clustering study was conducted. In this technique, the example images are dynamically clustered and the queries are decomposed according to the clustering result. In this dissertation, we used the technique in an image annotation scenario.

Due to the popularity of digital cameras and stronger communication infrastructure, digital images become widely available around the world. In order to provide better searching and sharing services of this massive amount of images, detailed and accurate annotations of images are required. However, manually annotating images is a tedious and expensive process. Automatic or semi-automatic annotation techniques are highly desirable [5, 11, 17, 22, 43, 49, 52, 80, 82, 85, 95].

In this dissertation, a *Multiple Directions Search* (MDS) technique is investigated. In this system, a user provides sample images with labels as query for annotation propagation: first, local clustering is performed on the example images based on their visual features. If the user's RF images have very different visual features (shows his intension to annotate groups of images with diverse visual characteristics), the current query is decomposed into multiple sub-queries accordingly to better benefit from the feedback information and provide more

precise annotation. Similar to CBIR, the neighboring images of the multiple sub-queries will be returned to the user for further RF, and the sub-queries might be decomposed again. This mechanism enables the proposed technique to explore all relevant neighborhoods in order to cover the k best matching images wherever they might be, rather than just the top k ranked images of the single best-matching cluster (while omitting other relevant clusters). The label of the example images will be propagated to the top-ranked images in the result set. This iterative process is repeated until the user stops.

CHAPTER 2: QUERY DECOMPOSITION: A MULTIPLE NEIGHBORHOOD APPROACH TO RELEVANCE FEEDBACK IN CBIR

2.1 Related Work

The weak correlation between the retrieval objects and their appearances in images is the most fundamental issue in CBIR. Although using relevance feedback to tackle this problem has been an active research area for several years [20, 28, 33, 34, 39–41, 44, 53–55, 73, 83, 84, 90, 96], it still remains a difficult and open problem. Most of the current techniques focus on finding the single cluster with the largest number of matching images, and modify the query contour to cover more relevant images. In Query Point Movement [41] (Figure 2.1(a)), during every round of feedback, the centroid of the relevant images is used as the new query point in the next iteration. The distance function is weighted such that the query contour, enclosing the result images, is shaped to optimize the precision and recall. In Multipoint Query [65] (Figure 2.1(b)), user's feedback images are grouped into clusters, each represented by the data point closest to its centroid. These representatives are treated collectively as a multipoint query. The technique expands the query contour, in accordance with the distribution of the centroids in the feature space. In Qcluster [39] (Figure 2.1(c)), a quadratic distance function is used to approximate any contour and only images near the cluster representatives are considered relevant. Multiple viewpoints approach [23] (Figure 2.1(d)) searches for relevant images using multiple query versions that are derived from the original query image by changing the color featuers: negative color image, black-white image, and black-white negative image. As a result, images differing slightly in color (e.g., a blue bus vs. a green bus) can still be found. But the images with other different features, for example, shape and texture, are still missing in the result.

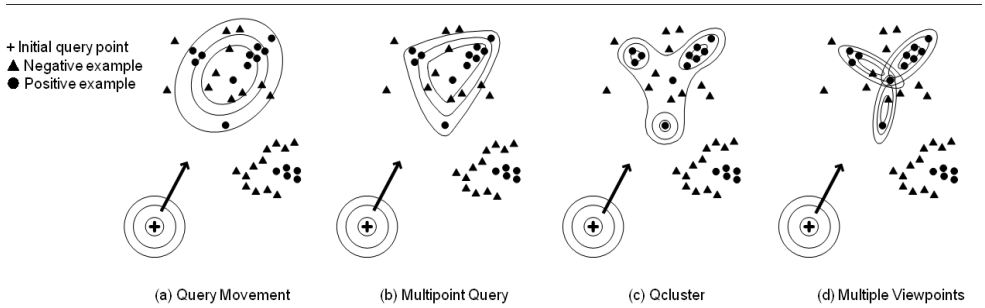


Figure 2.1: Query contours under different k-NN relevance feedback techniques.

The design of the above and some other existing CBIR techniques, e.g., [4, 40, 55, 70, 90, 96], are based on the k-NN model's assumption, i.e., objects of similar semantics look similar in many aspects. This strategy fails when the image characterization and similarity measure do not follow perceptual characteristics. Under this circumstance, images with similar semantics could scatter to distinct neighborhoods in the feature space. Relevance feedback techniques based on k-NN would navigate along a single path to select the one with the best precision and recall among these relevant neighborhoods. Due to the confinement characteristic of the query model used in the above techniques, some semantically similar images may never be explored in one relevance feedback session. This is illustrated in Figure 2 with five relevant images in the right-bottom corner not included in the query result. As the goal of these typical RF techniques is trying to find a locally optimal solution, the irrelevant images between the explored space and the bottom corner obstruct the discovering of the corner relevant images. Instead of the aforementioned techniques, machine learning methods have also been used in relevance feed-back systems [14, 20, 31, 33, 63, 64, 78, 79, 91, 94]. For exam-ple, Support Vector Machine (SVM) [63, 79, 94] can be ap-plied to find a hyperplane to seperate the relevant images from the irrelevants. This scheme has proved to improve retrieval performance. Its limitation is as follow. Although in theory there may exist a kernel to map the relevant images from different neighborhoods in the original feature space to a single neighborhood in the transformed high-dimensional space, it is difficult to determine such a kernel in practice. Furthermore, SVM defines the hyperplane based on a very small

set of feedback images, not the entire image database. Such limited training examples and high dimensionality may cause existing machine learning techniques such as SVM fail to give predictable results. In this chapter, a straight-forward and effective technique, Query Decomposition, with much less computation complexity is investigated.

2.2 Query Decomposition Approach with RFS Structure

To decompose user query to discover the semantically similar image clusters, Relevance Feedback Support (RFS) structure is investigated. The RFS structure is a hierarchical clustering structure of the database. With this RFS structure, queries can be efficiently decomposed to find the relevant subclusters according to the user's re-levance feedbacks.

2.2.1 Relevance Feedback Support Structure

The RFS structure is constructed in two stages: Data Clustering and Representative Images Selection. In the Data Clustering stage, a hierarchical clustering technique is used to organize the entire image database into a hierarchical tree structure. Then in the Representative Images Selection stage, a bottom-up representative-selection procedure is performed in the tree as follows. At the bottom level of the RFS structure, the images in each leaf node are clustered into subclusters by an unsupervised k-mean clustering algorithm. For each of these subclusters, one or more images nearest to its center are selected as the representative images. For each subsequent cluster in the upper levels of the hierarchy, the representative images of all its child clusters are again aggregated and clustered by an unsupervised k-mean clustering algorithm. The images nearest to these k-mean centers are selected as the repre-sentative images for the current node. The number of representative images for each cluster is chosen proportional to the number of images in that cluster. Finally, the list of the representative images and their corresponding cluster identifications to which they belong to are stored in the RFS structure as a part of the corresponding tree node.

2.2.2 Decomposition Technique

Given an RFS structure, the relevance feedback is per-formed starting at the root node as follows: 1. A user identifies initial relevant images (i.e. query images) from the set of images randomly selected from the representa-tive images corresponding to the root node in the RFS structure. (This step can be repeated until the user is satisfied with the initial query). 2. The images marked for the current query identify the relevant clusters in the next (lower) level of the RFS structure. A cluster in the next level is relevant if one or more of its representative im-ages match some of the query images. 3. Localized relevance feedback is processed independently in each relevant cluster. The user chooses query images from the set of images, randomly selected from the representative images from each relevant cluster. 4. Recursively repeat Steps 2 and 3 until the RF process reaches a relevant cluster at the leaf level (i.e., a semantically relevant cluster) of the RFS structure. At this time, the localized nearest neighbor computation is conducted for final image retrieval. At any time, the user also has the option to submit the final query before the search reaches the leaf level.

2.2.3 Localized Nearest Neighbor Computation

A distinct benefit of the QD approach is that the RFS scheme only needs to do k-NN computation over the much smaller data subspaces, typically represented by the relevant leaf nodes in the RFS structures, in the final round of relevance feedback. Existing RF techniques must perform the k-NN computation over the entire database for each round of RF. In QD technique, once the subqueries are submitted, they are computed in their according subclusters. In the situation when some of these local query images are located near the boundary of the given subcluster, some of their relevant images might actually reside in the sibling nodes of the RFS. In this case, the system computes the subquery over the parent node instead, in order to also consider possibly relevant images from the neighboring clusters. To determine whether a query image is near the boundary of corresponding node,

the ratio between the distance of this image to the center of the node is computed to compare with the diameter of the node. If this ratio exceeds a predetermined threshold, the search area is expanded to the parent node. The same process is repeated in the parent cluster. The threshold used in this test can be determined empirically based on the database size and content. If the images in the database are well clustered, this threshold can be set higher to reduce the probability of expanding the search to the parent cluster. For the later experiment database with 19,000 images from about 190 categories, the threshold is set to 0.4.

2.2.4 Similarity Ranking

After the localized retrieval process, the Query Decomposition technique obtains relevant image sets, one from each of the localized multipoint k-NN queries. To combine these results, the system includes several top-ranked images from each set of local results. The number of result images selected from each local set is proportional to the number of relevant images that the user had identified in the corresponding subcluster. To rank the images in each subcluster, any of the multipoint query processing techniques, discussed in Section 2.1, can be used. However, to emphasize the effect of query decomposition, Euclidian distance is used in this study as similarity score.

Various visualization techniques can be considered for displaying the final results on screen. In the prototype system, the result images are presented in groups as they are from different clusters. The system first compute the ranking score S_{g_j} $(j = 1, , N)$ of each image group as the sum of the similarity scores of its top m images, i.e.,

$$S_{g_j} = \sum S_{ji} \qquad \text{(Eq. 2.1)}$$

where S_{ji} is the similarity score of image i in group j; $m = k/N$, N is the total number of localized queries, k is the total number of images returned to the user. Then, the system

present these result groups in the order of their ranking scores S_{g_j}. Within each group, the result images are displayed in the order of their S_{ji}.

2.3 Experimental Studies

The dataset, test quires and evaluation protocol are described in Section 2.3.1 and 2.3.2. The comparison of the QD technique's performance with the MV technique is presented in 2.3.3. Then the effect of different clustering approaches on the QD systems is studied in 2.3.4. In 2.3.5, the performance sensitivity of the QD technique with RFS structure on various cluster size is investigated. The impact of the localized k-NN search threshold on performance is shown in Section 2.3.6. Finally, Section 2.3.7 discusses the performance results for computation efficiency.

2.3.1 Datasets and Test Queries

The first test database includes 15,000 images taken from the Corel image database. In order to test the performance of the QD system on an application-specific database, another Corel database which has 1,300 flower images is also used. In addition, a database with 59,900 images is used to test the system computation efficiency. The Corel images have been classified into distinct categories by domain professionals. Since users search for images based on high-level semantic concepts (as opposed to low-level image features), the Corel category information is used as the ground truth. Images from the same and related categories (such as yellow rose and red rose) are considered relevant.

To test the system's capability of addressing the semantic gap, it is compared with the Multiple Viewpoints (MV) approach using the queries listed in Table 1. This query set was designed to test the capability of bridging the semantic gap when all aspect or some aspects of the features of the desired images are different. For instance, "roses" have different colors; "cars" have different shapes and/or colors. Also, some queries are designed to investigate the

impact on performance under both general (e.g., "finding computers") and more specific (e.g., "finding laptop computers") queries. Each of these Query Concepts contains 100 images. In the following experiments, for each round of relevance feedback, the top 50 images are shown to the user.

Table 2.1: Testing Queries

Query Concept	Images in DB
1. A person	Hair-model, fitness, Kongfu
2. Airplane	single, multiple
3. Bird	eagle, owl, sparrow
4. Car	modern sedan, antique car
5. Horse	wild horse, race
6. Mountain view	snow, with water
7. Rose	yellow, red
8. Water Sports	surfing, sailing
9. Computer	server, desktop, laptop
10. Personal computer	desktop, laptop
11. Laptop	with clear background, with complicated background

2.3.2 Evaluation Metrics

To evaluate the effectiveness of the proposed system, precision and recall are used:

$$Precision = \frac{|relevantimages| \cap |retrievedimages|}{|retrievedimages|} \quad \text{(Eq. 2.2)}$$

$$Recall = \frac{|relevantimages| \cap |retrievedimages|}{|relevantimages|} \quad \text{(Eq. 2.3)}$$

$$GTIR = \frac{Numofretrievedsubconcepts}{Numoftotalsubconceptsingroundtruth} \quad \text{(Eq. 2.4)}$$

To illustrate the GTIR, the query "a person" from Table 2.1 is used as an example. In this case, the MV technique could only retrieve 1 out of 3 ground truth subcategories, thus

resulting in GTIR=1/3; while QD captured all three subcategories achieving a GTIR value of 1.

2.3.3 QD vs. MV

In this section, QD is compared with a recent technique, MV, to evaluate the capability of QD in addressing the semantic gap. For the QD approach, the R*-tree [6] method to hierarchically cluster the database. Then system selected about 80 representative images for each RFS clusters. The localized k-NN computation threshold was set to 0.4. As relevance feedback can be user subjective, 20 subjects are invited to test the systems by searching for the relevant images in the database. Each subjects searched for images 5 times on each of the test queries. The average is reported in this subsection.

The performance of QD and MV at each of the three rounds of relevance feedback is presented in Table 5.3. Note that the QD technique did not actually commit any similarity computation until Round 3 (i.e., the returned images in Round 1 and 2 are representative images randomly selected from the relevant clusters), so the the precision values are not calculated for Rounds 1 and 2. It can be observed from Table 2 that the QD technique performs significantly better than MV. MV's precision, recall and GTIR are not improved much after the second round of relevance feedback, which is due to the confinement characteristic of the traditional k-NN model as explained before.

Table 2.2: Image Retrieval Quality Comparison in 3 rounds

	MV			QD		
FeedbackRound	Precision	Recall	GTIR	Precision	Recall	GTIR
1	0.1	0.05	0.51	na	na	0.70
2	0.30	0.15	0.56	na	na	0.91
3	0.32	0.16	0.51	0.76	0.36	1

To give an overall performance comparison of QD and MV on addressing the semantic gap, the average precision and GTIR for MV and QD on the 11 test queries are presented in

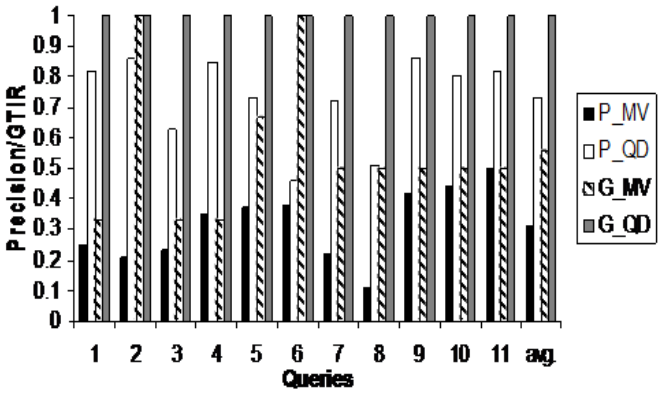


Figure 2.2: Precision and GTIR for MV and QD.

Figure 2.2, where P_MV and P_QD denote the precision for the MV and the QD techniques, respectively. Similarly, G_MV and G_QD are the GTIR for the MV and the QD techniques, respectively.

The result shows the advantage of using separate localized queries in the QD system. For most cases, the QD approach is able to capture all the semantic subclusters therefore achieving better precision and GTIR than the MV approach. This confirms that the QD technique searches all over the feature space using multiple localized queries to cover all the semantically similar images. On the contrary, the MV only focuses on searching images in the best neighborhood. On average, the QD systems precision is 0.76, while the precision of the MV system is 0.32; the QDs GTIR is 1, and the MVs GTIR is only 0.51.

From the results, for some cases, like the "Airplane", the GTIRs for QD and MV are the same. The reason is that most of the images with single airplane and the images with multiple airplanes have similar visual features, i.e., the background is clear. Although the QD's GTIR is the same as that of the MV, the precision is quite different. This can be explained as follows. The visual features of some of the images with multiple airplanes or single airplane in the sky are quite similar due to the sky background. As a result, a single neighborhood can cover all these images and capture the different subconcepts giving MV a GTIR measure comparable to that of QD. Nevertheless, there are still some airplane images of the aforementioned semantic categories, which do not have the same background and

therefore are not located in the same neighborhood. Missing these images results in a poor precision for MV. Moreover, MV approach would bring some unrelated images in the color-negative, black-white, and black-white negative channels. An extreme case occurs under the water sports query. In this case, MV returned many unrelated black-white images in the database. For the case of "mountain view", the QD approach was not significantly better than the MV approach due to the fact that the feature selected for this study is not sufficient to characterize the distant view of these images. As a result, neither technique performs well under this query, although QD could pick up both the "snow mountain" images and the "mountain with water" images by exploring the two corresponding subclusters independently giving it a small performance edge over MV in this case. The limitation of the visual features should not be viewed as a weakness of either technique as better features would improve performance using the same two retrieval methods.

Finally, the average results over all 11 test queries are presented in the last group of columns in Figure 2.2. For all the cases, the QD approach can catch all the semantic subcategories allowing it to outperform the MV approach in terms of both precision and GTIR.

To give an intuitive illustration of the advantage of the QD technique, some query results from QD and MV are presented. The top eight images retrieved for portable computer are presented in Figures 2.3 for the MV and QD techniques, respectively. Figure 2.3.a shows that the result images of MV only include one type of the "portable computer" images, i.e., "laptop with clear background". In contrast, it can be observe in Figure 2.3.b that the QD approach is able to capture all semantically relevant neighborhoods of "portable computer" scattered in the feature space.

The top 16 images for "personal computer" from MV are presented in Figure 2.4.a, and for QD in Figure 2.4.b. The personal computer includes "desktop" and "portable computer". Under the "personal computer" concept, there are two subcategories: "computer on a table" and "computer on the floor". Again, the MV approach can only find one neighborhood.

2.3.a: MV

2.3.b: QD

Figure 2.3: Retrieval result of "Portable computer from MV and QD"

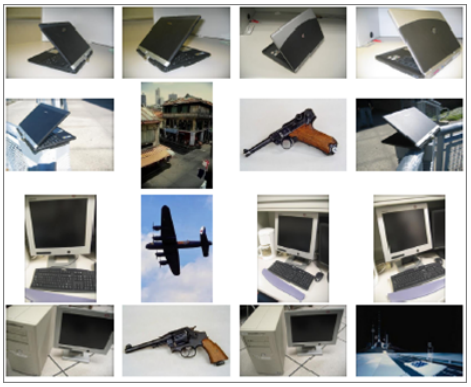
The feature space that MV covered by using multiple channels can only include one type of the computers in the ground truth. The set of desktop on the floor images and the set of desktop on a table images are too far apart in the feature space for the MV technique to capture them both.

In Figure 2.5.a, the top 24 images of "computer" returned by MV are shown. The top 24 images retrieved by QD are shown in Figure 2.5.b. In the QD approach, both the "Sun workstation" and "HP workstation" subcategories are covered, in contrast, MV only finds one subcategory. These experimental results confirm the assertion that the traditional k-NN model confines the query results to a single neighborhood in the feature space.

2.3.4 Sensitivity of Different Clustering Method to Build RFS Structure

To understand the effects of using different clustering techniques on performance, two well known and widely used methods, R*-tree and k-mean, are selected to built two different RFS structures. To use k-mean, the database is hierarchically clustered to form the tree structure.

2.4.a: MV

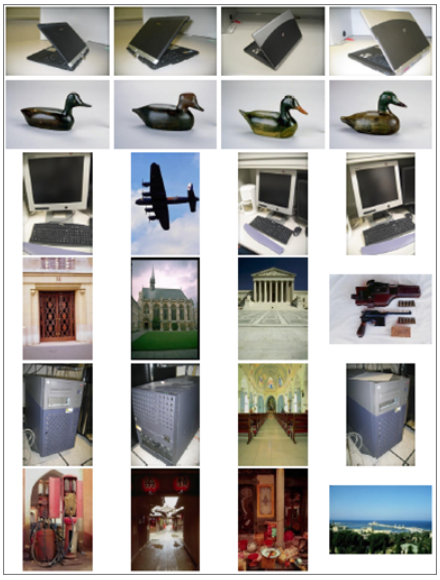


2.4.b: QD

Figure 2.4: Retrieval result of "Personal computer from MV and QD"

The representative image selection procedure is the same as using R*-tree as described in Section 3.1.

The performance comparison is reported in Table 2.3. The RFS structures are built in a way that the bottom-level subclusters contain approximately 80 images. The RFS structure has 3 levels with the top level cluster having about 900 representatives. In total, the RFS has 5% of the database images. In the experiment, 100 random queries are used on both systems. The average results are presented in Table 3. It indicates that both clustering techniques yield good performance and R*-tree is slightly better than the other. For the



2.5.a: MV

2.5.b: QD

Figure 2.5: Retrieval result of “Computer from MV and QD”

prototype in this work, R*-tree is chosen, since it clusters data significantly faster than the k-mean technique.

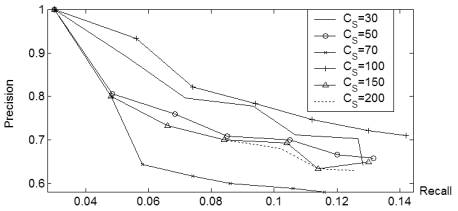
Table 2.3: system performance by using different clustering technique

Criteria	R*-tree	Kmean
1. Precision	0.76	0.75
2. Recall	0.41	0.40
3. GTIR	1	0.99

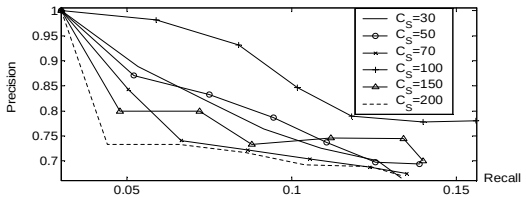
2.3.5 Effect of Cluster Size

Recall that there are two stages in constructing an RFS structure: data clustering and representative image selection. In the first stage, if the size for each RFS node (i.e., the number of representative images in each cluster) changes, both the total number of clusters at each level of the RFS structure as well as the depth of the RFS tree change. These changes may affect the selection of the representative images.

Two databases were used for this study. The first one is relatively large database with 19,000 general images. The second database is smaller application-specific database with 1,300 flower images. For the experiment on each database, the cluster size is changed between 30 and 150 and investigate its impact of the system performance. The results for the large and small databases are presented in Figure 2.6.a and Figure 2.6.b, respectively. It can be observed that the performance is the best when the cluster size is 80 for both databases; and all five curves are reasonably clustered together in both figures. This indicates that QD is robust with respect to the cluster sizes, i.e., it works well for a wide range of cluster sizes. This is a desirable property as the user has a lot of leeway to find a good cluster size for a given database.



2.6.a: 1,300 database

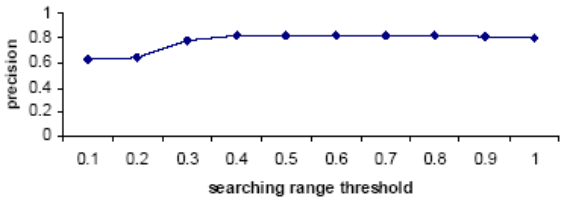


2.6.b: 19,000 database

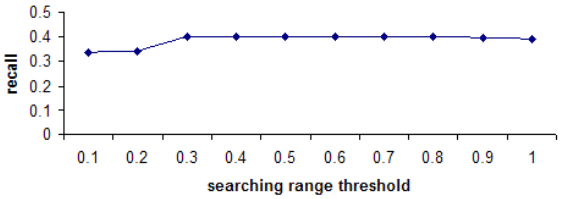
Figure 2.6: Precision and Recall vs. cluster size in 1,300 and 19,000 databases

2.3.6 Local k-NN Searching Threshold vs. Performance

In the last round of the image retrieval process, the system searches the image database for the final results in distinct neighborhoods. Recall that if the query image is far away from the cluster center, then the localized k-NN search is taken place in the parent cluster (Section 3.3). Here the threshold is used to control the searching range. In Figures 2.7.a and 2.7.b, the precision and recall are reported when the threshold for the localized k-NN search varies. It can be seen that when the threshold is around 0.4, the system performs the best on the retrieval accuracy. The systems performance decreases when the threshold is below 0.3 or above 0.8. When the threshold is set to less than 0.3, it makes the system expand the searching range to a larger space (i.e., the upper level cluster) too frequently. When the threshold is set to 0, all the retrievals take place in the whole database as in standard k-NN search. The larger the search space is, the slower the processing time. When the threshold becomes larger than 0.8, the system performance also decreases. The reason is that, when the query image is located near the cluster boundary, there might be some



2.7.a: Precision vs. Localized k-NN searching threshold



2.7.b: Recall vs. Localized k-NN searching threshold

Figure 2.7: Precision and Recall vs. cluster size in 1,300 and 19,000 databases

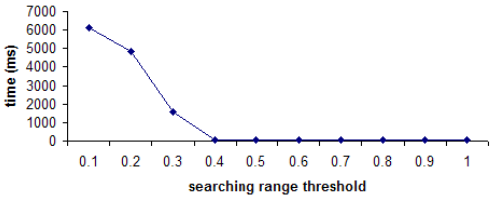


Figure 2.8: localized k-NN processing time vs. localized k-NN searching threshold.

similar images in the sibling clusters. The loose threshold limits the search within one small cluster. Therefore, the system might miss some relevant images in some sibling clusters.

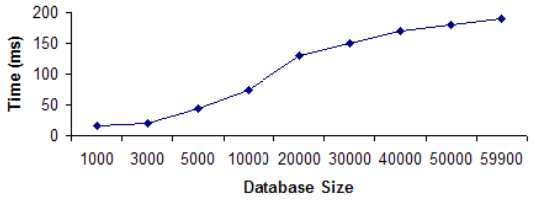
In Figure 2.8, the computation time of the localized k-NN verses the searching range threshold is shown. It can be observe that the k-NN processing time is quite long for threshold less than 0.4. This is due to the fact that the search took place in a larger cluster rather than the subcluster containing the relevant example. When the threshold is larger than 0.4, the processing time is almost the same. Combining the result shown in Figure 2.7.a, 2.7.b, and 2.8, it can be observed that the QD technique is robust with respect to the threshold. It performs well for a wide range of the threshold values.

2.3.7 Computation Efficiency

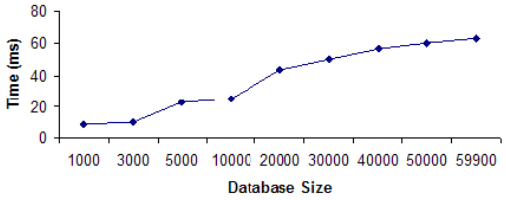
In this subsection, the efficiency of the proposed QD approach under various databases with different sizes is evaluated. As mentioned before, an important advantage of the QD approach is the efficiency attained by using the RFS structure. With this structure, expensive k-NN computation in each round of RF is avoided. If the cluster size is C and the size of a database is S, the depth of the RFS structure is $log_C S$ (For the database of 19,000 images with about 100 images in each subconcept, the RFS structure has 3 levels.) Therefore, in general, the user can find the desired images in $log_C S$ rounds. The whole RF processing time is $O(log_C S * Time(eachround))$. To confirm this, the proposed system is tested on different databases size, the largest one is 59,900 images.

This experiment was performed on a modest 2.5-GHz Pentium IV-based computer with 1GBytes of RAM. 100 initial queries are randomly generated and their average query processing time is evaluated (Figure 2.9.a), as well as the average relevance feedback processing time (Figure 2.9.b) for a single round. For each query, two rounds of relevance feedback are performed in addition to the initial query processing and the final localized k-NN computation. According to Figures 2.9.a and 2.9.b, the overall query processing time and the average iteration processing time increase slowly with database sizes. These results indicate that the QD approach is time efficient, suitable for very large databases with many concurrent users.

The efficiency of the Query Decomposition approach can also be attributed to low disk utilization. Since the information required for relevance feedback processing is stored in the RFS structure, when a representative image is marked by the user, only the corresponding RFS cluster will be accessed. This I/O cost is even less in the case that multiple relevant representative images are chosen from the same cluster. The only exception is when the query image is located far away from the centroid of the cluster. In such case, the searching of the parent/sibling nodes incurs additional disk accesses. Nevertheless, processing of all



2.9.a: Overall query processing time for different database sizes.



2.9.b: Average iteration processing time for different database

Figure 2.9: Query processing time vs. different database size

the localized k-NN subqueries need to access only a few neighborhoods in the feature space, which dramatically reduces the load on the storage system.

2.4 Conclusions

Confinement of the search results to a single neighborhood in visual-based feature space is the latent reason for the low recall/precision rate of typical nearest neighbor techniques. In this study a new image retrieval paradigm, the Query Decomposition (QD) model, is investigated. The technique facilitates retrieval of semantically similar images from multiple neighborhoods in the feature space. In this new model, a query is decomposed into multiple subqueries based on the users relevance feedbacks, so that multiple image clusters containing the semantically similar images can be covered. Compared to recent method (MV), the performance study indicates that the proposed QD technique is able to achieve significantly better performance.

The three primary contributions of the work in this chapter are as follows:

Better Query Result: The traditional k-NN image retrieval framework confines query results to a single neighborhood in the feature space. The QD approach considers multiple neighborhoods, and therefore can better address the inherent semantic gap problem in CBIR.

More Efficient Query Processing: Unlike traditional relevance feedback processing which performs k-NN computation on the entire database in each round of relevance feedback, the QD approach only performs localized k-NN computation on very small subclusters in the final round of the relevance feedback process, which dramatically reduces the computational cost.

More Scalable: Since the relevance feedback mechanism of the QD approach relies on only a tiny fraction of the database, it can be distributed to the client machines, which enables the server to focus on the final retrieval tasks. [93] has proved that this wireless relevance feedback mechanism is practical and effective.

CHAPTER 3: CLIENT-SIDE RELEVANCE FEEDBACK APPROACH FOR IMAGE RETRIEVAL IN MOBILE ENVIRONMENT

3.1 Introduction

Mobile devices with high resolution built-in camera are becoming ubiquitous. It is desirable to support multimedia information retrieval using the images taken from the camera. As examples, a student can use pictures of a plant to search for information of similar species in a remote digital library. Providing this capability calls for efficient techniques to facilitate Content-Based Image Retrieval (CBIR) in mobile environment. In a CBIR system, images are characterized by their low level features such as color, texture, and shape. Since it is difficult to "describe" a query through those features, RF is widely used and plays an important role [10,19,24,27,28,35,39]. It helps the system to understand the user's intention. In such a system, the user interacts with the system as follows: In each round, he helps by identifying the relevant images within the returned set; the system then utilizes this feedback to modify the query and thus to improve its retrieval results in the next round. This process can be repeated until the user is satisfied with the results.

Lots of information retrieval systems in mobile environment have been introduced in recent years [9, 45, 47, 50, 71, 72, 75]. In [9], a query is composed by visual sub-queries and keywords. First, the keyword is used as query to an existing search engine. Then the result images are processed in a query refinement agent based on the visual sub-queries in the client side. In [45] and [47], visual features are used to search in a large database and the result is returned to the user for relevance feedback. In summary, in those systems, the user needs to interact with the server for result. The mobile device works as an interface. However, this computation model is not suitable for RF because the wireless bandwidth is limited for exchanging images in each round of user RF. The long delay in feedback would affect

the usability of the mobile system. Moreover, communication is generally hundreds of times more demanding on mobile power than computation is [66, 76]. Constantly sending and receiving queries and results can quickly exhaust mobile device's battery power. Usually, CPU requires minimal power compared to sending data over the wireless radio. For instance, transmitting 1Kb message over a 100 meters distance is around 3 joules. On the other hand, a general processor with 300 MIPS/W power is able to execute 1 million instructions for the same amount of energy [66]. Also the storage power for mobile devices is always some hundred mW [97], which is 20-30% of the communication power consumption. Therefore, saving the communication power will be more urgent in mobile RF system.

Another issue in the existing image retrieval techniques is the semantic gap between the high level semantic concept and the low level visual features. In most of the current systems [39,41], the searching range is always confined in a single neighborhood. This is not consistent with the reality: a semantic concept can have very different representation (for example, side view of a sedan and front view of a sedan can be very different). Trying to bridge the gap, some mobile information management systems use tags generated at the point of picture was taken: In [21,48,56], location, temporal, and sometimes social contextual metadata are used as image features instead of the visual features. However, a picture is worth a thousand words. It is hard and tedious to fully describe an image using concise verbal terms. The most convenient and common way for a mobile user to express himself is to use the picture he took as the query [7].

To address the aforementioned issues, a Client-side Relevance Feedback (CRF) technique is investigated in this study. In this model, all RF except the last round is processed in the mobile device to avoid the communication cost. To achieve this, the Relevance Feedback Support (RFS) structure [38, 93] is adapted to address the storage and computation limitation of mobile devices. In the CRF system, the initial user query might be decomposed to cover more semantically relevant images, thus to bridge the semantic gap. The contributions are listed as below:

1. The CRF technique is detailed described and investigated.

2. The interface design is taken into consideration. Different interfaces are designed for different mobile users.

3. Extensive experimental studies are conducted on the CRF system to examine the mobile device power saving and the accuracy of the system.

3.2 CLIENT-SIDE RELEVANCE FEEDBACK TECHNIQUE

To support client side RF, Query Decomposition technique (QD) [38] is adapted, and RFS structure is used to facilitate the query processing: The system provides RF retrieval result to the user based on the RFS structure; Also, to cover more semantically similar image clusters, user's query is decomposed based on the RFS structure. In this section, RFS structure and the CRF retrieval technique is described in details.

3.2.1 RFS Structure

Since low-level visual features are not sufficient to capture the semantics of general images, semantically identical images may exhibit very different visual characteristics and thus may be projected to data points lying far apart in the multi-dimensional feature space. Standard retrieval techniques which are trying to find the cluster with the more relevant images cannot achieve good recall [38]. QD approach addresses this problem by finding all semantically-related clusters. To cover those clusters, the system decomposes the user's query based on the relevance feedback and the RFS structure.

The RFS structure is constructed in two stages: Data Clustering and Representative Images Selection. In the Data Clustering stage, a hierarchical clustering technique is used to organize the entire image database into a tree structure. Then in the Representative Images Selection stage, a bottom-up representative image selection procedure is performed as

follows. At the bottom level of the RFS structure, unsupervised k-mean clustering algorithm is applied to the images in each leaf node. For each of the resulted subclusters, one or more images nearest to its center are selected as the representative images. For each cluster in the upper level of the hierarchy, the representative images from its child clusters are aggregated and clustered again. Then representative images are selected for the current node. The number of representative images for each cluster is chosen proportional to the number of images in that cluster (i.e. 5% of the images). Finally, the list of the representative images and their corresponding subcluster identifications (i.e. to which they belong to) are stored in the RFS structure in the corresponding tree node. Thus, all of the information needed to support relevance feedback is self-contained in the RFS structure.

3.2.2 Query Processing with RFS structure

To help the user formulate the initial query, the system displays some randomly selected representative images from the root node of the RFS structure. From these images, the user identifies the most relevant ones. If necessary, this process can be repeated. To process the initial query, the system determines the relevant subclusters to which the selected representative images are from. If this process results in more than one relevant subcluster, the initial query is "split" into separate localized subqueries for RF, one for each relevant subcluster. Because real-world objects can have very different appearances when represented in 2-dimensional images, multiple independent subqueries are better suited for their retrieval.

In Figure 4.2, illustrate the query decomposition process with an example in Figure 1 (only tree nodes and images relevant to the discussion are presented in this figure). In this example, the RFS structure has three levels, with Node 1 representing the root cluster, which is the entire image database. Suppose the user wants to find images of cars. In the first feedback iteration, he found two relevant images in Node 1 - a steamed car and a modern car. Based on the RFS structure, the system determines that these two images came from Node

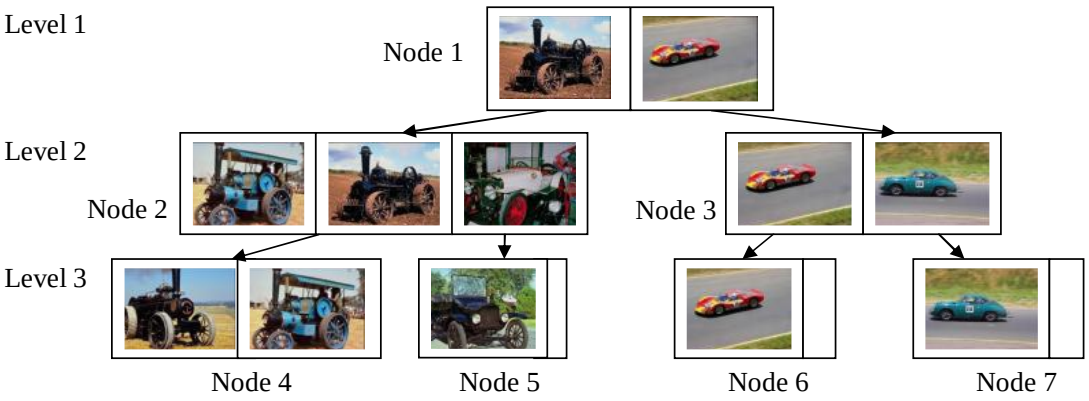


Figure 3.1: A 3 level RFS structure

2 and Node 3 in Level 2, respectively; then the initial query is split into two subqueries as follows. At the beginning of the second iteration, random representative images from Node 2 and Node 3 are presented to the user for RF. The user can now find more relevant images according to his interest, say, a steamed car and an antique car in Node 2, plus two modern cars with different colors in Node 3. With these new relevant images, the system recognizes the corresponding relevant subclusters to be Nodes 4, 5, 6, and 7 in Level 3. Finally, the user identifies the relevant images in these subclusters as illustrated in Figure 1. These images are then used as the final localized k-NN subqueries to search the corresponding subclusters independently; and their results are merged to form the overall result of car.

3.2.3 Procedures of Client-side Relevance Feedback

Given a RFS structure stored on the mobile device, the user's relevance feedback can be processed without any support from the remote server. The client-side relevance feedback is performed starting at the root node as follows:

1. A user has two ways to formulate the initial query: (1) By identifying initial relevant images (i.e. query images) from the set of images randomly selected from the representative images corresponding to the root node in the RFS structure. (This step can be repeated until the user is satisfied with the initial query.) (2) A set of pictures, taken

by the user, can be submitted as the initial query. In response to this initial query, the mobile device presents the top k similar images from the representative images in the root node; and the user marks the relevant images.

2. The images marked for the current query identify the relevant clusters in the next (lower) level of the RFS structure. A cluster in the next level is relevant if one or more of its representative images is selected as query images. With the help of RFS structure, it only takes O(1) time to determine which subcluster to explore next.

3. Localized relevance feedback is processed independently for each relevant subcluster. The user chooses query images from the set of images, randomly selected from the representative images from each relevant subcluster.

- 4. Recursively repeat Steps 2 and 3 until the RF process reaches a relevant cluster at the leaf level (i.e., a semantically relevant cluster) of the RFS structure. Then the query is sent to the server for final image retrieval. The user has the option to submit the final query before the search reaches the leaf level.

The CRF approach requires storing the RFS structure on the mobile device. Although the capacity of flash memory continues to quadruple each year, it is desirable to keep the size of this RFS structure small. Typically, the RFS structure contains less than 5image icons for feedback purposes, this RFS structure is generally very small. For instance, if each image icon requires 10 KB of storage space, a mobile device with 100MB of available flash memory can hold an RFS structure that can support an image database with 200,000 images. Considering 1 GB flash memory in a cell phone becomes a standard, the proposed approach is feasible.

3.2.4 Final Query Processing on Server

When the final query is sent to the server, localized k-NN queries are performed on the corresponding leaf cluster to retrieve the semantically relevant images. This time, the server searches all the images in the relevant cluster, not just the representative images. Finally, the candidates from different relevant clusters are merged and ranked to form the final result. A distinct benefit of CRF is that the RFS scheme only needs to do k-NN computation over the much smaller data subspaces, represented by the relevant leaf nodes in the RFS structure, in the final round of retrieval. Existing RF techniques must perform the k-NN computation over the entire database for each round of RF.

3.2.5 Active Learning in RF

According to the basic idea of QD technique, the relevance feedback procedure refines the initial query by decompose it into subqueries. However, the user might not satisfy with the images in the chosen subclusters. This might happen when the image's semantic meaning is different from the rest of the cluster, but similar on the features; or the selected relevant image is at the boundary of a cluster, the real relevant images should be in the nearby cluster. This situation is caused by the semantic gap. The user may not always be able to choose the right images. In the QD system [38], the system provides a back function to allow user to go back to the previous step to recover his wrong operation. Also, it is more difficult to use a mobile interface than a PC interface. Therefore, in order to save the mobile user's time and improve the system usability, the QD technique is adapted to learn the user's activity. This short term learning knowledge is used to adjust the relevance feedback procedure. In detail, if the user does not select any images from the returned subclusters, then the system considers that the user does not interested in any image contained within. Therefore, the system brings representatives to the user from other clusters that are not explored.

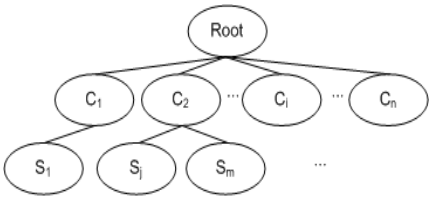


Figure 3.2: Adaptive search example.

Figure 3.2 shows a multi-level RFS structure. Under the root level, there are n clusters, which are $C_1, ...C_n$. Each of those clusters contains some subclusters. To simplify the graph, not all the clusters and subclusters are shown here. In the image retrieval procedure, user picks images from the representatives as examples. For instance, the user selects some images from C_1 and C_2. After a relevance feedback retrieval procedure on the RFS structure, the searching space is refined to the subclusters contains the selected relevant images. Say, S_{1i} $(1 \leq i \leq p)$, S_{21}, and S_{2q} are the identified subclusters. In the new round, the user does not further select any representative from S1i. There are two schemes to handle the situation.

Scheme 1: Due to the semantic gap, some unrelated images might be clustered together based on the similarity in the low-level features. Therefore, the selected subcluster, S_{1i}, does not contain the images that the user interested in. In this case, the system suggests representative image from the upper level clusters, which are siblings of S_{1i}'s parent cluster.

Scheme 2: when the selected representative image is near the boundary of S_{1i}, the images that the user interested in might be in the sibling of S_{1i}. In this case, the system needs to show images from the nearby subcluster of S1i. The nearby subcluster is determined by calculating the distance from the selected representative to the centers of the subclusters.

Considering these two cases, the system's RF result for the next round includes images from the subclusters containing the selected images and those from the clusters of the nearest sibling and upper level. After all, the relevance feedback procedure in CRF system can be summarized as in Algorithm 1.

Algorithm 3.1 Relevance Feedback in Query Decomposition

INPUT: features *F*, *DB*, *user's RF*
OUTPUT: Relevant Images as Retrieval Result

(i) $CandidateCluster = C_1, C_2, \ldots, C_n$;

(ii) $E = E_1, E_2, E_p$ = representative images randomly picked from *CandidateCluster*;

(iii) **Do**{

(iv) Show E to user for relevance feedback;

(v) User selects relevant images $I = I_1, I_2, \ldots, I_t$;

(vi) $NewCandidateCluster = \phi$;

(vii) **FOR** (each relevant image in I)

(viii) {

(ix) System identifies the corresponding RFS subcluster S_i, $0 \leq i \leq l$;

(x) $NewCandidateCluster = NewCandidateCluster \bigcup S_i$;

(xi) }

(xii) **IF** $((\exists S_j, \forall I_i, \neg (I_i \in S_j)), 0 \leq j \leq m, 0 \leq i \leq t;)$

(xiii) **THEN** E = images randomly picked from $NewCandidateCluster + \bigcup$ (representative images from closest siblings of S_j);

(xiv) $CandidateCluster = NewCandidateCluster \bigcup$ closest siblings of S_j;

(xv) $E = E_1, E_2, E_p$ = representative images randomly picked from *CandidateCluster*;

(xvi) }**WHILE**(User is not Satisfied)

3.3 3. INTERFACE DESIGN

To better help the user and present more information as possible, the interface design is also taken into consideration. There have been studies to investigate the design rules [13, 59]. With small screen space, mobile devices can only present limited amount of information at one time. So thumbnails are used for browsing. Smart phone and cell phone has different screen sizes. To maximize the screen usage on each device, different interfaces are designed

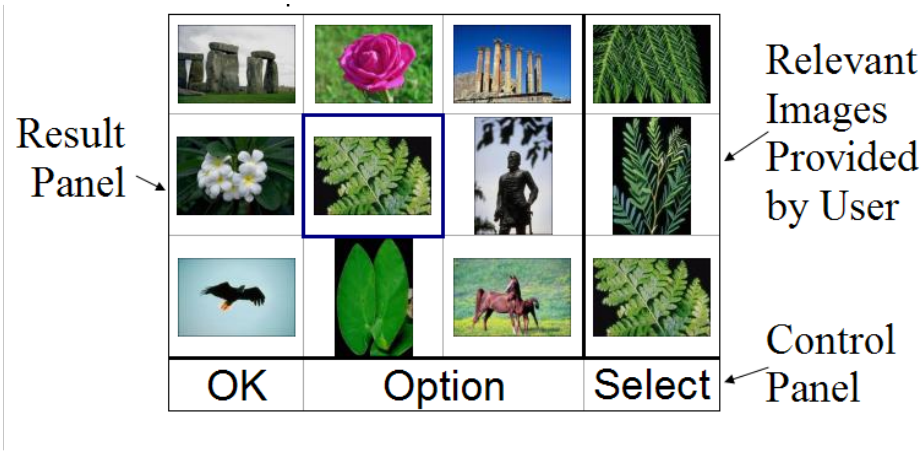


Figure 3.3: CRF mobile user interface

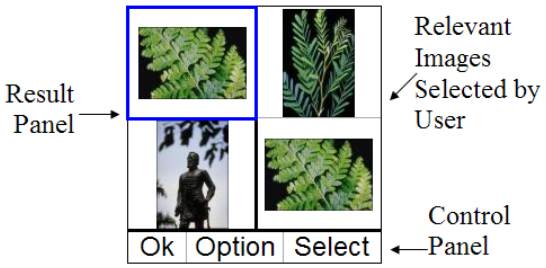


Figure 3.4: CRF mobile user interface

as shown in Figure 3.3 (for smart phone) and 3.4 (for cell phone user). The thumbnails shown in the Result Panel are the retrieval result from last round's relevance feedback, and are sorted according to the similarity to the query images. When the user is interested in one of the images, he can click "Select" to choose it as relevant images, then the image is added to the Relevant Image Panel. When the user finds enough relevant images, he can click "Option" from the Control Panel to choose whether to "submit" a query or "exit" the system.

3.4 EXPERIMENTAL RESULT

To evaluate the proposed CRF system's performance, it is compared with the existing client-server techniques. The experiment system interface is sized to fit smart phone and cell phones

as shown in Figure 3 and 4. the test images are from the Corel image database, which are classified into categories by domain professionals. There are 100 images under each category and the category information is used as the ground truth. A 19,200 image database with about 15,000 Corel images and new images is constructed. 37 features were extracted: 9 color moment features, 10 Wavelet-based Texture features, and 18 edge-based structural features. There were 100 queries for each experiment from which the average results were calculated.

3.4.1 Efficiency

The most important contribution of CRF is that it process the RF procedures all on the client-side mobile device. Therefore, there is no transmission overhead between the server and the client. The CRF user can get the RF result right after he submits the query. This advantage makes the system user friendly. To evaluate the efficiency, the amount of information transferred between server and client is captured in the CRF approach and the MV approach. 100 queries were randomly selected, each query involves 3 rounds of relevance feedback and the final round to retrieve the query results. 2 images were selected as relevant images for the first rounds; 2 more images were selected in the second rounds; and totally 6 images were selected for the third round. In the final round, 8 images were selected altogether. Within each round, the systems return 30 images as result.

To further save time for both systems, the system is designed such that if there were identical images in the previous rounds, the image would not be transmitted again. Furthermore, the system only transfers image icons, rather than the full size images which are 30KB each. The size of each image icon is about 16KB, with a resolution of 64x96 pixels. The objects in these images are recognizable for RF. For wireless transmission, assumed the system uses general cellular phone bandwidth, GPRS (General Packet Radio System), which is 5KBps.

Recall that processing RF in the CRF system incurs no data transmission. Therefore, there will be no waiting time in the RF processes. In Table 3.1, the information transferred between the server and client is listed, and also the time needed for CRF and the traditional system. To be specific, Total relevance feedback information being transferred, Total relevance feedback time, information transferred in each round, time spent in each round are computed.

Table 3.1: Table (Efficiency Comparison of CRF and Traditional Systems)

	Total RF Info. Trans (KB)	Total RF Time (s)	Info. per RF (KB)	Time per RF (s)
CRF	0	0	0	0
Traditional	992	198	330	66

From the result, it can seen that averagely, the user need to wait more than 1 minutes to get the result in each relevance feedback round, resulted in the user in the traditional server-client system needs 3.3 minutes to refine a query before the final round. However, there is no waiting time for the user in CRF system for refining a query. In fact, the delay in MV can be significantly longer since the server needs to multiplex among a large number of concurrent users to support their relevance feedback.

3.4.2 Interface Satisfaction

To test the interface's usability, 30 university students were invited to evaluate the proposed simulated interfaces: the PDA interface and the cell phone interface. They were asked to find images shown in Table 2.1, and whether they are satisfied with the interface and system's performance. The results are listed in Table 3.2, which indicate that the users are satisfied with the interface on the usability and speed. However, some of the users thought that the interface is designed for users with certain computer background.

3.5 Conclusion

The significance of the proposed Client-side Relevance Feedback (CRF) technique is twofold. First, it leverages distributed mobile computing to significantly reduce server load and therefore improve system throughput. Second, since processing of relevance feedback incurs no communication delay, CRF enhances system usability. The experimental studies indicate that substantial improvement in retrieval precision and recall can be achieved.

Table 3.2: Table (User's satisfaction of the CRF System)

Evaluation item	Satisfaction (out of 10)
Organization of information	9.1
Sequence of screens	9.1
Position of messages on screen	8.8
Learning to operate the system	8.5
System speed	9.8
System reliability	9.1
Designed for all levels of users	8.3

CHAPTER 4: AN IN-MEMORY RELEVANCE FEEDBACK TECHNIQUE FOR HIGH-PERFORMANCE IMAGE RETRIEVAL SYSTEMS

4.1 Introduction

In typical content-based image retrieval (CBIR) systems, visual features, such as color, texture, and shape are extracted to facilitate similarity computation. Such low-level features, however, are not sufficient to capture image semantics. To address this semantic gap, Relevance feedback (RF) has been identified as an effective solution. Allowing users to refine queries with RF significantly improves retrieval effectiveness, and has been widely adopted in recent CBIR systems [38, 39, 41, 65]. The effectiveness of the RF model depends on the efficiency of the feedback mechanism. Typically, multiple rounds of feedback are involved. In each round, the refined query must be processed over the database stored on disks. This is a lengthy process, particularly for large image databases, and may lead to high reneging user rates reducing the advantages of the RF system. Hence, scalability is a limitation of today's RF systems that manage large image collections.

To address the aforementioned problem, various data clustering [42] and indexing techniques [6, 8] have been investigated to reduce the number of disk accesses. This chapter explores an alternative approach by studying an in-memory RF technique, in which the high dimensionality of image feature vectors residing on disk are reduced to much lower one so that the compressed database fits in memory for fast RF processing. The main goal is to minimize the number of accesses to disk, executing similarity computation on the full feature set only in the final round of the retrieval. Moreover, short turn-around time for RF processing improves system usability and therefore encourages the user to try more rounds

for higher-quality results. The experimental results show that the proposed system significantly reduces response time, and outperforms conventional relevance feedback techniques by a significant margin. The contributions of this work are as follows:

1. A notion of RF with no false dismissal is introduced. That is, the set of images returned from the smaller database must include all the images that would be returned by the conventional RF mechanism on the original databases.

2. To ensure the 'RF with no false dismissal' property, an efficient retrieval technique is proposed. To the best of the author's knowledge, this technique is the first in-memory RF method in the literature.

3. The efficient configurations for system implementation is investigated. Experimental results to illustrate the benefits of the proposed in-memory RF approach are also provided.

The remainder of this chapter is organized as follow. Background and related work are reviewed in Section 2. The proposed dimensionality reduction technique is presented in Section 3. Section 4 introduces the proposed RF retrieval algorithm. The system prototype is presented in Section 5. Experiment results are discussed and some of the desirable properties of the selected dimensionality-reduction technique are examined in Section 6. Finally, Section 7 concludes this work and discusses possible extensions.

4.2 Background and Related Works

Recent research in CBIR focuses on query model designs. SIMPLIcity [83], SamMatch [37], and WALRUS [61] split each image into individual regions, and retrieve images with matching regions to the query image. MARS [65] and QCluster [39] group the relevant images into clusters, and draw query contours according to the clusters to eliminate as many

irrelevant images in the final result set as possible. While these techniques attempt to use a single contour to enclose the entire return set, Query Decomposition [38] decomposes queries into subqueries, each of which is independently executed. The retrieval set is no longer limited to a volume surrounding a single location, but a possible union of multiple, disconnected regions of the search space.

Note that the main objective of these above techniques is to improve the effectiveness of image retrieval using various similarity measures such as a weighed Euclidean distance [39]. In fact, they perform very well in terms of effectiveness and efficiency when the (full) feature set fits in memory. However, since they did not consider the issues of I/O accesses in response time, these approaches might not scale well for very large datasets that must reside on disk. One solution to this problem is to reduce the size of the feature file by reducing the dimensionality of its feature vectors. This solution incurs false matches (with respect to the similarity measure, such as the Euclidean distance). Consequently, the main criterion for selecting a dimensionality-reduction technique for this purpose is that it produces few false hits.

There are many well-known transformations that can be used for dimensionality reduction, such as Discrete Fourier Transform (DFT) [3] and Discrete Wavelet Transform (DWT) [51]. In those techniques, the reduction is accomplished by discarding the insignificant dimensions while keeping the principal dimensions. False matches could be further reduced by transforming data according to their distribution. Single Value Decomposition (SVD) [18,46] or PCA is the most widely used technique. It is optimal in the sense that, among all the possible linear transformations of a set of data into a lower dimensional space, it minimizes the mean squared error between reconstructed data and the original data. Unfortunately, SVD incurs high computation cost and high memory requirements for data analyses. Moreover, SVD is a data-dependent method, meaning that for datasets with frequent insertions and deletions, the feature set is required to be rebuilt regularly to maintain effectiveness.

In [81], a high-performance dimensionality reduction technique is proposed, referred as MS, to enable fast search over high dimensional data. It is based on a non-linear transformation, and can provide a closed volume around search spheres of arbitrary dimensionality. Bounded approximation has been proved to improve performance. MS is very competitive with SVD without data analysis because it is a data-independent technique, thus suitable for highly dynamic datasets such as image databases. Another advantage of this technique is that optimization is possible based on the important characteristic of queries (i.e., the standard deviation). Consequently, in this chapter, above technique is used for dimensionality reduction, which will be discussed in more detail in the next section.

Dimensionality reduction also has another advantage: the computation time can be reduced if main memory can accommodate the full feature set. Consider the Euclidean distance, which many RF systems use [38, 60, 69]. For a database of 60,000 images, each with 37 features (dimensions), one scan takes around 36 ms. If the data were reduced to 8 dimensions, it took less than 14 ms. If the complete set resided on disk, this would add expensive I/O accesses into the processing time. In this case, 2 seconds for scanning the 37 dimensions dataset. The total cost for the entire RF session using the full set would then be unacceptable.

Compression is an alternative to dimensionality reduction, which is widely used in many applications [25,58]. A compressed file is typically much smaller than the original file so that storage media can hold more digital contents. In supporting in-memory RF retrieval, a compressed feature file can be placed in main-memory to prevent expensive I/O. During query processing, feature vectors that are needed can be decompressed for similarity computation. As a result, the high cost of decompression must be added to response time. Although this effect can be avoided by executing comparison in the compressed domain, such a technique has drawbacks: the process is non-linear and the data structure syntax is rigid [26]. On the other hand, dimensionality-reduction approximation has a solid foundation (e.g., no false dismissals) and proven practical applications.

4.3 Dimensionality Reduction

In this section, the criteria for an effective dimension reduction technique to support RF is first discussed. Then, the details of how to use MS technique in the proposed system is discribed. In this work, the Euclidean distance (i.e. L_2 norm) is used as the measure of similarity. The reasons are: it is used widely in image retrieval [30, 38, 55, 60, 69], and more importantly, it has been proved that any finite metric space can be embedded into normed space L_2 [12].

4.3.1 Criteria for Selecting the Dimensionality Reduction Technique

For the in-memory RF purpose, a good dimensionality reduction technique should meet the following criteria:

1. **No False Dismissal**. When a reduction technique is applied, the original feature space G is transformed into G'; the query is also transformed and executed in the reduced space. The query result from the transformed space should be a superset of the query result from the original space. In other words, a good dimensionality reduction technique should be distance preserving.

2. **Data Independent**. Data deletions and insertions should not affect the current transformed data and the index structure that holds them.

3. **Optimization is Possible**. The performance characteristics of the technique are well known so that optimization for RF retrieval can be performed.

As mentioned in the preceding section, a recent technique, MS, satisfies all the above criteria. So, it is utilized in the proposed system. In the next subsection, important properties of the MS technique are summarized.

4.3.2 The MS Technique

Let ε be a user-defined threshold, image $P(p_1, p_2, ..., p_n)$ is similar to query image $Q(q_1, q_2, ..., q_n)$, if P satisfies the following equation (p_i and q_i, $1 \leq i \leq n$, are the i^{th} feature of image P and Q):

$$Distance(P, Q) = [\sum_{i=1}^{n}(p_i - q_i)^2]^{1/2} \leq \varepsilon. \quad \text{(Eq. 4.1)}$$

The central idea of MS is to approximate the retrieval set using two important parameters of data points: the mean μ and the standard deviation σ of data point coordinates. Consider a subset m of the n dimensions, $1 \leq m \leq n$. A dimension i belonging to that set, will be denoted as $i \subseteq dim(S_m)$. $\mu_p^{(m)}$ and $\sigma_p^{(m)}$ of S_m are defined as follows.

Definition 1. MEAN:

$$\mu_p^{(m)} = \frac{\sum_{i \subseteq dim(S_m)} P_i}{m} \quad \text{(Eq. 4.2)}$$

Definition 2. STANDARD DEVIATION:

$$\sigma_p^{(m)} = [\frac{\sum_{i \subseteq dim(S_m)} P_i^2}{m} - (\mu_p^{(m)})^2]^{(1/2)} \quad \text{(Eq. 4.3)}$$

When $m = n$, write $\mu_p^{(n)} = \mu_p$ and $\sigma_p^{(n)} = \sigma_p$. The computation of μ and σ are the same as the conventional mean and deviation. The terms have been adopted to describe these two important characteristics of points representing objects such as images [37] and time-series [29].

LEMMA 1. Consider points Q and P satisfying Equation (1) for some search distance ε. The inequality holds that:

$$\sqrt{(\mu_p - \mu_q)^2 + (\sigma_p - \sigma_q)^2} \leq \frac{\varepsilon}{\sqrt{n}} \quad \text{(Eq. 4.4)}$$

Proof: see [81].

For implementation, the dimensions are partitioned into l disjoint subsets of equal size $m=n/l$, assuming n is divisible by l. Thus there are l pairs of mean and standard deviation: $\mu_1, \sigma_1, \mu_2, \sigma_2, ..., \mu_l$, and σ_l. Let $d_1 = (\mu_{p1} - \mu_{q1}), d_2 = (\sigma_{p1} - \sigma_{q1}), ..., d_{2l} = (\sigma_{pl} - \sigma_{ql})$. The following lemma ensures no false dismissals of qualified images for similarity computation based on the reduced data.

LEMMA 2. Consider points Q and P satisfying Equation (1) for some search distance ε. The inequality holds that:

$$\sqrt{\Sigma_{i=1}^{2l} d_i^2} \leq \frac{\varepsilon}{\sqrt{n}} \qquad \text{(Eq. 4.5)}$$

Proof: see [81].

Summarizing the properties of the MS technique, Lemma 2 ensures that the MS technique guarantees no false dismissal; the transformation of MS is data independent; optimization is possible. The optimization issue is discussed later from the system view.

4.4 Relevance Feedback

According to the description in Section 3, the MS technique is able to approximate the nearest results based on the reduced dimensionality and the approximate set guarantees to contain the real result set executed on the original feature set. Therefore, this technique is utilized in the proposed system to reduce the feature set's dimensionality. Moreover, an in memory relevance feedback procedure is proposed.

4.4.1 Relevance Feedback Algorithm

To execute the RF procedure in memory, an reduced file is created to fit into the memory from the original data, see algorithm 4.1.

Algorithm 4.1 REDUCEDATA

INPUT: features F, DB, integer l

(i) $For(\forall P \subseteq Database)$

(ii) { group consecutive dimensions to form l disjoint subsets of size m=n/l}

(iii) Compute $\mu^{(m_i)}$ and $\sigma^{(m_i)}$ for set i. Thus, there are l pairs of μ, σ for each point.

(iv) Store these pairs into file $F_{reduced}$

Hence, algorithm 4.1 lowers the number of dimensions from n to $2l$, $1 \leq l \leq n/2$. In implementation, $d = 2l$ is no more than the intrinsic dimensionality of the data. When n is not divisible by l, zeros are appended to the end to make it to be.

According to the performance study in [81], MS is very effective for queries with small σ. In fact, when their $\sigma \rightarrow 0$, the precision approaches 100% (i.e., the approximation set is identical to the exact set). When the σ increases, the approximate set includes more false hits. Thus, to achieve very high precision for all queries, an σ threshold, σ_T, is set, below which the search is performed on the reduced data, whereas a higher value sends the search to the full set on disk. Algorithm 4.2 describes the details of the RF retrieval, which interactively searches for images similar to query Q within a user-defined search radius ε.

Algorithm 4.2 SEARCHDATA

INPUT:(features F, reduced_features $F_{reduced}$, query-point Q, radius ε, threshold σ_T)

(i) **repeat**
Analyze queries (computing their σ)
Search the in memory $F_{reduced}$ with a window query of range ε centered at Q
if $\sigma > \sigma_T$ **then**
apply Eq.1 to the full feature vector in F on disk to exclude false hits
end if
User identifies relevant images for next round of retrieval.

(ii) **until** user select the final relevance set for exact images

(iii) For all query images, apply Eq1 on F to determine final result set.

According to algorithm 4.2, the proposed RF search can occur both in memory (on the reduced data by Alg 4.1) or on disk (on the original set). These operations can be controlled by setting proper σ_T, to achieve the best tradeoff between precision and response time (discussed in Sec. 6). The search terminates when the user executes the final search (by hitting the *FinalQuery* button on the interface) or when the system detects that there is no changes in the refined query set. The precision of this step is equal to the highest precision produced by the conventional RF technique.

4.4.2 Determine the Searching Range

Note that, the system proposed in this chapter does not execute k-NN search directly as employed in conventional CBIR systems; instead, a spherical range search is used to approximate the k-NN result. Generally, the search radius needs to be sufficiently large so that the returned set covers at least the k nearest images. As a result, the approximate set may include some false hits. In other word, it sacrifices the precision to ensure the inclusion of all the exact images. Therefore, choosing a proper search radius is critical to ensure the search returns at least the k nearest images of the query, but not too many false hits. Different strategies can be used depending on the applications and the characteristics of their datasets. Below, three strategies are considered to determine ε and their advantages and disadvantages are discussed.

4.4.2.1 Exact Range

This strategy precomputes and stores the search range to retrieve the desired k for each point in the dataset. The calculation of ε is based on the original dataset. For example, if the system is mostly asked to return 10, 20, 30, and 40 nearest images, the minimum necessary thresholds to reach the nearest 10^{th}, 20^{th}, 30^{th}, and 40^{th} images are computed. These values are stored along with other parameters of the images. During query processing,

these thresholds are used to retrieve the specified number of nearest images. The advantage of this scheme is the system will return no false hits in the result by applying Eq. 5, then Eq. 1 on the approximate set. However, this scheme lacks flexibility (e.g., when executing *50*-NN), and insertions or deletions requires recomputation of the thresholds for affected points. Extra space is needed to store these thresholds.

4.4.2.2 Approximate Range

In this scheme, only the search distance to *10*-NN, and the distance increment *var* from *10*-NN to *20*-NN are precomputed and stored. When 20 nearest images are needed, search threshold of (10-NN + *var*) is used, (10-NN + $2 * var$) for 30 images, and so on. Observe that when the search threshold linearly increases, the search volume expands by many folds. For instance, in a 3-dimensional space, the volume of a ball with radius r to that of a ball with radius $2r$ is

$$\frac{Volumn_r}{Volumn_{2r}} = \frac{\frac{4}{3}\pi r^3}{\frac{4}{3}\pi(2r)^3} = \frac{1}{8} \qquad \text{(Eq. 4.6)}$$

As a result, this approximate scheme includes false hits but does not exclude relevant ones. In the experiments below, only few false hits are observed and precision is not significantly affected when the k-NN is close to the *10*-NN. With respect to the preceding scheme, this strategy also requires less storage, is more flexible, and demands less frequent updates of the thresholds due to insertions and deletions.

4.4.2.3 Pre-defined Incremental Range

This strategy does not store εs for individual images. Instead, it predetermines εs based on the entire dataset, and applies them to all search operations for every query. Let's assume that users typically expand their search starting with *10*-NN up to *40*-NN. This scheme works as follows. After collecting the distance information for *40*-NN search as in Scheme 1, the average ε: ε_a and the largest ε: ε_l are calculated to set the reach range of ε: $\varepsilon \subseteq [\varepsilon_a/2, \varepsilon_l]$.

Analysis of the database suggests that 10 εs in that range should be used: $\varepsilon_a/2$, $3\varepsilon_a/4$, ε_a, $(9\varepsilon_a + \varepsilon_l)/10$, $(7\varepsilon_a + \varepsilon_1)/8$, $(5\varepsilon_a + \varepsilon_l)/6$, $(3\varepsilon_a + \varepsilon_l)/4$, $(\varepsilon_a + \varepsilon_l)/2$, $(\varepsilon_a + 2\varepsilon_l)/3$, ε_l. At the start, the system uses $\varepsilon_a/2$ and the next value of ε if there is not enough result using the previous ε. Observe that the thresholds' increment is progressively larger toward the end of the range. This is because starting ε is typically small and a slowly initial increment helps avoid the inclusion of too many false hits. As the user continues to request for more images, a more aggressive expansion is needed to speedup the search. The searching range can be extended for k-NN, $k \geq 40$, image retrieval by using the current ε_a and ε_l. This scheme saves storage space, but sometimes, the system has to repeat the search to satisfy the need of large search. Comparing with the previous schemes, this strategy rarely needs to update the range provided that the thresholds are properly selected. They can be even used in other systems whose datasets share similar characteristics, such as image collections.

4.5 System Prototype

This section describes the system prototype. First, the structure of the in-memory index is defined. Then, the whole RF process is laid out in details. Third, how to optimize the system's performance is discussed. Finally, the last subsection presents the system's interface.

4.5.1 Structure of the In-Memory Index

The structure of the database is modeled based on the relational database concept - each image has a unique identification number (Image Identification number - IID). The compressed feature, the original feature, and the image icon are all identified by this IID. There are two tables:

1. Object Feature Table for the reduced feature data. Small image icons are stored in this table for relevance feedback. After the similarity computation, the resulted image icons are returned for user's further relevance feedback.

2. Image Feature Table for the original features, which is stored on disks.

4.5.2 Query Processing

The relevance feedback process, according to Alg. 4.2, is depicted in Figure 5.1. Upon presenting the initial query, the user starts an interactive retrieval session to search for relevant images. Similarity computation is performed on the reduced data in main memory and occasionally on the full set on disk. The results are returned for user RF so that the query is refined for the next round of search. The procedure is repeated until there is no change detected by the system or the user decides to execute the final query. The query is then evaluated on the original dataset as in the conventional k-NN processing (marked 2 in Figure 5.1). The result of this stage is of highest precision and equivalent to that obtained by k-NN search using the original database.

When there is more than one image in the query, the system will decide whether or not to decompose it into subqueries for higher retrieval effectiveness. The above procedure will be applied to each subquery, the results of which will be merged and presented to the user as the results of the entire query. More details of image retrieval with query decomposition is discussed in [38].

4.5.3 Optimization

The performance of the approximation technique has been studied in [81], and can be predicted based on the σ value of the query, or subquery. When σ is small, the approximate result is nearly identical to the exact set. Therefore, it can be used on the reduced data to

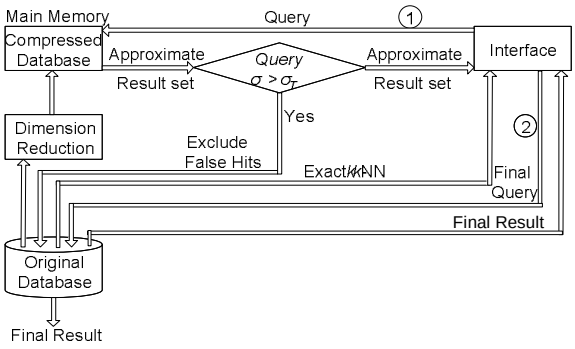


Figure 4.1: System prototype

determine relevant images. When σ exceeds a certain threshold σ_T, search based on the full-dimension set is performed to obtain the exact result, because the approximated set might be large. Here σ_T can be set to control how search is being conducted (either in memory or on disk), and thereby the performance of the system.

Observe that in the proposed system a user might have to examine some false hits in exchange for much faster response time due to the use of the approximation scheme. After the user has identified relevant image(s) from the returned images for the next iteration, false hits are discarded and the system starts in the same state as if k-NN had been executed, i.e., processing the same query at the beginning of a round. This implies that false hits in the proposed system are not accumulated, but are confined within individual rounds. With much faster response time, this system allows users to retrieve the target images more quickly, or to execute additional relevance feedback iterations for higher quality results.

4.5.4 Interface Considerations

The system interface based on the ImageGrouper [60] is shown in Figure 4.2. To help a user formulate the initial query, the system displays certain representative images and/or some randomly selected ones in the bottom-left panel. Relevant images are drag-dropped into the right panel, which form the first query. The results of its execution displace the images in the left panel, and the user then can pick relevant images and places them in the right panel

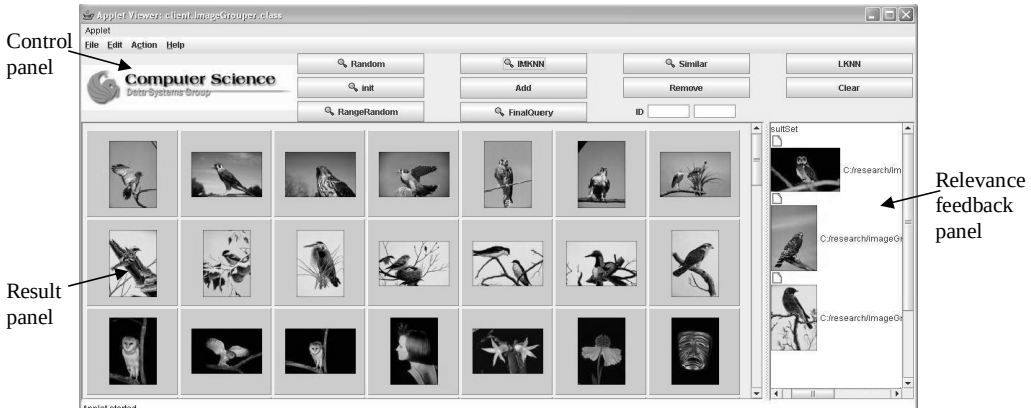


Figure 4.2: System Interface

to form a refined query for the next round, and so on. This process continues until the final query is executed. The final results should be identical to conventional system, but much faster.

As explained earlier, if a query contains multiple images, then it might be decomposed into subqueries. The simplest decomposition considers each image as a subquery. Result images are displayed in groups as follows. First, the ranking score for each set of the results are calculated as the sum of the similarity scores of the individual result images contained therein. Then these result groups are sorted in the order of their ranking scores. Within each group, the images selected for the final result are displayed in the order of their individual similarity scores. A screenshot given in Figure 4.2 shows how the results are organized. This system can also presents the images based on their individual similarity to make the model transparent to the user.

4.6 Experimental Study

This section discusses the results of the performance study, evaluates the effectiveness and efficiency of above in-memory technique.

4.6.1 Environment

The test database used in this work includes 59,900 images from Corel data-base. From each image, 37 features are extracted, which were categorized into three groups: 9 color moment features [77], 10 Wavelet-based Texture features [74], and 18 edge-based structural features [98]. These features have been shown to be effective in supporting image retrieval [38, 60]. Since there is no similar in-memory RF system in the literature, the proposed approach is compared with a technique that performs k-NN search using the full feature set, i.e., scanning the set and computing detailed similarity calculation. The original features were stored in a MySQL [2] database to simulate the traditional disk-based CBIR systems. The performance of such a technique, referred to as the conventional system, establishes the ground-truth in the following experiments. More than 500 random queries from the dataset were executed in the experiments. To divide the full feature set dimensions into subsets of equal sizes, zero elements are appended to some features, so that they can be groupped into *8* pairs *(m=5)*, *4* pairs *(m=10)*, or *2* pairs *(m=20)*. To be specific, one zero is added to the color features, and two zeros are added to the edge structure features.

4.6.2 Metrics

To evaluate the effectiveness of the proposed system, its performance is compared with that of the conventional system, under various range selecting schemes. Since the proposed system ensures the inclusion of the exact k-NN set in the approximate set, it is necessary to show this system does not overburden the users with many false hits. Since the false hits due to the use of this approximation method are confined within individual rounds, the average Relevance Feedback Precision (RF_Precision) for all rounds is defined as below to capture the overall effectiveness:

$$RF\_Precision = 1 - \frac{ApproxResult - kNNResult}{ApproxResult} \qquad \text{(Eq. 4.7)}$$

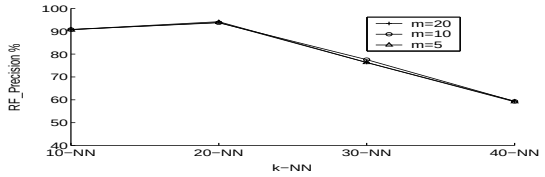


Figure 4.3: m vs. RF_Precision using scheme 2. The performance of these 3 settings are similar, resulting the overlap plots.

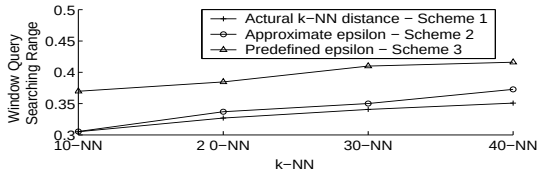


Figure 4.4: Comparison of ε determined by schemes1, 2, and 3

To demonstrate the efficiency of the proposed technique, the response time of above in-memory relevance feedback system is also measured. Since computation time depends on hardware configurations, the percentage of sequential scans on the original features is defined as the base measure of efficiency performance. In order to achieve various performance objectives, the size of the reduced data is also examined.

4.6.3 Effects of Dimensionality Reduction

The sizes of feature files, at various rates of reduction, are shown as follow. When $m = 5$, the size of the reduced set is about 1/2 of the original file. When $m = 10$, it is 1/4, and when $m = 20$, it is about 1/10. This confirms that the MS technique can reduce the feature set for in-memory storage. Figure 4.3 shows RF_Precision versus m; the precision remains almost unchanged for $m = 5$, $m = 10$, or $m = 20$. This is because the intrinsic dimensionality of the feature is around 4 (cf. Section 3). Thus, the selected reduction technique remains effective under very high rate of reduction. For subsequent experiments, $m = 20$.

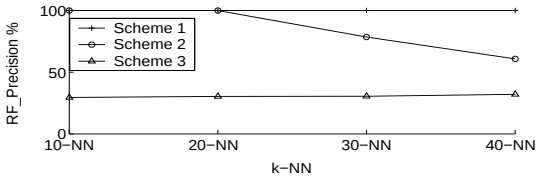


Figure 4.5: RF_Precision comparison by using different Schemes

4.6.4 Performance of Search Range Schemes

Figure 4.4 compares the εs determined by Scheme 1, Scheme 2, and Scheme 3 for some nearest neighbor searches. Recall that the εs in Scheme 1 is the actual distance to retrieve exactly the required number of nearest neighbors, while it is estimated in Scheme 2, and the εs in Scheme 3 are predefined based on an analysis of the dataset. This figure shows that the estimated and the predefined εs are always larger than the actual k-NN distances, indicating that Scheme 2 and Scheme 3 ensure the retrieval of the k-NN set. For small k-NN search, the distances in Scheme 2 are almost the same as the actual k-NN distances, while Scheme 3 seems to over-estimate the actual distance.

RF_Precisions of the three schemes are compared in Figure 4.5. Specifically, for Scheme 2, its RF_Precision improves as the number of requested images decreases. For instance, for $k = 40$, the user has to view about 60% more images than normal. When $k = 30$, there are only about 20% more, $k = 20$ about 6% more, and $k = 10$ there is no false hit. For Scheme 3, its RF_Precision is about 30% for all the cases. The performance of Scheme 3 can be improved by defining more intervals in the range of $[\varepsilon_a/2, \varepsilon_l]$. However, that would require the system to expand search more slowly to reach the desired number of relevant images. Using the current setting, the system performs on average 3 spherical range searches in order to obtain the required number of images, which is a reasonable number of iterations for most users.

Note that Scheme 1 and Scheme 2 use more space overhead than Scheme 3, which requires longer execution time. Thus, which scheme is selected depending on the availability of memory for storing search radii. Subsequent experiments will focus on the effectiveness

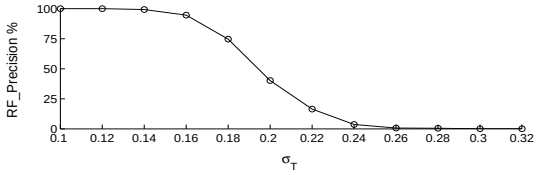


Figure 4.6: RF_Precision versus σ_T

of similarity computation using the reduced data, therefore, Scheme 1 is chosen for the implementation.

4.6.5 Effects of deviation threshold

As discussed in Section 4, standard deviation threshold σ_T can be used to control against which data set a query is performed, the full set or the reduced set. Figure 4.6 plots RF_Precision for various query images' σ using the reduced set. This figures shows that the RF_Precision decreases when σ_T increases. For example, when $\sigma_T = 0.16$, RF_Precision is over 80%. However, when $\sigma_T = 0.2$, the precision is about 40%. To improve the overall performance, the proposed system forces the similarity measure to be evaluated on the full set for queries with $\sigma > \sigma_T$. When a query image's σ is greater than σ_T, the system needs to fetch the required feature vectors from disk for similarity computation, which incurs disk accesses. Thus, σ_T should be set so that a good balance between effectiveness and efficiency of the system is achieved. The next subsection will discuss the effects of σ_T on system response time and similarity computation time.

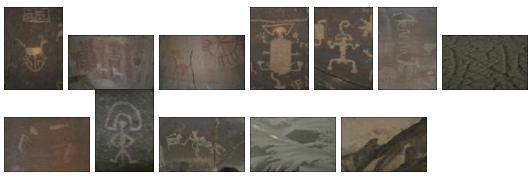


Figure 4.7: Relevance feedback from the proposed system

To illustrate the effectiveness of the system with $\sigma_T = 0.2$, a sample query and its results after a round of relevance feedback computation are presented in Figures 4.7 and 4.8. The

Figure 4.8: k-NN relevance feedback

results are ranked based on the reduced features (see Figure 4.7). Comparing the retrieved images with the top ten k-NN on original features, shown in Figure 4.8, the result set from the proposed system is similar to the conventional k-NN result: the k-NN images can be found at the top of the result set, but might be at the same or some lower ranked places. For instance, the 1st k-NN image is at the 1^{st} place in the proposed system's results, the 3^{rd} image at the 4^{th}, the 2^{nd} k-NN image at the 5^{th}, and the 6^{th} image at the 7^{th}. All the other images in Figure 4.7 can be found in Figure 4.8 at similar ranked places. When the final query is executed, the results from the proposed system are identical to those from the conventional system.

4.6.6 Efficiency

In the following experiments, search is executed by sequentially scanning the data files. Although an index structure such as R*-tree [6] could be used to accelerate the search, it does not scale linearly with the size of the data set; it favors small data set (e.g. the reduced data) over the large set (e.g. the full set). The time complexity of sequential scans, on the other hand, is linearly proportional to the size of datasets, thus giving a more accurate comparison of response times.

Figure 4.9 plots the response time of query execution under various σ_T and compares them with the execution time of the conventional system (with the original features stored in a MySQL database). As shown in the figure, when $\sigma_T < 0.16$, the proposed system takes about 80% of the execution time of the conventional system to process the same query. This indicates that many disk accesses occurred for this setting of σ_T. On the other hand, when

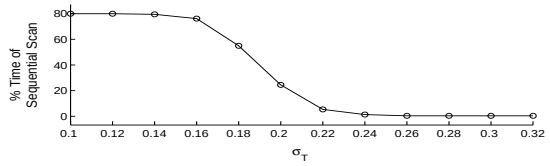


Figure 4.9: Execution time versus σ_T

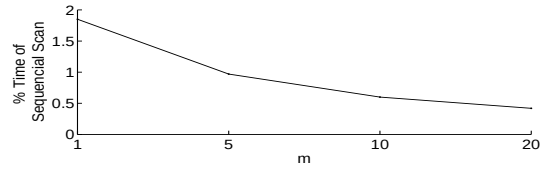


Figure 4.10: Execution time versus m

$\sigma_T > 0.22$, the execution time of the proposed system is less than 10% of the conventional system, and it is about 1% when $\sigma_T > 0.24$. Considering the precision performance at various σ_T (see Figure 4.6), $\sigma_T = 0.2$ is a good trade-off for the data set: RF_Precision is acceptable, which is 40%, and the execution time is 24% that of the conventional system.

Figure 4.10 compares the response time of various reduction configurations relative to that of the conventional system, with the assumption that the reduced data fits in memory while the original set resides on disk. As seen, the response time of the proposed technique is less than 1% of the conventional disk-based CBIR system. It confirms that for large datasets which must reside on disk, an efficient in-memory RF technique can ensure good performance and system usability. Even if the full set can be fully accommodated in memory, the use of a dimensionality-reduction technique is still beneficial. A scan of the reduced data with $m = 20$ takes less than 1/3 that of the full set ($m = 1$) if they are in memory.

In practice, not all queries in the proposed system are executed in memory: for some queries, the system needs to access the original set to improve precision. The response time is therefore longer than the ideal, no-disk-access scenario. However, the increase is expected to be small since only specific part (determined by IID, see Section 5) of the full dataset needs to be fetched. With $\sigma_T = 0.2$ and $m = 20$, the proposed system's retrieval time is 24.51% of the retrieval time of the traditional system.

4.7 Conclusion

This chapter presents a novel in-memory relevance feedback technique to address the scalability and usability problems of today's CBIR systems with relevance feedback. A non-linear approximation method is utilized to reduce the dimensionality of the feature set so that it fits in the memory to facilitate efficient feedback processing. The proposed system guarantees no false dismissals with respect to the similarity measure based on the Euclidean distance. Experimental results show that this technique significantly reduces response time while maintaining high precision of retrieval. With the growing interest in Internet-scale image search applications, it offers an effective solution to support very large image databases.

CHAPTER 5: A MULTI-DIRECTIONAL SEARCH TECHNIQUE FOR IMAGE ANNOTATION PROPAGATION

5.1 Introduction

Recently, most of the annotation researches focus on learning the mapping between images and keywords [5,17]. However, those systems can only learn the correlations between a small image database and a very limited vacabularies. It is not applicable to the web-scale image database with potentially unlimited vocabularies.

Another image annotation approach is based on the image labels of an annotated database [53, 95]. The basic idea is to build a initial image database with labels, the labels are from the images' bounding text on a web page or a vocabulary. Then the labels are propagated to the visually similar images. Those images are from a content-based image retrieval or clustering based on the images' visual features. A potential weakness of these approaches is that the bounding text can be incomplete and/or inconsistent compared to the real image content, especially on the spam pages. From this point of view, the bounding text is not always reliable.

From the above discussion, it can be observed that to annotate a large scale image database with unlimited vocabularies, user's *Relevance Feedback* (RF) should be utilized as a trust-worthy source for annotations. At the same time, an accurate CBIR technique with user's RF should be utilized for efficient annotation propagation. CBIR with RF has been an active area of research [27,35,38,79,83] for the last decades. Essentially most of the existing techniques are based on the kNN (k nearest neighbor) model. Two images are considered similar if they are near each other in the feature space based on some distance measure. For each round of RF, the user's identified relevant images are used to retrieve k nearest images for the next round of user feedback. Different techniques have been proposed for the computation of these k nearest images.

However, the standard kNN model, which is employed by today's relevance feedback techniques, has one inherent weakness. Due to the semantic gap, semantically similar images might have different appearances and are not necessarily located within close proximity to each other in the feature space. Since traditional CBIR techniques focus on searching for the single best-matching neighborhood, their performance suffers when the relevant images are in multiple neighborhoods far apart from each other in the feature space. For instance, the user provides visually different but semantically identical example images, then those images should be treated separately in the CBIR process.

To address the aforementioned limitations, A *Multi-Directional Search* (MDS) technique is proposed in this chapter. In this system, a user provides sample images with labels as query for annotation propagation: first, local clustering is performed on the example images based on their visual features. If the user's RF images have very different visual features (shows his intension to annotate groups of images with diverse visual characteristics), the current query is decomposed into multiple sub-queries accordingly to better benefit from the feedback information and provide more precise annotation. Similar to CBIR, the neighboring images of the multiple sub-queries will be returned to the user for further RF, and the sub-queries might be decomposed again. This mechanism enables the proposed technique to explore all relevant neighborhoods in order to cover the k best matching images wherever they might be, rather than just the top k ranked images of the single best-matching cluster (while omitting other relevant clusters). The label of the example images will be propagated to the top-ranked images in the result set. This iterative process is repeated until the user stops.

The primary contributions of this work are as follows:

- The MDS technique leverages user's RF to annotate images with the same semantic meanings but different visual characters precisely and efficiently.

- As the user's query can be hierarchically split into multiple sub-queries, the annotation speed increases in the new rounds.

- With the user's interaction, the annotations are propagated to the retrieved images near and on the query pathes. In this manner, the proposed technique can handle uncaptioned database with unlimited vocabularies effectively.

The remainder of this chapter is organized as follows. The related works is discussed in Section 5.2. Section 5.3 gives a system overview and explains the proposed Multi-Directional Search technique in detail in Section 5.4. The annotation mechanism is describe in Section 5.5. The experimental results are discussed in Section 5.6. Finally, Section 5.7 conclude the study.

5.2 Related Work

Traditional annotation researches focus on learning the mapping between images and keywords [5, 17]. For example, in [22] a translation model is proposed to tag image blobs. In [43], a cross-media relevance model is utilized to predict the probability of generating a word based on the the image blobs. The basic idea is for a given keyword, learn the most representative image region. In [36], multiple existing ontologies are used for semantic annotation and search in a collection of art images. The ontologies include the *Art and Architecture Thesaurus*, WordNet, ULAN and Iconclass.

Recent image annotation approaches are based on the image labels of an annotated database. In [80], a weighted nearest neighbor model is proposed to predict the term relevance of images by taking a weighted sum of the annotations of the visually most similar images in an annotated training set. In [53, 86], for example, images crawled from the Web are annotated with keywords extracted from their web page title and surrounding text; and they are used to build an initial annotated database. To label a new image subsequently, a

Content Based Image Retrieval (CBIR) mechanism is used. That is, the image to be labeled is used as a query image to retrieve similar images from the labeled database. The returned images' labels can then be used to determine the keywords for the new image. In [85], a guaranteed accurate keyword is used for text-based search. The result is a set of semantically similar images. Then a content-based retrieval is conducted on this set for visually similar images. Finally, annotations are extracted from the image surrounding texts, titles, and URLs. In [82], a probabilistic model is proposed to model the relationship in the image, its annotations, and its class label. A new image's annotation is determined by the classification result and the existing images' annotation will be propagated to the new image. In [95], for a given image, proper words are chosen from a vocabulary as tags. Then the tags are refined with the help of user's relevance feedback. For an given image, the user's feedback on the tags will be used for correcting the classifiers' outputs. Then in the next round, the system recommends more proper tags.

Most recently, near duplicate image search has become a popular research topic in the image retrieval and annotation community, since the majority of high quality images on the web may have plenty of (near) duplicates from different web pages. Similar to [53], the bounding texts of these duplicate images can be propagated to each other, after efficient filtering and/or clustering. Based on the classic visual vocabulary [62], some works even embed more spatial information in visual word [16, 87], or focus on semantic clues [57]. Some research studies [15] suggest that the user-generated semantic texts are usually more reliable than the visual features. Quack et al. [67] propose to leverage the surrounding texts with Wikipedia knowledge for geographical estimation. Yin et al. [89] further examine the geographical distribution of different topics in the world map.

To use the user-generated texts as the annotations for image, the the work in this chapter propose the use of RF when refine/propagate user's annotations to similar images. So, first of all, it is crucial to find a RF system that can give reliable result on image search. In [41], the relevant (feedback) images form a proximity area; and the centroid of this query contour

is used as the query point. In [65], the relevant images are grouped into clusters, and the k nearest images are those images closer to all the centroid of these clusters. In [39], a quadratic function is used to approximate the discontinued query contour. But still, the result images are in one neighborhood covered by the quadratic function. In [23], the query image's negative color image, black and white image, and the negative black and white image are used as queries in addition to the original query image. Although for technique like [23] expands the search range by considering multiple versions of the original query image, these different versions still share most of the visual features (e.g., shapes) and the effective search range is therefore limited to a single neighborhood.

The MDS technique remedies the above constrains by leveraging user's RF to annotate images with the same semantic meanings. As the user's query can be hierarchically split into multiple sub-queries, the semantically identical images can have different visual presentations. Also, the annotation speed increases in the new rounds. With the user's interaction, the annotations are propagated to the retrieved images near and on the query pathes. In this manner, the proposed technique can handle uncaptioned database with unlimited vocabularies effectively.

5.3 System Overview

The proposed MDS annotation process is depicted in Figure 5.1. The whole process starts from an unlabeled database. Upon presenting the randomly selected database images, the user selects relevant images and annotates them. Then local clustering (Multiple Direction Search, described in the next section) is performed on the relevant images. If there are more than one clusters resulted, the initial query is decomposed into multiple sub-queries, each represents a cluster. And the sub-query points are calculated as below: before retrieving relevant images to the user, the system will first revise the sub-queries according to the queries in the previous iteration to integrate the historical navigation information, and then

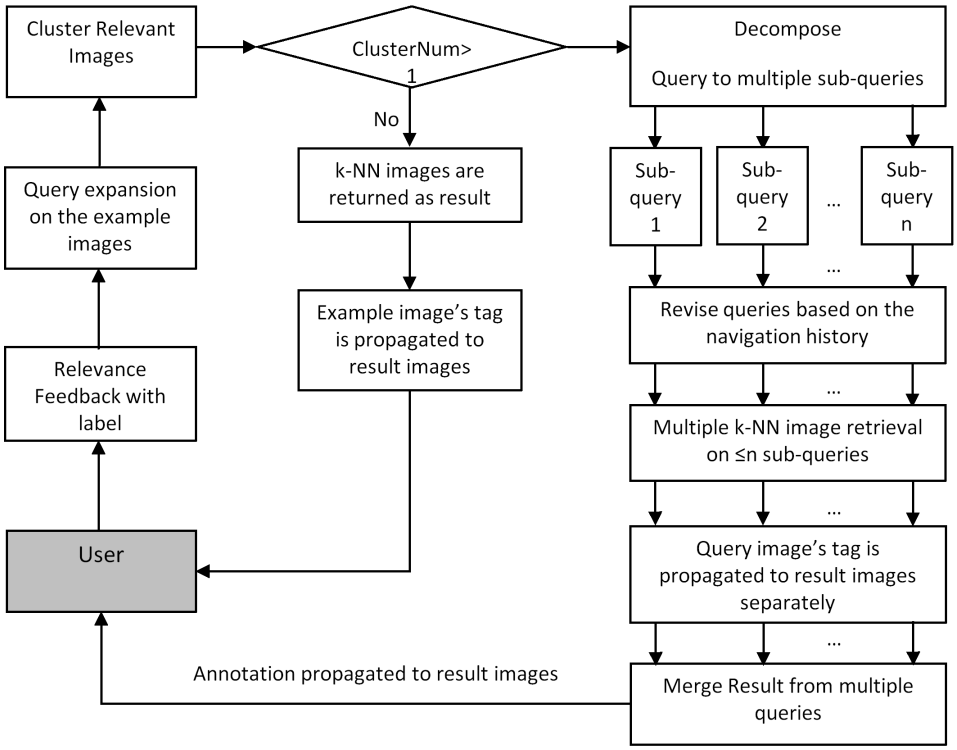


Figure 5.1: System Overview - Multi-Directional Search for Annotation Propagation.

check whether some of the sub-query points can be merged. If two query points are close to each other, they will be merged to form one sub-query. If no query is merged, the revised sub-queries are conducted separately, the result from those queries are combined and returned to the user (the details can be found in Section 5.4). Along with the retrieval process, the user can label different query images. As all the images retrieved during the process are similar to the corresponding sub-query, the query images' semantic labels are propagated to those images. The above procedure is repeated until the user is satisfied with the returned result. Ideally, all the images with similar semantic meaning will be labeled with the same annotation even if they are not close to each other in the feature space.

The proposed approach fits the mpeg-7 annotation standard which depends on standardized content-based descriptions: it captures the user's intention by clustering and decomposing the user's query to cover contents which has the same semantic concept but different visual features.

Similar to the works in earlier chapters, following content based image retrieval descriptors are extracted: 9 color moment features, 10 Wavelet-based texture features, and 18 edge-based structural features. While annotating the images, free text was used by users. Then the annotations are stored in XML files.

5.4 Multi-Directional Search (MDS)

This section describes the detail process of MDS: First, the user provides example images. Second, the system determines whether the images are diverse and should be process separately based on their visual features. If they are diverse, the dynamic directional clustering mechanism decomposes the user's initial query into multiple sub-queries, which is to split the annotation direction into multiple paths. Third, combining the user's previous behavior by exploiting the directional information from the previous rounds of relevance feedback, the multiple direction navigation scheme predicts the query points for the next round. Fourth, the system retrieves images to show to the user for further annotation. Along the relevance feedback iteration, the annotation propagation mechanism assigns user specified labels to the visually similar images.

5.4.1 Dynamic Directional Clustering

Suppose a user selects $X = \{x_1, x_2, ..., x_t\}$ as relevant images. Usually, the number of images selected by the user is relatively small compare to the visual features' high dimensionality. Therefore, to get stable clustering result, query expansion is conducted on X to get the extended image set $X' = \{x_1, x_2, ..., x_r\}$. X' includes the example images and their neighboring images. Then X' is clustered based on the image visual features using k-mean clustering. Each cluster is examined to determine the desired number of clusters. If the data points in a cluster does not appear Gaussian distributed, then this local cluster is split. The splitter

continues until all the clusters appear Gaussian [32]. Suppose there are K clusters and the objective function ϕ, which is the mapping of each image x_t to one of the K clusters:

$$\phi(x_t) = arg \min_{1 \leq k \leq K} d_2(c_k, x_t) \quad \text{(Eq. 5.1)}$$

where d_2 is the Euclidean distance for two d-dimensional images x_i and x_j,

$$d_2(x_i, x_j) = \sqrt{\sum_{m=1}^{d} (x_{im} - x_{jm})^2} \quad \text{(Eq. 5.2)}$$

n is the number of features, c_k is the center for cluster k, and c_{km} is the m^{th} feature of c_k.

Let π_k denotes a set of images that belongs to cluster k, the partitioning of X' into K disjoint clusters satisfies

$$\bigcup_{j=1}^{K} \pi_j = \{x_1, x_2, ..., x_r\}; \pi_j \cap \pi_l = \varnothing, if j \neq l \quad \text{(Eq. 5.3)}$$

According to above clustering mechanism, all the K clusters must be Gaussian distributed or will be split, the cluster number K is based on the relevant images' distribution in the feature space. If there are multiple clusters found, then that means the user selected images have quite different visual appearance. For the accuracy of the annotation, those image clusters should be processed separately. As the proposed system is a semi-automatic annotation system, it will help the user to annotate images that are similar to his/her example images in multiple iterations. Section 5.4.2 will discuss how to use the clustering result to navigate and help users annotate the candidate images.

During multiple iterations, the user's initial query might be hierarchically decomposed into multiple direction navigation. An example navigation process of a given relevance feedback session is shown as in Figure 5.2. The example in Figure 5.2 illustrates that three

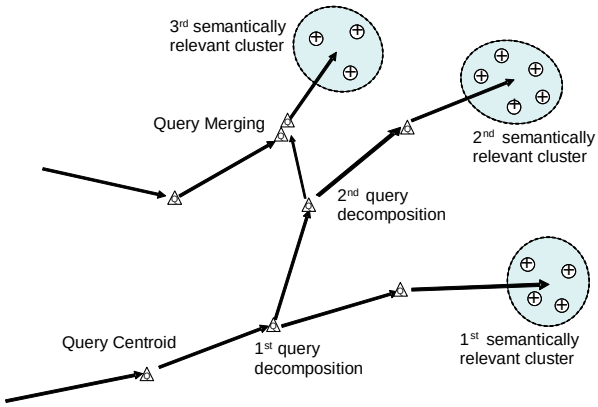


Figure 5.2: An example navigation process of a given relevance feedback session.

semantically similar clusters are discovered and finally three sub-queries are carried out, as a result of a sequence of relevance feedbacks.

5.4.2 Dynamic Directional Navigation

This section describes how to use the clustering result from the Dynamic Directional Clustering step to help user to propagate annotation to more images efficiently and accurately.

After the clustering, the mean vectors of each cluster are calculated. For each $1 \leq j \leq k$, the mean vector of the centroid of π_j is

$$m_j = \frac{\Sigma_{x \in \pi_j} x}{n_j} \qquad \text{(Eq. 5.4)}$$

where n_j is the number of image vectors in π_j.

In the author's earlier study [92], this centroid was used to retrieve similar images for user's further annotation. Considering the fact that during multiple feedback iterations, the user is modifying the query implicitly for his/her interest, therefore, the centroid in the previous rounds should also be taken into consideration. To be specific: After each feedback session, there is a mean vector for each of the clusters. As the feedback sessions proceed, a navigation path is formed in the feature space for each of the sub-queries. In

each navigation step, the user is expressing his/her intention implicitly about where to find the desired images in the feature space. Therefore, these dynamic directional information should be used and integrated with the previous session's information to update the query points. I.e.,after getting the new centroid of π_j, which is m_j, the new query points will be updated as follow:

$$p_j^{(i)} = [m_j + p_j^{(i-1)}]/2 \qquad \text{(Eq. 5.5)}$$

where $p_j^{(i)}$ is the predicted query point for the i^{th} RF round. For the first iteration, $p_j^{(1)} = m_j$.

Overall, when a query is decomposed, the new sub-queries' $p_j^{(i)}$ is calculated based on the ancestor query:

$$sub - queries_{j_1} : p_{j_1}^{(i)} = [m_{j_1} + p_j^{(i-1)}]/2 \qquad \text{(Eq. 5.6a)}$$

$$... \qquad \text{(Eq. 5.6b)}$$

$$sub - queries_{j_n} : p_{j_n}^{(i)} = [m_{j_n} + p_j^{(i-1)}]/2 \qquad \text{(Eq. 5.6c)}$$

The updated query point $p_{j_r}^{(i)}, 1 \leq r \leq n$, n is the number of sub-queries in the current round, reflects its new direction and also the historical direction information.

5.4.3 Sub-query Merge

While the RF processing continues, there will be more and more sub-queries. Some of them might converge to the same neighborhood. For example, in Figure 5.2, one of the sub-queries from the query path in the upper left of the figure moves to the nearby place of the sub-query to the 3rd image cluster. In this case, if the two sub-queries are close enough, they will be merged to form one sub-query. To be specific, suppose in the current iteration, there are two sub-queries: q_1 and q_2. If they are close to each other, that means they are exploring the same feature space and the system will combine these two queries to one: If

$$d_2(q_1, q_2) < \varepsilon, \qquad \text{(Eq. 5.7)}$$

then $q_{new} = (q_1 + q_2)/2$. Where ε is a small value, q_{new} is the new query point that replaces q_1 and q_2.

If q_1 was in a previous session, q_2 is in the current iteration, and

$$d_2(q_1, q_2) < \varepsilon, \qquad \text{(Eq. 5.8)}$$

then q_2 is ignored (the sub-query is not computed and do not return any result).

5.5 Annotation Propagation with MDS

This section describes how the user specified annotations are propagated to images in the database. By utilizing the MDS technique, the user can navigate into multiple image neighborhood according to his/her interest. Along those navigation paths, the annotations of the example images are propagated.

5.5.1 Annotation Propagation

At the beginning of each RF session, the user selects images of interests and provide the annotation of these images. Those images are used as query for image retrieval. After the query expansion, each image in X' gets the same label of the according example image. Based on the technique described above, the dynamic clustering result is combined with the history query points to predict the new query points, $p_j^{(i)}$. From now on, $p_j^{(i)}$ works as a seed to retrieve more similar images. CBIR is conducted on $p_j^{(i)}$ to retrieve multiple groups of images. Finally, each images in those sets gets the same label as the according $p_j^{(i)}$. Meanwhile, those retrieved images are shown to the user for further annotation and relevance

feedback. With the user's input, the above steps repeats. Therefore, the user specified label is propagated to the database images.

In the propagation process, one image can be returned to the user in different queries, and therefore one image can be assigned with many different annotations. For the convenience of indexing and storage, for each image, a certain number of annotations will be kept. And those annotations should be the ones that are visually closest to the query images that propagate the annotation to the image. In specific, following criteria is used to determine which annotations to keep:

Suppose keeping n annotations for each image in the system, for image x_l if there are n annotations, $A_{x_l}^{(i-1)} = \{a_1, a_2, ..., a_n\}$, at round *(i-1)*. In the new query session, x_l is retrieved as result of query x_q which has label a_q, where $a_q \notin A_{x_l}^{(i-1)}$. Define the distance from x_q to x_l as *Annotation Distance* ($AnnoDist_q$) and compare it with $AnnoDist_r$, where $1 \leq r \leq n$. $AnnoDist_i$ is the distance between x_l and the query image that propagates a_i to it. If $AnnoDist_q < max(AnnoDist_1, AnnoDist_2, ..., AnnoDist_n)$, then $A_{x_l}^{(i)} = \{a_1, a_2, ..., a_n\} - a_s + a_q$, where a_s has the largest *AnnoDist*. Above criteria will keep the "closest" annotations for each image.

Figure 5.3 shows an example of how the proposed MDS annotation system propagate user's example labels. At the first round, a user was looking for "animal" images. This user selected two images as examples and gave "animal" as the label to those two images. The system first propagated the annotation "animal" to all the neighboring images (covered by the big oval in Figure 5.3). Then it clustered those images. As there were two clusters, the initial query was decomposed into two sub-queries. As the process went on, the user annotated one of the returned images as "eagle" and another one as "horse". The system again propagated those annotations to the sub-queries' neighboring images separately (as shown in Figure 5.3: "eagle to the left group of images", "horse" to the right group of images).

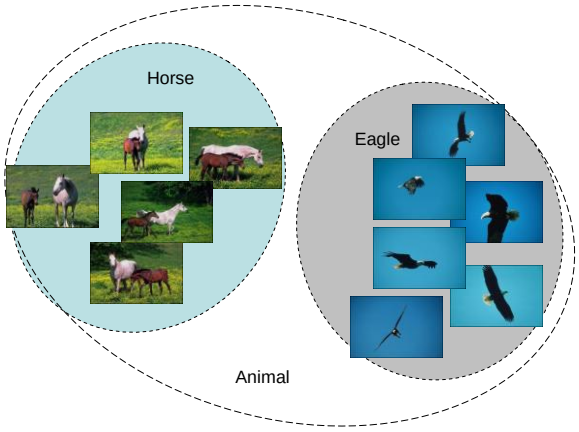


Figure 5.3: An example of MDS with annotation propagation

5.5.2 Annotation Algorithm

Based on the above descriptions of the MDS, the interactive annotation propagation algorithm is presented as below.

Algorithm 5.1 Annotation propagation in MDS

INPUT: features F, DB, *user's RF*
OUTPUT: Annotations on the query's similar images
System presents randomly selected images;
while User want to provide more input **do**
  User selects relevant images and annotate them with labels L;
  Cluster the relevant images;
  if cluster number > 1 **then**
    Decompose query into multiple sub-queries;
    Calculate $p_j^{(i)}$ as new query point;
    if Two sub-queries are close **then**
      Merge them to form one sub-query;
    end if
    if One sub-query is close to a previous sub-query **then**
      Ignore the current sub-query;
    end if
  end if
  Retrieve kNN for each of the $p_j^{(i)}$;
  Assign annotations L to the returned images;
  Present the top images from each kNN set to the user;
end while

Upon presenting the initial query, the user starts an interactive retrieval session to search for relevant images. Clustering is performed on the example images. At the same time, the query images' annotations are propagated to the retrieved images on the navigation path. If there are more than one clusters, then the initial query is decomposed into multiple queries and updated by integrating the predicted query points according to the navigation history; in the next step, separate queries are conducted, the result from those queries are combined and returned for user's relevance feedback; annotations are propagated to those returned images. The procedure is repeated when the user wants to provide more example images and annotations. Although MDS has been proposed as an image annotation technique, it is obvious that it can also be used for CBIR without the annotations.

5.6 Experimental Study

This section discusses the experimental study on the proposed MDS annotation propagation system. The system's accuracy is evaluated in terms of precision and recall; the systerm's efficiency on propagating annotations is also observed.

5.6.1 Annotation Evaluation Metrics

The following two criteria were used to evaluate the annotation accuracy - annotation precision (p_a) and recall (r_a)

$$p_a = \frac{1}{n}\sum_{k=1}^{n}\frac{num\ of\ correctly\ annotated\ words\ in\ I_k}{num\ of\ annotated\ words\ in\ I_k} \qquad \text{(Eq. 5.9)}$$

$$r_a = \frac{1}{n}\sum_{k=1}^{n}\frac{num\ of\ correctly\ annotated\ words\ in\ I_k}{num\ of\ ground\ truth\ words\ in\ I_k} \qquad \text{(Eq. 5.10)}$$

5.6.2 Model Comparison

The proposed MDS annotation technique is compared with two models: the Machine Translation model (MT) [22] and the Multi-Instance Learning model (MIL) [88]. The same dataset is used by these models, which is a Corel database including 5,000 images and 371 keywords. The same features as [22,88] are used so that the three algorithms have the same input. The experiment used 500 images as query and the remaining 4,500 images as target as if they had no annotation. The partitioning is consistent with [88]. Because the images are not available, in MDS, the 4,500 images' blobs ground truth annotations were returned to the user for relevance feedback. If an image was selected, the new annotation was assigned to it as the whole image's annotation. The average per-word precision and recall is reported in Table 5.1 for the best 49 keywords as [88] did. Table 5.1 shows that the MDS approach has higher precision and recall than MT and MIL. The reason is that MDS can propagate annotations to images with different visual features but identical semantic meanings. That highly enhanced the annotation accuracy.

Table 5.1: Performance Comparisons

Model	MT	MIL	MDS
Average precision	0.20	0.31	0.32
Average recall	0.34	0.46	0.49

5.6.3 Annotation Accuracy

In the proposed MDS system, the database starts from images without any annotations, as the relevance feedback interactions proceed, annotations are propagated to the similar images in the database. The ultimate goal of this technique is to annotate database images and provide better search service on text queries to users. Therefore, it is important to guarantee the annotation quality. This section will discuss the proposed system's annotation accuracy.

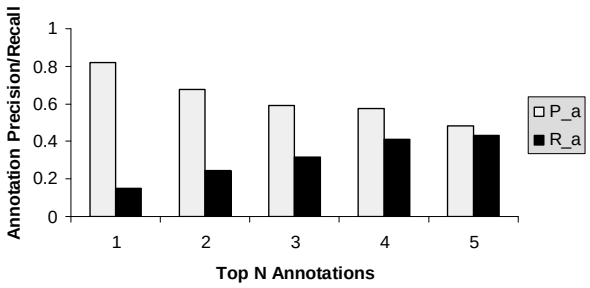


Figure 5.4: Annotation Precision and Recall for the MDS system.

5.6.3.1 Experiment on UW dataset

First, the experiment results on the UW database [1] are presented. These images have about 5 manually labeled ground truth annotations; not all objects are annotated. The annotation accuracy is evaluated by comparing the result image labels to the ground truth annotations provided in the database. To evaluate the proposed system, 10 subjects were asked to perform the relevance feedback image annotation task on the database as follow: Firstly, the subject selected images in the UW database and a corresponding label from the UW annotation list as the initial query. Then the MDS system returned some similar images for user's further annotation. At the meantime, the query image's label were propagated to these returned images. The RF process continued if the subject was willing to.

Figure 5.4 presents the annotation precision-recall vs. top N annotations for the UW dataset. Since the annotation methods is based on the relevance feedback image retrieval model, there might be some images "get less attention" than the others and assigned fewer annotations. Therefore, in the plot, when N varies, only those images with enough number of annotations are considered.

It can be seen that when considering the top 5 annotations of the image, p_a=0.48, r_a=0.43. Because exact annotation matching is used on calculating p_a and r_a, some keywords like "Geneva" and "Switzerland" are considered not matching, so the result could be better.

Also, with more user using the system, the annotations will be more accurate based on the mechanism presented in Section 5.5.1. I.e., the most closest annotations are kept.

5.6.3.2 Experiment on Corel database

This experiment study used another Corel image data set to test the proposed system's annotation accuracy. This Corel database includes 15,000 images for image annotation. The images are from 150 categories, the category information is used as ground truth. Again, from each image, 37 features are extracted and categorized into three groups: 9 color moment features, 10 Wavelet-based texture features, and 18 edge-based structural features. These features have been shown to be effective in supporting image retrieval systems [38].

After every image in the database is annotated, image retrieval by text queries is conducted. The retrieved images' annotations are compared with the ground truth information. Figure 5.5 reports the average precision and recall rate on 200 queries. Because the Corel images only have one category name as their annotation, in this test, when calculating precision and recall, if an image's annotation describes an object in the image, it is counted as correct. Figure 5.5 shows that the annotation accuracy of the MDS system is very high on the top annotations. Note that the recall is not high at the precision and recall by only considering the first annotation. The reason is that sometimes even two images' visual features are similar, but their semantic meaning will be different. Giving an example: an image with skyscraper can has a ground truth annotation as "New York". If a user selected this image, then some other images who has skyscraper could also get the annotation "New York". But those images are not New York images. Therefore, the annotation "New York" to those images will be incorrect.

To show the result of the proposed annotation mechanism, some of the example outputs (images and their annotations from the experiments) are also presented in Figure 5.6. The

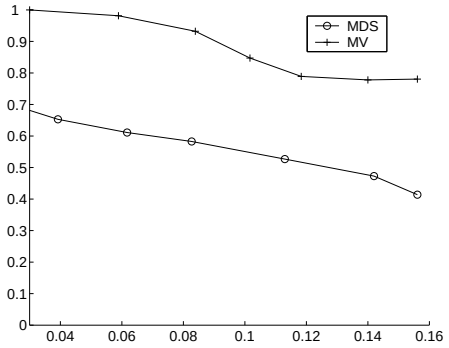


Figure 5.5: Precision and Recall for Image Retrieval using Text Query in the MDS system.

				
Mountain Surf_up Hairstyle	**Rose** Flower Garden	**Fitness** Kungfu Bird_art	**Dino_art** Bird_art Plane	**Pyramid** Ruin Horse

Figure 5.6: Some example images with labels in MDS system

keywords in bold are the labels with the highest votes, followed by more labels sorted in descend order according to their votes.

As MDS is capable of decomposing user's initial query into multiple sub-queries for better coverage, the proposed system is able to find the top k images without the confinement of one neighborhood, therefore, provides the flexibility for the user to provide detail annotations. An example of annotating "rose" is shown in Figure 5.7. The two rows show the retrieval result from two sub-queries in the same session. And the user can annotate those images as "roses" and "rose".

5.6.4 Annotation Efficiency

Because MDS considers multiple clusters in each query session, so the annotations are propagated to multiple kNNs, which is much faster than just one kNN. Figure 5.8 plots the image percentage in the database that has at least one label on it vs. the number of user iterations.

Figure 5.7: "Rose" from the MDS annotation system (result of "single rose" in the top row, and "multiple roses" in the bottom row)

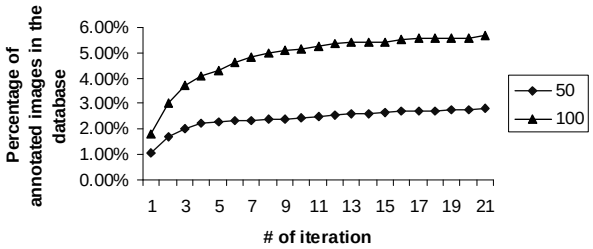


Figure 5.8: Annotation propagation speed in MDS system

In each query session, the system propagates image label to 50 and 100 images. In each iteration, 3 relevant images are annotated. In the first iterations, the number of images annotated increased faster than in the later iterations. This is because the result converges at the last iterations, therefore fewer new images will be included in the retrieval set and hence get annotations. The experiment results indicate that after about 100 independent queries, each with 4 iterations of RF, 90% of the images in the database are assigned at least one annotation.

5.6.5 Image Retrieval Performance

Because the proposed system is based on the content based image retrieval schema to find similar images, so it is important to show that this system is accurate on the image retrieval task. As describe previously, the current image retrieval models suffer the confinement of kNN computation, to provide better coverage for annotation propagation, the new image retrieval technique must be able to handle the case when the semantically similar top k

images are far apart in the feature space. To test the system's capability of bridging the semantic gap in image retrieval, it is compared with the Multiple Viewpoints (MV) [23] approach. In MV approach, the query image's negative color image, black and white image, and the negative black and white image are used as queries in addition to the original query image. It decomposes query to sub-queries based on color feature and try to cover more images with the same semantic concept but different color representations. The proposed approach decomposes query based on the clustering result on all the visual features.

To facilitate a fair comparison, both the MDS and MV techniques were evaluated using the same queries, and the same image data set with the same features. Both techniques retrieved the same number of images for each test query. For the MV approach, the images returned by the four color channels were combined to form the final result set.

5.6.5.1 Bridge the semantic gap

A total of 100 queries are conducted on both systems. The example queries are listed in Table 5.2. Table 5.3 reports the average. Another metric is used here to evaluate the two system's performance on bridging the semantic gap - Ground Truth Inclusion Ratio:

$$GTIR = \frac{number\ of\ retrieved\ relevant\ subconcepts}{number\ of\ relevant\ subconcepts\ in\ the\ database} \qquad \text{(Eq. 5.11)}$$

Table 5.2: Table (Testing Query)

Query	Sub-concept
A person	Hair-model, fitness, Kongfu
Airplane	single, multiple
Bird	eagle, owl, sparrow
Car	modern sedan, antique car, steamed car
Horse	polo, wild horse, race
Mountain	snow, with water
Rose	single, multiple
Water Sports	surfing, sailing

For most of the queries, MDS is able to capture all the sub-concepts, resulting GTIR=1; while MV only capture part of the sub-concept(s). For the query "rose", the MV could also capture both the sub-concepts. The reason is that in addition to the original query image, MV also used color negative, black and white, and negative black and white images to query the database. The dominant feature of rose images is color, so MV could pick the images from different sub-concepts if they have similar color features. But those additional color queries also brought irrelevant images which causes a lower precision than the MDS. Overall, the proposed MDS system better bridges the semantic gap by using multiple sub-queries. It is able to cover the semantically similar images even they are in multiple clusters in the feature space. To better illustrate the idea of the query decomposition technique, refer to Figure 5.7.

5.7 Conclusion

The MDS approach dynamically analyzes the user's relevance feedback and considers multiple neighborhoods by decomposing initial query into separate sub-queries. As the sub-queries covers distinct image clusters scattered in the feature space, MDS addresses the inherent weakness of kNN based RF techniques - confining the search in one neighboring image cluster. Therefore, MDS can simultaneously annotate images with different visual characteristics but the same semantic concept. The MDS techniques also considers user's previous query intention to better exploit the intent of user feedback, which greatly improved the search and annotation efficiency. The experimental studies showed that the proposed technique helps user annotate large uncaptioned database efficiently and accurately.

Table 5.3: Image Retrieval Quality Comparison

MV		MDS	
GTIR	Precision	GTIR	Precision
0.56	0.32	1	0.65

REFERENCES

[1] http://www.cs.washington.edu/research/imagedatabase/groundtruth/.

[2] http://www.mysql.com.

[3] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, pages 69–84. Springer Verlag, 1993.

[4] Jrgen Assfalg, Alberto Del Bimbo, and Pietro Pala. Three-dimensional interfaces for querying by example in content-based. *IEEE TRANS. VISUALIZATION AND COMPUTER GRAPHICS*, 8:2002, 2002.

[5] Kobus Barnard, Pinar Duygulu, David Forsyth, and Nando De Freitas. Matching words and pictures. *Journal of Machine Learning Research*, 3(2003):1107–1135, 2003.

[6] Norbert Beckmann, Hans peter Begel, Ralf Schneider, and Bernhard Seeger. The r*-tree: an efficient and robust access method for points and rectangles. In *Proceeding of SIGMOD*, pages 322–331. Springer Verlag, 1990.

[7] F. Bentley, C. Metcalf, and G. Harboe. Personal vs. commercial content: the similarities between consumer use of photos and music. In *Proceedings of International Conference on Human Factors in Computing Systems*, pages 667–676, 2006.

[8] Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. The x-tree: An index structure for high-dimensional data. In *Proceedings of the 22th International Conference on Very Large Data Bases*, pages 28–39, 1996.

[9] F. N. Bezerra, E. Werbet, and W.B. Silva. Client-side content-based refinement for image search in the web. In *Proceedings of Brazilian Symposium on Multimedia and the web*, pages 1–3, 2005.

[10] M. O. Binderberger and S. Mehrotra. Relevance feedback techniques in the mars image retrieval system. *Multimedia System*, 9(6):535–547, 2004.

[11] David M. Blei, Michael I, David M. Blei, and Michael I. Modeling annotated data. In *Proceedings of the 26th Intl. ACM SIGIR Conference*, pages 127–134, 2003.

[12] J. Bourgain. On lipschitz embedding of finite metric spaces into hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[13] S. Brewster. Overcoming the lack of screen spaces on mobile computers. *Personal and Ubiquitous Computing*, 6(3):188–205, 2002.

[14] Deng Cai, Xiaofei He, and Jiawei Han. Spectral regression: A unified subspace learning framework for content-based image retrieval. In *Proceedings ACM Conf. Multimedia*, pages 403–412, 2007.

[15] Liangliang Cao, Jie Yu, Jiebo Luo, and Thomas S. Huang. Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 125–134, 2009.

[16] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. In *Proceedings of CVPR*, pages 3352–3359, 2010.

[17] Gustavo Carneiro. A database centric view of semantic image annotation and retrieval. In *Proceedings of the 28th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, pages 559–566, 2005.

[18] Kaushik Chakrabarti, Eamonn Keogh, Sharad Mehrotra, and Michael Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. In *proceedings of ACM SIGMOD Conference on Management of Data*, pages 151–162, 2002.

[19] X. Chen, C. Zhang, S.-C. Chen, and S. H. Rubin. A human-centered multiple instance learning framework for semantic video retrieval. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(2):228–233, 2009.

[20] Ritendra Datta, Dhiraj Joshi, Jia Li, James, and Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 39:2007, 2006.

[21] M. Davis, S. King, N. Good, and R. Sarvas. From context to content: Leveraging context to infer media metadata. In *Proceedings of ACM Multimedia*, pages 188–195, 2004.

[22] Pinar Duygulu, Kobus Barnard, Nando de Freitas, P. Duygulu, K. Barnard, and David Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of European Conference on Computer Vision*, pages 97–112, 2002.

[23] James C. French, Xiangyu Jin, and W. N. Martin. W.: An empirical investigation of the scalability of a multiple viewpoint cbir system. In *Proceedings of CIVR*, pages 252–260, 2004.

[24] Y. Fu, Z. Li, T. S. Huang, and A. K. Katsaggelos. An empirical investigation of the scalability of a multiple viewpoint cbir system. In *Proceedings of CIVR*, pages 252–260, 2004.

[25] Shih fu Chang. Compressed-domain content-based image and video retrieval. In *Proc Symposium on Multimedia Communications and Video Coding*, pages 1–8, 1995.

[26] Shih fu Chang, John R. Smith, and Jianhao Meng. Efficient techniques for feature-based image/video access and manipulation. In *Proceedings, 33 rd Annual Clinic on Library Applications of Data Processing: Digital Image Access and Retrieval*, pages 86–99, 1996.

[27] T. Gevers and A. Smeulders. Content-based image retrieval: An overview. *Emerging Topics in Computer Vision*, 2004.

[28] K. Goh, E. Chang, and W.-C. Lai. Concept-dependent multimodal active learning for image retrieval. In *Proceedings of ACM Multimedia*, pages 564–571, 2004.

[29] Dina Goldin and Paris C Kanellakis. On similarity queries for time-series data: Constraint specification and implementation. In *Proceedings of the First International Conference on Principles and Practice of Constraint Programming*, pages 137–153, 1995.

[30] Guo-Dong Guo, Anil K. Jain, Wei-Ying Ma, and Hong-Jiang Zhang. Learning similarity measure for natural image retrieval with relevance feedback. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 13(4):811–820, 2002.

[31] Zhen Guo, Z. Zhang, E. Xing, and C. Faloutsos. A max margin framework on image annotation and multimodal image retrieval. In *Proceedings of ICME*, 2007.

[32] Greg Hamerly and Charles Elkan. Learning the k in k-means. In *Proceedings of NIPS*, page 2003, 2003.

[33] Xiaofei He, Wei ying Ma, and Hong jiang Zhang. Learning an image manifold for retrieval. In *Proceedings ACM Multimedia*, pages 17–23, 2004.

[34] Nualsawat Hiransakolwong, Kien A. Hua, and Yao H. Ho. Asia: An automatic annotation technique for querycby-concept in image retrieval systems, 2003.

[35] Steven C. H. Hoi, Michael R. Lyu, and Rong Jin. A unified log-based relevance feedback scheme for image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):509–524, 2006.

[36] Laura Hollink, Guus Schreiber, Jan Wielemaker, and Bob Wielinga. Semantic annotation of image collections. In *In Workshop on Knowledge Markup and Semantic Annotation*, pages 1–3, 2003.

[37] Kien A. Hua, Khanh Vu, and Jung-Hwan Oh. Sammatch: A flexible and efficient sampling-based image retrieval technique for large image databases, 1999.

[38] Kien A. Hua, Ning Yu, and Danzhou Liu. Query decomposition: A multiple neighborhood approach to relevance feedback in content-based image retrieval. In *Proceedings of the 22nd International Conference on Data Engineering*, pages 84–95, 2006.

[39] Deok hwan Kim. Qcluster: Relevance feedback using adaptive clustering for content-based image retrieval. In *Proceedings of the ACM SIGMOD Int. Conf. on Management of Data*, pages 599–610, 2003.

[40] Hyung il Koo and Nam Ik Cho. A relevance feedback algorithm based on the clustering and parzen window. In *ICIP (2)'03*, pages 551–554, 2003.

[41] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. Mindreader: Querying databases through multiple examples, 1998.

[42] A K Jain, M N Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3), 1999.

[43] J. Jeon and R. Manmatha. Automatic image annotation of news images with large vocabularies and low quality training data. In *Proceedings of ACM Multimedia*, number 368-375, 2004.

[44] Sangoh Jeong, Chee Sun Won, and Robert M. Gray. Image retrieval using color histograms generated by gauss mixture vector quantization, 2004.

[45] M. Jia, X. Fan, X. Xie, M. J. Li, and W. Y. Ma. Photo-to-search: Using camera phones to inquire of the surrounding world. In *Proceedings of Mobile Data Management*, pages 46–48, 2006.

[46] K.V. Ravi Kanth, Divyakant Agrawal, Amr El Abbadi, K. V. Ravi, Kanth Divyakant, Agrawal Amr, El Abbadi, and Ambuj Singh. Dimensionality reduction for similarity searching in dynamic databases, 1998.

[47] Suckchul Kim, Yoonsik Tak, Yunyoung Nam, and Eenjun Hwang. mclover: mobile content-based leaf image retrieval system. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 215–216, 2005.

[48] Janne Lahti, Utz Westermann, Marko Palola, Johannes Peltola, and Elena Vildjiounaite. Mobicon: integrated capture, annotation, and sharing of video clips with mobile phones. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 798–799, 2005.

[49] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Proceedings of NIPS*, pages 553–560, 2003.

[50] Jeannie S. A. Lee and Nikil Jayant. Mixed-initiative multimedia for mobile devices: a voting-based user interface for news videos. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 611–614, 2006.

[51] Yi leh Wu, Divyakant Agrawal, and Amr El Abbadi. A comparison of dft and dwt based similarity search in time-series databases. In *Proceedings of the 9 th International Conference on Information and Knowledge Management*, pages 488–495, 2000.

[52] Beitao Li, Kingshy Goh, and Edward Y. Chang. Confidence-based dynamic ensemble for image annotation and semantics discovery, 2003.

[53] Xirong Li, Le Chen, Lei Zhang, Fuzong Lin, and Wei ying Ma. Image annotation by large-scale content-based image retrieval. In *Proceedings of the 14th Annual ACM international Conference on Multimedia*, pages 27–30, 2006.

[54] Yen-Yu Lin, Tyng-Luh Liu, and Hwann-Tzong Chen. Semantic manifold learning for image retrieval. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 249–258, 2005.

[55] Danzhou Liu, Kien A. Hua, Khanh Vu, and Ning Yu. Fast query point movement techniques with relevance feedback for content-based image retrieval. In *Proceedings of International Conference on Extending Database Technology*, pages 700–717, 2006.

[56] Xiaotao Liu, M. Corner, and P. Shenoy. Seva: Sensor-enhanced video annotation. In *Proceedings of the 13th ACM Annual Conference on Multimedia (MM05), Singapore*, pages 618–627, 2005.

[57] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Supervised dictionary learning. In *Proceedings of NIPS*, pages 1033–1040, 2008.

[58] M. K. Mandal, F. Idris, and S. Panchanathan. A critical evaluation of image and video indexing techniques in the compressed domain. *IMAGE AND VISION COMPUTING*, 17:513–529, 1999.

[59] Iqbal Mohomed, Jim Chengming Cai, Sina Chavoshi, and Eyal de Lara. Context-aware interactive content adaptation, 2006.

[60] Ljubomir Manola Munehiro Nakazato and Thomas S. Huang. Imagegrouper: a group-oriented user interface for content-based image retrieval and digital image arrangement. *Journal of Visual Languages and Computing*, 14:363–386, 2003.

[61] Apostol Natsev, Rajeev Rastogi, and Kyuseok Shim. Walrus: A similarity retrieval algorithm for image databases. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 395–406, 1999.

[62] David Nistr and Henrik Stewnius. Scalable recognition with a vocabulary tree. In *Proceedings of CVPR*, pages 2161–2168, 2006.

[63] Navneet Panda, Kingshy Goh, and Edward Y. Chang. Active learning in very large databases. *Multimedia Tools and Applications*, 31(3):249–267, 2006.

[64] Jing Peng and Douglas R. Heisterkamp. Kernel indexing for relevance feedback image retrieval. In *Proceedings of the IEEE International Conference on Image Processing (ICIP03*, 2003.

[65] Kriengkrai Porkaew, Kaushik Chakrabarti, and Sharad Mehrotra. Query refinement for multimedia similarity retrieval in mars. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 235–238, 1999.

[66] G. Pottie and W. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43:51–58, 2000.

[67] Till Quack, Bastian Leibe, and Luc Van Gool. World-scale mining of objects and events from community photo collections, 2008.

[68] C. J. Van Rijsbergen. *Information Retrieval*. 1979.

[69] Yong Rui and Thomas Huang. Optimizing learning in image retrieval. In *Proceedings IEEE ICCVPR*, pages 236–243, 2000.

[70] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Content-based image retrieval with relevance feedback in mars. In *Proceedings IEEE Int. Conf. on Image Proc*, pages 815–818, 1997.

[71] Risto Sarvas, Mikko Viikari, Juha Pesonen, and Hanno Nevanlinna. Mobshare: Controlled and immediate sharing of mobile images. In *Proceedings of Multimedia 2004*, pages 10–16, 2004.

[72] Mei-Ling Shyu, Shu-Ching Chen, Min Chen, Chengcui Zhang, and Chi-Min Shu. Probabilistic semantic network-based image retrieval using mmm and relevance feedback. *Multimedia Tools Appl.*, 30:131–147, 2006.

[73] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1349–1380, 2000.

[74] J. R. Smith and S.-F. Chang. Transform features for texture classification and discrimination in large mage databases. In *Proceedings of IEEE International Conference on Image Processing*, pages 407–411, 1994.

[75] H. Sonobe, S. Takagi, and F. Yoshimoto. Image retrieval system of fishes using a mobile device. In *Proceedings of International Workshop on Advanced Image Technology*, pages 33–37, 2004.

[76] Mark Stemm, Randy H. Katz, and Y H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transaction on Communication*, E80-B:1125–1131, 1997.

[77] Markus Stricker and Markus Orengo. Similarity of color images. In *Proceedings of SPIE 2420,381*, pages 381–392, 1995.

[78] Dacheng Tao, Xiaoou Tang, Xuelong Li, and Xindong Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1088–1099, 2006.

[79] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, 2001.

[80] Jakob Verbeek and Thomas Mensink. Image annotation with tagprop on the mirflickr set. In *ACM Multimedia Information Retrieval*, pages 537–546, 2010.

[81] Khanh Vu, Kien A. Hua, Hao Cheng, and Sheau dong Lang. A non-linear dimensionality-reduction technique for fast similarity search in large databases. In *Proceedings of the 2006 SIGMOD Conf*, pages 527–538, 2006.

[82] Chong Wang, David Blei, and Li Fei-fei. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition*, pages 1903–1910, 2009.

[83] James Z. Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.

[84] Wei Wang and Aidong Zhang. Extracting semantic concepts from images: a decisive feature pattern mining approach. *Multimedia Syst.*, 11(4):352–366, 2006.

[85] Xin-Jing Wang, Lei Zhang, Feng Jing, and Wei-Ying Ma. Annosearch: Image auto-annotation by search. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, pages 1483–1490, 2006.

[86] Xin-Jing Wang, Lei Zhang, Ming Liu, Yi Li, and Wei-Ying Ma. Arista - image search to annotation on billions of web photos. In *CVPR*, pages 2987–2994, 2010.

[87] Z. Wu, Q. Ke, and J. Sun. Bundling features for large-scale partial-duplicate web image search. In *Proceedings of CVPR*, pages 25–32, 2009.

[88] Changbo Yang, Ming Dong, and Farshad Fotouhi. Region based image annotation through multiple-instance learning. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 435–438, 2005.

[89] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In *Proceedings of the 20th international conference on World wide web*, pages 247–256, 2011.

[90] J. Yoon and N. Jayant. Relevance feedback for semantics based image retrieval. In *Proceedings of ICIP*, pages 42–45, 2001.

[91] Jie Yu, Yijuan Lu, Yuning Xu, Nicu Sebe, and Qi Tian. Integrating relevance feedback in boosting for content-based image retrieval. In *Proceedings of ICASSP*, 2007.

[92] Ning Yu, Kien A. Hua, and Hao Cheng. Dynamic directional navigation in content-based image retrieval. In *Proceedings of the 2008 IEEE International Conference on Multimedia and Expo*, pages 1253–1256, 2008.

[93] Ning Yu, Kien A. Hua, and Danzhou Liu. Client-side relevance feedback approach for image retrieval in mobile environment. In *Proceedings of ICME*, pages 552–555, 2007.

[94] Lei Zhang, Fuzong Lin, and Bo Zhang. Support vector machine learning for image retrieval. In *Proceedings IEEE Int. Conf. on Image Processing*, pages 721–724, 2001.

[95] Shile Zhang, Bin Li, and Xiangyang Xue. Semi-automatic dynamic auxiliary-tag-aided image annotation. *Pattern Recognition*, 43(2):470–477, 2010.

[96] Rong Zhao and William I. Grosky. Narrowing the semantic gap - improved text-based web document retrieval using visual features. *IEEE Transactions on Multimedia*, 4:189–200, 2002.

[97] Fengzhou Zheng, Nitin Garg, Sumeet Sobti, and Chi Zhang. Considering the energy consumption of mobile storage alternatives. In *Proceedings of International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems*, pages 36–45, 2003.

[98] Xiang Sean Zhou and Thomas S. Huang. Edge-based structural features for content-based image retrieval. *Pattern Recognition Letters*, 22:457–468, 2001.