12-2013

Effective Graph-Based Content–Based Image Retrieval Systems for Large-Scale and Small-Scale Image Databases

Ran Chang

Utah State University

EFFECTIVE GRAPH-BASED CONTENT-BASED IMAGE RETRIEVAL SYSTEMS FOR LARGE-SCALE AND SMALL-SCALE IMAGE DATABASES

by

Ran Chang

A dissertation submitted in partial fulfillment
of the requirements for the degree
of

DOCTOR OF PHILOSOPHY

in

Computer Science

Approved:

Dr. Xiaojun Qi Major Professor	Dr. Stephen Clyde Committee Member
Dr. Kyumin Lee Committee Member	Dr. Vicki Allan Committee Member
Dr. Yangquan Chen Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNVIVERSITY
Logan, Utah

2014

ABSTRACT

Effective Graph-based Content-Based Image Retrieval Systems for Large-Scale And Small-Scale Image Databases

by

Ran Chang, Doctor of Philosophy

Utah State University, 2014

Major Professor: Dr. Xiaojun Qi
Department: Computer Science

This dissertation proposes two novel manifold graph-based ranking systems for Content-Based Image Retrieval (CBIR). The two proposed systems exploit the synergism between relevance feedback-based transductive short-term learning and semantic feature-based long-term learning to improve retrieval performance. Proposed systems first apply the active learning mechanism to construct users' relevance feedback log and extract high-level semantic features for each image. These systems then create manifold graphs by incorporating both the low-level visual similarity and the high-level semantic similarity to achieve more meaningful structures for the image space. Finally, asymmetric relevance vectors are created to propagate relevance scores of labeled images to unlabeled images via manifold graphs. The extensive experimental results demonstrate two proposed systems outperform the other state-of-the-art CBIR systems in the context of both correct and erroneous users' feedback.

(144 pages)

PUBLIC ABSTRACT

Effective Graph-based Content-Based Image Retrieval Systems for Large-Scale And Small-Scale Image Databases

Digital imaging was a great invention in the last century. Since digital cameras became popular in the public, a large amount of digital images emerged in the late of the twentieth century. How to manage the huge amount of images and find desired images among them became an urgent issue during the same period.

Techniques of retrieving a desired image are generally categorized into two basic classes. One relies on text-based key words to retrieve desired images in the image database. The other one relies on image-based queries to retrieve desired images in the image database. The second technique is usually named the content-based image retrieval technique. Major techniques involved in the content-based image retrieval technique include the image feature extraction, the feature matching algorithm, and the similarity calculation. Each technique plays an important role in the content-based image retrieval, and they have their own challenge issues as well. For instance, how to find an efficient and accurate feature matching algorithm is still a hot topic in the content-based image retrieval.

This dissertation addresses certain challenge issues that exist in the content-based image retrieval technique and proposes two different retrieval systems that can be applied in the small-scale and the large-scale image databases.

Ran Chang

This work is dedicated to

my families, Qiao Xu,

Lingmin Chang and Shaohua Lu,

Zhi Xu and Lianrong Yao

ACKNOWLEDGMENTS

I appreciate Dr. Xiaojun Qi for her detailed direction and encouragements for the research in this dissertation. I would like to thank my committee members, Dr. Vicki Allan, Dr. Stephen Clyde, Dr. Kyumin Lee, Dr. Yanquan Chen and previous committee members, Dr. Dan Watson and Dr. Changhui Yan.

I cordially express my appreciation to my family, friends, and colleagues for their encouragements, moral support, and patience as I worked my way from writing the initial proposal to this final document. I could not have done it without all of you.

Ran Chang

CONTENTS

Page

CHAPTER

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

With the rapidly growing number of digital images found on the Internet and housed in digital libraries, the need for effective and efficient tools to manage large image databases has grown dramatically. Specifically, the development of efficient image retrieval systems to find images of interest in this haystack of data has become an active research area in recent years [1].

1.1 New Era of Images and the Need for Digital Image Retrieval

Since the first digital image (shown in Figure 1.1) was generated in 1957 by Russell Kirsch, a scientist working in the research institute now known as the National Institute of Standards and Technology (NIST), people have entered a new era of digital imaging. Rapid development of digital imaging was stimulated by the emergence of microprocessors in the early 1970s. Millions of digital imaging devices equipped with the Charge-Coupled Device (CCDs) have launched a revolution for conventional photography and generated billions of digital images since the last decade of the twentieth century. Due to the huge amount of digital images on the Internet and in different kinds of large or small digital libraries across the world, the need for image database management and effective image retrieval tools has been growing rapidly.

Image retrieval systems meet the above need of acquiring the desired images from digital image libraries for human users. Basically, an image retrieval system is a computer system along with necessary hardware and software to search through a relatively large digital image database or library and retrieve similar images according to the user’s query. In general, there are two kinds of image retrieval systems, namely, classical image retrieval systems and localized image retrieval systems. Specifically, classical image retrieval systems aim to find images that are similar to the query images in terms of semantic concepts. Localized image retrieval systems aim to find duplicated or near-duplicated objects in an image collection, which are the same objects contained in the query images. For example, if the query image contains a red rose with a pure background (i.e., pure black or white background), classical image retrieval systems consider any returned images containing a flower or multiple flowers with different colors and backgrounds as good retrieval results while localized image retrieval systems consider any returned images containing the rose(s) with different scales, rotation angles, and locations as good retrieval results.

In classical image retrieval systems, the query could be either text-based keywords or a certain image that the user is interested in. Most conventional techniques first require a large amount of manual labor to annotate images in the database with certain relevant keywords, descriptions, tags, or captions. They then match annotated words of the database images with the text-based query keywords submitted by the user to return similar images. Google Image (images.google.com), which is used daily by billions of people, applies this text-based conventional technique to retrieve similar images. Obviously, this kind of conventional image retrieval system requires database images to be annotated before they are added into the corresponding digital image library. Otherwise, images without any annotation will never be retrieved when a text-based query is submitted. However, manual annotation of digital images in a large digital library is an unimaginable, time-consuming task. No organizations or companies can afford this kind of labor and expense. In addition, manual annotation has other weaknesses such as the user's subjectiveness, erroneous image annotation, and inconsistent annotation from different users for the same images, etc. Thus, researchers start exploring techniques that perform automatic annotation for a large amount of digital images. Usually, these techniques learn a statistical model that is trained by using sufficient annotated images. With the aid of the trained model, they then perform automatic annotation for other images. The downside of automatic image annotation is that the trained model greatly relies on the quality and the number of annotated training images. If training images have inaccurate, insufficient, unevenly distributed, or low quality tags, the trained statistical model cannot provide accurate annotation for other images. Moreover, the trained statistical model cannot learn more accurate semantic concept of the images if human feedback on the automatically annotated keywords is not provided. In the 1990s, several researchers at the Massachusetts Institute of Technology (MIT), including Banireddy Prasaad, Amar Gupta, Hoo-min Toong, and Stuart Madnick, invented the first microcomputer-based digital image retrieval system for a large digital image database, wherein each image is automatically annotated [2]. This system is an initial experimental image retrieval system based on the automatic image annotation. Since the early 2000s, automatic image annotation has become a popular research topic and attracted more and more researchers to build image retrieval systems upon the automatically annotated images. These described digital image retrieval systems are also called concept-based, or "text-based" or "description-based" image retrieval systems, whose searching and retrieving process relies on automatically annotated keywords or tags of the digital images. Later on, web-based image search engines also apply concept-based image indexing techniques to retrieve similar images from the web.

Almost at the same time of the emergence of concept-based image retrieval systems, another kind of image retrieval systems, namely, Content-Based Image Retrieval (CBIR) systems also emerged in the early 1990s. Both concept-based and CBIR systems have evolved significantly since the 1990s. In the following, I will describe the background of the CBIR system, since my proposed system belongs to this branch of the classical image retrieval systems.

Figure1.1. The first digital image in the world

1.2 Background of Content-Based Image Retrieval (CBIR)

Unlike concept-based image retrieval systems, CBIR systems perform the image retrieval task by submitting image(s) as a query and making use of low-level visual image features (e.g., color, texture, shape, etc.) instead of keywords to represent images, where each feature can be automatically and consistently extracted without human intervention. In other words,

Typical CBIR systems perform the searching and retrieving task by analyzing colors of images, shapes of objects in images, the textures distribution of images, or any other representative information extracted from images, rather than any metadata such as keywords, tags or captions etc. Back in 1992, researcher Kato T [3] first used the term of CBIR to describe the experiment of automatic digital image retrieval by comparing image color and shape features of each database image with query's color and shape features. Since then, this term has been wildly used to refer to all similar techniques and processes of searching and retrieving images from a digital image library using the common representative features such as colors, shapes, and textures, etc.

Early CBIR systems usually rely on image feature extraction and matching strategies to retrieve relevant images from a database. For example, Flickner *et al.* [4] from IBM invented the QBIC system in 1995, Gupta and Jain invented VIRAGE [5] in 1997, and Mukherjea *et al.* [6] invented NEC AMORE in 1999. The above three CBIR systems are the earliest systems for the commercial purpose. During the same period, some other researchers invented CBIR systems for the academic purpose, such as the MIT Photobook by Pentland *et al.* [7], Columbia VisualSEEK and WebSEEK by Smith and Chang [8], UCSB NeTra by Ma and Manjunath [9], and Standford WBIIS by Wang *et al.* [10].

Meanwhile, researchers found the advantages of employing CBIR systems in several real-world applications [11]. The following is the list of a few sample applications:

1. Architectural and engineering design: CBIR can help the designers to find similar buildings, or landscape designs by providing certain sample designs.
2. Art collections: CBIR can be applied in digital art museums and help the user to find the desired art work such as painting, drawing, photography or even sculpture by sending a sample image.
3. Criminal prevention: CBIR can help law enforcement officers to quickly find similar crime scenes, or suspects by uploading the evidence images into the system.
4. Geographical information field: CBIR can help geologic researchers to easily find desired mineral resources by grouping similar physiognomy.
5. Intellectual property: CBIR can help authors of drawing or photography to easily locate any copyright violation of their work on the Internet by submitting a digital copy of their work to the system.
6. Medical treatment: CBIR can provide the doctors great help in early diagnosis by retrieving similar pathological photos in a large medical image database.
7. Military: CBIR can help commanding officers or intelligence officers to quickly define hostile vehicles by sending the live picture of the potential enemy vehicle into the system.

8. Retail catalogs: CBIR can help customers to quickly and easily retrieve their desired merchandise by uploading the photo or picture of the merchandise to the system.

1.3 Basic Content-Based Image Retrieval Systems

CBIR techniques are viable solutions to find desired images from digital image libraries. In a basic CBIR system, all digital images in a library are represented by their visual features (e.g., visual contents of images). Typical visual features include colors, shapes, edges, and textures to represent an image from different visual perspectives. Initially, these visual features are extracted from each image and stored in a feature database corresponding to the digital image library to facilitate the future use. When a query image is submitted to the system, visual features of the query image are first extracted. A matching method is then employed to compare the similarity between visual features of the query image and visual features of all digital images in the image database. Only those images having higher similarity scores are returned to the user as the retrieval results. Figure 1.2 shows the high-level block diagram of a basic CBIR system.

However, as the ranking of retrievals is calculated based on selected image features, the retrieval accuracy may be unsatisfactory due to the semantic gap between low-level visual features and high-level semantic concepts. This semantic gap exists because images of similar semantic content may be scattered far away from each other in the feature space, while images of dissimilar semantic content may share similar low-level features. For example, given a query image with a black horse in the front view, an image with a white horse in a side view is considered similar to the query image from the view point of the semantic concept. However, the front view of a black horse looks very different from the side view of a white horse, so are their visual features (i.e., their visual features are different). On the other hand, the cruise in the ocean and the airplane in the blue sky, as shown in Figure 1.2, are two distinct objects with the similar low-level features. The cruise is wrongly retrieved by the CBIR system when the airplane is submitted as a query due to the high similarity of their low-level feature vectors. Humans bridge this gap without even noticing that they are doing it. However, computer vision techniques have been struggling to bridge this gap ever since the advent of the computer vision.

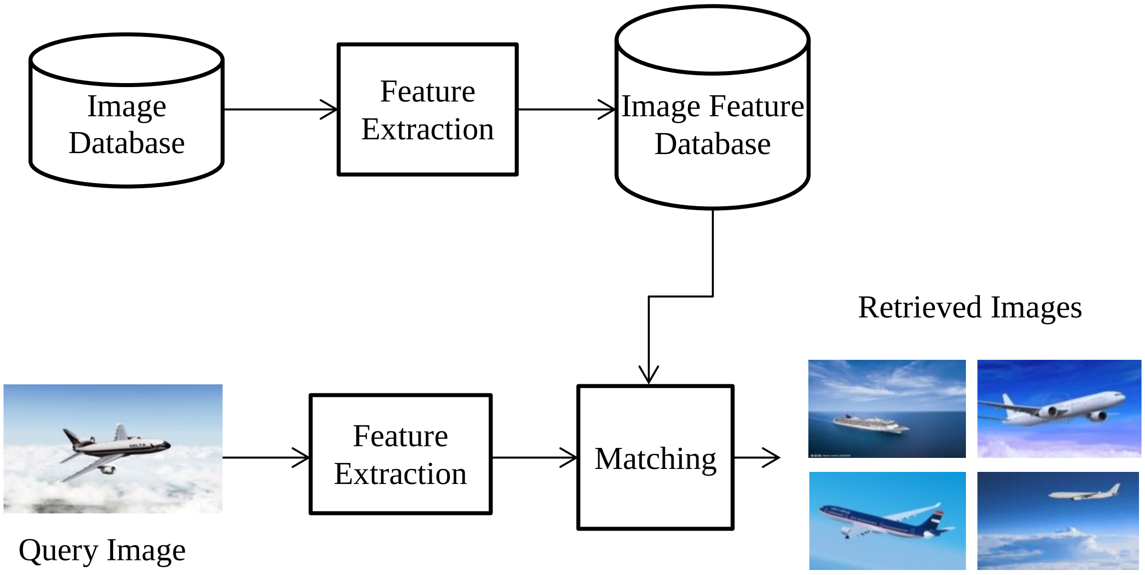


Figure 1.2. A basic CBIR system

Therefore, the existence of the semantic gap makes basic low-level feature-based CBIR systems have limited use. This also motivates researchers to study other techniques to bridge the semantic gap for CBIR systems.

CHAPER 2

RELATED WORK

As mentioned in the previous chapter, the semantic gap is the primary issue to be considered by researchers when they develop CBIR systems. Many novel techniques have been proposed to overcome this stubborn challenge. Representative techniques that tackle with the semantic gap problem are reviewed.

Present CBIR techniques can be generally classified into four major categories according to the survey papers by Antani *et al.* [12], Smeulders *et al.* [13], and Zhou *et al.* [14]. These four categories are global feature-based [4, 7, 15, 16], region-level feature-based [1, 17 - 25], object-level feature-based [19, 25 - 30], and Relevance Feedback (RF)-based [16, 27, 31 - 34], respectively.

Global feature-based techniques rely on the visual features extracted from a whole image and treat each part or object in the image without discrimination. With the aid of these global features, CBIR systems deploy variable matching strategies to find most relevant images in the database to the query image based on the similarities of global features. For example, the QBIC system [4] uses the average color and texture of an image as low-dimensional features and the 20-dimensional moment-based shape features as high-dimensional features to represent an image for the retrieval task.

Unlike global feature-based techniques, region-level feature-based techniques usually separate an image into several regions and treat regions with different attentions according to the importance of the content in each region. The size of the regions can be either equal or different. In other words, methods of dividing the images vary depending on real applications. Figure 2.1 shows an example of dividing an image into five different regions according to the importance of the content in each region [35]. This division scheme assumes that the central region in an image likely contains the most important visual content, the upper and lower regions normally contain the background information, and the most left and right regions possibly contain some other objects that are not the key content of an image. After extracting visual features for every region (e.g. the color, shape and texture), regional-level feature-based techniques apply various matching algorithms to calculate the similarities between the images in the digital image database and the query image at the regional level and fuse the similarities of all regions to produce a final relevance score to measure the overall similarity. For instance, Qi and Han [36] propose a CBIR system which uses a fuzzy feature representation to represent the characteristics of an image based on a set of color-clustering-based regions. The final relevance score is calculated based on a fuzzy region matching scheme.

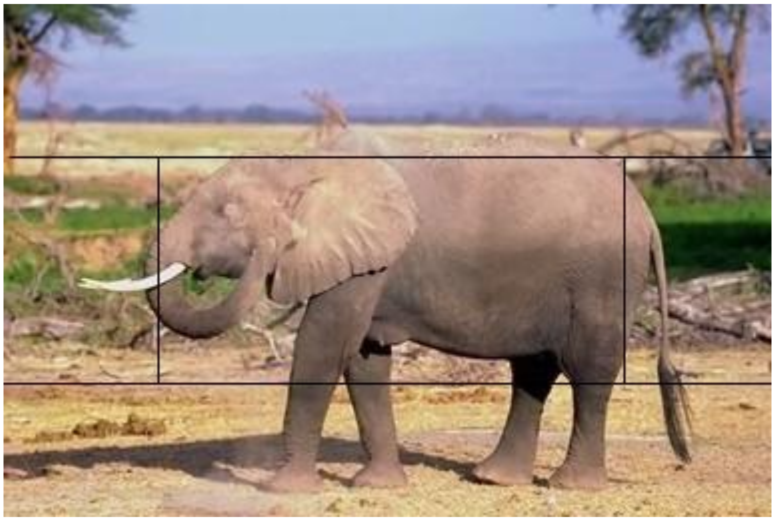


Figure 2.1. An example of dividing an image into five regions

Compared to region-level feature-based techniques, object-level feature-based techniques focus on more detailed content information. This kind of technique first applies an image segmentation method to obtain independent objects in an image. They extract visual characteristics of these objects such as color, texture, shape etc. to form a low-level visual feature vector for an image. Finally, they apply a matching algorithm on these object-level features to calculate the final relevance score for each image in the image database. Wang *et al.* [25] apply object-level color, texture, and shape features in their proposed SIMPLIcity CBIR system and demonstrate their effectiveness. However, image segmentation remains to be a challenging research topic in the computer vision field. There is not a universal segmentation solution for all type of images. Therefore, object-level feature-based systems suffer from the degraded quality of the segmented images.

Relevance Feedback (RF)-based techniques [37] are online supervised learning techniques which have been widely adopted in CBIR systems to bridge the semantic gap. RF repeatedly modifies the query descriptive information (feature, matching models, metrics or any meta knowledge) as response to the users' feedback on retrieved results. Therefore, it learns the query close to its optimal and returns more user-desired images (i.e., improves the retrieval precision) after each round. Figure 2.2 provides a simple flow diagram of a RF-based CBIR system. The first RF-based interactive CBIR system is proposed in [33], where the user's provided judgment upon the retrieved images in previous retrieval iterations is used to overcome two major weaknesses in non-RF-based systems: 1) the semantic gap between high-level semantic concepts and the low-level visual features of images, and 2) the subjectivity of human perception of visual content (i.e., inconsistent relevance judgments of the same image from different human users). Specifically, this first RF-based system dynamically updates the corresponding feature weights to capture the user's query intention and perception subjectivity after each query iteration. As a result, this RF-based CBIR system improves the retrieval performance of other non-RF-based CBIR systems.

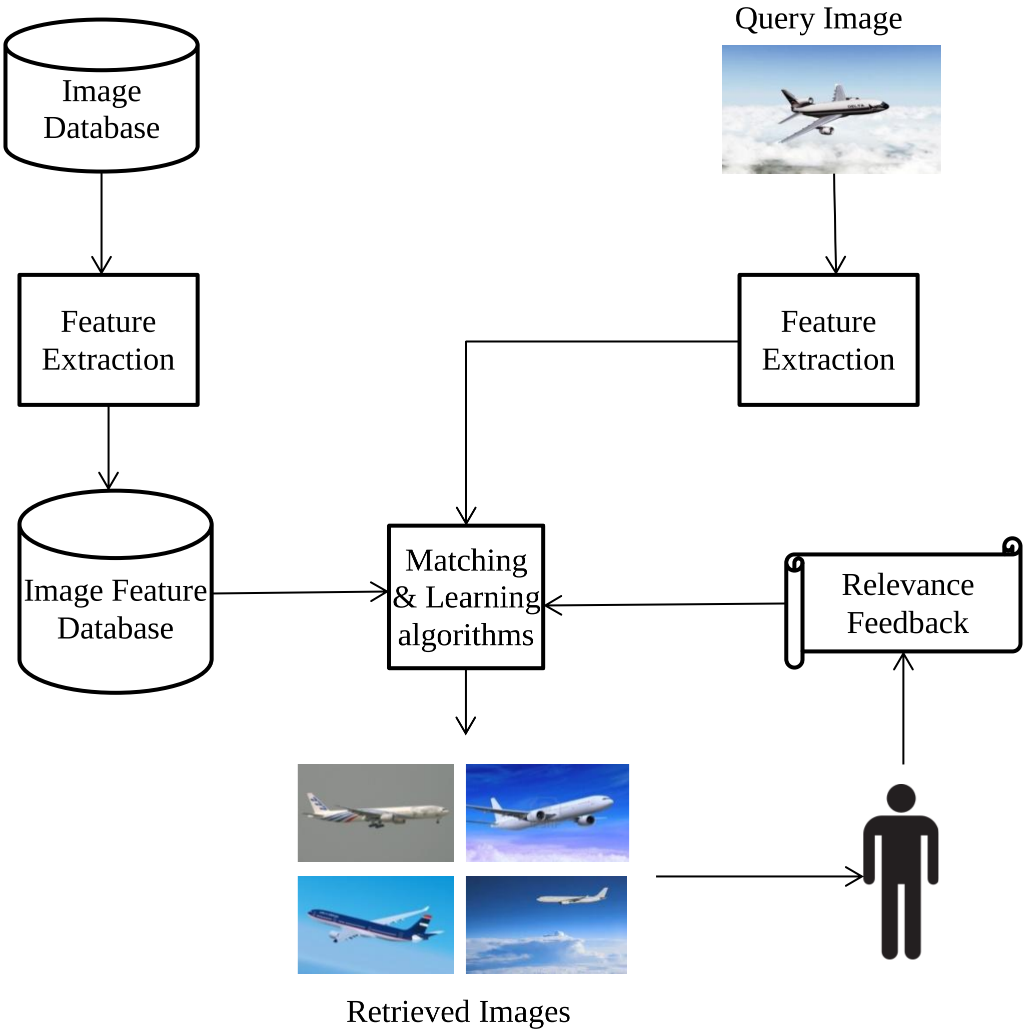


Figure 2.2. The illustration of a RF-based CBIR system

Users play an important role in such RF-based CBIR systems. Correct feedback from users greatly boosts the performance of the CBIR system to capture the desired search intention of users. As a result, researchers have been focusing on applying learning algorithms on the user's RF to improve the retrieval performance. These learning algorithms can be generally categorized into short-term learning techniques and long-term learning techniques. Selecting the proper learning techniques depends on the real retrieval applications. There is not a clear answer of whether short-term learning is better than long-term learning or vice versa. The review of short-term learning and long-term techniques is provided in the following subsections.

2.1 Short-term Learning Techniques

Short-term learning techniques aim to find out which images are relevant to the user's query over the course of a single query session. Query updating and statistical learning techniques are two common categories of short-term learning techniques.

2.1.1 Query Updating Techniques

Query updating techniques improve the representation of the query itself by using the user's subjectively labeled information. Examples of query updating techniques include query re-weighting [38], query shifting [39], and query expansion [40]. Specifically, Kushki *et al.* [38] apply the query re-weighting technique to learn an optimal mapping between low-level visual features and high-level semantic concepts of an image by adjusting the weights (or importance) of each feature component or by modifying the corresponding similarity measure. Muneesawang and Guan [39] apply the query shifting technique to allow the user to directly modify the query image's characteristics, which correspond to some components in the query's feature vector, by specifying their attributes in the form of relevant or irrelevant retrieved training images marked by the user. In other words, the characteristics of the query image's content are changed according to a more accurate semantic representation provided by the user during the retrieval process. Widyantoro *et al.* [40] apply the query expansion technique to include a set of relevant non-user-labeled images to compensate for the lack of the user-labeled images and help the system capture more accurate meaning of the query image.

2.1.2 Statistical Learning Techniques

Statistical learning techniques improve the classification boundary between relevant and irrelevant images or predict the relevance of unlabeled images which are attainable during the training stage. Examples of statistical learning techniques include inductive learning and transductive learning.

2.1.2.1 Inductive Learning: Inductive learning [41] is defined as a process of acquiring knowledge by drawing inductive inferences from teacher or environment-provided facts. Such a process involves operations of generalizing, transforming, correcting, and refining knowledge representations. Inductive learning techniques applied in CBIR systems create various classifiers which separate the relevant (i.e., positive) and irrelevant (i.e., negative) images and generalize well on unlabeled images. Here, relevant and irrelevant images are respectively positively and negatively retrieved images labeled by the users during the query retrieval session. Typical inductive learning techniques include decision tree learning [42], Bayesian learning [43 - 45], support vector machine (SVM) learning [46], fuzzy SVM (FSVM) learning [47], and boosting [48].

MacArther *et al.* [42] apply a decision tree in the CBIR application. They use the relevant and irrelevant images marked by the user to partition the feature space until all instances in a partition are of the same class. Su *et al.* [44] feed the relevant and irrelevant feedback from the user into a Bayesian classifier. Relevant images are used to estimate a Gaussian distribution which represents the user desired images for a query image, while irrelevant images are used to revise the ranking of retrieved candidates. Tong and Chang [46] propose a CBIR system with the aid of the SVM which learns a proper boundary to separate images in the database into relevant and irrelevant partitions using relevant and irrelevant samples collected from previous retrieval iterations. The mechanism of the SVM will be described in Chapter 3, since it is a related technique employed in my proposed CBIR system. Wu and Yap [47] apply a FSVM to learn a decision boundary to separate positive and negative training images which are assigned corresponding fuzzy weights to determine its importance in the classification task. The learned decision boundary is then used to partition the database images into relevant and irrelevant images, where the relevant images with the largest distance to the boundary are considered to be the most similar images to the query. Tieu and Viola [48] propose a CBIR system which a “boosting” learning mechanism is used to generate a very large number (e.g., 46,875) of highly selective features to capture as many as possible aspects of an image’s visual concept. A series of weak learners based on a small number of features are trained during the query time. By combining all of these weak classifiers, the system eventually obtains a strong classifier which is well-correlated with the ideal classification.

However, query updating methods [39] do not fully utilize the information embedded in feedback images and therefore cannot achieve satisfactory retrieval results. Inductive learning methods [46, 47] yield degraded retrieval results when the chosen classifier is trained with insufficient labeled training samples. Moreover, these two categories of techniques ignore the manifold structure of image features. Therefore, the latest trend has been moving towards RF-based transductive learning.

2.1.2.2 Transductive Learning: Transductive learning techniques explore the relationship of all database images in the feature space and propagate ranking scores of labeled images to unlabeled images via a weighted graph. In this way, the information of the entire database, instead of labels that are assigned to images by users, is efficiently utilized to facilitate the future learning. Manifold-ranking-based learning [16, 49 - 56] techniques are the representative transductive learning techniques. They use all unlabeled images as vertices in a weighted graph to propagate the ranking score of labeled images. The following paragraphs provide a brief review of representative transductive learning techniques in CBIR systems.

He *et al.* [49] propose the Manifold Ranking Based Image Retrieval (MRBIR) algorithm to represent images and their relationships as a graph. This system propagates the labeled image information through the graph structure of the image database and exploits the distribution of unlabeled images to improve the retrieval accuracy. Cai *et al.* [50] incorporate a locality preserving regularizer into the manifold structure to learn a classification function in the image manifold. They then apply the user's RFs to update the manifold structure for better classification. He *et al.* [16] propose the generalized MRBIR (gMRBIR) algorithm to improve the MRBIR algorithm by allowing the user to submit any query image that is either inside or outside of the database. Wang *et al.* [51] apply the Affinity Propagation Clustering (APC) algorithm to reduce the manifold graph and preserve its manifold structure. This reduced graph damps the effect of noisy images while emphasizing the effect of reliable images. However, the retrieval performance may be degraded when clusters do not resemble the semantic concept. Lin *et al.* [52] propose a so-called Augmented Relation Embedding (ARE) method to transform an image space into a semantic manifold. By applying this semantic manifold structure, the system can obtain the user's query preferences. Meanwhile, a new image representation based on the augmented feature is also deployed to adapt the ARE learning. Wan [53] proposes to divide every database image in equal-sized blocks and then apply the MRBIR algorithm on each block. The retrieval score of each image is a fusion of ranking scores of all blocks in the image. Extensive experimental results show that this block-based manifold ranking method outperforms the conventional manifold ranking method. Liu *et al.* [54] invent a novel manifold ranking system, named Bidirectional-Isomorphic Manifold Learning, to acquire more semantic representation from web images to overcome the imprecise semantic content representation caused by noisy and redundant information from textual and visual aspects. This method eventually optimizes the visual feature and textual spaces and unifies adjustments in both spaces to a topological structure called reversed manifold mapping. This new system also combines the image annotation and keywords correlation analysis to boost the final retrieval accuracy. Han *et al.* [55] come up with a novel image classification framework, which adopts Local and Global

Regressive Mapping (LGRM) in manifold learning to learn the low-dimensional embedding of the input data and the mapping function for out-of-sample data at the same time. Eventually, it predicts the class labels for a test image by applying the supervised classifier in the learned low-dimensional manifold. Xu *et al.* [56] propose to project the conventional manifold ranking into a Bregman divergence optimization framework by using an equivalent optimal kernel matrix. Based upon their new formulation, two effective and efficient extensions called DMR_E and DMR_C are created to boost the retrieval accuracy and shorten the computational time.

All above transductive methods achieve better retrieval precision in each iterative step. However, they do not apply users' accumulated historical RF information to improve the manifold graph. They also cannot run on a computer when the number of images in the database reaches a certain level due to the use of several large square matrices. Furthermore, all these short-term learning techniques cannot capture the semantic meaning of an image and therefore cannot achieve satisfactory retrieval results. They also cannot remember users' historical feedback and therefore cannot utilize it in future retrievals.

2.2 Long-term Learning Techniques

Recently, long-term learning or inter-query learning extends short-term learning by utilizing the information gathered from the past retrieval sessions to improve the retrieval results in future retrieval sessions. Specifically, these long-term learning techniques first store the accumulated feedback history collected from multiple query sessions in a feedback log. They then aggregate the information in the feedback log into a semantic matrix, relevance matrix, or affinity matrix, which can be further used to discover extra knowledge. Finally, they infer relationships between images by analyzing the transformed matrix and estimate the semantic relevance level of a database image to the current query. In general, long-term learning techniques can be categorized into six categories [57]: Latent Semantic Indexing (LSI)-based, correlation-based, clustering-based, feature representation-based, similarity measure modification-based, and manifold-based techniques. In the following subsections, a review of the representative long-term learning techniques is presented.

2.2.1 Latent Semantic Indexing-based Long-Term Learning

Latent Semantic Indexing (LSI) technique was proposed by Deerwester *et al.* [58] for document query in 1990. The core of LSI is to construct a term-by-document matrix and apply the Singular Value Decomposition (SVD) on this term-by-document matrix to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. In document query, the query is a document which is represented by several known terms. Given these terms, a number of similar documents can be retrieved. When the LSI techniques are used in CBIR systems, terms refer to query images and documents refer to the retrieved relevant images in the digital image database. The term-by-document matrix in CBIR is a matrix M with the size of $m \times n$, where m is the number of queries and n is the number of images. The SVD technique can then be applied on $M_{m\times n}$ to acquire an approximation term-by-document matrix $\tilde{M}_{m\times n}$, which is defined by the following formula:

$$M_{m\times n} = U_{m\times r} S_{r\times r} V_{r\times n} \approx \tilde{M}_{m\times n} = \tilde{U}_{m\times k} \tilde{S}_{k\times k} \tilde{V}_{k\times n} \quad (2.1)$$

where r is the rank of $M_{m\times n}$, $\tilde{M}_{m\times n}$ is composed of the top k largest singular values ($k < r$) corresponding to the singular vectors, U contains the eigenvectors of MM^T, V contains the eigenvectors of M^TM (e.g., U and V are orthogonal), and S contains non-zero singular values of M at the diagonal of the matrix, which are the square roots of the non-zero eigenvalues of both MM^T and M^TM. This approximated decomposition greatly reduces the sizes of $\tilde{U}_{m\times k}, \tilde{S}_{k\times k}$ and $\tilde{V}_{k\times n}$, so the storage requirement and the computational cost are reduced.

Heisterkamp [59] invent a new method to construct the term-by-document matrix M based on three assumptions: 1) images in a digital image database can be treated as the words of vocabulary of the system; 2) an image has multiple semantic concepts; and 3) many images share similar semantic concepts. SVD is then utilized on M to generate corresponding approximated $\tilde{U}_{m\times k}, \tilde{S}_{k\times k}$ and $\tilde{V}_{k\times n}$. When an unknown query is submitted, a *pseudo-document* T_q is constructed for this query and projected into the latent semantic space by $F_q = \tilde{U}T_q$. The system finds K-Nearest Neighboring documents of Fq and selects the most probable and informative terms to form the retrieval set which is returned to the user. The new retrieval iteration starts after T_q is updated based on the user's RF.

2.2.2 Correlation-based Long-Term Learning

Correlation-based long-term learning techniques aim to explore the semantic correlation for each pair of images in an image database using accumulated historical log files obtained from query sessions. Researchers demonstrated that the efficiency of the CBIR system can be improved with the help of these log files.

Zhou *et al.* [60] introduce a correlation matrix $R_{m \times n}$ to store the relevance information labeled by the user, where m is the number of query sessions, and n is the number of images in the image database. Specifically, each element r_{ij} in $R_{m \times n}$ denotes the relevance label of an image I_j at the ith query session F_i. Figure 2.3 shows an example of this correlation matrix. Here, if an image is labeled as relevant by the user at a certain retrieval iteration, the corresponding element in R is set as 1's. Suppose that image I_1 is labeled as a relevant image by the user in query sessions F_1, F_2, and F_3. Corresponding elements in the first column are set as 1's, respectively. For other images (e.g., images labeled as irrelevant by the user or not returned in a query session), corresponding elements are set as 0's in the correlation matrix. Finally, the system applies the collaborative filtering method to measure the correlation between database images and the current relevant images. However, this work does not involve irrelevant images labeled by the user in the construction of the correlation matrix.

	I_1	I_2	I_3	I_4	I_5	I_6	•••	I_n
							•••	
F_1	1	0	0	0	0	0	•••	0
F_2	1	0	0	0	1	0	•••	0
F_3		1	0	0	0	1	•••	0
•••	•••	•••	•••	•••	•••	•••	•••	•••
Fm	0	1	0	0	0	1	•••	0

Figure 2.3. An example of the correlation matrix

Yin *et al*. [61] design a virtual-features-based technique to explore the long-term historical feedback information to estimate the semantic relationship between images. Specifically, their proposed system combines the short-term and long-term learning to utilize users' historical feedback to form virtual features of images to represent the semantic meaning of images. It then adaptively calculates the similarity between each image and the query based on the semantic relevance calculated by virtual features. In addition, their proposed system is capable of handling the dynamic database by adapting the concepts according to the user's new subjective concepts.

He *et al.* [62] propose a RF-based CBIR framework to combine short-term and long-term learning. Specifically, short-term learning employs the query refinement technique to update the image's low-level visual features. Long-term learning focuses on constructing the semantic space which contains the relevant feedbacks in the previous retrieval iterations. An image in the database is then represented by a semantic vector, which can be updated according to the future accumulated users' RF. A SVD technique is also applied on the semantic space to reduce its size. However, this system does not consider the irrelevant samples in the RF in the construction of the semantic space.

Xiao *et al.* [63] propose a short-term (intra-query) and long-term (inter-query) combined learning strategy by applying users' historical RF semantic knowledge to create dynamic semantic features for database images. This system builds an adaptive semantic matrix to store the similarity of the relevant and irrelevant images in historical query sessions during the cross-session learning process. The high-level semantic similarity can benefit from the updated semantic features to boost the overall retrieval accuracy.

2.2.3 Clustering-based Long-Term Learning

Clustering-based long-term learning techniques rely on the large amount of the accumulated historical retrieval information to group database images into several clusters, where each cluster has a unique semantic concept.

Han *et al.* [64] propose a memory learning technique to form a knowledge memory model to store the semantic information and learn semantic relations. Specifically, the semantic correlation between a pair of images is measured as a ratio of co-positive-feedback frequency and co-feedback frequency, where co-positive-feedback frequency represents the number of times that this image pair is both labeled as positive to the same query and co-feedback frequency represents the number of times that this image pair is labeled as either both positive or one positive and one negative. With the aid of semantic correlations computed in the previous labeling, the system applies the *k*-means clustering method to divide database images into several semantic-correlated clusters, where images in the same cluster are regarded as sharing the same semantic concept. It then explores the semantic relations between images according to the correlation ranking learned from low-level visual feature-based short-term learning and high-level semantic-based long-term memory learning. Finally, it measures the relevance similarity between each image and query image by applying a probabilistic model.

Recently, Qi *et al.* [35] enhanced the retrieval performance by developing a short-term block-based FSVM and long-term dynamic semantic clustering (DSC) technique, which adaptively learns and updates semantic categories using users' positively and negatively labeled RF. Specifically, in short-term learning, the system applies the nearest neighbor mechanism to choose additional similar blocks. A fuzzy metric is computed to measure the fidelity of the actual class information of the additional blocks. The FSVM is finally applied on the enlarged training set to learn a more accurate decision boundary for classifying images. Long-term learning addresses the large storage problem by building dynamic semantic clusters to remember the semantics learned during all query sessions. In detail, it applies a cluster-image weighting algorithm to find the images most semantically related to the query. It then applies a DSC technique to adaptively learn and update the semantic categories.

2.2.4 Feature Representation-based Long-Term Learning

Feature representation-based long-term learning techniques use the user's historical feedback to adjust the weights for the low-level visual feature vectors.

Cord and Gosselin [65] use the subset of labels which are accumulated in the historical query sessions to modify low-level visual feature vectors of images to refine images' feature representation. A supervised optimization of a subset of feature vectors is employed to improve the representation of the image collection without any prior information.

2.2.5 Similarity Measure Modification-Based Long-Term Learning

Similarity measure modification-based long-term learning techniques aim to adaptively modify the similarity measure based on the accumulated historical user's RF knowledge.

Hoi *et al.* [66] propose a CBIR system which applies the statistical correlation on the retrieval log to analyze the relationship among images based on the current and past query sessions. This correlation of images is stored in a retrieval log-based correlation matrix. In details, the relevance similarity is calculated by the formula $f_q(I_i) = 0.5 \cdot f_{LG}(I_i) + 0.5 \cdot f_{LL}(I_i)$, where q is the query image, I_i is a database image, $f_q(I_i)$ calculates the relevance score of image I_i with regards to q, $f_{LG}(I_i)$ calculates the relevance score using the correlation difference between user labeled relevant samples and irrelevant samples in the log-base correlation matrix, and $f_{LL}(I_i)$ calculates the relevance score based on low-level visual feature vectors.

2.2.6 Manifold-based Long-Term Learning

Unlike the above five categories of long-term learning techniques which use the piecewise distance calculation, manifold-based long-term learning techniques explore the relationship of all database images in the feature space.

Chang and Qi [67] and Chang *et al.* [68] create semantic clusters based on users' historical RF to group semantically similar images. They then construct a weighted semantic clusters-based manifold structure to represent image relationship in low-level visual feature space [67] and both low-level visual and high-level semantic feature spaces [68] for better retrieval performance. However, these two learning techniques cannot be directly applied to a large scale CBIR system due to the use of several large square matrices whose size equals to the square of the number of database images. To make the system scalable, Chang and Qi [69] propose a novel hierarchical manifold ranking system which constructs a two-layer intrinsic weighted structure using the visual space at the first layer and the visual and semantic spaces at the second layer. The relevance scores of labeled images are propagated to unlabeled images via this hierarchical manifold. However, a relatively large matrix is used to store semantic features of each database image. The size of this matrix equals to the number of database images multiplying by the number of training queries (e.g., 10% of the number of database images). As the size of the image database grows, the size of this matrix grows too. Eventually, computer may not have enough memory space to store this matrix. So the scalability issue still presents in all existing long-term manifold-based CBIR systems.

2.3 Evaluation Measures

Two kinds of evaluation measures, *Precision* and *Recall*, are used extensively to evaluate the retrieval performance in CBIR.

Precision is the ratio of the number of retrieved relevant images and the number of all retrieved images during a single retrieval iteration. It is computed as follows:

$$\mathrm{Pr}ecision = \frac{Sum(\mathrm{Re}levant)}{Sum(\mathrm{Re}trieved)} \tag{2.2}$$

Recall is the ratio of the number of retrieved relevant images in a query session over the number of all relevant images in the image database. It is computed by:

$$\mathrm{Re}\,call = \frac{Sum(\mathrm{Re}levant\_in\_Session)}{Sum(\mathrm{Re}levant\_in\_Database)} \tag{2.3}$$

Both precision and recall are normally represented by a percentage. They also have an inversed relationship. That is, the precision decreases when the number of retrieved images increases, while the recall conversely increases.

Another often adopted measure is an evolution of the precision, which is called *Average Retrieval Precision (ARP).* Given multiple queries, the retrieval performance of each query is first measured by precision. The overall retrieval performance of a system is measured by averaging all of the precisions. This *ARP* is computed as follows:

$$ARP = average(\sum P_i) \qquad (2.4)$$

where P_i is the precision of each query. It is a powerful measure to represent the performance of the CBIR system, especially when a thorough test is deployed.

Combining precision and recall generates another commonly applied measure, namely, *Precision-Recall curve* (PR curve), to evaluate the performance of CBIR systems. Specifically, in a PR curve, the x-axis represents recalls which are achieved by using different number of returning images in a query session, and y-axis represents the corresponding precision associated with each recall.

2.4 Outline of the Proposed Method

In this dissertation, I propose two novel manifold-based long-term learning CBIR frameworks. The first framework is named as the scalable graph-based CBIR framework, can be effectively applied to large-scale image databases. This scalable graph-based ranking system requires comparatively small memory space to construct two-layer hierarchical graphs for the image database. Therefore, this proposed system is efficient to perform retrieval tasks in large-scale image databases (i.e., image databases with more than 10,000 images). On the other hand, this proposed system has to sacrifice certain computational efficiency to build such hierarchical graphs during the offline training stage. Specifically, it takes the advantages of both RF based transductive short-term learning and semantic feature-based long-term learning techniques to improve retrieval performance. Major contributions are: 1) Quickly constructing a compact dynamic feedback log to store retrieval patterns of each past query session. 2) Efficiently merging similar semantic concepts to maintain a reasonable number of representative semantics for all images in a database. 3) Creatively constructing two-layer hierarchical graphs to represent the inherent structure of the large-scale image database during the system offline training stage. 4) Effectively combining low-level visual and high-level semantic similarity measure to build a scalable manifold graph, which explores the intrinsic structure of images in both low-level visual and high-level semantic feature spaces. 5) Effectively designing a layered relevance vector to propagate the relevance scores from anchor images to the second layer graphs and further propagate relevance scores of labeled images to unlabeled image via the hierarchical graph-based structure.

The second framework, named as the single weighted semantic manifold graph ranking framework, can be effectively used in small databases. This proposed system requires less computation to construct the graph structure of the image database. Therefore, it's an efficient CBIR system when users perform retrieval tasks in relatively small databases (i.e., image databases with less than 10,000 images etc.). Specifically, this framework builds a more accurate intrinsic structure for the proper image space by combining low-level and high-level relations. Major contributions are: 1) Applying the learning mechanism to explore semantic concepts of the image database. 2) Extracting high-level semantic features of each image based on users' retrieval experiences. 3) Incorporating the importance score of each image into the affinity matrix to build the weighted semantic manifold structure. 4) Constructing the asymmetric relevance vector to propagate ranking scores of its labeled images via the manifold to images with high similarities.

The rest of this dissertation is organized as follows: Chapter 3 describes several key related techniques used in the proposed frameworks. Chapter 4 presents the proposed scalable graph-based CBIR framework together with its corresponding extensive experiments, analysis, conclusions, and future work. Chapter 5 describes the proposed single weighted semantic manifold graph-based CBIR system together with its corresponding extensive experiments, analysis, conclusions, and future work. Chapter 6 concludes the dissertation and presents the future work to improve both frameworks.

CHAPTER 3

RELATED TECHNIQUES

In this chapter, I provide a detailed background of techniques that are employed in my proposed systems. Remaining sections in this chapter are arranged as follows: section 3.1 describes the low-level visual feature used in proposed systems; section 3.2 explains the kernel-based soft margin SVM technique that is applied in the offline training phase in proposed systems; section 3.3 describes the conventional manifold techniques.

3.1 Low-Level Feature Extraction

In computer vision, effectively representing an image is a critical issue. Powerful image features can greatly reduce the semantic gap between the human perception-based semantic concepts and the machine-based visual features. As a result, proper low-level image features should contain complementary information to represent an image or objects within an image from different perspectives. They should be of a reasonable length and easy to compute. In other words, inefficient low-level features decrease the efficiency of deploying CBIR systems in real-world applications.

Low-level features usually include color, edge, and texture features. Specifically, the color feature can be represented by the color histogram and the color moment; the edge feature can be represented by the edge histogram; and the texture feature can be represented by co-occurrence matrices and statistics of each significant subband in the wavelet transformed image.

3.1.1 HSV Color Space and Color Histogram.

In CBIR systems, color is the commonly adopted feature to represent the characteristics of an image. Researchers have explored many techniques to categorize the color into different color spaces. Red, Green, and Blue (RGB) space is a well-known common color space for the public because it works similarly as the human visual system [70]. Mixing three primary colors (e.g., red, green, and blue) can generate countless colors. Figure 3.1 shows a RGB color cube as an example. The RGB color space has many variants including ISO RGB, Extended ISO RGB, standard RGB (sRGB), Adobe RGB (1998), Apple RGB, NTSC RGB (1953), etc. [71].

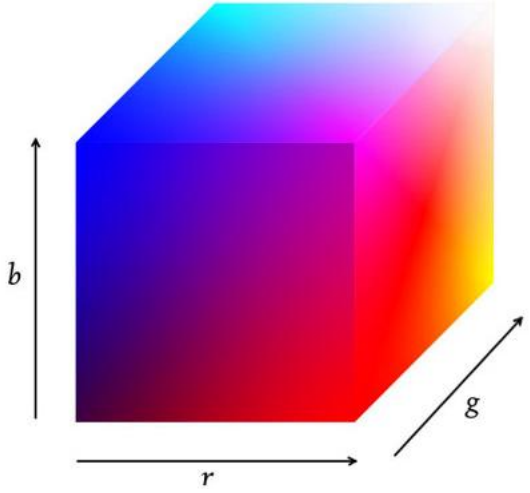


Figure 3.1. RGB color cube

However, the RGB color space is not suitable for color image processing, because of the following three reasons:

(1) The colors R, G, and B have tight relationships among themselves.

(2) It is not easy for an inexperienced user to customize its own desired color.

(3) In computer vision, R, G, and B colors of an object in a digital image highly rely on the reflecting lights of the object, which makes the object discrimination very difficult.

As a result, Smith *et al*. [72] propose an HSV color space, where H, S, and V represent Hue, Saturation, and Value, respectively. Here, Hue indicates the color type, Saturation indicates the color purity, and Value indicates the color brightness. Compared to the RGB color space, the HSV color space makes the object discrimination easier because the information in three channels is relatively independent to each other. In addition, the HSV color space closely models the natural human perception and has been proven to be effective in many previous CBIR research studies. Figure 3.2 shows the HSV color cylinder as an example.

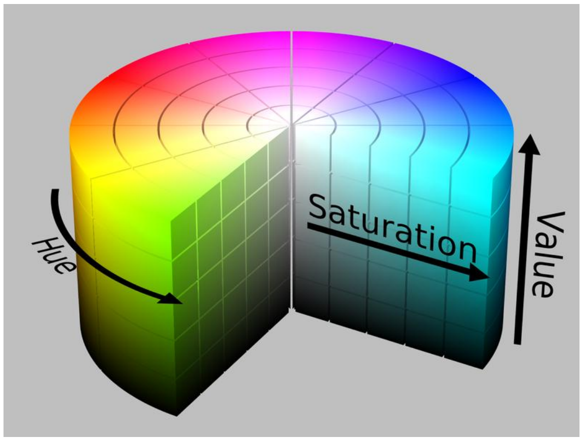


Figure 3.2. HSV color cylinder

Each point in the RGB color space can be mapped into a point in the HSV color space using the following formulas [73].

$$H = \begin{cases} 0 & if \;\; Max = Min; \\ \frac{G-B}{6(Max-Min)}, & if \;\; Max = R, \;\; and \;\; G \geq B; \\ \frac{G-B}{6(Max-Min)} + 1, & if \;\; Max = R, \;\; and \;\; G < B; \\ \frac{B-R}{6(Max-Min)} + \frac{1}{3}, & if \;\; Max = G; \\ \frac{R-G}{6(Max-Min)} + \frac{2}{3}, & if \;\; Max = B; \end{cases} \tag{3.1}$$

$$S = \begin{cases} 0, & if \;\; Max = 0; \\ 1 - \frac{Min}{Max}, & Otherwise; \end{cases} \tag{3.2}$$

$$V = Max \tag{3.3}$$

where *Max* and *Min* are the maximum and minimum value of the R, G, and B components at a point, respectively.

Normalized color histogram is one of the most commonly used color features. The image histogram refers to the count of different image intensities. For a color digital image, the normalized color histogram captures the joint probabilities of the intensities of the three color channels. It is defined as follows:

$$H_{X,Y,Z}(a,b,c) = \frac{1}{N} \times Counts\ (X = a,\ Y = b,\ Z = c) \tag{3.4}$$

where *X*, *Y*, *and Z* respectively represent the three different color channels, such as R, G, and B in the RGB color space, and H, S, and V in the HSV color space, and *N* is the total number of pixels in the image. Computationally, the color histogram is generated by discretizing the color within a digital image and counting the number of the pixels of each color in each bin. Finally, this histogram is normalized in the range of [0, 1] by dividing the total number of pixels (e.g., *N*) in the image.

Figure 3.3 shows an example image and its 64-bin normalized HSV color histogram. Here, the image is discretized into 8 bins for H channel, 2 bins for S channel, and 4 bins for V channel. In the following, I explain the detailed steps to obtain the 64-bin normalized HSV color histogram for the example horse image.

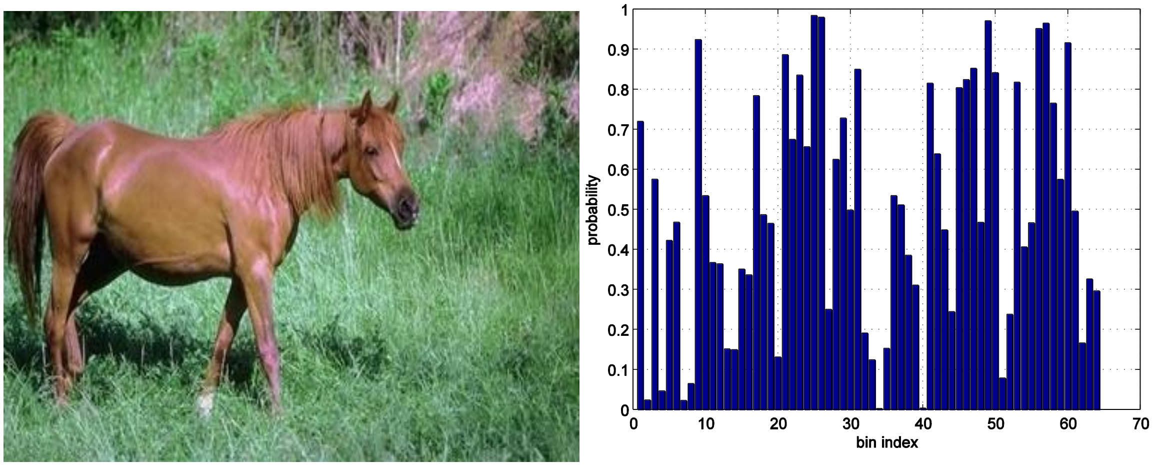


Figure 3.3. Example of a horse and its normalized HSV color histogram

Firstly, the horse image is converted from the RGB color space to the HSV color space using Equations (3.1) through (3.3). Secondly, the values in H channel are discretely scaled into one of the eight integer values ranging from 0 to 7, the values in S channel are discretely scaled into one of the two integer values ranging from 0 to 1, and similarly the values in V channel are discretely scaled into one of the four integer values ranging from 0 to 3. Thirdly, represent each bin in a 64-bin color histogram by a 3-dimensional vector (*Hvalue*, *Svalue*, *Vvalue*) with *Hvalue*, *Svalue*, and *Vvalue* being any of the scaled integer values in the range of [0, 7], [0, 1], and [0, 3], respectively. Examples of these 3-dimensional vectors include (0, 0, 0), (1,0,0),…, (0,1,0),…(7, 1, 3). Fourthly, for each bin corresponding to its unique (*Hvalue*, *Svalue*, *Vvalue*) vector, record the number of pixels whose values in the three channels H, S, and V respectively are *Hvalue*, *Svalue*, and *Vvalue*. One sample result of this counting is shown in Figure 3.4.

Finally, the pixel count in each bin is divided by the total number of pixels in the horse image to obtain its probability. The final 64-bin normalized HSV color histogram is shown in Figure 3.3, which represents the color distribution of an image.

H	S	V	Pixel Count
0	0	0	3,588
1	0	0	120
…	…	…	…
5	1	3	1,910
…	…	…	…
7	1	3	2,188

Figure 3.4. The count of pixels in each bin for the example color image

3.1.2 Color Moments

Color moments are another common color features. The first three order moments of an image are respectively Mean, Variance, and Skewness in each color channel. The first order moment, Mean, measures the average color in each channel. The larger mean value indicates most pixels in the image tend to have the brighter intensity. The second order moment, Variance, measures the spreadness of the color in each channel. A small variance indicates that the pixel intensities tend to be very close to the average intensity while a high variance indicates that the pixel intensities are very spread out from the average intensity. The third order moment, Skewness, measures the shape of the color distribution in each channel. They are computed as follows:

$$MEAN \quad E_i = \sum_{j=1}^{N} \frac{1}{N} p_{i,j} \tag{3.5}$$

$$STANDARD\ DEVIATION \quad \sigma_i = \sqrt{\left(\frac{1}{N}\sum_{j=1}^{N}(p_{i,j} - E_i)^2\right)} \tag{3.6}$$

$$SKEWNESS \quad s_i = \sqrt[3]{(\tfrac{1}{N}\sum_{j=1}^{N}(p_{i,j} - E_i)^3)} \tag{3.7}$$

where $p_{i,j}$ is the intensity of the j-th pixel in the i-th color channel, and N is the total number of pixels in an image.

3.1.3 Edge Direction Histogram

Edge direction histogram is one common edge feature which complements color features to provide more accurate representation of an image. It measures the edge distribution in a digital image. To this end, the color image is converted to a grayscale image. The Sobel edge detector is then applied to the grayscale image to obtain its edge image. A simple example of Sobel edge detector is shown in the following equations.

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \circledast A \tag{3.8}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \circledast A \tag{3.9}$$

where G_x and G_y are respectively two images which are filtered with Sobel filters in horizontal and vertical directions, the operator $\circledast$denotes the 2-dimensional convolution operation and A is the original digital image. Based on G_x and G_y, the gradient direction (or the direction angle) of each pixel's edge is computed by:

$$\Theta = \operatorname{atan}\,(\tfrac{G_y}{G_x}) \tag{3.10}$$

If Θ is 0 for a pixel, it means the edge at this pixel is estimated in a horizontal direction.

Each edge orientation can be quantized into one of the specified bins. For example, if the bin number of the edge histogram is set to be 18, each bin corresponds to the edge orientations in the intervals of 20 degrees. An edge direction histogram can be generated by counting the number of pixels whose edge direction angle falls into each corresponding direction bin. A normalized edge direction histogram is then obtained by dividing the counts in each bin by the total number of pixels in the image.

3.1.4 Discrete Wavelet Transform based Texture Features

Image texture features provide the information about the spatial arrangement of color or intensities in an image [74]. Discrete Wavelet Transform (DWT) becomes popular in the presentation of the texture characteristic of an image in recent research. DWT is capable of removing the redundant texture information of an image, so that the image keeps the core texture information after the transformation. Figure 3.5 shows an example of the 2-level DWT on a source image. Correspondingly, this DWT generates 7 sub-bands, High-High detail subband at level 1 decomposition (HH_1), Low-High detail subband at level 1 decomposition (LH_1), High-Low detail subband at level 1 decomposition (HL_1), High-High detail subband at level 2 decomposition (HH_2), Low-High detail subband at level 2 decomposition (LH_2), High-Low detail subband at level 2 decomposition (HL_2), and approximation subband at level 2 decomposition (LL_2). Here, HH detail subbands are also called diagonal detail subbands since they contain the edge information in the diagonal directions at different resolutions. HL detail subbands are also called horizontal detail subbands since they contain the edge information in the horizontal directions at different resolutions. LH detail subbands are also called vertical detail subbands since they contain the edge information in the vertical directions at different resolutions.

Entropy of any detail subband is a statistical measure of randomness that can be used to evaluate the texture characteristic of an image. For the example in Figure 3.5, the entropy of the six detail subbands can be computed to form a 6-dimensional vector to present the texture characteristic of a digital image. Specifically, the entropy is calculated as the formula defined as follows:

$$E = -\sum p * \log(p) \qquad (3.11)$$

where p is the probability of the values in each sub-band after DWT.

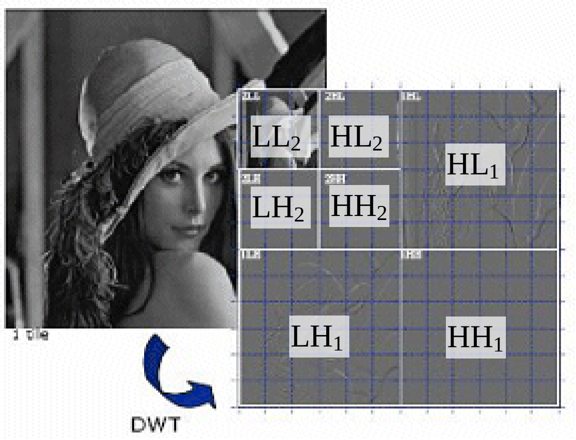


Figure 3.5. An example of 2-level DWT on an image

In my proposed system, I calculate the entropy of the nine detail subbands after applying Daubechies wavelet transform (e.g., db2) on the original grayscale image. The nine entropy values form a 9-dimensional texture feature to represent the texture characteristics of the image.

3.2 Kernel-based Soft Margin SVM Learning

In 1995, Corinna and Vapnik introduced a new supervised SVM learning technique [75] for data classification and regression analysis. Since the birth of the SVM, this technique has rapidly gained the attention in computer vision to solve the handwriting recognition problem.

SVM has several advantages:

(1) It uses a subset of training points, also called support vectors, in the decision function to save memory requirement.

(2) It can powerfully solve the high-dimensional data point classification.

(3) It can use different kernel functions to generate various decision functions.

Generally, SVM requires *N* training data in a set ***D*** shown as follows.

$$\boldsymbol{D} = \{(\boldsymbol{X_i}, y_i) |\ i = 1\ to\ N,\ \ \boldsymbol{X_i}\,\epsilon\mathbb{R}^p,\ \ y_i\ \epsilon\ \{+1, -1\}\} \tag{3.12}$$

where $\boldsymbol{X_i}$ is a *p*-dimensional features vector, and y_i is the label indicating which class $\boldsymbol{X_i}$ belongs to. If ***D*** is linearly separated, a linear hyper-plane classifier separating the training data points into the positive and negative classes can be written as a set of points $\boldsymbol{X_i}$ satisfying the following.

$$\boldsymbol{H} = \{\boldsymbol{X}\ | g(\boldsymbol{X}) = \boldsymbol{W}^T\boldsymbol{X}\ + b\ = 0\} \tag{3.13}$$

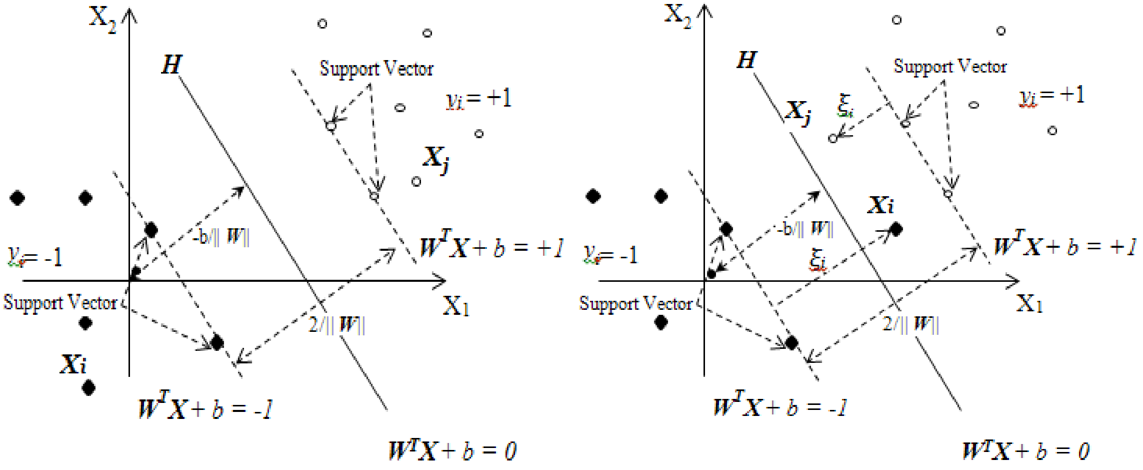
where *g(**X**)* is a discriminant function, ***W*** is a normal vector perpendicular to ***H***, *b* is a constant, and–*b*/||***W***|| is the offset of the hyper-plane from the origin (see Figure 3.6(a)). For any positive data point $\boldsymbol{X_i}$ in ***D***, $g(\boldsymbol{X_i}) \geq 0$; similarly, for any negative data point $\boldsymbol{X_j}$ in ***D***, $g(\boldsymbol{X_j}) \leq \boldsymbol{0}$.

The two hyper-planes that are parallel to $\boldsymbol{H}$ separate the data points without any points lying between them and maximize the margin between them. They are defined as follows:

$$\boldsymbol{W}^T\boldsymbol{X} + b = +1 \tag{3.14}$$

$$\boldsymbol{W}^T\boldsymbol{X} + b = -1 \tag{3.15}$$

The data points $\{X_i, Y_i\}$ for which the equalities hold are called support vectors as marked in Figure 3. 6 (a).



(a) (b)

Figure 3.6. An example of 2-D hyper-plane for a binary classification (a) without wrong classification and (b) with wrong classification

In most real pattern classification applications, training data points cannot be completely linearly separated. In this case, the hyper-plane allows the existence of mislabeled data points (See $\boldsymbol{X_i}$ and $\boldsymbol{X_j}$ in Figure 3.6 (b)). Thus, the soft margin SVM provides a constraint by incorporating *n* non-negative variables ξ_i.

$$y_i g(\boldsymbol{X_i}) = y_i(\boldsymbol{W}^T\boldsymbol{X_i} + b) \geq 1 - \xi_i,\ \ \xi_i \geq 0,\ 1 \leq i \leq N \tag{3.16}$$

To find the optimal hyper-plane, an approximated cost function is used.

$$\Phi(\boldsymbol{W},\xi) = \min_{\boldsymbol{w},\xi,b}\max_{\boldsymbol{\alpha},\boldsymbol{\mu}}\left\{\frac{1}{2}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W} + C\sum_{i=1}^{N}\xi_i - \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}\alpha_i(y_i(\boldsymbol{W}^{T}\boldsymbol{X_i} + b) - 1 + \xi_i) - \sum_{i=1}^{N}\mu_i\,\xi_i\right\} \qquad (3.17)$$

where C is a positive regularization constant, which represents the tradeoff between error and margin. It is also known as *slack penalty*. $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ are *Lagrange Multipliers*.

With the aid of the quadratic programming, the final model to find the hyper-plane in the soft margin SVM is as follows:

$$\begin{cases} \quad \text{Maximize } Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{X^T}_{\boldsymbol{i}}\boldsymbol{X_j} \\ \text{Subject to: } \sum_{i=1}^{N}\alpha_i y_i = 0 \quad \text{and } 0 \le \alpha_i \le C, \ \text{ for } i = 1,2,\dots,N \end{cases} \qquad (3.28)$$

When training data can not easily be separated in the *p*-dimensional feature space, mapping them to a higher dimensional feature space makes the classification task easier (see Figure 3.7). Data points in *p*-dimensional feature space are mapped into a higher-dimensional feature space by a transform function φ (.).

$$\boldsymbol{Z_i} = \varphi(\boldsymbol{X_i}) \qquad (3.18)$$

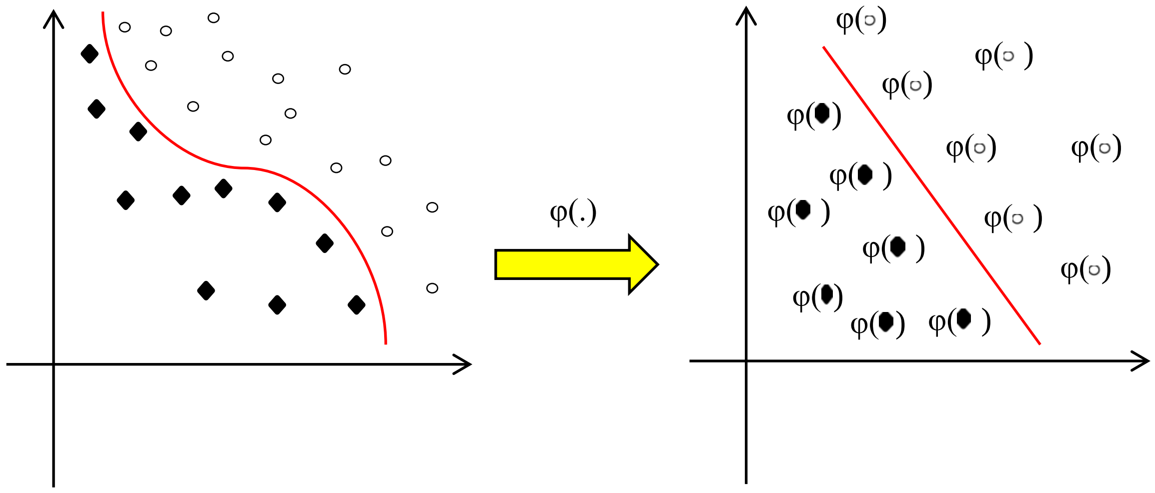


Figure 3.7. Mapping data points from the low-dimensional feature space to the high-dimensional feature space

Similarly, by employing *Lagrange Multiplier* and quadratic programing, the final model to define the hyper-plane is:

$$\begin{cases} \text{Maximize } Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{Z^T}_{\boldsymbol{i}} \boldsymbol{Z_j} \\ \text{Subject to: } \sum_{i=1}^{N} \alpha_i y_i = 0 \quad \text{and } 0 \le \alpha_i \le C, \text{ for } i = 1,2,\dots,N \end{cases} \tag{3.19}$$

where $\boldsymbol{Z_i}^{\boldsymbol{T}}\boldsymbol{Z_j}$ can be represented as a kernel function *K(.)*.

$$\boldsymbol{Z_i}^{\boldsymbol{T}}\boldsymbol{Z_j} = \varphi(\boldsymbol{X_i})^{\mathrm{T}} \varphi(\boldsymbol{X_j}) = K(\boldsymbol{X_i}, \boldsymbol{X_j}) \tag{3.20}$$

Due to its decent classification performance, the most popular kernel function is Gaussian Radial Basis Function (RBF) that is defined as follows:

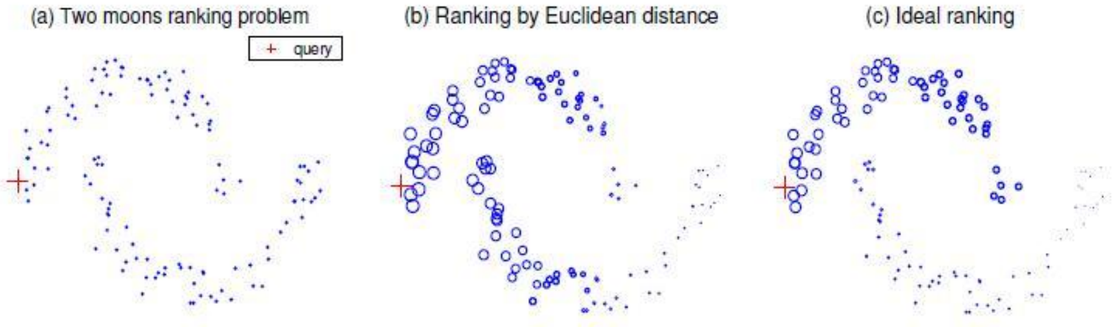
$$K(\boldsymbol{X_i}, \boldsymbol{X_j}) = \exp(-\frac{\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2}{2\sigma^2}),\ \sigma > 0 \tag{3.21}$$

where σ is selected a priori parameter.

3.3. CBIR Systems Based on Conventional Manifold Techniques

Traditional CBIR systems use a pair-wise perceptual similarity measure (e.g., Euclidean distance) to measure the similarity between the query image and each database image. On the other hand, the manifold-ranking-based CBIR systems rely on a relevance measure between the query image and database images to explore the relevance relationship of all data points in the given feature space and propagate ranking scores of labeled images to unlabeled images via a weighted graph.

Many real-world data are more suitably represented in a global manifold structure space rather than in other distance based structure spaces, such as Minwoski distance-based structure spaces. Figure 3.8 presents a toy example to reveal the suitability of the manifold structure. In this example, a set of points form a two-moon pattern. Suppose that a query in the upper moon is given, the task is to rank the remaining points according to their relevance to the query. We may easily claim that all points in the upper moon are more relevant to the query than points in the lower moon. However, if we measure the similarity of points to the query in the Euclidean space, the lower left points in the lower moon are more similar to the query than the upper right points in the upper moon. Obviously, this result as shown in Figure 3.8 (b) is not satisfactory based on human's perception.



(a) (b) (c)

Figure 3.8. A toy example to illustrate the advantage of the manifold technique: (a) Toy data set with a single query marked by a red plus sign; (b) Euclidean distance-based ranking result; (c) Human perception-based ideal ranking result. In both (b) and (c), larger empty dots represent the ranking results.

In the following sections, I explain the basic manifold ranking technique and its three variations.

3.3.1 The Conventional Manifold Ranking Technique

Ranking data in the manifold structure belongs to semi-supervised learning [76]. Given an assumption that each data point in a certain feature space has a relationship to other data points in the same space, there should be an edge to connect each pair of points, where the edge is assigned a weight to represent how relevant the two data points are. Therefore, the system first constructs such a weighted graph for all data points in the feature space. Then, each query data point is initially assigned a ranking score, and the remaining data points, whose relevance is unknown with respect to the query data points, are assigned zero to be their ranking scores. Second, all data points spread their ranking scores to their neighboring data points via the weighted graph. The propagation of the ranking scores iteratively runs until it converges to a global stable status. All data points in the database eventually have their own final ranking scores after the propagation reaches its convergence. These final ranking scores represent the similarities between each data point and the query points. The data points that are similar to the query points are the ones having the largest ranking scores.

Zhou *et al.* [77] provided an explanation of this basic manifold ranking algorithm. Given a set of points $\chi=\{x_1, \dots, x_q, x_{q+1}, \dots, x_n\} \subset \mathbb{R}^p$, where n is the number of points. The first q points are the queries, and the rest are the points to be ranked according to their relevance to the queries. For each pair of points x_i and x_j, a distance $d(x_i, x_j)$ is defined as $d: \chi \times \chi \rightarrow \mathbb{R}$, which is a metric on the point set χ. This distance could be Euclidean distance or Manhattan distance, etc. Correspondingly, every point x_i has a ranking value f_i defined as $f_i = f(x_i)$,where $f: \chi \rightarrow \mathbb{R}$ denotes a ranking function. Finally, a vector $y = [y_1, \dots, y_n]^T$ is defined, in which $y_i = 1$ if x_i is a query, and $y_i = 0$, otherwise.

The algorithmic view of the basic manifold-based ranking scheme is summarized below in Figure 3.9 [77].

1. Sort the pair-wise Euclidean distance among points in the ascending order. Repeatedly connecting the two points with an edge according to the order until a connected graph is obtained.
2. Form the affinity matrix W defined by
$$W_{ij} = \exp[-d^2(x_i, x_j)/2\sigma^2] \qquad (3.22)$$
where $d(x_i, x_j)$is the Euclidean distance between point x_i and point x_j, and σ is the overall standard deviation of χ. Note that $W_{ii} = 0$ because there are no loops in the graph.
3. Symmetrically normalize W by $S = D^{-1/2}WD^{-1/2}$ in which D is the diagonal matrix with (i, i)-element being the sum of the i-th row of W.
4. Iterate $f(t+1) = \alpha S f(t) + (1-\alpha)y$ until it convergences, where α is a parameter in [0,1).
5. Let f_i^* denote the limit of the sequence $\{ f_i(t)\}$. Rank each point x_i according to its ranking scores f_i^* (largest ranking scores will be ranked first).

Figure 3.9. The algorithmic view of the conventional manifold ranking technique

First of all, a graph is constructed to connect all points in the database. Then the edges in this graph are assigned corresponding weights by Equation (3.22). The normalization is performed in step 3 to ensure convergence. Afterwards, the points are ranked according to their final ranking scores. Here, parameter α specifies the contribution to the ranking scores from its neighbors, and (1- α) specifies the contribution to the initial ranking scores. The ranking score is propagated symmetrically because S is a symmetric matrix.

According to Cox *et al* [78], the sequence $\{ f(t)\}$ in step 4 converges to

$$f^* = \beta(1-\alpha S)^{-1}y \qquad (3.23)$$

where β is a common scaling factor for every point when calculating the ranking score and is set to be *1-* α. As a result, β can be skipped in computing the ranking score, and Equation 3.23 can be simplified by

$$f^* = (1-\alpha S)^{-1}y \qquad (3.24)$$

Variant manifold ranking techniques modify step 2 or step 4 or both of them. In the following subsections, I explain several representative variations of the manifold systems that exclusively modify step 4.

3.3.2 Variation 1: Propagating with Only Positive Feedback

In the above basic manifold ranking algorithm, the ranking scores spread iteratively until a final global stable status is achieved. In each iteration, the system integrates users' feedback for the next iteration of the ranking score propagation. When users only submit positive feedback for returned examples, or when only relevant images to the queries are concerned, the newly returned positive examples are added into the query set, and the ranking score propagation will repeatedly refine the retrieval results. To this end, Equation (3.24) can be revised as follows:

$$f^* = (1 - \alpha S)^{-1} y = (1 - \alpha S)^{-1} \sum_{i=1}^{n^{+}+1} y^{i} \qquad (3.25)$$

where y^i is an *n*-dimensional vector with the *i-th* component equal to 1 and others equal to 0, and n^+ is the number of positive feedback examples. In other words, non-zero components in y correspond to positively labeled returned images and contribute to the spreading of ranking scores in the propagation process.

3.3.3 Variation 2: Propagating with Positive and Negative Feedback

Since users' feedback probably contain both positive and negative judgments for the retrieved examples, some manifold systems use both information to propagate the labels based on the following two observations:

1) Relevant images tend to form certain clusters in the feature space.

2) Irrelevant images may form some other clusters with different semantic meanings.

These systems consider that the knowledge learned from both relevant and irrelevant images is helpful to refine retrieval results and achieve a decent final retrieval result. To accommodate positive and negative feedbacks, two vectors y^{+} and y^{-} are introduced. The first vector is similar to the previously defined vector $y^{+} = [y_1^{+}, \dots, y_n^{+}]^T$, whose elements are set to 1's if the corresponding image is the query itself or a positively labeled returned image. The elements in the second vector $y^{-} = [y_1^{-}, \dots, y_n^{-}]^T$ are set to -1's if the corresponding image is a negatively labeled returned image. All the remaining elements in both vectors are set to 0's. Equation (3.25) can be refined as follows:

$$f^{*} = f^{+^{*}} + f^{-^{*}} = Ay^{+} + Ay^{-} \qquad (3.26)$$

where $A = (1 - \alpha S)^{-1}$, $f^{+^{*}}$ and $f^{-^{*}}$ are the ranking scores obtained from the positive and negative feedback, respectively.

3.3.4 Variation 3: Propagating with Weighted Positive and Negative Feedback

Furthermore, different weights can be applied to Equation (3.26) based on the following two observations:

1) The farther an unlabeled image lies from positive examples in the feature space, the less likely it is positive.
2) If an unlabeled image lies far from negative examples in the feature space, its likelihood of being positive is uncertain, since it may not be close to positive examples either.

As a result, positive examples generally make more contributions to the final ranking scores than negative examples, and Equation (3.26) can be refined as follows:

$$f^{*} = f^{+*} + \gamma * f^{-*} = Ay^{+} + \gamma * Ay^{-} \qquad (3.27)$$

Here, parameter $\gamma \in (0,1]$ weakens the contribution of negative ranking scores to f^{-*}. The smaller the γ , the less impact negatively labeled examples in the final ranking scores. When $\gamma = 1$, negatively labeled examples make the same contribution as positively labeled examples in the propagation of ranking scores. The system becomes the second variation system as explained in section 3.3.3.

From the above analysis of three variant manifold systems, it's obvious to claim that the variant 3 is better than other 2 variants because it utilizes both positive and negative feedback from the user, and treats these two kinds of feedback with different emphasis in the ranking score calculation. Therefore, my proposed system is built upon the foundation of the variant 3.

CHAPTER 4

A SCALABLE MANIFOLD GRAPH-BASED CONTENT-BASED IMAGE RETRIEVAL APPROACH

In this chapter, I propose a novel scalable graph-based ranking system for CBIR. This proposed system extends the short-term learning by utilizing the RF information gathered from the past retrieval sessions to hierarchically explore the relationship of all database images in both low-level visual feature space and high-level semantic feature space. It treats both labeled and unlabeled images as vertices in their respective graph and builds pairwise edges between these vertices, which are weighted by both visual and semantic affinities between the corresponding image pairs. The small portion of vertices carrying seed labels (e.g., the users' RF information) is then harnessed via information propagation to predict the labels of the unlabeled vertices (images). Positively predicted images are finally returned as the retrieval results. Specifically, the proposed system first learns semantic features of each database by using the users' historical RF. It then builds a two-layer manifold graph ranking system which models the intrinsic structure for the image space in several manageable small scales. The first layer manifold graph ranking system is constructed using both low-level visual similarity and high-level semantic similarity of the anchor images in the database. These anchor images are chosen based on the users' RF. They normally contain key semantic concepts of the image database. The number of anchor images approximately corresponds to the number of semantic concepts contained in all images in a database. The second layer manifold graph ranking system is constructed based on the clusters formed around anchor images. For each cluster, both low-level visual and high-level semantic similarities of the images in the cluster are integrated to construct its manifold graph to achieve a more meaningful structure in the image space. The size of these graphs is significantly smaller compared to the size of the traditional manifold graph, which makes the proposed system scalable. Finally, an asymmetric relevance vector is created for each second layer graph by assigning initial scores from the first layer graph. This vector then propagates the relevance scores of labeled images to unlabeled images via the hierarchical graph-based structure. In the proposed RF-based CBIR system, the training and retrieval processes have the following advantages over other common RF-based CBIR systems:

- Quick construction of a compact dynamic feedback log to store unique retrieval patterns (i.e., the similarity of relevant and irrelevant images) of historical query sessions.
- Efficient merging of similar retrieval patterns to maintain a reasonable number of meaningful semantic concepts to represent all images in a database.
- Creative construction of two-layer hierarchical graphs to represent the inherent structure of the large-scale image database.
- Effective composition of low-level visual and high-level semantic similarity measure to build the manifold graph, which explores the intrinsic structure of images in both low-level visual feature space and high-level semantic feature space.
- Effective layered design of the relevance vector to propagate the relevance scores from anchor images to the second layer graphs and further propagate the relevance scores of labeled images to unlabeled image via the hierarchical graph-based structure.

The rest of this chapter is organized as follows: Section 4.1 presents the proposed scalable graph-based ranking system. Section 4.2 compares the proposed CBIR system and its variant systems with four manifold-based CBIR systems and five representative long-term-based CBIR systems on five databases. Section 4.3 draws conclusions and presents future directions.

4.1 The Proposed Scalable Graph-Based CBIR Approach

The proposed scalable graph-based CBIR approach consists of offline training and online retrieval phases, which are demonstrated in Figure 4.1. One of the aims of offline training is to collect users' historical RF to learn semantic features of each database image. Specifically, SVM active learning is first applied to select the most informative unlabeled positive images based on the decision boundary learned from user's positively and negatively labeled images. The relevancy information for each retrieved image in each query session is stored in a dynamic feedback log, which is updated after each query session. To this end, the relevancy information of the current query may be iteratively merged with the relevancy information of past query sessions if they contain sufficient overlapping information. The final merged relevancy information is then used to update the feedback log. On the other hand, the relevancy information of the current query may contain unique information which is not present in the past query sessions. This new relevancy information is then appended to the feedback log. After all query sessions have been performed, this dynamic feedback log holds the semantic information of each database image. Another aim of offline training is to build a scalable graph-based ranking system for future retrievals. To this end, anchor images are first

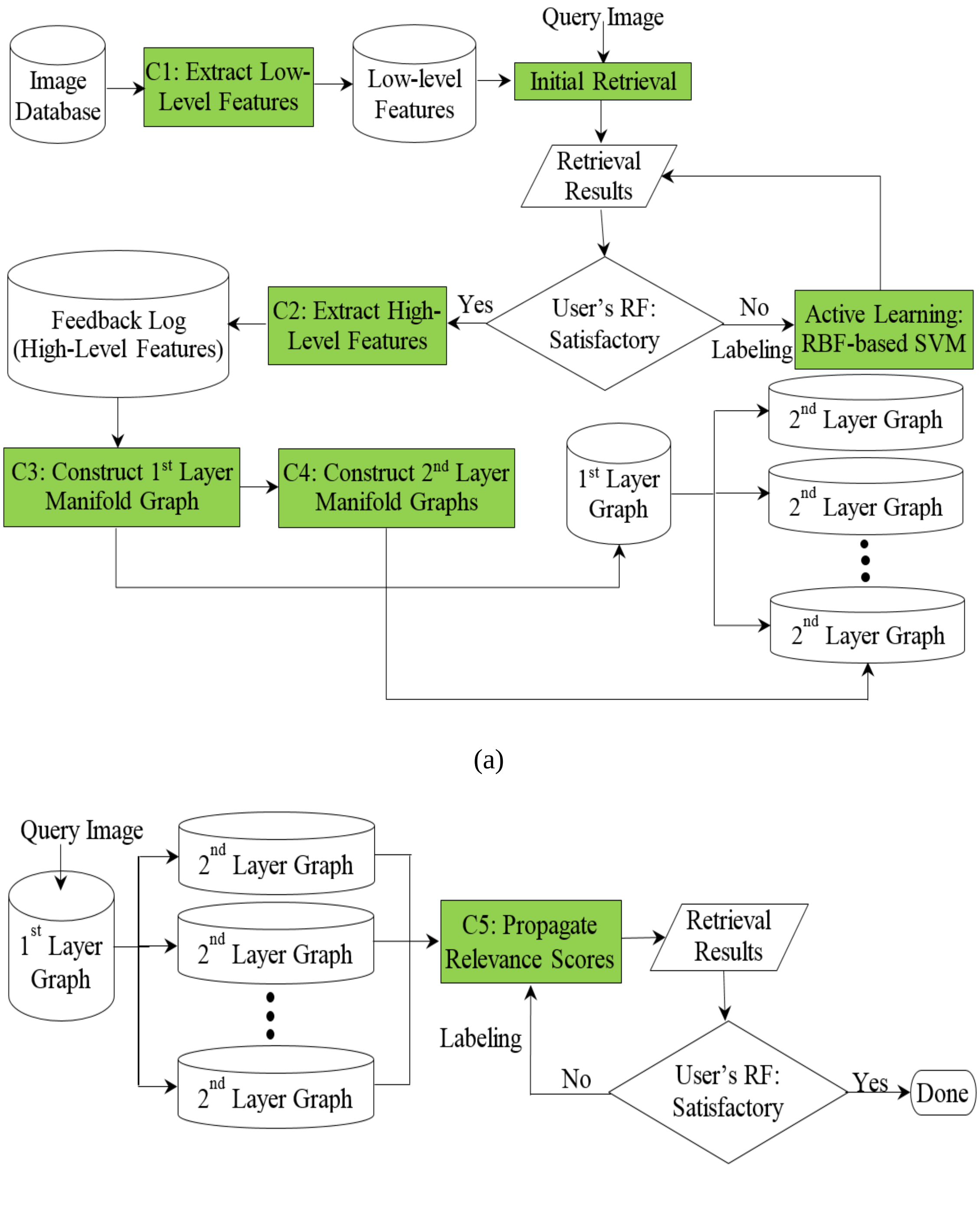


Figure 4.1. Block diagram of the proposed system: (a) Offline training phase, (b) Online retrieval phase.

located based on the feedback log. A cluster is formed around each anchor image and any image outside of any cluster is assigned to an appropriate cluster using the minimum-distance-based strategy. In this way, the image database is divided into several clusters (categories). Finally, the first layer manifold graph is constructed by incorporating low-level visual and high-level semantic features of all anchor images. Several second layer manifold graphs are also constructed by using low-level visual and high-level semantic features of all images in their respective clusters. The online retrieval phase focuses on designing a strategy to asymmetrically propagate the relevance scores of labeled positive and negative images through the hierarchical manifold graphs. Specifically, the first layer graph is capable of quickly identifying the potential clusters that a query image belongs to and propagating its relevance scores to the second layer graphs. The final relevance scores can then be propagated to unlabeled images via the hierarchical manifold structure. In the following, I explain the major components of each phase in detail.

4.1.1 Offline Training Phase

The ultimate goal of the offline training process is to construct the hierarchical scalable graph-based structure of the image database which stores the learned relationship between each image pair. The algorithmic view of the offline training phase is summarized in Figure 4.2.

```
Input: All images in the database
Output: Two-layer hierarchical graphs.
  1. Apply "Extract Low-level Features" on each image in the image database to represent images from the visual perspectives.
  2. Randomly choose 10% of database images as training images to perform the training task.
  3. For each training query image,
     3.1. Perform "Initial Retrieval" to return top v relevant images
     3.2. Allow the user to select relevant (i.e., positive) images from the retrieved images
     3.3. Treat non-selected images as irrelevant (i.e., negative) images.
     3.4. Apply Active Learning (e.g., RBF-based SVM) on the accumulated positive and negative images to find a better classification boundary to discriminate positive images from negative images in the database.
     3.5. Return top v relevant images based on the distance to the classification boundary.
     3.6. Repeat step 3.2 through step 3.6 for a few feedback iterations until the query session finishes (i.e., the maximum number of iterations is achieved or the user is satisfied with the retrieval results).
     3.7. Store the relevancy information for each retrieved image in the current query session in a dynamic feedback log.
  4. Apply "Extract High-Level Features" on the dynamic feedback log to obtain high-level semantic features for each database image.
  5. Apply "Construct 1st Layer Manifold Graph" on anchor images obtained from the feedback log to obtain one manifold graph.
  6. Apply "Construct 2nd Layer Manifold Graphs" on clusters around each anchor image to obtain several manifold graphs.
```

Figure 4.2. The algorithm view of the offline training phase

4.1.1.1 Extract Low-Level Features: All three important features, e.g., color, edge, and texture features, are utilized to represent each image in the database. The proposed system uses a 100-dimensional vector to represent low-level features of an image. These global features were proven to be effective in Qi and Chang's work [79], and they are easy to compute and complementary to each other. Specifically, the 100-

dimensional vector includes a 64-bin ($8\times2\times4$) HSV-based color histogram and a 36-dimensional complementary feature vector [66], which contains 9 color, 18 edge, and 9 texture components. To this end, it computes the first three moments in each HSV color channel to represent color features; computes the 18-bin edge direction histogram in the grayscale image to represent the edge features; and computes the entropy of each of nine detail subbands of a 3-level wavelet transform to represent texture features in the grayscale image. These extracted low-level features are then globally normalized into [0, 1] by the linear scaling to unit range technique [80]. This technique first finds the lower bound *l* and the upper bound *u* for a feature component *x*. It then normalizes *x* by:

$$\tilde{x} = \frac{x-l}{u-l} \tag{4.1}$$

4.1.1.2 Initial Retrieval: For initial retrieval, the Euclidean distance is applied on the normalized low-level features to measure the similarity between the query image *Iq* and each database image *Ii* by:

$$Sim = \| LVF(I_q) - LVF(I_i) \|_2 = \sqrt{\sum_{k=1}^{n} \left(LVF(I_q(k)) - LVF(I_i(k)) \right)^2} \tag{4.2}$$

where *LVF(Ii)* represents normalized low-level visual features of an image I_i, *LVF(Iq)* represents normalized low-level visual features of the query image *Iq*, *LVF(Ii(k))* represents the *k*-th value of normalized low-level visual features of an image *Ii*, *LVF(Iq(k))* represents the *k*-th value of normalized low-level visual features of the query image *Iq*, and *n* is the dimensionality of normalized low-level visual features. An image with a smaller distance to the query image is more similar to the query image. According to the Euclidean distances of all images to the query images, top *v* images which are most similar or relevant to the query image are retrieved for the user to provide RF information.

4.1.1.3 Active Learning: RBF-Based SVM: The aim of SVM-based active learning is to apply the statistical active learning technique to refine the decision boundary after each RF iterative step to find relevant images more accurately. After initial retrieval, the relevant (positive) images among top *v* images are marked by the user. The non-marked retrieved images are automatically considered as the irrelevant (negative) images. For the remaining iterations, the Gaussian Radial Basis Function (RBF) kernel-based SVM is applied on the accumulated positively and negatively labeled images to find a classification boundary. The distance from a database image to the classification boundary is used to measure the similarity between the query image and each database image. Top *v* images which have the largest positive distances to the classification boundary are returned to the user for the next round of labeling. This process continues for a few iterations till the maximum number of iterations is reached. A query session is completed at this time as well.

In order to speed up the initial learning and maximize the amount of the semantic relationship information that could be learned on the training set, a retrieved image will not be returned in the following iterations during a query session. To limit the number of user interactions, 25 images are returned at each iteration because these images can be easily fit into one screen for the user's RF. In addition, four iterations are set as the maximum number of iterations in the training phase.

4.1.1.4 Extract High-Level Features: Each image is also represented by high-level semantic features, which are learned from the users' historical RF. Since high-level semantic features are closely related to the high-level semantics of an image, I also call semantic features as high-level semantic features. The more images in the database, the more possible semantic concepts are. So I fix the maximal length of high-level semantic features (e.g., maximal semantic concepts) to be linked with the total number of images in the database. In the proposed system, I initially confine the maximal length of high-level semantic features for a database image to be 10% of the total number of images in the database, which is a reasonable and conservative estimate for the maximal number of semantic concepts contained in all images in a database. These high-level semantic features are directly constructed from the dynamic feedback log $\boldsymbol{R}$. This $\boldsymbol{R}$ is a 2-D matrix whose row number equals to the total number of images in the database (e.g., N) and column number starts with 0 and is updated with each training query. It should be noted that the number of training queries equals to the confined maximal length of high-level semantic features (i.e., 10% $\times N$) so sufficient learning can be achieved to learn the semantic features of each image.

After each query session, the system creates a candidate column with all 0's. It then marks the cells corresponding to the rows of positive images as 1's and marks the cells corresponding to the rows of negative images as -1's. A merging technique is then carried out to iteratively combine this candidate column with other similar columns in $\boldsymbol{R}$. If no column in $\boldsymbol{R}$ is similar to this candidate column, the merging operation does not take place and the candidate column is added as the last column in $\boldsymbol{R}$. The basic idea of this merging strategy is as follows:

1) The candidate column is sequentially compared with each column in $\boldsymbol{R}$.
2) If the candidate column is similar to an existing column in $\boldsymbol{R}$, these two columns are combined to form a new candidate column by performing an addition operation.

3) This newly merged candidate column is continuously examined against the remaining columns in ***R*** until there is no merging operation occurs.

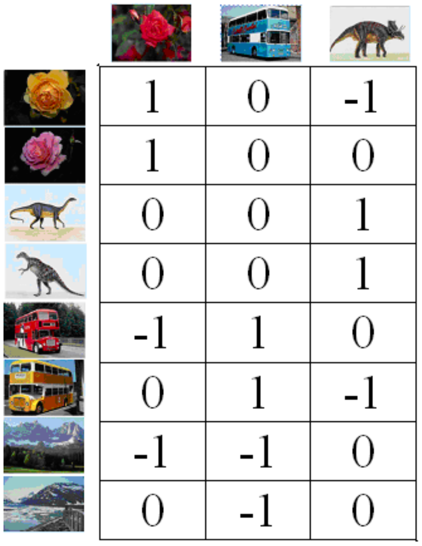
All the columns that have been iteratively merged with the candidate column are deleted from ***R***. The final merged candidate column is then added as the last column in ***R***.

The algorithmic view of this iterative merging strategy is shown in Figure 4.3.

```
Input: Dynamic feedback log $\boldsymbol{R}$ and the new column $c_{new}$
Output: Updated $\boldsymbol{R}$
    1. Put the IDs of cells of 1's in $c_{new}$ into a set A
    2. Generate a vector $VecID$ with all 0's. The length of $VecID$ is $m$,which is the total
       number of columns in $\boldsymbol{R}$.
    3. For each column $c_i$ ($1 \leq i \leq m$) in $\boldsymbol{R}$
       3.1. Put the IDs of cells of positive values in $c_i$ into a set B
       3.2 If |intersect (A, B)| > 0.5 ×min(|A|, |B|)
               Merge $c_{new}$ and $c_i$ by $c_{new} + c_i$
                       Update $c_{new}$ with the newly merged results
               Update the $i$-th element of $VecID$ as 1's. That is $VecID(i) = 1$.
                   Endif
      Endfor
    4. For each element in $VecID$ whose value is 1, remove its corresponding column from
       $\boldsymbol{R}$.
    5. Append $c_{\text{new}}$ to the last column of $R$.
```

Figure 4.3. The algorithm view of the iterative merging strategy

Figure 4.4 shows an example of ***R*** for a database with eight images after three query sessions. In this example, four images are returned for each query and the user gives two positive feedbacks and two negative feedbacks for each query. The three columns constructed from the three query sessions are not similar to each other, so they cannot be merged. As a result, ***R*** has eight rows and three columns. For each column representing each query session, the elements corresponding to positive feedback is set as 1's and the elements corresponding to negative feedback is set as -1;s. The other elements corresponding to the non-returned images are set as 0's. This ***R***, an 8×3 matrix, stores the semantic information of the database image. Specifically, each row stores the semantic information of each database image. Each column stores the learned semantic information for a particular query. In this example, the first row indicates that the first image has the same semantic information as the first query (e.g., flower) and does not have any semantic information regarding the third query (e.g., dinosaur). The first column indicates that the current query (e.g., flower) shares the same semantic information as the first and second database images (e.g., flower) and does not have any semantic information regarding the fifth and seventh database images (e.g., bus and mountain). The value of 0 at (*x*, *y*) location in ***R*** indicates that nothing has been learned about the relationship between the *x*-th database image and *y*-th query's concept.



1	0	-1
1	0	0
0	0	1
0	0	1
-1	1	0
0	1	-1
-1	-1	0
0	-1	0

Figure 4.4. An example of ***R*** for an eight-image database after three distinct query sessions.

Figure 4.5 shows an example of the merging process. To ease discussion, I only show the relevant images together with their corresponding values in each column since these relevant images are used to decide whether the merging process takes place. In this example, there are four images (i.e., fireworks) marked as relevant in the new column c_{new}, as shown in the top row. An existing column c_i has six images (i.e., fireworks) marked as relevant (e.g., the cells corresponding to these six images contain positive values), as shown in the middle row. The merging process is carried out to count the overlapping relevant images that coexist in both c_{new} and c_i. Here, two relevant images coexist in both columns. This number is a half of the total number of images in c_{new}. As a result, these two columns should be merged by the addition operation described in Figure 4.3. The newly merged column containing merged relevant images is shown in the bottom row. It should be noted that the cells in the newly merged column corresponding to the irrelevant images in c_{new} and c_i are updated by the addition operation. This newly merged column is further compared with the remaining existing columns to perform the same merging process.

After performing the query session for all the training images, ***R*** holds the possible semantic information for each database image. The column number of ***R*** equals to the number of learned semantic concepts (e.g., foreground objects or background implicitly marked by the users as a set of relevant images in the RF step). High-level semantic features of an image correspond to the respective row in ***R***. Each value in ***R*** represents the relationship between a database image and the semantic concept corresponding to the respective queries encapsulated in the corresponding column. For example, the first row in ***R*** represents semantic features of the first database image. If the

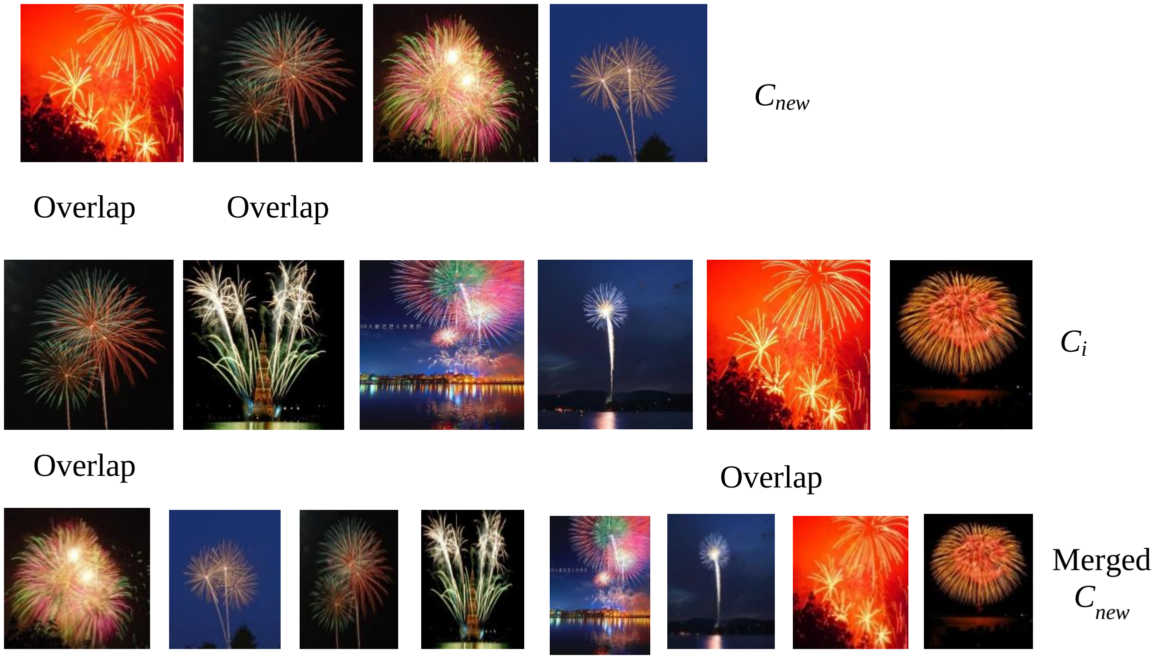
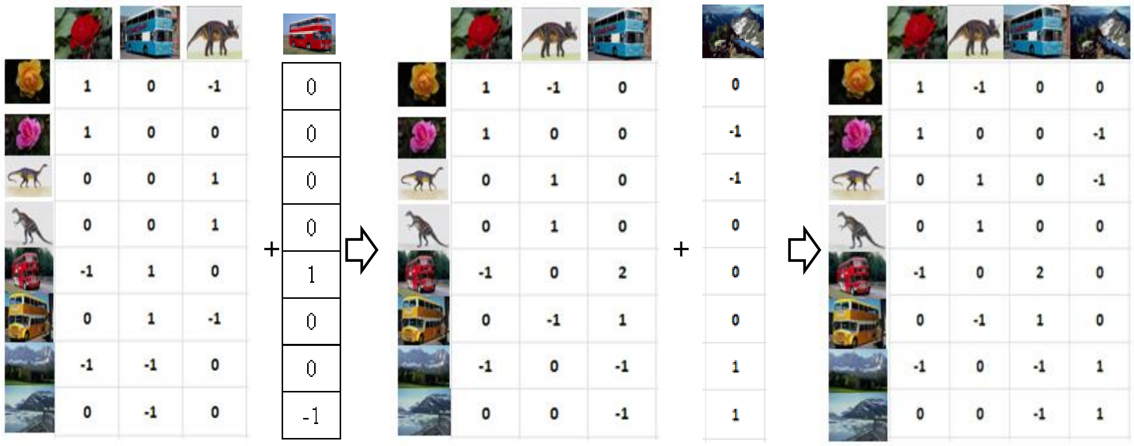


Figure 4.5. An example of merging a new column with an existing column

first column in ***R*** represents the semantic concept (e.g., sky and mountain) of the first (merged) query, the value at (1, 1) in ***R*** means the relevance of the first database image to the sky and mountain concepts. A larger positive value indicates the database image likely to possess the corresponding semantic concept. A smaller negative value indicates the database image unlikely to possess the corresponding semantic concept.

Figure 4.6 shows an example ***R*** that performs the merging when two new query sessions are added on the eight-image database. ***R*** initially is filled with +1's and -1's based on the user's RF after three query sessions. When the first new query session (i.e., bus) is added, the existing second column in the ***R*** is qualified to be merged with the new query session. After merging, the new merged column with new values is appended as the last column in ***R*** and the old second column is removed correspondingly. When the second new query session (i.e., mountain) is added, there is no existing column is qualified to be merged. As a result, the new query session with the user's RF is automatically appended to ***R***.

To this end, the number of learned semantic concepts equals to the number of columns (e.g., 4), when all query sessions finish during the offline training phase. Each row in ***R*** represents a high-level semantic-based feature of the corresponding image.



$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & 1 & -1 \\ -1 & -1 & 0 \\ 0 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \\ 0 & -1 & 1 \\ -1 & 0 & -1 \\ 0 & 0 & -1 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 2 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Figure 4.6. An example of the compact ***R*** after merging

The high-level semantic relevance relation $S_{i,j}$ between images *i* and *j* is computed by the semantic-correlation-based distance:

$$S_{i,j} = HSF_i \bullet HSF_j = \sum_{k=1}^{p} HSF_i(k) \times HSF_j(k) \tag{4.3}$$

where HSF_i and HSF_j respectively represent semantic features of images *i* and *j*, $HSF_i(k)$ and $HSF_j(k)$ respectively are the k^{th} element of semantic features of images *i* and *j*, *p* is the dimensionality of semantic features of each image (i.e., the number of columns in ***R***), and the $\times$ operation is defined as follows:

$$HSF_i(k) \times HSF_j(k) = \begin{cases} HSF_i(k) \times HSF_j(k) & \text{if } HSF_i(k) > 0, HSF_j(k) > 0 \text{ or } HSF_i(k) \times HSF_j(k) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

This operation yields positive results when $HSF_i(k)$ and $HSF_j(k)$ are positive values (i.e., both images have the k^{th} semantic meaning represented in the k^{th} column of $\boldsymbol{R}$), yields negative results when $HSF_i(k)$ and $HSF_j(k)$ have different signs (i.e., one image has the k^{th} semantic meaning and the other image does not have the k^{th} semantic meaning), and yields 0's otherwise (i.e., both images do not have the k^{th} semantic meaning or no semantic meaning is learned for either of the two images or both).

4.1.1.5 Construct the First Layer Manifold Graph: The first layer manifold graph is constructed from $\boldsymbol{R}$. Suppose that there are p columns in $\boldsymbol{R}$ after performing the query session for all training images, the proposed system sequentially investigates each of p columns to find its anchor image. To this end, it first records the IDs of the images that have positive values in the corresponding column. It then computes the centroid of these images (i.e., the average of their low-level visual features) and finds the image in the recorded set that has the closest distance to this centroid. The found image is considered as the anchor image for the respective column (i.e., the representative image for the respective semantic concepts). The other images in the recorded set are considered as the members for the respective column. They share similar semantic meanings as their anchor image. In total, there are p anchor images. These anchor images contain key semantic concepts of the image database, which are learned from the users' historical RF.

The system constructs the first layer manifold graph using p anchor images. It builds a $p \times p$ affinity matrix, in which each element represents the relationship between each pair of anchor images. The constructed first layer graph is capable of spreading relevance scores of the query to all anchor images. The algorithmic view of constructing the first layer manifold graph is summarized in Figure 4.7.

1. Initialize the first layer manifold graph *FMG* and three intermediate graphs (e.g., *GW*, *GD*, and *GN*) as all 0's. The sizes of these four graphs are all $p \times p$.
2. For each pair of anchor images A_i and A_j, $1 \leq i \leq p$, $1 \leq j \leq p$, $i \neq j$, compute their distance by using the respective low-level visual features and high-level semantic features. The computed distance is stored in the i^{th} row and j^{th} column of *GW* (e.g., $GW_{i,j}$).
3. Update the diagonal element in *GD* as the sum of all elements in its corresponding row in *GW*. That is, $GD_{i,j} = \sum_{k=1}^{p} GW_{i,k}$
4. Update each element in *GN* by symmetrically normalizing *GW*. That is, $GN = GD^{-1/2} \times GW \times GD^{-1/2}$.
6. Update each element in *FMG* by computing $(1 - \alpha \times GN)^{-1}$, where α is a parameter in [0, 1).

Figure 4.7. The algorithmic view of constructing the first layer manifold graph

In step 2, two popular Minkowski distances, e.g., the Manhattan (L_1) distance and the Euclidean (L_2) distance, can be used to compute each element in $GW_{i,j}$. If the L_1 distance is employed, $GW_{i,j}$ is computed by the Laplacian kernel-based 100-dimensinoal low-level visual and high-level semantic features:

$$GW_{ij} = \prod_{l=1}^{100} \exp\left(-\frac{\left|lvf_{il} - lvf_{jl}\right|}{\sigma_L}\right) \times \exp\left(-\frac{1 - NS_{i,j}}{\sigma_H}\right) \tag{4.5}$$

where lvf_i and lvf_j are respectively normalized low-level visual features of two anchor images A_i and A_j, lvf_{il} and lvf_{jl} are respectively the l^{th} element of normalized low-level visual features lvf_i and lvf_j, σ_L is a positive parameter reflecting the standard deviation of the low-level visual similarity, $NS_{i,j}$ is the normalized high-level semantic relevance relation between A_i and A_j, and σ_H is a positive parameter reflecting the standard deviation of the high-level semantic similarity. If the L_2 distance is employed, $GW_{i,j}$ is computed as follows:

$$GW_{ij} = \exp\left(-\frac{\left[(1-w_h)\times d\left(lvf_i, lvf_j\right) + w_h \times \left(1-NS_{i,j}\right)\right]^2}{2\sigma^2} \right) \tag{4.6}$$

where $d(lvf_i,\ lvf_j)$ represents the Euclidean distance between normalized low-level features of A_i and A_j, σ is a positive parameter reflecting the standard deviation of the low-level visual and high-level semantic similarity, and w_h is the contribution factor of high-level semantic features.

4.1.1.6 Construct the Second Layer Manifold Graphs: The second layer manifold graphs are constructed from the clusters around anchor images. For each of *p* anchor images, the system forms a cluster around it and constructs a second layer manifold graph. As a result, there are *p* second layer manifold graphs in total.

Each anchor image and its associated positively labeled images form the initial cluster. Other database images that are not retrieved from the system or are negatively labeled in all query sessions are assigned to their appropriate cluster using the minimum-distance-based strategy. I denote the set of these other database images as *UnassignedSet* and the set of images in *p* clusters as *AssignedSet*. For each image *Imx* in *UnassignedSet*, the system computes its distances to all images in *AssignedSet* and finds the image *Imy* in *AssignedSet* that has the closest distance to *Imx*. The system then assigns *Imx* to the same cluster as *Imy*. In this way, all images in *UnassignedSet* are assigned to exactly one cluster. Each of *p* clusters contains positively labeled images and some images in *UnassignedSet.* Each image has its own anchor image, which represents the characteristic semantic concepts of the cluster. In this way, the images with the same anchor image are considered to be in the same cluster since they assume to share similar semantic concepts as the anchor image. The system then uses a vector *AnchorVec* to store the ID of the anchor image for each database image so that the cluster related information can be easily acquired. The number of clusters equals to the number of anchor images or the number of columns in ***R***, where each column is obtained by the merging strategy explained in Section *4.1.1.4*.

Figure 4.8 shows an example of how to assign an unlabeled image *Imx* in *UnassignedSet* to a proper *AssignedSet*. For ease of discussion, this example shows two clusters, namely, *AssignedSet* 1 and *AssignedSet* 2, which are obtained during the offline training phase. *AssignedSet 1* contains *n* positively labeled images, namely, $Child_{1,1}$, $Child_{1,2}$, $Child_{1,3}$, …, and $Child_{1,n}$. *AssignedSet 2* contains *m* positively labeled images, namely, $Child_{2,1}$, $Child_{2,2}$, $Child_{2,3}$, …, and $Child_{2,m}$. For each *AssignedSet*, two of positively labeled images lie near the edge of its ellipse. In other words, these two positively labeled images are far from the centroid of their corresponding cluster *AssignedSet*, but still share the same semantic concept with other positively labeled images within the same cluster. Three unlabeled images, which are represented by three points, e.g., *Point 1*, *Point 2*, and *Point 3*, need to be assigned to one of these two clusters. If assigning an unlabeled image to a cluster according to the smallest distance between this unlabeled image and the centroid of the cluster, *Point* 2 and *Point 3* should be assigned to *AssignedSet 2* with *Centroid 2*, and *Point 1* should be assigned to *AssignedSet 1* with *Centroid 1*. However, re-examining the assignment of *Point 3* from the view point of transitive semantic relationship among all positively labeled images within a cluster, this assignment of *Point 3* is not proper because any image in the same *AssignedSet* equally shares the same semantic meaning. Specifically, *Point 3*'s nearest neighbor image in *AssignedSet* 1 is *Child1,1* and *Point 3*'s nearest neighbor image in *AssignedSet* 2 is *Child2,2*. *Point 3* is closer to $Child_{1,1}$ of the *AssignedSet 1* than C$hild_{2,2}$ of the

AssignedSet 2 Therefore, the better choice is to assign *Point 3* into *AssignedSet 1* with *Centroid 1*. In summary, the proposed system finds the nearest neighbor of the unlabeled image *Imx* among all images within *AssignedSet*’*s*, and then assigns the unlabeled image *Imx* to the *AssignedSet* that contains the nearest neighbor. In this way, each unlabeled database image *Imx* is assigned to a proper *AssignedSet*.

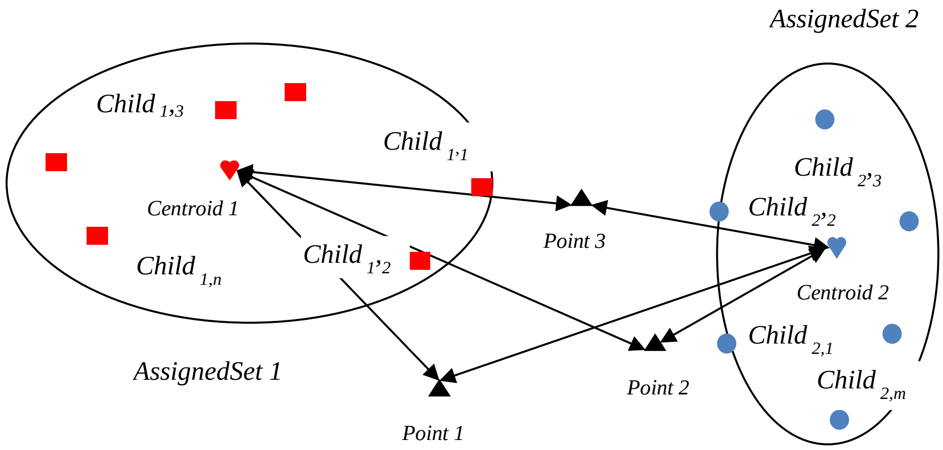


Figure 4.8. Strategy of assigning unlabeled images

Suppose a cluster k contains n_k images including positively labeled images and some images in *UnassignedSet*. The system builds an $n_k \times n_k$ affinity matrix as the second layer manifold graph SMG_k for cluster k. The construction of each second layer manifold graph SMG_k, $1 \leq k \leq p$, follows the same five steps as summarized in Section *4.1.2.3* with two exceptions: 1) The size of SMG_k is $n_k \times n_k$. 2) Each element in graph represents the relationship between each pair of n_k images in cluster k.

Figure 4.9 shows the structure of the proposed scalable graph-based ranking system generated at the end of the training phase. In the first layer, one manifold graph *FMG* is constructed using *p* anchor images, where *p* (i.e., $p << N$) is the number of columns in ***R*** and is also the number of clusters. In the second layer, there are *p* manifold graphs $SMG_{k,}\ 1 \le k \le p$. Each graph is constructed using all member images in its respective cluster. For example, if there are n_1 images in the first cluster, the size of the corresponding graph SMG_1 is $n_1 \times n_1$. The sum of all images in *p* manifold graphs at the second layer is *N*. Here, $n_i << N$ $(1 \le i \le p)$. It is clear that the size of each of $p + 1$ graphs at both the first layer and the second layer is significantly smaller than the size of the traditional manifold graph, which equals to $N \times N$. As a result, the need for a computer to allocate several large consecutive $N \times N$ memory spaces to store the graph is eliminated. It should be noted that a computer runs out of memory or swap space to satisfy such a need and therefore the proposed scalable manifold graphs can be employed

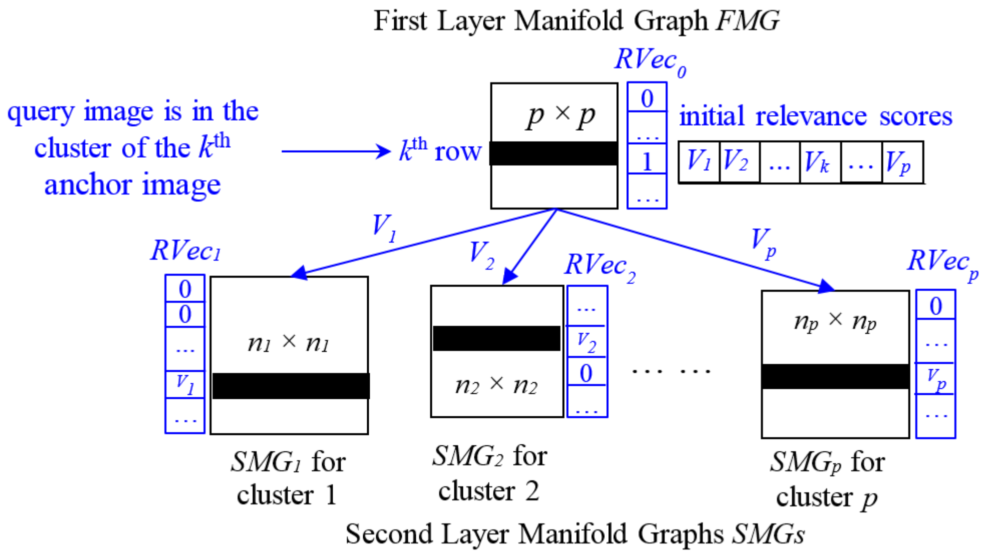


Figure 4.9. The structure of the proposed scalable graph-based ranking system and illustration of the layered design of the relevance vectors together with their initialization.

for a large scale image database as long as each graph does not exceed the memory capability of the running machine.

At the end of the offline training process, two-layer hierarchical manifold graphs are constructed. There are one first-layer manifold graph and *p* second-layer manifold graphs in total.

4.1.2 Online Retrieval Phase

The aim of the online retrieval process is to propagate the ranking scores of positively and negatively labeled images collected during RF iterations to unlabeled images through the proposed scalable hierarchical manifold graphs. These propagated relevance scores are also used as similarity scores between query and database images.

4.1.2.1 Propagate Relevance Scores: Since one *FMG* is constructed to represent the relationship between anchor images and *p SMGs* are constructed to represent the relationship between images in their corresponding clusters, *p*+1 relevance vectors, i.e., $RVec_i$, $0 \le i \le p$, are used to propagate the relevance scores among images in their respective graphs. Here, $RVec_0$ denotes the relevance vector for *FMG* and $RVec_i$ $(1 \le i \le p)$ denotes the relevance vector for SMG_i. Initially, the system sets all relevance vectors as all 0's. That is, $RVec_i = 0$ for $0 \le i \le p$.

For each submitted query image, the system first locates its anchor image from *AnchorVec*. If the index of the query's anchor image in *FMG* is *k*, the system then sets the k^{th} element of $RVec_0$ as 1's. That is, $RVec_{0,k} = 1$. Next, the system propagates $RVec_0$ through *FMG* (i.e., $[FMG]_{p\times p}\times[RVec_0]_{p\times 1}$) to obtain the relevance score of each anchor image to the query. These relevance scores correspond to the values in the k^{th} row (or the

k^{th} column) of *FMG*. The system then propagates each value in the k^{th} row (e.g., V_i ($1 \le i \le p$)) as the initial relevance score for its corresponding second layer manifold graph SMG_i ($1 \le i \le p$). Specifically, the system sets $RVec_{i,m}$ as V_i, where $1 \le i \le p$ and *m* is the index of the anchor image in its respective graph SMG_i. For example, V_1 is put at the row of the anchor image of cluster 1 in $RVec_1$ and V_2 is put at the row of the anchor image of cluster 2 in $RVec_2$, etc. Finally, the system performs one more operation if the query image is a positive image in a cluster *k* ($1 \le k \le p$). To this end, the system finds respective rows of all positive images in cluster *k* and then puts V_k at these same rows in $RVec_k$. This layered design of the relevance vectors together with their initialization is also demonstrated in Figure 4.8 in blue color.

After initializing all *p* relevance vectors, the relevance score of each image is determined by propagating $RVec_i$ through each SMG_i. A relevance score vector T_i for SMG_i is computed by:

$$T_i = [t_{ij}]_{n_i \times 1} = [SMG_i]_{n_i \times n_i} \times [RVec_i]_{n_i \times 1} \quad (4.5)$$

where SMG_i is the i^{th} second layer manifold graph whose size is $n_i \times n_i$ and $RVec_i$ is its initialized relevance vector. The system finally concatenates all relevance score vectors T_i's computed from the second layer manifold graphs into a long relevance score vector *T* with a length of *N*. It then returns *v* images with the highest relevance scores in *T*.

Based on the user's RF information on *v* returned images, the system first finds the anchor images for all labeled images and their respective clusters. It then updates the relevance vectors of these pertinent clusters using the following strategies:

1) For positive images, set the corresponding cells in their relevance vector as 1's.
2) For negative images, set the corresponding cells in their relevance vector as -0.25.

This assignment is empirically determined to be optimal and ensures that the propagation on the negative images is not dominated since negative images do not provide sufficient information as the positive images. The system continues to use updated relevance vectors to compute the relevance scores in T to propagate these relevance scores to unlabeled images and return top v images for the user to label. This process iterates several times until the user is satisfied with retrieval results.

It should be mentioned that the following rules should be employed to update a value in the relevance vector: 1) The cells corresponding to positively labeled images are assigned positive values; 2) The cells corresponding to negatively labeled images are assigned negative values; 3) The magnitude of the values assigned to positively labeled images should be larger than the magnitude of the values assigned to negatively labeled images. The experimental results show that setting positive image cells as 1's and negative image cells as -0.25's achieves the optimal retrieval performance with the minimal computational cost. This asymmetrical assignment also ensures the propagation on the negatives is not dominated.

The error resulted from the first layer manifold graph usually can be corrected based on users' RF information. Since this kind of error comes from the possibly inaccurate cluster assignment of the query image, the user's correct RF makes the system have a higher chance to select potentially correct clusters. In addition, when the query image is assigned to the appropriate cluster, the initial score is only assigned to the query itself in the relevance vector. Therefore, the possible error will not affect other images in the same cluster and the error propagation is prohibited. In other words, clusters with more positively labeled images are likely to be returned in the next iteration based on the asymmetric propagation of positively and negatively labeled images.

To accommodate the possible erroneous RF from the user, the proposed system incorporates a cross-iteration checking and correction strategy to asymmetrically set values for the relevance vector associated with each second-layer manifold graph using RF information in all iterations within a query session. This cross-iteration checking and correction method is described as follows:

- For positive images labeled in the current RF iteration, if they are also labeled as positive images in the previous RF iterations, the proposed system set the corresponding cells in the relevance vector as 1's.
- For positive images labeled in the current RF iteration, if they are labeled as negative images in the previous RF iterations, the proposed system set the corresponding cells in the relevance vector as 0's.
- For negative images labeled in the current RF iteration, if they are labeled as positive images in the previous RF iterations, the proposed system set the corresponding cells in the relevance vector as 0's.
- For negative images labeled in the current RF iteration, if they are also labeled as negative images in the previous RF iterations, the proposed system set the corresponding cells in the relevance vector as -0.25's.

In summary, the proposed cross-iteration checking and correction method utilizes the contradictory RF information in a query session to prevent the possible wrongly labeled images from propagating their labels. As a result, it is helpful to suppress the possible erroneous RF from the user and improve the retrieval accuracy when erroneous feedback is involved.

4.2 Experiments and Results

I conduct a set of carefully designed experiments to evaluate the performance of the proposed scalable graph-based ranking system on five image databases. In Section *4.2.1*, I explain these five image databases. In Section *4.2.2*, I evaluate the effectiveness of the proposed CBIR system by comparing with seven variant systems on the benchmark database. In Section *4.2.3*, I evaluate the performance of the proposed CBIR system together with four manifold-based ranking systems, five state-of-the-art long-term-based CBIR systems, and several representative variant systems on five image databases. In section *4.2.4*, I evaluate the complexity and the storage effectiveness of the compared long-term-based CBIR systems.

4.2.1 Five Image Databases

To simplify the retrieval process and reduce the burden of soliciting user's labeling, I manually organize database images into several semantic classes. As a result, the image relevance can be automatically determined by checking whether returned images belong to the same manually defined class as the query. It should be noted that this ground truth is exclusively used to evaluate the retrieval performance during each iterative RF process and is not assumed to provide additional class-related information for the proposed system. Thus, the proposed technique can be directly applied in any new unorganized database. I collect the following images to evaluate retrieval performance:

- 6000 COREL images: I select 60 distinct categories from the COREL database. Each category contains 100 images covering various real-world scenes.
- 2000 Flickr images: I download a large collection of images from the social photography site http://www.flickr.com. Flickr's API is used to download top 200 images (based on relevance) for each of the chosen 20 categories. I then manually choose 100 images that best represent the category.
- 4000 online images: I download another set of images from http://images.google.com and http://picasa.google.com through their APIs. Similarly to Flickr images, I download top 200 images for each of 40 distinct keywords and manually pick the most appropriate 100 images for each keyword.
- 22000 NUS-WIDE images: I download a set of real-world web images from National University of Singapore [81]. I randomly choose 100 images from each of 81 concepts, which are used for annotation evaluation. I then choose 100 images from each of additional 139 concepts, which contain a sufficient number of images.

Three graduate students are asked to check the appropriateness for each image in its semantic class based on the majority of the agreement. The inappropriate images are replaced by appropriate images approved by at least two graduate students. I then build five image databases as follows:

1) 6000-image database containing COREL images;

2) 2000-image database containing Flickr images;

3) 8000-image database containing 6000 COREL images and 2000 Flickr images;

4) 12000-image database containing 6000 COREL, 2000 Flickr, and 4000 online images;

5) 22000-image database containing NUS-WIDE images.

Each image in the database is represented by a 100-dimensional low-level visual feature vector and a high-level semantic feature vector, whose dimensionality is known after the training phase.

4.2.2 Effectiveness Evaluation

To simulate the practical retrieval process of online users, I randomly generate a sequence of query images to conduct various experiments. At each query session, the proposed CBIR system refines its retrievals by taking advantages of both RF-based transductive short-term learning and semantic feature-based long-term learning techniques and exploiting the synergism between them for several iterations. I use the retrieval precision (RP), which is defined as the ratio of the number of relevant images retrieved to the total number of irrelevant and relevant images retrieved in iterations, as the performance measure. I then compute the average RP (ARP) of the chosen sequence of query images as the final performance measure to evaluate the overall retrieval performance for a large set of query images. The ARP is defined as the total of RP of all query images divided by the total number of queries. In each experiment, I perform four iterations of RF with the top 25 images returned in each iterative step.

Due to the difficulty to recruit a lot of volunteers who are willing to provide the judgment of the relevance of retrieved images for a large amount of query sessions, I design an automatic RF scheme to simulate the users' feedback. Here, I assume that all images in the same category share a common semantic meaning and all query images are from the image database. Under this valid assumption, a retrieved image is automatically defined as relevant or irrelevant to the current query image based on the known categorical information. If the retrieved image belongs to the same category as the query, it is considered as relevant. Otherwise, it is considered as irrelevant. Figure 4.10 shows sample retrieval results using a Coke-Cola can image as the query image.

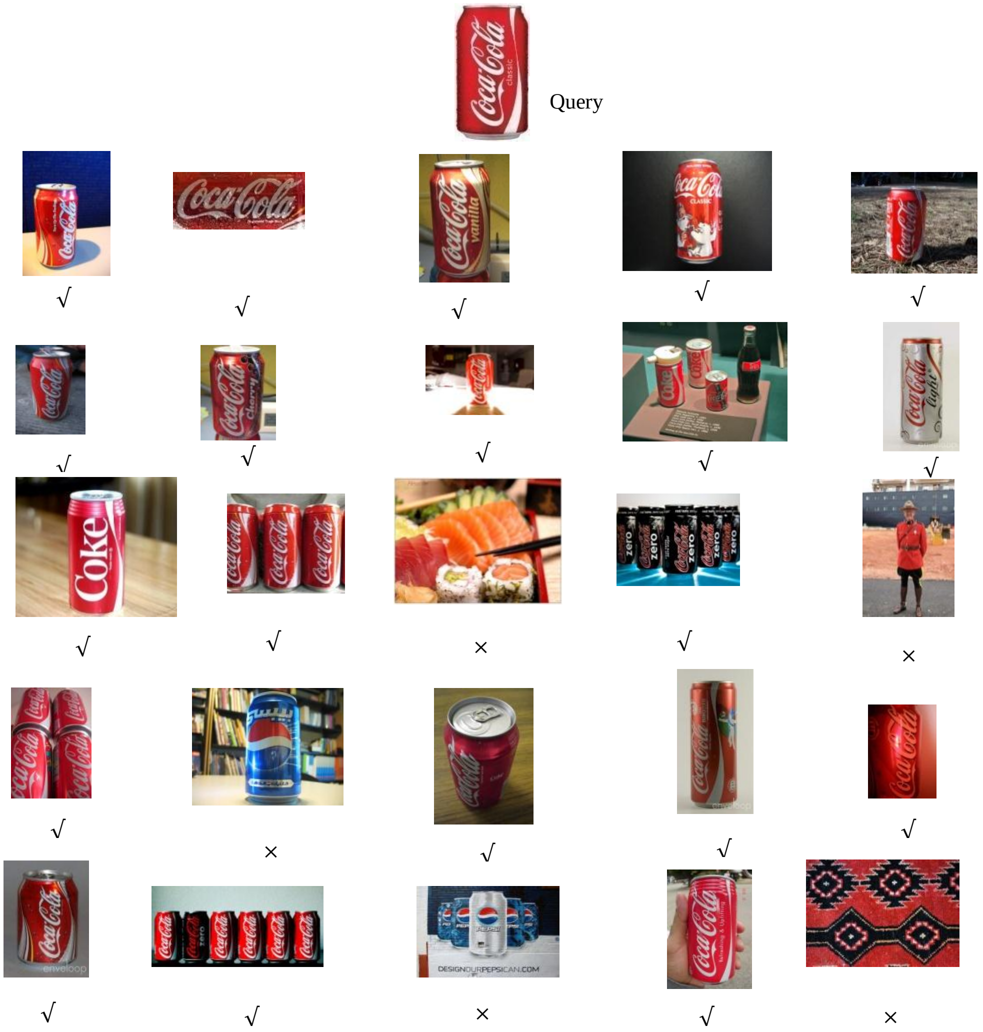


Figure 4.10. Example of automatic RF scheme

Here, the system automatically judges the retrieved images in a retrieval iteration as relevant or irrelevant to the query image, where images marked with $\surd$ are relevant images if they belong to the same category as the query image and images marked with $\times$ are irrelevant images if they do not belong to the category of the query image.

In the proposed system, I incorporate L_2-based low-level visual similarity and high-level semantic similarity into both the first layer manifold graph and the second layer manifold graphs to build the scalable graph-based ranking system. The positive parameter σ_L and σ_H in Equation (4.5) are respectively set to be 0.05, the positive parameter σ in Equation (4.6) are respectively set to be 0.05, the convergence rate α of the affinity matrix is set to be 0.99, and the parameter γ in the RBF kernel is set to be 0.5. These values are empirically chosen to achieve the optimal retrieval performance.

To evaluate the effect of positive parameter w_h , which is used to combine the low-level visual features and high-level semantic features in Equation (4.6), on the proposed retrieval system, I experimentally test several values from 0 to 1 with a step size of 0.1 on the 6000-COREL benchmark database. Table 4.1 compares the retrieval performance in terms of ARP using different w_h's for four iterations.

Table 4.1 clearly shows that the system achieves the best ARP at iterations 2, 3 and 4 by using w_h = 0.5. Specifically, the system achieves the ARP of 88.83%, 94.23% and 97.89% at iterations 2, 3, and 4, respectively. According to the performance difference obtained by using different w_h's on the 6000-COREL benchmark database, I choose w_h as 0.5 for the proposed system.

Table 4.1. Comparison of retrieval performance using different w_h*'s*

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**
$\boldsymbol{w_h}$ **= 0**	80.74 %	87.22 %	91.92 %	95.53 %
$\boldsymbol{w_h}$ **= 0.1**	80.73 %	87.31 %	92.89 %	96.62 %
$\boldsymbol{w_h}$ **= 0.2**	80.68 %	87.91 %	94.06 %	97.19 %
$\boldsymbol{w_h}$ **= 0.3**	80.63%	88.13%	94.12%	97.23%
$\boldsymbol{w_h}$ **= 0.4**	80.62%	88.20%	94.20%	97.56%
$\boldsymbol{w_h}$ **= 0.5**	80.60%	88.83%	94.23%	97.89%
$\boldsymbol{w_h}$ **= 0.6**	80.58%	88.56%	93.75%	97.79%
$\boldsymbol{w_h}$ **= 0.7**	80.57%	88.11%	93.59%	97.18%
$\boldsymbol{w_h}$ **= 0.8**	80.56%	87.91%	92.95%	96.94%
$\boldsymbol{w_h}$ **= 0.9**	80.57%	87.94%	92.71%	96.10%
$\boldsymbol{w_h}$ **= 1**	80.60%	87.69%	90.54%	94.43%

Furthermore, to evaluate the effectiveness of the proposed system, I implement its three L_2-based variants:

- Variant 1: The CBIR system that incorporates L_2-based low-level visual and high-level semantic similarities into the first layer graph and L_2-based low-level visual similarity into the second layer graphs.
- Variant 2: The CBIR system that incorporates L_2-based low-level visual similarity into the first layer graph and L_2-based low-level visual and high-level semantic similarities into the second layer graphs.

- Variant 3: The CBIR system that incorporates L_2-based low-level visual similarity into both first layer and second layer graphs.

Similarly, I implement four L_1-based counterpart variant systems:

- Variant 4: The CBIR system that incorporates L_1-based low-level visual and high-level semantic similarities into both first layer and second layer graphs.
- Variant 5: The CBIR system that incorporates L_1-based low-level visual and high-level semantic similarities into the first layer graph and L_1-based low-level visual similarity into the second layer graphs.
- Variant 6: The CBIR system that incorporates L_1-based low-level visual similarity into the first layer graph and L_1-based low-level visual and high-level semantic similarities into the second layer graphs.
- Variant 7: The CBIR system that incorporates L_1-based low-level visual similarity into both first layer and second layer graphs.

Figure 4.11 compares the retrieval performance of the proposed system and its seven variant systems on the COREL benchmark database. It clearly shows that the proposed system achieves the best ARP of 88.83% at iteration 2, 94.23% at iteration 3, and 97.89% at iteration 4. At the last iteration, the proposed system improves the second-best system (variant 1) by 0.9%, the third-best system (variant 7) by 3.04%, and the worst system (variant 3) by 13.35%. It clearly demonstrates the effectiveness to include both L_2-based visual and semantic similarities in the first layer graph since the proposed system and its variant 1 achieve the best ARP. However, four L_1-based variant systems interestingly achieve comparable retrieval performance regardless of the incorporation of the semantic similarity. These four L_1-based variant systems outperform variant 2 and variant 3. These L_1-based results are consistent with the experimental results of [81] and [67], where L_1 distance outperforms other distances on color images. On the other hand, these L_1-based results indicate that incorporating semantic similarity does not significantly improve the retrieval performance since the semantic similarity is computed by the correlation measure, which cannot be evaluated on a dimensionality basis. In other words, the semantic similarity is evaluated by one value and the visual similarity is evaluated by multiple values (e.g., 100 values) as shown in Equation 4.5, which significantly reduces the effect of the semantic similarity.

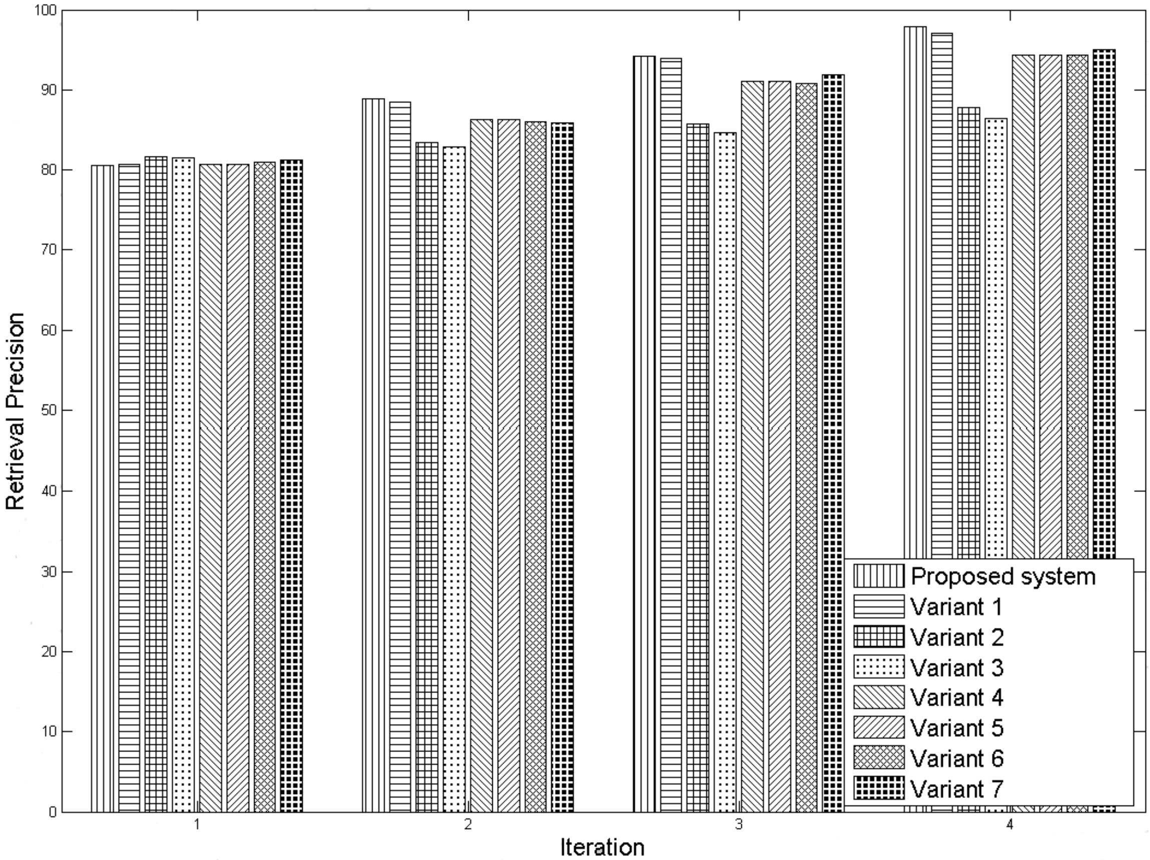


Figure 4.11. Comparison of the proposed system and its seven variant systems, which are built from the compact feedback log

To further prove the effectiveness of the construction of the compact feedback log to extract high-level semantic features, I implement eight respective systems which are built on the full feedback log without applying any merging operations. I call these eight systems as full feedback log based systems. Figure 4.12 compares the retrieval performance of these eight full feedback log based systems on the *6000-COREL* benchmark database.

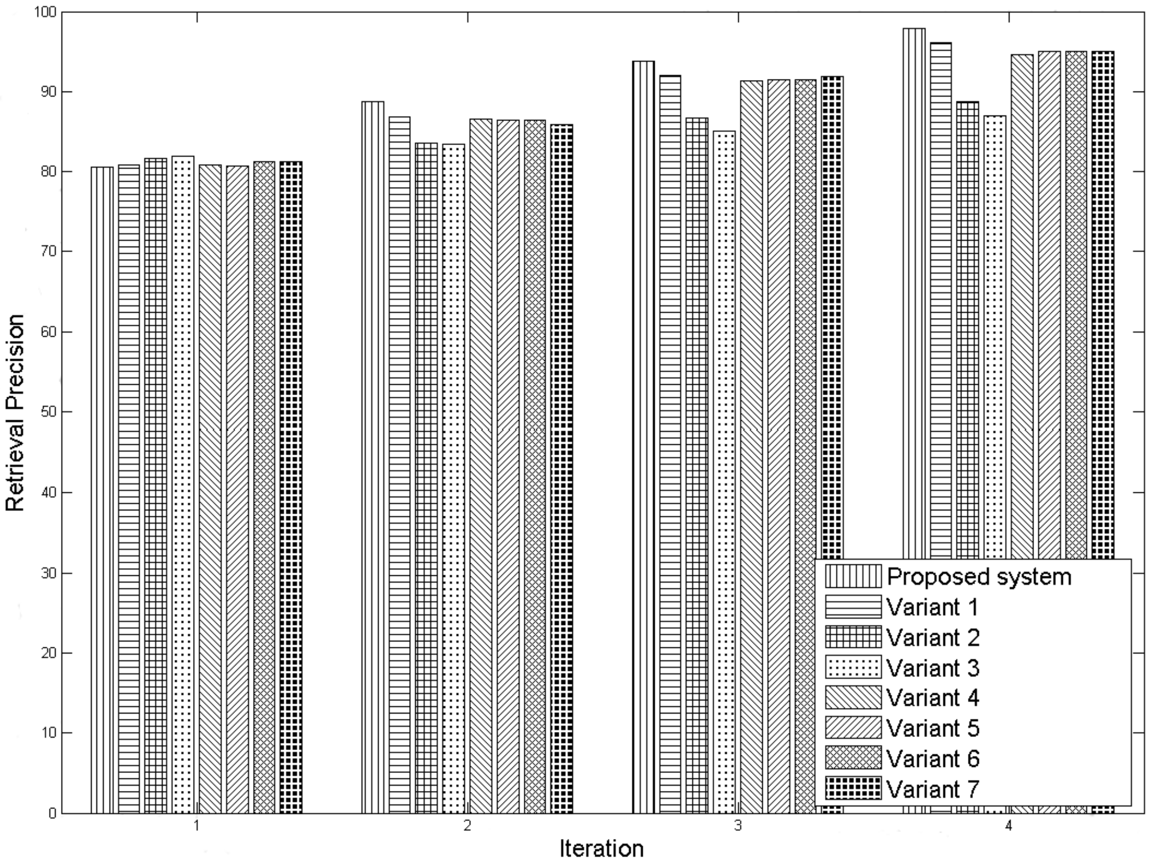


Figure 4.12. Comparison of the proposed system and its seven variant systems, which are built from the full feedback log

It clearly shows that these eight systems demonstrate the same retrieval performance as their counterpart systems built from the compact feedback log. In addition, the proposed system and its seven variant systems achieve similar ARP as their eight counterpart systems built from the full feedback log. Specifically, the proposed system achieves ARP of 80.60%, 88.83%, 94.23% and 97.89% at iterations 1, 2, 3, and 4 on the compact feedback log, respectively. The same system built from the full feedback log achieves ARP of 80.62%, 88.78%, 93.74%, and 97.84% at iterations 1, 2, 3, and 4, respectively. The difference in ARP for these two systems is less than 0.5% at each RF iteration. As a result, I claim that the simple merging strategy works well to reformulate the users' historical feedback in a compact feedback log and extract representative semantic concepts of the image database.

4.2.3 Performance Evaluation

For a comprehensive performance evaluation, I compare the proposed system with nine state-of-the-art long-term-based CBIR systems on five image databases. These compared systems can be categorized into two groups:

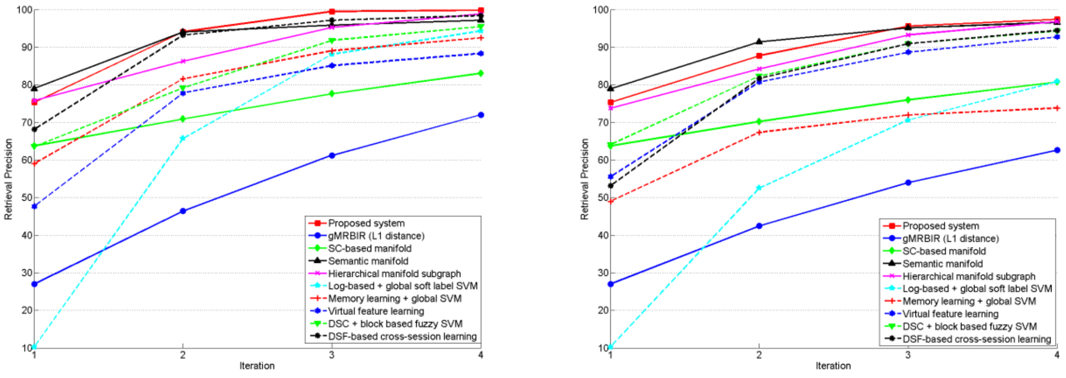
- Manifold-based long-term learning systems: L_1-distance based gMRBIR [16], semantic clusters based manifold ranking system (i.e., SC-based manifold) [67], single weighted semantic manifold ranking system (i.e., Semantic manifold) [68], and hierarchical manifold subgraphs ranking system (i.e., Hierarchical manifold subgraph) [69].
- Other long-term learning systems: log-based system (i.e., Log-based + global soft label SVM) [66], memory learning system (i.e., Memory learning + global SVM) [64], virtual feature-based system (i.e., Virtual feature learning) [61], dynamic semantic clustering system (i.e., DSC + block based fuzzy SVM) [35], dynamic

semantic feature-based long-term cross-session learning system (i.e., DSF-based cross-session learning) [63].

The following five figures show the retrieval performance of the compared systems at each of four iterations in the context of having correct feedback and having a level of 5% erroneous feedback on 2000-image, 6000-image, 8000-image, 12000-image, and 22000-image databases, respectively. In all these figures, manifold-based systems are shown in solid lines and other long-term systems are shown in dashed lines. For the three smaller databases (i.e., 2000-image, 6000-image and 8000-image databases), the proposed system and all the aforementioned systems are included in the comparison. For the 12000-image database, gMRBIR, SC-based manifold, and semantic manifold systems cannot run on a computer due to its requirement of several matrices of 12000×12000. For the 22000-image database, the same three manifold-based systems and memory learning system cannot run on a computer due to its requirement of several matrices of 22000×22000. As a result, these systems are not included in the comparison for either 12000-image or 22000-image or both databases. Instead, I include the variant 3, variant 4, and variant 7 systems in the comparison for these two larger databases. For the erroneous RF, I let the simulated "user" misclassify some relevant images as irrelevant and irrelevant images as relevant during the online retrieval phase. I choose the level of 5% erroneous RF since it is similar to the real noise level of the non-malicious human users.

Figure 4.13 compares ten state-of-the-art long-term-based CBIR systems on 2000-image database in the context of correct feedback and 5% erroneous feedback. Figure 4.13 (a) clearly shows that the proposed system achieves the best ARP at RF iterations 3 and 4. Specifically, the proposed system achieves ARP of 75.29%, 94.07%, 99.47%, and

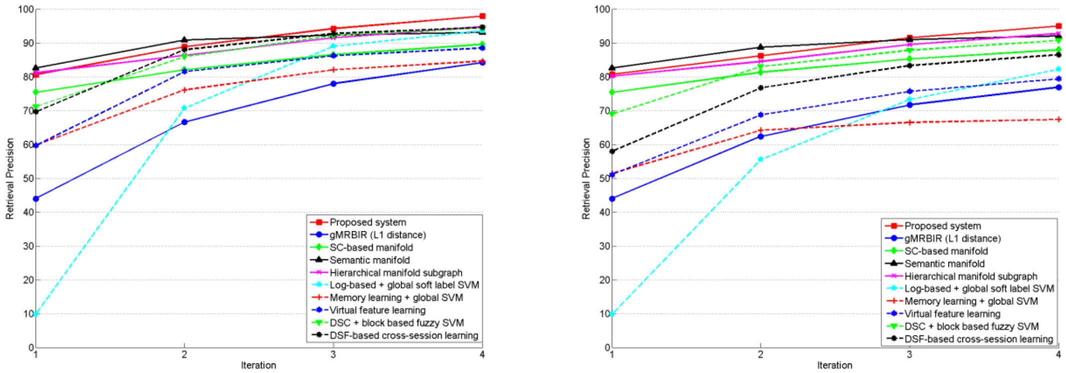
99.84% at iterations 1, 2, 3, and 4, respectively. Furthermore, the proposed system improves the second-best system (i.e., hierarchical manifold subgraph) by 1.04% at iteration 4. Additionally, the proposed system also starts to achieve the best ARP since iteration 2, which is 94.07%. This impressive retrieval performance at early iterations is one of the advantages of the proposed system. As a result, it reduces the burden for the user to label returned images by quickly yielding satisfactory retrieval results. Figure 4.13(b) shows the performance comparison when 5% erroneous feedback is introduced. It clearly shows that the proposed system has the best ARP of 95.16% and 97.41% at iterations 3 and 4, respectively. Moreover, the performance of the system involved with erroneous feedback decreases only by 2.43% at the last iteration when comparing with the performance of the proposed system with correct feedback. It demonstrates the robustness of the proposed system to the erroneous on the 2000-image database.



(a) (b)

Figure 4.13. Comparison of ten state-of-the-art long-term-based CBIR systems on 2000-image database with (a) correct feedback and (b) 5% erroneous feedback

Figure 4.14 compares ten state-of-the-art long-term-based CBIR systems on 6000-image database in the context of correct feedback and 5% erroneous feedback. It clearly demonstrates the proposed system outperforms ten state-of-the-art long-term-based CBIR systems. Specifically, Figure 4.14 (a) shows the proposed system achieves the best ARP of 94.23% and 97.89% at iterations 3 and 4, respectively. At the last RF iteration, it improves the second-best system (i.e. hierarchical manifold subgraph) by 3.29%. But the semantic manifold approach has a little bit better retrieval performance than the proposed approach at iterations 1 and 2. Specifically, the semantic manifold system improves the proposed system by 2.47% and 2.27% at iterations 1 and 2. Figure 4.14 (b) shows the performance comparison when 5% erroneous feedback is introduced. It clearly shows that the proposed system has the best ARP of 91.48% and 94.95% at iterations 3 and 4, respectively. Moreover, the performance of the system involved with the erroneous feedback decreases only by 2.94% at the last RF iteration when comparing with the

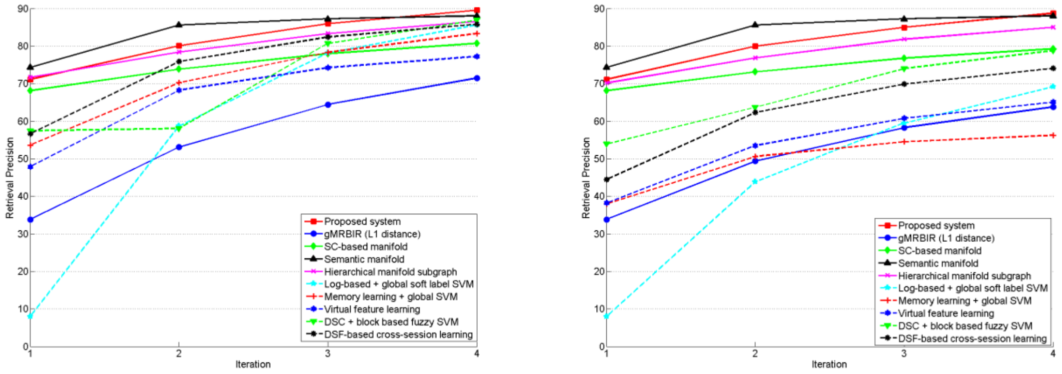


(a) (b)

Figure 4.14 Comparison of ten state-of-the-art long-term-based CBIR systems on 6000-image database with (a) correct feedback and (b) with 5% erroneous feedback

performance of the proposed system with correct feedback. It demonstrates the robustness of the proposed system to the erroneous feedback on the 6000-image database.

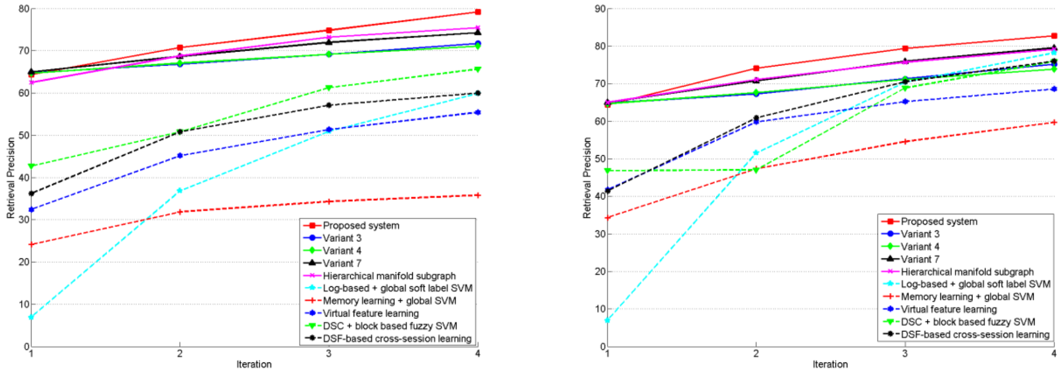
Figure 4.15 compares ten state-of-the-art long-term-based CBIR systems on 8000-image database in the context of correct feedback and 5% erroneous feedback. Figure 4.15 (a) clearly demonstrates that the proposed system has the best ARP which is 89.57% at the last RF iteration. It improves the second-best system (i.e., semantic manifold) by 1.73% at the last iteration. Semantic manifold approach achieves the best ARP at the first three iterations. It improves the proposed system by 4.51%, 6.96%, and 1.54% at iterations 1, 2 and 3, respectively. However, the semantic manifold approach cannot be employed on a larger database due to the large memory requirement to store several big matrices. The proposed system is scalable and only requires several small matrices to store the relevance information between corresponding image pairs. Therefore, the proposed system can be employed on a larger database. Figure 4.15 (b) shows the performance comparison when 5% erroneous feedback is introduced. It clearly shows that the proposed system has the best ARP of 88.75% at the last RF iteration. Moreover, the performance of the system with the erroneous feedback introduced decreases only by 0.92% at the last RF iteration when comparing with the performance of the proposed system with no erroneous feedback. It demonstrates the robustness of the proposed system on the 8000-image database when users make erroneous RF during the online retrieval phase.



(a) (b)

Figure 4.15. Comparison of ten state-of-the-art long-term-based CBIR systems on 8000-image database with (a) correct feedback and (b) with 5% erroneous feedback

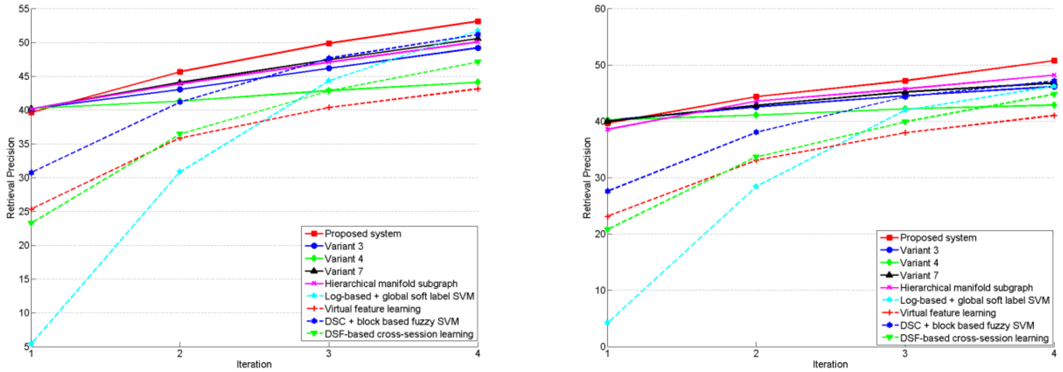
Figure 4.16 compares the proposed system, its three variants, and other six state-of-the-art long-term-based CBIR systems in the context of the correct feedback and 5% erroneous feedback. Figure 4.16 (a) clearly shows that the proposed system achieves the ARP of 64.35%, 74.08%, 79.36%, and 82.69% at iterations 1, 2, 3, and 4, respectively. Moreover, it improves the second-best system (i.e., variant 7) by 4.72%, 4.49%, and 4.00% at RF iterations 2, 3, and 4, respectively. Figure 4.16 (b) shows the performance comparison when 5% erroneous feedback is introduced. It clearly shows that the proposed system has the best ARP of 70.74%, 74.87% and 79.20% at iterations 2, 3 and 4, respectively. Moreover, it decreases the ARP of the proposed system with correct feedback only by 4.22% at the last RF iteration. It demonstrates the robustness of the proposed system on the 12000-image database when users make erroneous RF during the online retrieval phase.



(a) (b)

Figure 4.16. Comparison of state-of-the-art long-term-based CBIR systems on 12000-image database with (a) correct feedback and (b) with 5% erroneous feedback

Figure 4.17 shows the performance comparison of proposed system, its three variants, and other six selected state-of-the-art long-term-based CBIR systems on the 22000-image database in context of correct feedback and 5% erroneous feedback. Figure 4.17 (a) clearly shows that the proposed system achieves the best ARP at RF iterations 2, 3 and 4. Specifically, it achieves ARP of 45.66%, 49.84%, and 53.11 at iterations 2, 3 and 4; and the proposed system improves the second best system (i.e., Log-based + global soft label SVM) by 2.7% at the last RF iteration. Figure 4.17 (b) shows the performance comparison when 5% erroneous feedback is introduced. It clearly shows that the proposed system has the best ARP of 44.36%, 47.19% and 50.73% at iterations 2, 3, and 4, respectively. Moreover, the performance of the system with the erroneous feedback introduced decreases only by 4.48% when comparing with the performance of the proposed system with no erroneous feedback introduced at the last RF iteration. It demonstrates the robustness of the proposed system on the 22000-image database when users make erroneous RF during the online retrieval phase.



(a) (b)

Figure 4.17. Comparison of state-of-the-art long-term-based CBIR systems on 22000-image database with (a) correct feedback and (b) with 5% erroneous feedback

From Figures 4.13 through 4.17, it is clear that the proposed system increases the ARP after each retrieval iteration on all five image databases. Meanwhile, the ARP of the proposed system decreases when the size of the image database increases. Specifically, in the context of the correct feedback, the ARP of the proposed system at the last iteration is 99.84% for the 2000-image database, 97.89% for the 6000-image database, 89.57% for the 8000-image database, 82.69% for the 12000-image database, and 53.11% for the 22000-image database. However, the second best system achieves the ARP of 98.82% for the 2000-image database, 94.77% for the 6000-image database, 88.05% for the 8000-image database, 79.51% for the 12000-image database, and 51.72% for the 22000-image database. It should be mentioned that the second best system is different for different databases. Specifically, the second best system is “Hierarchical manifold subgraph” for the 2000-image database, “Hierarchical manifold subgraph” for the 6000-image database, “Semantic manifold” for the 8000-image database, “Variant 7” for the 12000-image database, and “Log-based + global soft label SVM” for the 22000-image database. This clearly shows that the effectiveness of the proposed system on databases with different number and types of images. In the context of the erroneous feedback, the ARP of the proposed system at the last iteration is 97.41% for the 2000-image database, 94.95% for the 6000-image database, 88.75% for the 8000-image database, 79.20% for the 12000-image database, and 50.73% for the 22000-image database. Meanwhile, the second best system achieves the ARP of 96.82% for the 2000-image database, 92.77% for the 6000-image database, 88.01% for the 8000-image database, 75.43% for the 12000-image database, and 48.16% for the 22000-image database. Specifically, the second best system is “Hierarchical manifold subgraph” for the 2000-image database, “Hierarchical manifold subgraph” for the 6000-image database, “Semantic manifold” for the 8000-image database, “Hierarchical manifold subgraph” for the 12000-image database, and “Hierarchical manifold subgraph” for the 22000-image database. It clearly demonstrates that the robustness of the proposed system on databases with different number and types of images.

Figure 4.18 plots the precision and recall curves of the proposed system in the context of correct feedback on the 22000-image database when a different number of images (e.g., 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, and 155) are returned at each of four RF iterations. Here, precision represents ARP. Recall represents the mean recall that is computed as the total of recall values of all query images divided by the total number of queries, where recall is defined as the ratio of the number of relevant images retrieved to the total number of relevant images (e.g., 100) in the database (refer to Equation (2.3)). This figure clearly shows recall increases when the number of retrieved images increases for each iteration. However, precision for each iteration drops along with the increasing number of retrieved images. It also shows the proposed system is effective in returning above 40% of relevant images at the last iteration (i.e., ARP is above 40%) when the number of images returned is less than 95.

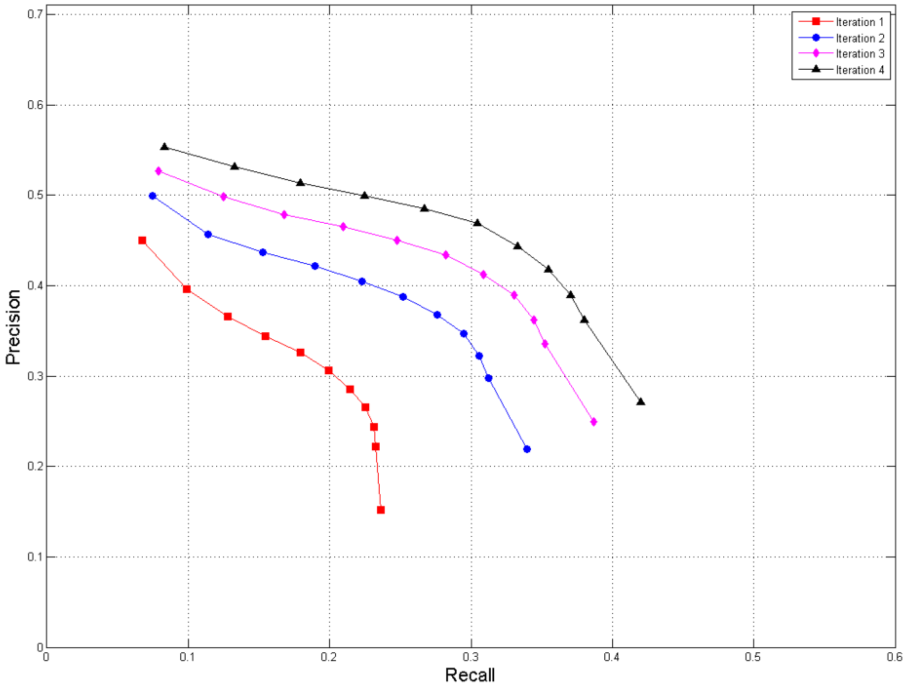


Figure 4.18. Precision and recall curve of the proposed system on the 22000-image database in the context of correct feedback

The precision-recall curve of the proposed system in the context of the erroneous feedback follows the same trend as the precision-recall curve of the proposed system in the context of the correct feedback. As shown in Figures 4.13 through 4.17, the retrieval performance under the erroneous feedback decreases a little bit compared to the retrieval performance under the correct feedback. As a result, the precision-recall curve of the proposed system under the erroneous feedback is moving downward. In other words, all the curves of the proposed system involved with the erroneous feedback are similar to the curves shown in Figure 4.18, except that the area under the curve area is smaller than its counterpart in Figure 4.18.

Finally, I summarize the ARPs for several representative categories of the 22000-image database when correct RF is involved. Specifically, two NUS-WIDE categories (e.g., flags with different backgrounds and water/water drops with different backgrounds) achieve the worst ARP of 7.33% at the last iteration. Five COREL categories (e.g., dinosaurs with pure backgrounds, elephants, masks with pure backgrounds, mineral samples with pure backgrounds, and molecular diagram) and the skyscraper category from 4000 online images achieve the best ARP of 100% at the last three iterations. The pills category of the COREL database achieves the average ARP of 53.33% at the last iteration. The portraits category of the COREL database achieves the median ARP of 54.44% at the last iteration. This clearly shows the effectiveness of the proposed retrieval process on a majority of semantic categories (classes).

4.2.4 Comparative Complexity and Storage Evaluation

I compare the above ten CBIR systems from the perspectives of the storage and computational complexity. The proposed CBIR system requires $O(N \times p)$ space to store historical RF information in a compact feedback log for extracting semantic knowledge, where N denotes the total number of images in the database and p denotes the number of columns in the feedback log. Based on the experiments, p is 27, 89, 192, 361, and 1188 for the 2000-image, 6000-image, 8000-image, 12000-image, and 22000-image databases, respectively. Dynamic semantic clustering system requires $O(N \times NumC)$ space, where $NumC$ is the number of learned clusters and is approximately 21, 68, 98, 139, and 326 for the 2000-image, 6000-image, 8000-image, 12000-image, and 22000-image databases, respectively. All the other long-term-based CBIR systems require $O(\mathrm{c}\times N\times N)$ space. The c's in hierarchical manifold subgraph, virtual feature learning, and DSF-based cross-session learning systems are a fractional number (e.g., 0.1). The c's in log-based and memory learning systems are 1 and 3, respectively. The c's in gMRBIR, SC-based manifold, and semantic manifold systems are all equal to 4. It clearly shows that the proposed CBIR system requires a little more storage space than dynamic semantic clustering system and a small fraction of storage space as required by the other eight long-term-based CBIR systems. This efficient storage is necessary for real-world situations with databases of millions of images.

The complexity of the proposed retrieval algorithm is $O(N\times p)$. The complexity of dynamic semantic clustering system is $O(N\times NumC+NumC\times NumC)$. The complexity of the other eight long-term-based CBIR systems is $O(\mathrm{c}\times N\times N)$. It clearly shows that the proposed system is computationally efficient.

4.3 Conclusions and Future Work

I propose a novel scalable manifold graph-based CBIR system for image retrieval. It takes the advantages of both RF based transductive short-term learning and semantic feature-based long-term learning techniques to improve retrieval performance. The major contributions are:

- Quickly constructing a compact dynamic feedback log to store retrieval patterns of each past query session.

- Efficiently merging similar semantic concepts to maintain a reasonable number of representative semantics for all images in a database.
- Creatively constructing two-layer hierarchical graphs to represent the inherent structure of the large-scale image database during the system offline training stage.
- Effectively combining low-level visual and high-level semantic similarity measure to build a scalable manifold graph, which explores the intrinsic structure of images in both low-level visual and high-level semantic feature spaces.
- Effectively designing a layered relevance vector to propagate the relevance scores from anchor images to the second layer graphs and further propagate relevance scores of labeled images to unlabeled image via the hierarchical graph-based structure.

I plan to test the proposed technique for its effectiveness and scalability on a larger database by comparing with emerged state-of-the-art systems. I will first investigate the usefulness of incorporating other sophisticated and distinguishable features such as histograms of oriented gradients (HOG) to extract low-level visual features. Next, I plan to obtain a sufficient number of human subject tests to simulate the user's query log information and investigate how the proposed system would do with real human feedback. Finally, I may explore the potential of applying the proposed technique in the image annotation task.

CHAPTER 5

A SINGLE WEIGHTED MANIFOLD GRAPH-BASED CONTENT-BASED IMAGE RETRIEVAL APPROACH

The conventional manifold ranking techniques discussed in Chapter 3 explore the relationship of all database images in the feature space and propagates ranking scores of labeled images to unlabeled images via a weighted graph. However, they still have several drawbacks including:

1) The weighted graph is not powerful enough to represent what database images look like in the feature space since only low-level visual features are involved.
2) The semantic gap between visual features and semantic concepts of images still exists when visual features of images are used to construct the manifold graph and perform the retrieval task.
3) Accumulated feedback from historical query sessions are not used to improve the manifold graph.

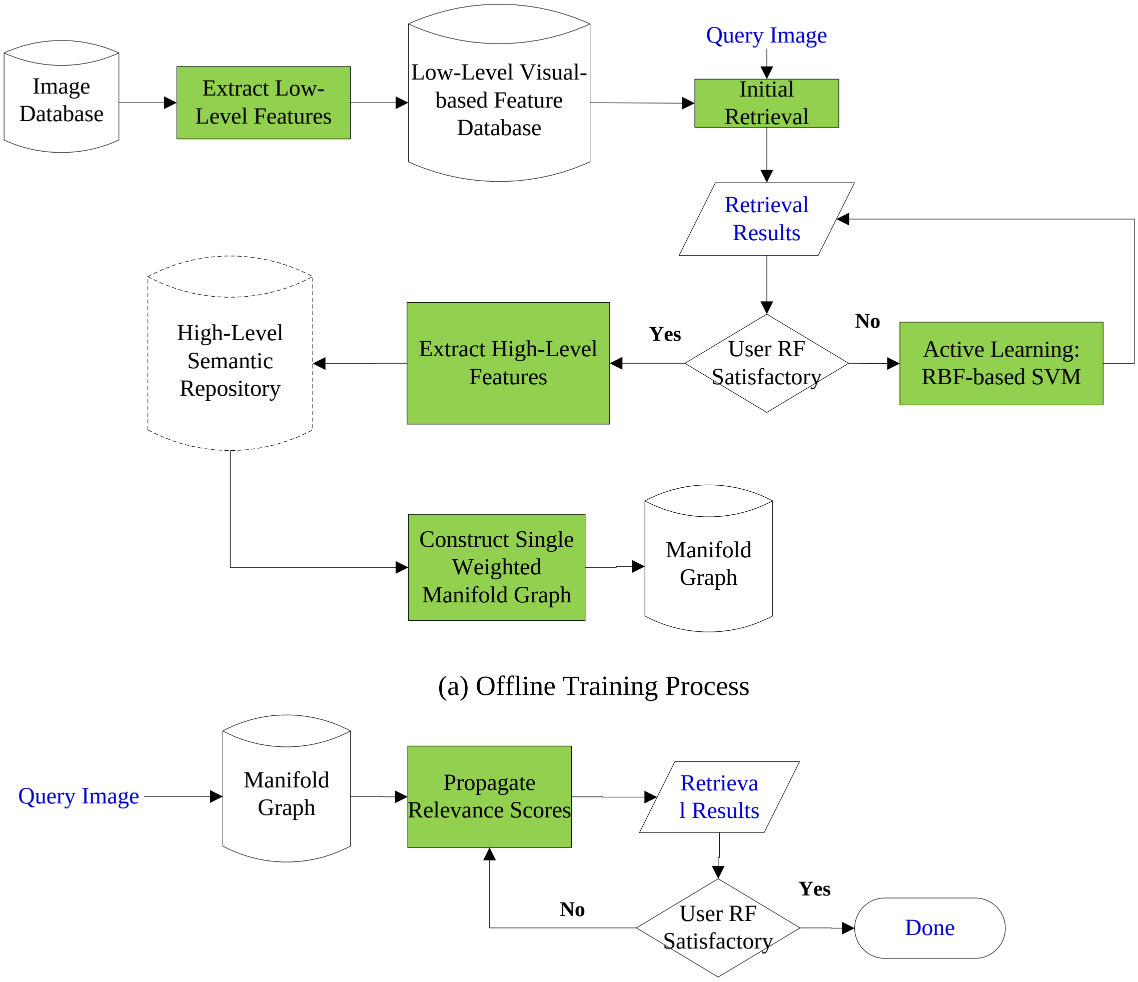
In Chapter 4, I introduced the proposed scalable manifold graph-based CBIR system that has the capability to perform retrieval tasks in large-scale image databases. The construction of the two-layer hierarchical manifold graphs in the scalable graph-based CBIR system requires more computation in the offline training phase. Similarly, it requires more computational time to propagate the ranking scores from labeled images to unlabeled images by going through two-layered manifold graphs. For small image databases, I propose a single weighted manifold graph-based CBIR system that only requires constructing a single graph to save the computational cost.

The aforementioned shortcomings of the conventional manifold ranking systems and the proposed scalable manifold graph-based CBIR systems motivate me to develop a novel technique to enhance the weighted graph by incorporating both visual and semantic information together with the importance scores to obtain more satisfactory results within fewer query iterations. The major contributions are:

1) Applying the SVM-based RF technique to construct a dynamic feedback log to store the user's RF. Based on this feedback log, the proposed system can explore semantic concepts of the image database, and then applies a minimum-distance-based strategy to assign each non-labeled image into a proper semantic concept. These explored semantic concepts properly divide the database images into meaningful semantic categories to facilitate future learning.
2) Computing the importance score of each image. The higher importance score an image has, the more semantic information we know about an image, and the more propagation power an image possesses. As a result, the importance score can be used to suppress the decayed effects of erroneous feedback.
3) Extracting high-level semantic features of each database image based on users' historical retrieval experiences. These features are used to estimate the high-level semantic relations among images.
4) Incorporating the importance scores, high-level semantic scores, and low-level visual scores into the affinity matrix to construct the single weighted manifold graph. In this way, the proposed system significantly suppresses the noise propagation among images and is therefore more robust than the traditional manifold graph.

5) Constructing an asymmetrical relevance vector based on the user’s RF and propagating the ranking scores of labeled images in the relevance vector to unlabeled images via the weighted manifold graph. This asymmetrical assignment ensures the propagation on the positive images is dominated and helps unlabeled images to obtain more proper ranking scores.

5.1 The Framework of the Single Weighted Manifold Graph Approach


Query Image
Image Database
Extract Low-Level Features
Low-Level Visual-based Feature Database
Initial Retrieval
Retrieval Results
High-Level Semantic Repository
Extract High-Level Features
Yes
User RF Satisfactory
No
Active Learning: RBF-based SVM
Construct Single Weighted Manifold Graph
Manifold Graph
Query Image
Manifold Graph
Propagate Relevance Scores
Retrieva l Results
No
User RF Satisfactory
Yes
Done

(a) Offline Training Process

(b) Online Retrieval Process

Figure 5.1. Block diagram of the proposed system: (a) offline training process and (b) online retrieval process

The block diagram of the proposed system is shown in Figure 5.1. The goal of the offline training process is to construct a single weighted manifold graph, which stores the learned similarity between each image pair. The goal of the online retrieval process is to propagate ranking scores of labeled images to unlabeled images via the learned weighted manifold graph. The following subsections explain each component in detail.

5.1.1 Offline Training Phase

```
Input: All images in the database.
Output: Single weighted manifold graph.
    1. Apply "Extract Low-level Features" on each image in the image database to
       represent images from the visual perspectives.
    2. Randomly choose 10% of database images as training images to perform the training
       task.
    3. For each training query image,
        3.1 Perform "Initial retrieval" to return top $v$ relevant images
        3.2 Allow the user to select relevant (i.e., positive) images from the retrieved images
        3.3 Treat non-selected images as irrelevant (i.e., negative) images.
        3.4 Apply "Active Learning (e.g., RBF-based SVM)" on the accumulated positive
            and negative images to find a better classification boundary to discriminate
            positive images from negative images in the database.
        3.5 Return top $v$ relevant images based on the distance to the classification boundary.
        3.6 Repeat step 3.2 through step 3.6 for a few feedback iterations until the query
            session finishes (i.e., the maximum number of iterations is achieved or the user is
            satisfied with the retrieval results).
        3.7 Store the relevancy information for each retrieved image in the current query
            session in a dynamic feedback log.
    10. Apply "Extract High-Level Features" on the dynamic feedback log to obtain high-
        level semantic features for each database image.
    11. Apply "Construct the Single Weighted Manifold graph" based on the feedback log
        obtained in the RF iterations to construct the single weighted manifold graph.
```

Figure 5.2. The algorithmic view of the offline training process

The goal of the offline training process is to construct a single weighted manifold graph-based structure of the image database which stores the learned correlation between each image pair. The algorithmic view of the offline training process is summarized in Figure 5.2.

Since the proposed single weighted manifold graph-based CBIR system is evolved from the scalable manifold graph-based CBIR system described in Chapter 4, most key components in the offline training process are the same as the ones in the scalable graph-based system. Readers may refer to the details of these key components in the previous chapter as listed below:

- Extract Low-level Features (Section 4.1.1.1)
- Initial Retrieval (Section 4.1.1.2)
- Active Learning: RBF-based SVM (Section 4.1.1.3)
- Extract High-level Features (Section 4.1.1.4)

For each image pair Im_i and Im_j:

1. Compute the low-level visual feature-based distance $d_{i,j}$ (refer to Equation (4.2)).
2. Compute the high-level semantic feature-based distance $S_{i,j}$ (refer to Equation (4.2)).
3. Construct an affinity matrix $W = [W_{i,j}]_{N \times N}$ where each element W_{ij} represents the correlation of each image pair Im_i and Im_j in the database and N is the total number of images in the database. Specifically, $W_{i,j}$ is computed by incorporating importance scores, low-level visual based features and high-level semantic-based features.
4. Compute the symmetrically normalized affinity matrix S by *D-1/2WD-1/2*, where D is a diagonal matrix with the i-th diagonal element $D(i, i)$ being the sum of the i-th row of W. That is, $D_{i,i} = \sum_{k=1}^{N} W_{i,k}$, where k is the index of elements of i-th row in W.
5. Compute the final manifold graph MG as *(1-αS)-1*, where α is set to be 0.99 in the system.

Figure 5.3. The algorithm view of constructing the single weighted semantic manifold graph

In the following, I explain the novel component, namely, Construct Single Weighted Manifold Graph, proposed in the single weighted manifold graph-based CBIR system. Figure 5.3 summarizes the algorithmic view of constructing this single weighted manifold graph.

In step 3, I incorporate three components, namely, low-level visual feature-based similarity, high-level semantic feature-based similarity, and importance scores, to compute the distance between each image pair. The importance score measures the level of correctness of assigning each image to its corresponding assigned set. Since positive images in *AssignedSet* are labeled by the user during the RF iteration, they share similar semantic concepts and are assigned an importance score of 1's. On the other hand, for each image *Imx* in *UnassignedSet*, the system estimates the level of correctness of the assignment to suppress the possible wrong assignment of images in *UnassignedSet*. Specifically, the importance score for the image *Imx* in *UnassignedSet* is calculated by the standard Cauchy distribution function [83]. I choose the standard Cauchy distribution function over some commonly used cone and exponential functions due to its good expressiveness and its high computational efficiency. The original Cauchy distribution function is defined by the following formula.

$$f(x) = \frac{1}{s\pi[1+(\frac{x-t}{s})^2]} \tag{5.1}$$

where *t* is the location parameter and *s* is the scale parameter. When *t = 0*, and *s = 1*, the above formula becomes the standard Cauchy distribution function whose values range between 0 and 1. Thus, it is suitable to evaluate the correctness level of the assignment of unlabeled images to their *AssignedSet*. This standard Cauchy distribution, shown in Figure 5.4, is defined as follows:

$$f(x) = \frac{1}{\pi(1+x^2)} \tag{5.2}$$

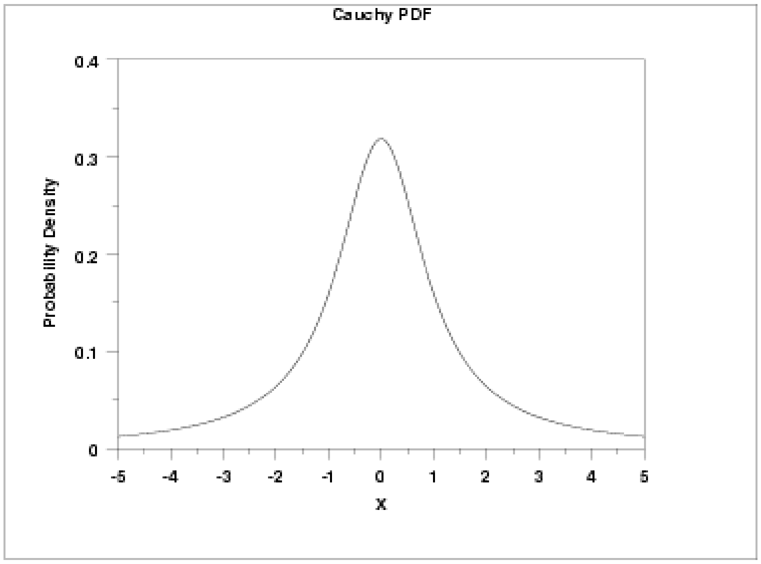


Figure 5.4 Standard Cauchy distribution function

Based on the standard Cauchy distribution function, I calculate the importance score of an unlabeled image in the single weighted manifold graph-based CBIR system by:

$$IS_i = \frac{1}{1+\left(\frac{\left\|x_i - C(x_i)\right\|}{\frac{2}{N_C(N_C-1)}\times\sum_{i=1}^{N_C}\sum_{j=i+1}^{N_C}\left\|C(i)-C(j)\right\|}\right)^2} \tag{5.3}$$

where ||x_i – $C(x_i)$|| denotes the distance between an unlabeled image x_i and the corresponding centroid of its assigned set (i.e., cluster), $\frac{2}{N_c(N_c-1)} \times \sum_{i=1}^{N_c}\sum_{j=i+1}^{N_c}\|C(i) - C(j)\|$ denotes the average of the distance between each pair of centroids for all assigned sets. I omit π which appears in the original standard Cauchy distribution function, because it equally contributes to the computation of all importance scores of unlabeled images. The computed importance score is in the range of [0, 1]. The value of 0 indicates the assignment is incorrect and therefore the distance between any image and this wrongly assigned image is 0. The value of 1 indicates that the assignment is correct and therefore the distance between any image and this correctly assigned image is kept the same. The higher importance score, the more propagation power of images gained in the online retrieval phase.

The proposed system can flexibly apply two popular Minkowski distances, e.g., the Euclidean (L_2) distance and the Manhattan (L_1) distance, to calculate each element $W_{i,j}$ by combining low-level visual features and high-level semantic features.

If the L_2 distance is employed, $W_{i,j}$ is computed as follows:

$$W_{i,j} = IS_i \times IS_j \times \exp\left(- \frac{\left[(1 - w_h) \times d(lvf_i, lvf_j) + w_h \times (1 - NS_{i,j}) \right]^2}{2\sigma^2} \right) \tag{5.4}$$

where $d(lvf_i, lvf_j)$ represents the Euclidean distance between normalized low-level features of i-th image and j-th image, σ is a positive parameter reflecting the standard deviation of the low-level visual and high-level semantic similarity, w_h is the contribution factor of high-level semantic features within the range from 0 to 1, $NS_{i,j}$ is the normalized high-level semantic relevance relation between i-th image and j-th image, IS_i is the importance score for i-image, and IS_j is the importance score for j-image.

If the L_1 distance is employed, $W_{i,j}$ is computed as follows:

$$W_{i,j} = IS_i \times IS_j \times \prod_{l=1}^{100} \exp\left(- \frac{\left| lvf_{il} - lvf_{jl} \right|}{\sigma_L} \right) \times \exp\left(- \frac{1 - NS_{i,j}}{\sigma_H} \right) \tag{5.5}$$

where lvf_i and lvf_j are respectively normalized low-level visual features of i-th image and j-th image, lvf_{il} and lvf_{jl} are respectively the l-th element of normalized low-level visual features lvf_i and lvf_j, σ_L is a positive parameter reflecting the standard deviation of the low-level visual similarity, $NS_{i,j}$ is the normalized high-level semantic relevance relation between i-th image and j-th image, σ_H is a positive parameter reflecting the standard deviation of the high-level semantic similarity, IS_i is the importance score for *i*-image, and IS_j is the importance score for *j*-image.

At the end of the offline training process, a single weighted semantic manifold graph is composed by incorporating low-level visual- feature-based similarity, high-level sematic feature-based similarity, and importance scores.

5.1.2 Online Retrieval Process

The aim of the online retrieval process is to propagate the ranking scores of positively and negatively labeled images collected during RF iteration to unlabeled images through the proposed weighted semantic manifold graph. These ranking scores also serve as the similarity scores between the query image and database images.

Initially, the system encodes a relevance vector Y=$[y_i]_{N\times1}$by setting the row corresponding to the query image as 1's and setting the remaining elements as 0's. If the query image is a positively labeled image in *AssignedSet*, I also set the rows corresponding to all the other positive images in *AssignedSet* as 1's. The ranking score of each image is determined by the propagation of vector *Y* through the manifold graph *MG* constructed in step 5 in Figure 5.2. Let $P = [p_i]_{N\times1}$represent the ranking score vector for all images, where p_i is the ranking score of each image, and *N* is the number of database images. The system computes *P* by $MG\times Y$. Here, the images with higher scores are considered more similar to the query image. As a result, the system returns top *v* images with highest *v* ranking scores. The user then labels the returned images as relevant or irrelevant to the query. These labeled images are then incorporated into $Y = [y_i]_{N\times1}$ using the same cross-iteration checking and correction method to prevent the possible wrongly labeled images from propagating their labels.

The manifold graph *MG* is then multiplied with this updated *Y* to compute ranking scores for the next round. This process continues for a few iterations or until the user is satisfied with retrieval results.

5.2 Experiments and Results

I conduct a series of carefully designed experiments to evaluate the performance of the proposed system. The three smaller databases described in Chapter 4, namely, 2000-image database, 6000-image database, and 8000-image database, are used in my experiments since this single weighted manifold graph-based CBIR system cannot be employed in the two larger databases. In subsection *5.2.1*, I evaluate the effectiveness of the proposed system by comparing with variant systems on the benchmark 6000-image database. In subsection *5.2.2*, I evaluate the performance of the proposed system with selected peer systems on the three smaller image databases.

5.2.1 Effectiveness Evaluation

In the proposed system, I incorporate L_2-based low-level visual feature-based similarity, high-level semantic feature-based similarity, and importance score *IS* to build the single weighted manifold graph. The positive parameter σ in Equation (5.4) is set to be 0.05, the positive parameter σ_L and σ_H in Equation (5.5) are respectively set to be 0.05, the convergence rate α of the affinity matrix is set to be 0.99, and the parameter γ in the

RBF kernel is set to be 0.5. These values are empirically chosen to achieve the optimal retrieval performance.

To evaluate the effect of positive parameter w_h, which is used to combine the low-level visual features and high-level semantic features in Equation (5.4), on the proposed retrieval system, I experimentally test several values from 0 to 1 with a step size of 0.1 on the 6000-image benchmark database. Table 5.1 compares the retrieval performance in terms of ARP using different w_h's for four iterations.

Table 5.1. Performance difference with different w_h's

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**
w_h = 0	81.26%	87.54%	89.27%	90.33%
w_h = 0.1	81.51%	88.58%	90.01%	91.07%
w_h = 0.2	81.94%	89.61%	91.00%	91.90%
w_h = 0.3	82.13 %	90.51%	91.74%	93.10%
w_h = 0.4	82.31%	91.17%	92.79%	94.06%
w_h = 0.5	82.50%	91.52%	93.05%	94.33%
w_h = 0.6	82.63%	91.56%	93.05%	94.04%
w_h = 0.7	82.69%	91.62%	92.87%	93.82%
w_h = 0.8	82.64%	91.47%	92.69%	93.53%
w_h = 0.9	82.51%	90.64%	91.36%	92.04%
w_h = 1	81.52%	86.40%	86.80%	87.15%

Table 5.1 clearly demonstrates that the system achieves the best ARP at iterations 3 and 4 by using w_h = 0.5. Specifically, the system achieves 93.05% and 94.33% at

iterations 3 and 4, respectively. According to the performance difference obtained by using different w_h's on 6000-image benchmark database, I choose w_h as 0.5 for the proposed system.

To evaluate the effectiveness of the importance score *IS* which is defined in Equation (5.3), I implement its L_2-based and L_1-based variants:

- Variant 1: The CBIR system that incorporates L_2-based low-level visual features without *IS*.
- Variant 2: The CBIR system that incorporates L_1-based low-level visual features with *IS*.
- Variant 3: The CBIR system that incorporates L_1-based low-level visual features without *IS*.

Figure 5.5 compares the retrieval performance of the proposed system and its three variant systems on the 6000-image benchmark database. It clearly shows that the proposed system achieves the best ARP of 82.50% at iteration 1, 91.52% at iteration 2, 93.05% at iteration 3, and 94.33% at iteration 4. At the last iteration, the proposed system improves its opponent system without *IS* (variant 1) by 1.47%; the L_1-based system (variant 2) with *IS* improves its opponent system without *IS* (variant 3) by 1.26%. Meanwhile, it also clearly shows that the proposed system is not sensitive to the L_2 and L_1 kernel when building the weighted manifold graph, because the proposed system improves its L_1-based counterpart system (variant 2) by less than 0.9% at all four iterations. This can conclude that the proposed framework has the capability to apply L_2-based or L_1-based distance flexibly and it won't decrease the retrieval performance.

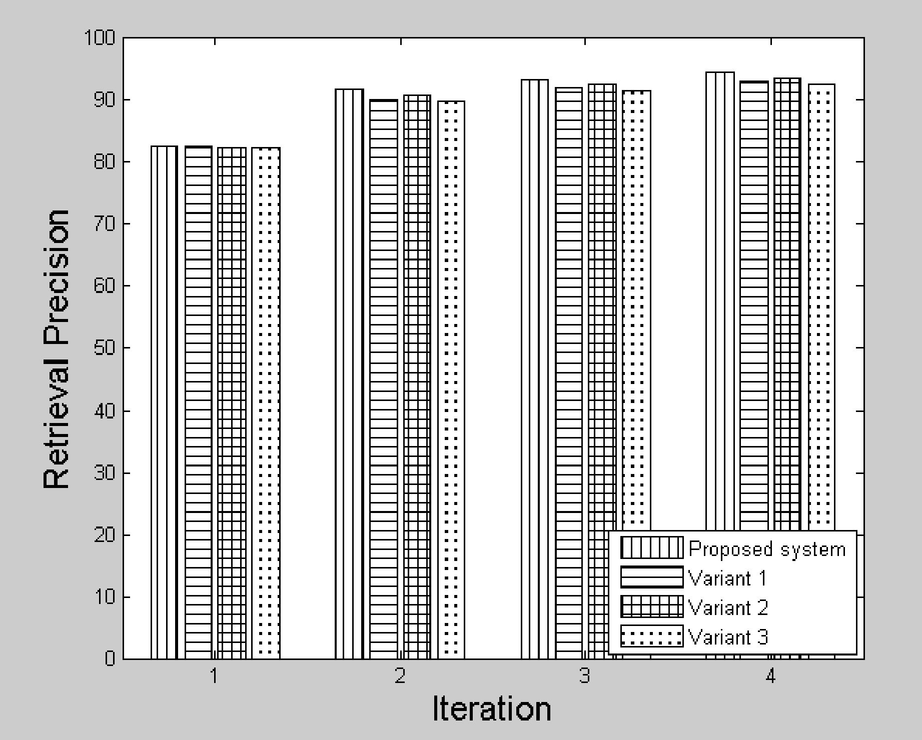


Figure 5.5. Comparison of the proposed system and its three variants, which are built from the compact feedback log

To further prove the effectiveness of the construction of the compact feedback log to extract high-level semantic features, I also implement four respective systems which are built on the full sized feedback log without applying any merging operation. These four systems are called as full feedback log based systems. Figure 5.6 compares the retrieval performance of these four full feedback log based systems on the 6000-image benchmark database. It clearly shows that the four systems demonstrate the same retrieval performance as their counterpart systems built from the compact feedback log whose retrieval performance is shown in Figure 5.5. Specifically, at the last iteration the proposed system built from compact feedback log has ARP of 94.33%, and its counterpart system built from full feedback log has ARP of 94.91%. The performance difference is less than 0.6%. In addition, the L_1-based system with *IS* (variant 2) built from compact feedback log has ARP of 93.49% at the last iteration, and its counterpart system (variant 2) built from full feedback log has ARP of 93.79% at the last iteration. The performance difference is only 0.3%. As a result, I claim that the merging method that is defined in section *4.1.1.4* works well to reformulate users' historical RF in a compact feedback log and extract representative semantic features for database images without bringing down the retrieval performance.

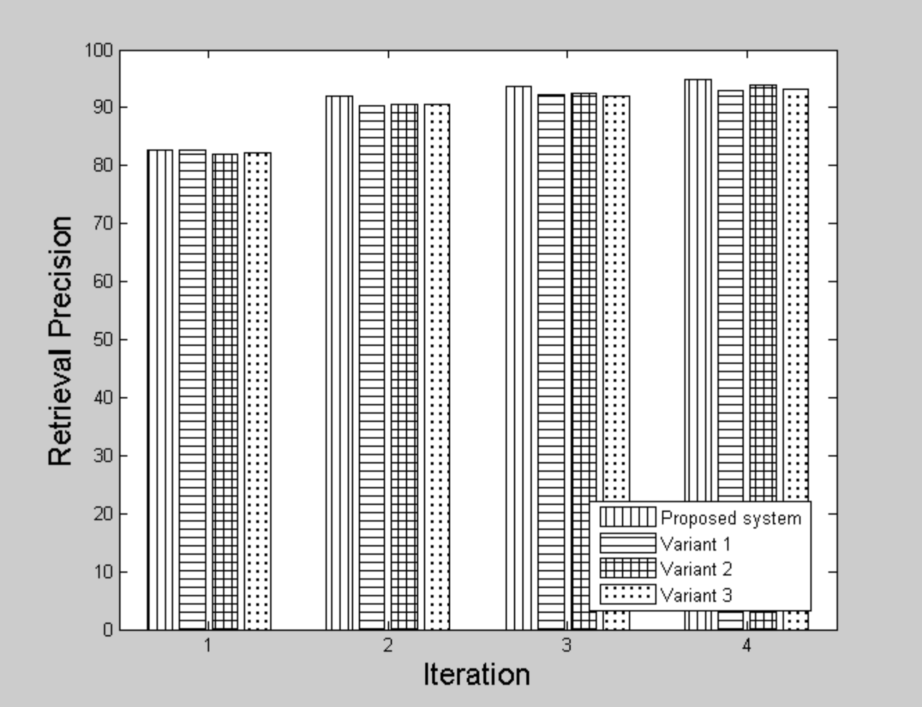


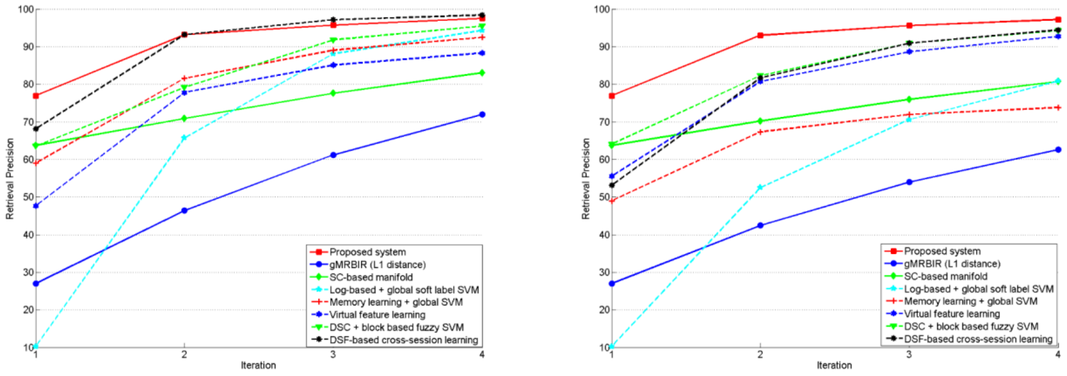
Figure 5.6. Comparison of the proposed system and its three variants, which are built from the full sized feedback log

5.2.2 Performance Evaluation

For a comprehensive performance evaluation of the proposed system, I compare the proposed system with seven state-of-the-art long-term-based CBIR systems on three image database. These compared systems that are clearly described in section 4.2.3 can be categorized into two groups:

- Manifold-based long-term learning systems: L_1-distance based gMRBIR [16], and semantic clusters based manifold ranking system (i.e., SC-based manifold) [67].
- Other long-term learning systems: log-based system (i.e., Log-based + global soft label SVM) [66], memory learning system (i.e., Memory learning + global SVM) [64], virtual feature-based system (i.e., Virtual feature learning) [61], dynamic semantic clustering system (i.e., DSC + block based fuzzy SVM) [35], dynamic semantic feature-based long-term cross-session learning system (i.e., DSF-based cross-session learning) [63].

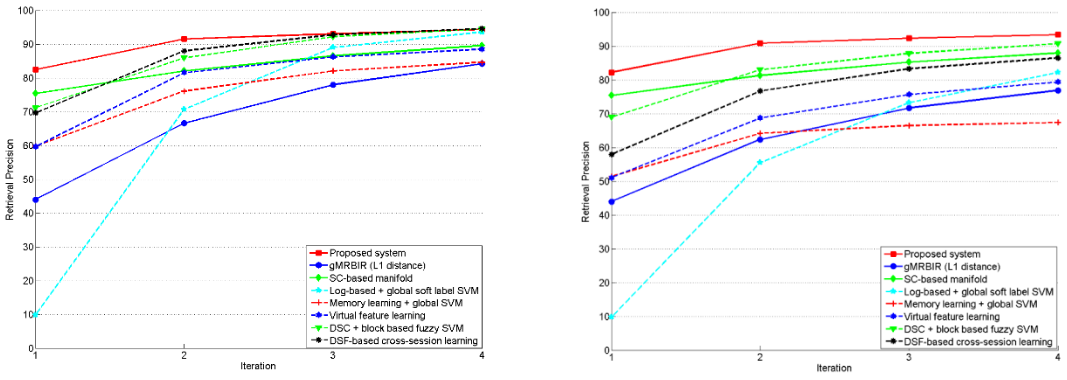
Figure 5.7 compares eight state-of-the-art long-term-based CBIR systems on 2000-image database in the context of correct feedback and 5% erroneous feedback. Figure 5.7 (a) shows the retrieval performance of the compared systems at each iteration on 2000-image database, where manifold-based systems are shown in solid lines and other long-term systems are shown in dashed lines. Specifically, the proposed system achieves ARP of 76.93% at iteration 1, 93.20% at iteration 2, 95.71% at iteration 3, and 97.48% at iteration 4. The proposed system has the best ARP at iterations 1 and 2 and it improves the performance than the second-best system (e.g., DSF-based cross-session learning approach) by 9% and 0.5% at iterations 1 and 2, respectively. It also achieves to be the second-best system at iterations 3 and 4, and the ARP is less than the DSF-based cross-session learning approach by just 1.9% and 0.9% at iterations 3 and 4, respectively. Figure 5.7 (b) shows the performance comparison among selected CBIR systems with users' erroneous feedback during the online retrieval phase. It clearly shows that the proposed system dominates the ARP at all four RF iterations. Specifically, it achieves ARP of 76.93%, 93.03%, 95.56%, and 97.21% at iterations 1, 2, 3, and 4, respectively. Moreover, the proposed system with users' erroneous feedback drops the performance by only 0.29% when comparing with the proposed system with correct feedback. This proves the robustness of the proposed system when the system resists the user's erroneous feedback during online retrieval phase. However, all other selected CBIR systems drop much ARP when the user's erroneous feedback is involved, including the system (i.e., DSF-based cross-session learning approach) that has the best performance at iterations 3 and 4 when users provide complete correct feedback.



(a) (b)

Figure 5.7. Comparison of eight state-of-the-art long-term-based CBIR systems on 2000-image database with (a) correct feedback and (b) with 5% erroneous feedback

Figure 5.8 compares eight state-of-the-art long-term-based CBIR systems on 6000-image database in the context of correct feedback and 5% erroneous feedback. Figure 5.8 (a) shows the retrieval performance of the compared systems at each iteration on 6000-image image database, where manifold-based systems are also shown in solid lines and other long-term systems are shown in dashed lines. The proposed system achieves the best performance at all four iterations. Specifically, it achieves ARP of 82.5% at iteration 1, 91.52% at iteration 2, 93.05% at iteration 3, and 94.33% at iteration 4. It improves the second-best system (e.g., SC-based manifold approach) at iteration 1 by 7.11%; improves the second-best system (e.g., DSF-based cross-session learning approach) at iteration 2 and 3 by 3.62% and 0.4%, respectively; and achieves the same ARP of 94.33% as DSF-based cross-session learning approach at iteration 4. Figure 5.8 (b) shows the performance comparison among selected CBIR systems with users' erroneous feedback during the online retrieval phase. It clearly shows that the proposed system achieves the best ARP at all iterations. Specifically, it achieves ARP of 82.21%, 90.86%, 92.30%, and 93.34% at iterations 1, 2, 3, and 4, respectively. Moreover, the proposed system with users' erroneous feedback drops the performance by only 0.99% when comparing with the proposed system with correct feedback. This proves the robustness of the proposed system when the system resists the user's erroneous feedback

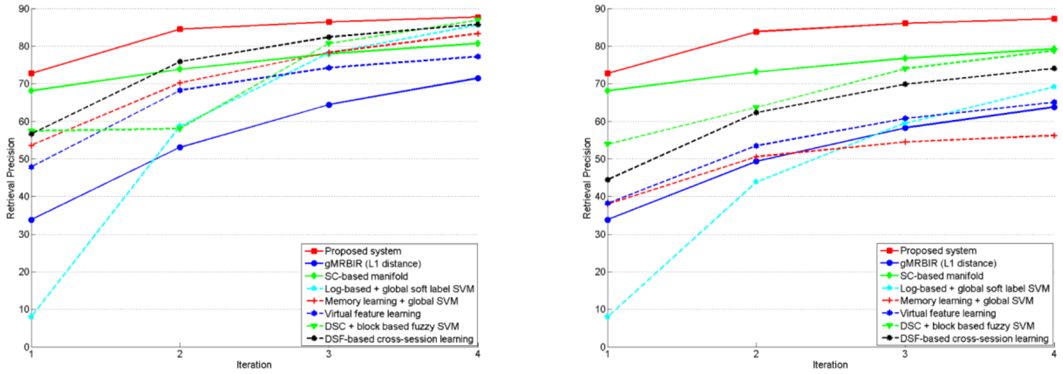


(a) (b)

Figure 5.8. Comparison of eight state-of-the-art long-term-based CBIR systems on 6000-image database with (a) correct feedback and (b) with 5% erroneous feedback

during online retrieval phase. Comparing with systems (e.g., SC-based manifold approach, and DSF-based cross-session learning approach) that achieve close retrieval performance at iterations 1, 2, 3, and 4 when the user provide correct feedback, both of them drop the ARP much more than the proposed system when the user's erroneous feedback is involved.

Figure 5.9 compares eight state-of-the-art long-term-based CBIR systems on 8000-image database in the context of correct feedback and 5% erroneous feedback. Figure 5.9 (a) shows the retrieval performance of the compared systems at each iteration on 8000-image database, where manifold-based systems are also shown in solid lines and other long-term systems are shown in dashed lines. The figure clearly shows that the proposed system achieves the best retrieval performance at all iterations. Specifically, it achieves ARP of 72.75% at iteration 1, 84.52% at iteration 2, 86.43% at iteration 3 and 87.72% at iteration 4. In details, at iteration 1, the proposed system improves the second-best system (e.g., SC-based manifold approach) by 4.6%; at iterations 2 and 3, it improves the second-best system (e.g., SC-based manifold approach) by 8.67% and 4.03% respectively; and at iteration 4, it improves the second-best system (e.g., DSC + block based fuzzy SVM) by 1.02%. Figure 5.9 (b) shows the performance comparison among selected CBIR systems with users' erroneous feedback during the online retrieval phase. It clearly shows that the proposed system achieves the best ARP at all iterations. Specifically, it achieves ARP of 72.75%, 83.87%, 86.06% and 87.28% at iterations 1, 2, 3 and 4, respectively. Moreover, the proposed system with users' erroneous feedback drops ARP by only 0.44% when comparing with the proposed system with correct feedback. This proves the robustness of the proposed system when the system resists the user's erroneous feedback during online retrieval phase.



(a) (b)

Figure 5.9. Comparison of eight state-of-the-art long-term-based CBIR systems on 8000-image database with (a) correct feedback and (b) with 5% erroneous feedback

From Figure 5.7 through 5.9, it is clear that the proposed system increases the ARP after each retrieval iteration on all three image databases. Meanwhile, the ARP of the proposed system decreases when the size of the image database increases. Specifically, in the context of the correct feedback, the ARP of the proposed system at the last iteration is 97.48% for the 2000-image database, 94.33% for the 6000-image database, and 87.72% for the 8000-image database. However, "DSF-based cross-session learning" achieves a little bit better ARP of 98.40%, 94.57% for the 2000-image database and 6000-image database, respectively; "DSC + block-based fuzzy SVM" achieves the second-best ARP of 86.69% for the 8000-image database. This clearly shows that the effectiveness of the proposed system on databases with different number and types of images. In the context of the erroneous feedback, the ARP of the proposed system at the last iteration is 97.21% for the 2000-image database, 93.34% for the 6000-image database, and 87.28% for the 8000-image database. Meanwhile, the second best system achieves the ARP of 94.44% for the 2000-image database, 90.68% for the 6000-image database, and 79.28% for the 8000-image database. Specifically, the second best system is “DSF-based cross-session learning” for the 2000-image database, “DSC + block-based fuzzy SVM” for the 6000-image database, and “SC-based manifold” for the 8000-image database. It clearly demonstrates that the robustness of the proposed system on databases with different number and types of images.

Figure 5.10 plots the precision and recall curves of the proposed system on the *8000-image* database when a different number of images (e.g., 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, and 155) are returned at each of four iterations. Specifically, precision represents ARP. Recall represents the average recall that is computed as the total of recall values of all query images divided by the total number of queries, where recall is defined as the ratio of the number of relevant images retrieved to the total number of relevant images (e.g., 100) in the database (see subsection 2.3). This figure clearly shows recall increases when the number of returned images increases for each iteration. However, precision for each iteration drops along with the increasing number of returned images. It also shows the proposed system is effective in returning above 70% of relevant images at the last iteration (i.e., ARP is above 70%) when the number of images returned is less than 105; meanwhile the recall is about 75%, which means the majority of relevant images from a category in the image database can be retrieved successfully (i.e., more than 75 relevant images can be retrieved out of total 100 relevant images in a category).

The precision-recall curve of the proposed system in the context of the erroneous feedback follows the same trend as the precision-recall curve of the proposed system in the context of the correct feedback. As shown in Figures 5.7 through 5.9, the retrieval performance under the erroneous feedback decreases a little bit compared to the retrieval performance under the correct feedback. As a result, the precision-recall curve of the proposed system under the erroneous feedback is moving downward. In other words, all the curves of the proposed system involved with the erroneous feedback are similar to the curves shown in Figure 5.10, except that the area under the curve area is smaller than its counterpart in Figure 5.10.

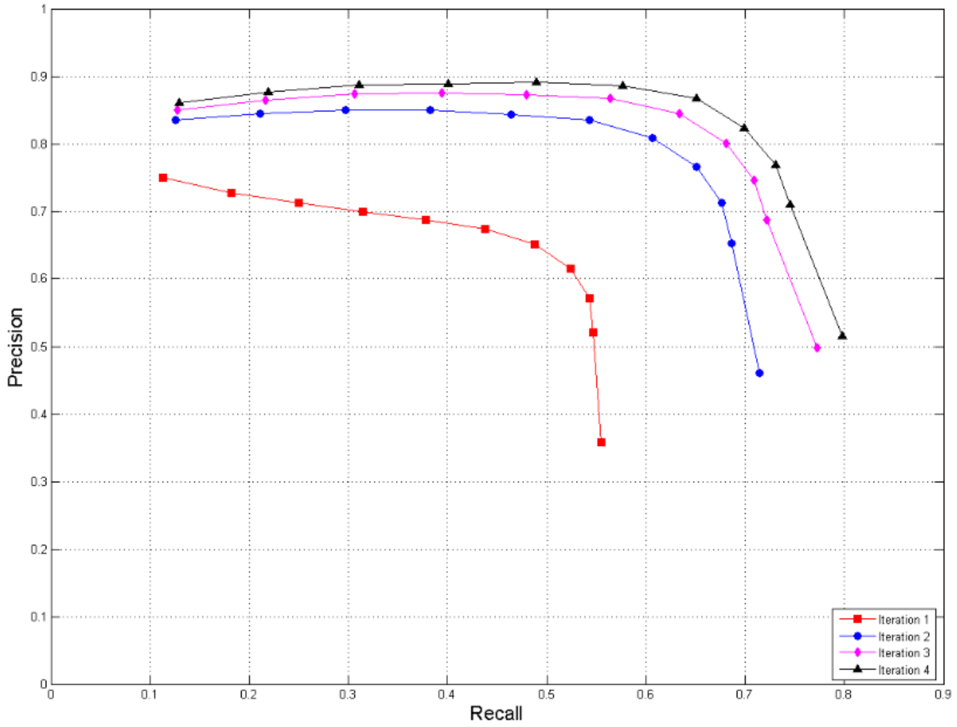


Figure 5.10. Precision and recall curve of the proposed system on 8000-image database

5.3 Conclusions and Future Work

In this chapter, I proposed a single weighted semantic manifold graph-based system for CBIR. The proposed system builds a more accurate intrinsic graph-based structure for the proper image space by combining low-level and high-level relations. Major contributions are: 1) Apply the learning mechanism to explore semantic concepts of the image database and approximately categorize database images into meaningful semantic categories. 2) Extract high-level semantic features of each image based on users' retrieval experiences. 3) Incorporate importance score and the composite relation into the affinity matrix to build the weighted semantic manifold graph. 4) Construct the asymmetric relevance vector to propagate ranking scores of its labeled images via the manifold to images with high similarities. Extensive experiments demonstrate the effectiveness of importance score and the updating strategy of the feedback log; and also shows the proposed system outperform two manifold-based and five long-term-based CBIR systems.

To address the scalability of the proposed system, I will investigate other strategies to reduce the size of the single manifold graph to be applicable in a large-scale database. I will also investigate other strategies to replace or eliminate the use of importance scores such that the system can save computation cost.

CHAPTER 6

CONCLUSIONS

With the rapidly growing number of digital images found on the Internet and housed in digital libraries, the need for effective and efficient tools to manage large image databases has grown dramatically. CBIR techniques are promising solutions to find desired images from image databases. However, the semantic gap is a challenge issue in CBIR systems. In this dissertation, I conduct the study of the CBIR technique and propose two novel CBIR systems that employ the historical user's RF to effectively bridge the semantic gap. The first system is the scalable manifold graph-based CBIR system, and the second system is single weighted manifold graph-based CBIR system.

The first system has the capability to perform the retrieval task in large-scale image databases but it requires more computation cost to construct powerful semantic graphs for the image database. Therefore, this system is suitable to carry out the retrieval task in the large-scale database. Major contributions of this CBIR system are summarized as follows:

- Quickly constructing a compact dynamic feedback log to store retrieval patterns of each past query session.
- Efficiently merging similar semantic concepts to maintain a reasonable number of representative semantics for all images in a database.
- Creatively constructing two-layer hierarchical graphs to represent the inherent structure of the large-scale image database during the system offline training stage.

- Effectively combining low-level visual and high-level semantic similarity measure to build a scalable manifold graph, which explores the intrinsic structure of images in both low-level visual and high-level semantic feature spaces.
- Effectively designing a layered relevance vector to propagate the relevance scores from anchor images to the second layer graphs and further propagate relevance scores of labeled images to unlabeled image via the hierarchical graph-based structure.

The single weighed semantic manifold graph-based CBIR is an effective graph-based CBIR system to perform retrieval task in small-scale image databases. This approach requires less computation cost to build a single manifold graph for the image database. Therefore, for retrieval tasks in small-scale image databases, this CBIR system is the right choice, since it is quicker to construct the manifold graph in the offline training phase and faster to retrieve images in the online retrieval phase. Major contributions of the single weighted semantic manifold graph-based CBIR system are summarized as follows:

- Applying the learning mechanism to explore semantic concepts of the image database.
- Extracting high-level semantic features of each image based on users' retrieval experiences.
- Incorporating the importance score of each image into the affinity matrix to build the weighted semantic manifold structure.

- Constructing the asymmetric relevance vector to propagate ranking scores of its labeled images via the manifold to images with high similarities.

In summary, this study effectively solves the great challenge issue existing in CBIR systems, which is the semantic gap. Meanwhile, it also shows powerful CBIR systems that obtain the promising potential to be applied in the real-world retrieval system for large-scale and small-scale image databases.

REFERENCES

[1] B. Thomée, "A picture is worth a thousand words: content-based image retrieval techniques," Ph.D. dissertation, Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden University, 2010.

[2] B. E. Prasad, *et al.,* "A microcomputer-based image database management system," *IEEE Trans. Industrial Electronics,* pp. 83-88, 1987.

[3] T. Kato, "Database architecture for content-based image retrieval," in *SPIE/IS&T 1992 Symp. on Electronic Imaging: Science and Technology*, pp. 112-123. International Society for Optics and Photonics, 1992.

[4] M. Flickner, *et al.,* "Query by image and video content: The QBIC system," *Computer,* vol. 28, no. 9, pp. 23-32, 1995.

[5] A. Gupta and R. Jain, "Visual information retrieval," *Communications of the ACM,* vol. 40, no. 5, pp. 70-79, 1997.

[6] S. Mukherjea, *et al.,* "Amore: A world wide web image retrieval engine," *World Wide Web,* vol. 2, no. 3, pp. 115-132, 1999.

[7] A. Pentland, *et al.,* "Photobook: Tools for content-based manipulation of image database," in *Proc. of the Conf. on Storage and Retrieval for Image and Video Database II, SPIE*, San Jose, CA., *1994.*

[8] J. Smith and S.–F. Chang, "VisualSEEK: A Fully Automated Content-Based Image Query System," in *Proc. of the ACM Int. Conf. on Multimedia,* 1997.

[9] W. Ma, and B. Manjunath, "Netra: A toolbox for navigating large image databases," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP),*1997.

[10] J. Z. Wang, *et al.*, "Content-based image indexing and searching using Daubechies' wavelets," *Int. J. on Digital Libraries,* vol. 1, no. 4, pp. 311-328, 1998.

[11] M. Kokare, *et al.,* "A survey on current content based image retrieval methods," *IETE J. of Research,* vol. 48, no. 3-4, pp. 261-271, 2002.

[12] S. Antani, *et al.,* "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," *Pattern recognition,* vol. 35, no. 4, pp. 945-965, 2002.

[13] A. W. Smeulders, *et al*., "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 22, no. 12, pp. 1349-1380, 2000.

[14] X. S. Zhou and S. H. Thomas, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia systems,* vol. 8, no. 6, pp. 536-544, 2003.

[15] T. Gevers and A. W. Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 102-119, 2000.

[16] J. He, *et al.*, "Generalized manifold-ranking-based image retrieval," *Image Processing, IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3170-3177, 2006.

[17] J. Brank, “Image Categorization Based On Segmentation And Region Clustering,” in *Proc. of the 1st Starting AI Researchers Symp. (STAIRS)*, Lyon, France, 2002, pp. 145-154.

[18] C. Carson, *et al.*, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol.24, no. 8, pp. 1026-1038, 2002.

[19] Y. Chen and J. Z. Wang, "A region-based fuzzy feature matching approach to content-based image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol.24, no.9, pp.1252-1267, 2002.

[20] S. Kumar, *et al.*, "An observation-constrained generative approach for probabilistic classification of image regions," *Image and Vision Computing*, vol.21, no.1, pp.87-97, 2003.

[21] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," *Multimedia systems,* vol. 7, no. 3, pp. 184-198, 1999.

[22] O. Maron and A. L. Ratan, "Multiple-Instance learning for natural scene classification," in *ICML*, vol. 98, pp. 341-349, 1998.

[23] J. R. Smith and S. F. Chang, "VisualSEEk: a fully automated content-based image query system," in *Proc. of the 4th ACM Int. Conf. on Multimedia*, pp. 87-98, ACM, 1997.

[24] J. R. Smith and C. S. Li. "Image classification and querying using composite region templates," *Computer Vision and Image Understanding*, vol. 75, no. 1, pp. 165-174, 1999.

[25] J. Z. Wang, *et al.*, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 23, no. 9, pp. 947-963, 2001.

[26] J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 25, no. 9, pp. 1075-1088, 2003.

[27] Y. Lu, *et al.*, "Joint semantics and feature based image retrieval using relevance feedback," *IEEE Trans. Multimedia,* vol. 5, no. 3, pp. 339-347, 2003.

[28] Sheikholeslami, *et al.,* "SemQuery: semantic clustering and querying on heterogeneous features for visual data," *IEEE Trans. Knowl. Data Eng.* vol. 14, no. 5 pp. 988-1002, 2002.

[29] A. Vailaya, *et al.,* "Image classification for content-based indexing," *Image Processing, IEEE Trans. Image Process.* vol. 10, no. 1, pp. 117-130, 2002.

[30] N. Vasconcelos, "Exploiting group structure to improve retrieval accuracy and speed in image databases," in *Image Processing 2002. Proc. 2002 Int. Conf.*, vol. 1, pp. 980-983, IEEE, 2002.

[31] I. J. Cox, *et al.,* "The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments," *Image Processing, IEEE Trans. Image Process,* vol. 9, no. 1, pp. 20-37, 2000.

[32] M. Ortega, *et al.*, "Supporting similarity queries in MARS," in *Proc. of the 5th ACM int. conf. on Multimedia*, pp. 403-413. ACM, 1997.

[33] Y. Rui, *et al.,* "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.* vol.8, no.5, pp.644-655, 1998.

[34] J. R. Smith, “Integrated spatial and feature image systems: Retrieval, Compression and analysis,” *Ph.D. dissertation, Graduate School of Arts and Sciences,* Columbia University, 1997.

[35] Qi, Xiaojun, *et al.,* "A noise-resilient collaborative learning approach to content-based image retrieval," *Int. J. of Intelligent Systems,* vol. 26, no. 12, pp. 1153-1175, 2011.

[36] X. J. Qi and Y. T. Han, "A novel fusion approach to content-based image retrieval," *Pattern Recognition,* vol. 38, no. 12, pp. 2449-2465, 2005.

[37] Y. Liu, *et al.,* "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition,* vol. 40, no. 1, pp. 262-282, 2007.

[38] A. Kushki, *et al.,* "Query feedback for interactive image retrieval." *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 644-655, 2004.

[39] P. Muneesawang and L. Guan, "An interactive approach for CBIR using a network of radial basis functions," *IEEE Trans. Multimedia,* vol. 6, no. 5, pp. 703-716, 2004.

[40] D. H. Widyantoro and J. Yen, "Relevant data expansion for learning concept drift from sparsely labeled data," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 401-412, 2005.

[41] R. S. Michalski, "Readings in knowledge acquisition and learning," *Morgan Kaufmann Publishers Inc.*, San Francisco, CA., USA, 1993, pp. 323-348.

[42] S. D. MacArthur, *et al.*, "Relevance feedback decision trees in content-based image retrieval," in *Content-based Access of Image and Video Libraries, 2000. Proc. IEEE Workshop*, pp.68-72, IEEE, 2000.

[43] E. Chang, *et al.*, "CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 26-38, 2003.

[44] Z. Su, *et al.*, "Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning," *Image Processing, IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 924-937, 2003.

[45] S. Wilson and G. Stefanou, "Bayesian approaches to content-based image retrieval," *Proc. of the Int. Workshop/Conf. on Bayesian Statistics and Its Applications, Varanasi, India (January 2005)*, 2005.

[46] S. Tong and E. Chang. "Support vector machine active learning for image retrieval," in *Proc. of the 9th ACM Int. Conf. on Multimedia*, pp. 107-118, ACM, 2001.

[47] K. Wu and K. H. Yap, "Fuzzy SVM for content-based image retrieval: a pseudo-label support vector machine framework," *IEEE Comput. Intell. Mag.*, vol. 1, no. 2, pp. 10-16, 2006.

[48] K. Tieu and P. Viola, “Boosting image retrieval,” in *Proc. of IEEE Int. Conf. of Computer Vision and Pattern Recognition*, 2000, pp. 228-235.

[49] J. R. He, *et al.*, "Manifold-ranking based image retrieval," in *Proc. of the 12th annual ACM Int. Conf. on Multimedia*, pp. 9-16, ACM, 2004.

[50] D. Cai, *et al.*, "Regularized regression on image manifold for retrieval," in *Proc. of the int. workshop on Workshop on Multimedia Information Retrieval*, pp. 11-20, ACM, 2007.

[51] F. Wang, *et al.*, "Inequivalent manifold ranking for content-based image retrieval," in *Image Processing, 2008. ICIP 2008. 15th IEEE Int. Conf.*, pp.173-176, IEEE, 2008.

[52] Y. Y. Lin, *et al.*, "Semantic manifold learning for image retrieval," in *Proc. of the 13th annual ACM int. conf. on Multimedia*, pp. 249-258, ACM, 2005.

[53] X. J. Wan, "Content based image retrieval using manifold-ranking of blocks," in *Multimedia and Expo, 2007 IEEE Int. Conf.*, pp. 2182-2185, IEEE, 2007.

[54] X. M. Liu, *et al.*, "Bidirectional-isomorphic manifold learning at image semantic understanding & representation," *Multimedia Tools and Applications*, vol. 64, no. 1, pp. 53-76, 2013.

[55] Y. H. Han, *et al.*, "Image classification with manifold learning for out-of-sample data," *Signal Processing*, vol. 93, no. 8, pp. 2169-2177, 2013.

[56] B. Xu, *et al.*, "A Bregman Divergence Optimization Framework for Ranking on Data Manifold and Its New Extensions," in *Proc. on the 26th AAAI Conf. on Artificial Intelligence*, pp. 1190 – 1196, 2012, Toronto, Ontario, Canada.

[57] J. Li and N. M. Allinson, "Long-term learning in content-based image retrieval," *Int. J. of Imaging Systems and Technology*, vol. 18, no. 2-3, pp. 160-169, 2008.

[58] S. C. Deerwester, *et al.*, "Indexing by latent semantic analysis," *JASIS* 41, no. 6, pp. 391-407, 1990.

[59] D. R. Heisterkamp, "Building a latent semantic index of an image database from patterns of relevance feedback," in *Pattern Recognition, 2002. Proc. 16th Int. Conf.*, vol. 4, pp. 134-137, IEEE, 2002.

[60] X. D. Zhou, *et al.*, "A relevance feedback method in image retrieval by analyzing feedback log file," in *Machine Learning and Cybernetics, 2002. Proc. 2002 Int. Conf.*, vol. 3, pp. 1641-1646, IEEE, 2002.

[61] P. Y. Yin, *et al.*, "Long-term cross-session relevance feedback using virtual features," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 3, pp. 352-368, 2008.

[62] X. F. He, *et al.*, "Learning a semantic space from user's relevance feedback for image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 39-48, 2003.

[63] M. Cord and P. H. Gosselin, "Image retrieval using long-term semantic learning," in *Image Processing, 2006 IEEE Int. Conf.*, pp. 2909-2912, IEEE, 2006.

[64] J. W. Han, *et al.*, "A memory learning framework for effective image retrieval," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 511-524, 2005.

[65] Z. M. Xiao, *et al.*, "Dynamic semantic feature-based long-term cross-session learning approach to content-based image retrieval," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE Int. Conf.*, pp. 1033-1036, IEEE, 2012.

[66] S. C. Hoi, *et al.*, "A unified log-based relevance feedback scheme for image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 4, pp. 509-524, 2006.

[67] R. Chang and X. J. Qi, "Semantic clusters based manifold ranking for image retrieval," in *Image Processing (ICIP), 2011 18th IEEE Int. Conf.*, pp. 2425-2428, IEEE, 2011.

[68] R. Chang, *et al.*, "Learning a weighted semantic manifold for content-based image retrieval," in *Image Processing (ICIP), 2012 19th IEEE Int. Conf.*, pp. 2401-2404, IEEE, 2012.

[69] R. Chang and X. J. Qi, "A hierarchical manifold subgraph ranking system for content-based image retrieval," in *Multimedia and Expo (ICME), 2013 IEEE Int. Conf.*, pp. 1-6, IEEE, 2013.

[70] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 2nd,* Prentice Hall: Upper Saddle River, NJ, 2002.

[71] D. Pascale, "A review of rgb color spaces... from xyy to r’g’b’," *Babel Color,* 2003.

[72] A. R. Smith, "Color gamut transform pairs," in *ACM Siggraph Computer Graphics*, vol. 12, no. 3, pp. 12-19, ACM, 1978.

[73] R. C. Gonzalez, *et al.*, *Digital image processing using MATLAB*, Pearson Education India, 2004.

[74] L. Shapiro and G. C. Stockman, *Computer Vision. 2001,* Englewood Cliffs, NJ : Prentice Hall, 2001.

[75] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning* 20, no. 3, pp.273-297, 1995.

[76] D. Y. Zhou, *et al.*, "Learning with local and global consistency," in *NIPS*, vol. 16, pp. 321-328, 2003.

[77] D. Zhou, *et al.*, "Ranking on data manifolds," in *Neural Information Processing systems*, vol. 16, pp. 169-176, 2003.

[78] I. J. Cox, *et al.,* "The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments," *IEEE Trans. Image Process.*, vol. 9, no.1, pp. 20-37, 2000.

[79] X. J. Qi and R. Chang. "Image retrieval using transaction-based and SVM-based learning in relevance feedback sessions," in *Image Analysis and Recognition*, Berlin Heidelberg: Springer, pp. 638-649, 2007.

[80] S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognition Letters,* vol. 22, no. 5, pp. 563-582, 2001.

[81] T. S. Chua, *et al.*, "NUS-WIDE: a real-world web image database from National University of Singapore," in *Proc. of the ACM Int. Conf. on Image and Video Retrieval*, pp. 48, ACM, 2009.

[82] G. Pass, *et al.*, "Comparing images using color coherence vectors," in *Proc. of the 4th ACM Int. Conf. on Multimedia*, pp. 65-73, ACM, 1997.

[83] N. L. Johnson, *et al.*, "Chapter 16," in *Continuous Univariate Distributions, Vol. 1*. New York: Wiley, 1994, pp. 298-336.

CURRICULUM VITAE

Ran Chang
(2013)

EDUCATION

Ph.D. Computer Science May 2014(expected)
Utah State University (USU), Logan, UT, U.S.A. GPA: 3.61
Dissertation: "*Effective Graph-Based Content-Based Image Retrieval Systems for Large-Scale and Small-Scale Image Databases*"
Advisor: Xiaojun Qi

M.S. Electrical Engineering May 2007
Utah State University (USU), Logan, UT, U.S.A. GPA: 3.42

M.S. Electrical Engineering July 2004
University of Electronic Science and Technology of China (UESTC), Chengdu, China
GPA: 3.50
Advisor: Yubai Li

B.S. Electrical Engineering July 2001
University of Electronic Science and Technology of China (UESTC), Chengdu, China
GPA: 3.63

RESEARCH INTERESTS

- Image Processing
- Computer Vision
- Machine Learning
- Pattern Recognition

RESEARCH EXPERIENCE

Graduate Student/Research Assistant

- Dr. Xiaojun Qi's computer vision research lab, Department of Computer Science, Utah State University

  Aug. 2006 – Present

  - Research topics include *Content-based Image Retrieval* with multiple techniques such as methods based on supervised, unsupervised and semi-supervised learning techniques, users' online relevance feedback techniques and manifold techniques.
  - Working on the *Hierarchical User Feedback Content-based Image Retrieval* system, fully responsible for the system design, algorithm development, debugging and testing.
  - Working on *Semi-Fragile Water Marking* technique, responsible for algorithm development, debugging and testing.

- Experienced in many aspects of machine learning, computer vision and image processing, including motion detection and tracking, image forensics, and object/pattern recognition.

➢ WAVE, Inc., Utah State University Commercial Enterprise

Oct. 2012 – Jan.2014

- Working on the project of Wireless Power Transmission for Electrical Vehicles which is dedicated for the advanced electric energy transferring from the power station to the battery system on the electrical vehicle in a transdutive way. Taking charge of the communication module design and implementation among the power receiver pad on the vehicle, the battery manager system and other related ECUs in the vehicle based on the CAN bus communication protocol and SAE J1939.

➢ Energy Dynamics Laboratory, Utah State University Research Foundation

Oct. 2010 – Aug. 2012

- Working on the *Multiple-person Occupancy Sensor* which will be used to assist in the reduction of energy consumption in office buildings, responsible for algorithm development, implementation, debugging and testing of the sensor software.
- Working on the development of the *Human Activity Sensor* which will be used to assist in taking care of the elder people or people with disabilities in apartments.

➢ College of Electrical and Computer Engineering, Utah State University

Aug. 2004 – Aug. 2006

- Research topic includes *Direction of Arrival and Ray Tracing* by using sensor-array based on variant algorithms.

➢ TI's DSP Laboratory, School of Communication and Information Engineering, University of Electronic Science and Technology of China (UESTC)

Sep. 2002 – Aug. 2004

- Participated in the project on *GSM Mobile Integration Testing Device*, supported by Qianfeng Electronic Corporation, responsible for the receiver module design, debugging and testing based on TI's DSPs.
- Participated in writing the book *TI's DSP Integrated Development Environment CSS handbook* (published in 2005 by Tsinghua University).
- Participated in writing the book *OMAP Application Platform*, responsible for the sections "Introduction", "System Initialization", and "Usage of OMAP 1510".

Ungraduate Research Assistant

➢ School of Electrical Engineering, University of Electronic Science and Technology of China (UESTC)

Aug. 2000 – Aug. 2001

- Senior project on *Manufacture Control Application on a Certain Type of Fighter's Windshields without Frames*, responsible for the development of the mutual control part and real-time dynamic 3D visualization of the windshield manufacture by using OPENGL/C++.

TEACHING EXPERIENCE

Instructor

- ➢ School of Further Education, University of Electronic Science and Technology of China (UESTC)

  Sep. 2003 – Feb. 2004

  - • Teaching the course "*UNIX/LINUX Operating System, Fundamental Knowledge*" for seniors, also responsible for weekly homework discussion & grading and midterm, final exams design & grading.

Teaching Assistant

- ➢ Department of Computer Science, Utah State University

  Sep. 2006 – Dec. 2012

  - • Taking charge of answering questions and grading for "*Computer Vision, Pattern Recognition, and Image Processing*" (CS5650), "*Computer Organization and Architecture*" (CS2810) and "*Computer Science II: Introduction to Computer Science with C++*" (CS1410)

- ➢ College of Electrical and Computer Engineering, Utah State University

  Sep. 2004 – Dec. 2015

  - • Taking charge of answering questions and grading for "*Electromagnetic I*" (ECE 3870), "*Stochastic Processes in Electronic Systems*" (ECE6010), "*Mathematical Methods for Signals and Systems*" (ECE6030)

- ➢ School of Electrical Engineering, University of Electronic Science and Technology of China (UESTC)

  Sep. 2000 – Aug. 2001

  - • Student advisor for freshmen

PUBLICATIONS

Patents

- • Aravid Dasu, **Ran Chang**, Chenguang Liu, Pranab Banerjee, Bruce Christensen, Juan De la Cruz, and Doug Ahlstrom, "Systems, devices, and methods for monitoring and controlling a controlled space." United States WO2013013079 A3, Filed July 19, 2012

- • Aravid Dasu, **Ran Chang** and Chenguang Liu, "Systems, devices, and methods for multi-occupant tracking." United States WO2013013082 A1, Filed July 19, 2012

Referred Journal Articles

- • Xiaojun Qi and **Ran Chang**, "A Scalable Graph-Based Semi-Supervised Ranking System for Content-Based Image Retrieval," to appear in *International Journal of Multimedia Data Engineering and Management*, Vol. 4, No. 4, 2014.

- Xiaojun Qi, Samuel Barrett, and **Ran Chang**, "A Noise-Resilient Collaborative Learning Approach to Content-Based Image Retrieval," *International Journal of Intelligent Systems*, Vol. 26, No. 12, pp. 1153-1175, 2011.

Referred Conference Articles

- **Ran Chang** and Xiaojun Qi, "A Hierarchical Manifold Subgraph Ranking System for Content-Based Image Retrieval," *IEEE Int. Conf. on Multimedia and Expo (ICME'13)*, July 15-19, San Jose, California, 2013.

- **Ran Chang**, Zhongmiao Xiao, KokSheik Wong, and Xiaojun Qi, "Learning a Weighted Semantic Manifold for Content-Based Image Retrieval," *IEEE Int. Conf. on Image Processing (ICIP'12)*, pp. 2401-2404, Sept. 30-Oct. 3, Orlando, Florida, 2012.

- **Ran Chang** and Xiaojun Qi, "Semantic Clusters Based Manifold Ranking for Image Retrieval," to appear in *IEEE Int. Conf. on Image Processing (ICIP'11)*, Sept. 11-14, Brussels, Belgium, 2011.

- Adam D. Gilbert, **Ran Chang**, and Xiaojun Qi, "A Retrieval Pattern-Based Inter-Query Learning Approach for Content-Based Image Retrieval," to appear in *IEEE Int. Conf. on Image Processing (ICIP'10)*, Sept. 26-29, Hong Kong, 2010.

- Scott Fechser, **Ran Chang**, and Xiaojun Qi, "Inter-query Semantic Learning Approach to Image Retrieval," to appear in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'10)*,March 14-19, Dallas, Texas, USA, 2010.

- Xiaojun Qi, Xing Xin, and **Ran Chang**, "Image Authentication and Tamper Detection Using Two Complementary Watermarks," Proc. of *IEEE International Conference on Image Processing (ICIP'09)*, Nov. 7-11, Cairo, Egypt, 2009.

- Samuel Barrett, **Ran Chang**, and Xiaojun Qi, "A Fuzzy Combined Learning Approach to Content-Based Image Retrieval," *Proc. of IEEE International Conference on Multimedia & Expo*, pp.838-841, June 28-July 3, New York, USA, 2009.

- Xiaojun Qi and **Ran Chang**, "A Fuzzy Statistical Correlation-Based Approach to Content-Based Image Retrieval," *Proc. of IEEE International Conference on Multimedia and Expo (ICME'08)*, pp.1265-1268, June 23-26, Hannover, Germany, 2008.

- Xiaojun Qi and **Ran Chang**, "Image Retrieval Using Transaction-Based and SVM-Based Learning in Relevance Feedback Sessions," *Proc. of International Conference on Image Analysis and Recognition (ICIAR'07)*, LNCS 4633, pp. 638-649, August 22-24, Montreal, Canada, 2007.

- Matthew Royal, **Ran Chang**, and Xiaojun Qi, "Learning From Relevance Feedback Sessions Using a K-Nearest-Neighbor-Based Semantic Repository," *Proc. of IEEE International Conference on Multimedia and Expo (ICME'07)*, pp. 1994-1997, July 2-5, Beijing, China, 2007. Working under NSF REU Funding

Referred Published Book

- Qicong Peng, Shiya Zhang, and **Ran Chang**, Book "TI's DSP Integrated Development Environment CSS handbook," published in 2005 by Tsinghua University, ISBN: 7302121494, 9787302121497

SKILLS

Software

- Languages: C/ C++, C#, Python, Assembly Language (x86 CPUs and TI's DSPs)
- Professional tools: MATLAB, OPENCV, Code Fire (Freescale), CCS(Code Composer Studio) of TI.
- Other: TEX/ LATEX.

Hardware

- Protocol: SAE J1939, CAN bus communication in Automobile Industrial
- Languages: VHDL, Verilog.
- Professional tools: Protel, Modelsim, ISE.
- Devices: Signal generator, Spectrum analyzer, Mixed signal oscilloscope (Agilent)

Professional Activies

- Reviewer of International Journal of Software and Informatics (IJSI), 2012
- Reviewer of IEEE International Conference on Image Processing (ICIP'10)

AWARDS AND HONORS

- Summer 2005 Intramural Championship of Men's Score in Utah State University
- 2003 Outstanding Graduate Student Fellowship in UESTC
- 2002, 2003 Basketball Championship of Graduate School in UESTC
- 2001 Excellent Undergraduate Thesis Award in UESTC
- 1998, 2000, 2001 Outstanding Undergraduate Student Fellowship in UESTC

LANGUAGES

- English (Fluent)
- Chinese (Native)