2007

Semantic image retrieval using relevance feedback and transaction logs

Amin Shah-hosseini
Louisiana State University and Agricultural and Mechanical College

SEMANTIC IMAGE RETRIEVAL USING RELEVANCE FEEDBACK AND TRANSACTION LOGS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical Collage
In partial fulfillment of the
Requirements for the degree of
Doctor of Philosophy

In

The Interdepartmental Program in Engineering Science

By
Amin Shah-hosseini
B.S. in Industrial Engineering, Sharif University of Technology, 2000
M.S. in Industrial Engineering, Sharif University of Technology, 2002
August 2007

TABLE OF CONTENTS

ABSTRACT

Due to the recent improvements in digital photography and storage capacity, storing large amounts of images has been made possible, and efficient means to retrieve images matching a user's query are needed. Content-based Image Retrieval (CBIR) systems automatically extract image contents based on image features, i.e. color, texture, and shape. Relevance feedback methods are applied to CBIR to integrate users' perceptions and reduce the gap between high-level image semantics and low-level image features. The precision of a CBIR system in retrieving semantically rich (complex) images is improved in this dissertation work by making advancements in three areas of a CBIR system: input, process, and output. The input of the system includes a mechanism that provides the user with required tools to build and modify her query through feedbacks. Users behavioral in CBIR environments are studied, and a new feedback methodology is presented to efficiently capture users' image perceptions. The process element includes image learning and retrieval algorithms. A Long-term image retrieval algorithm (LTL), which learns image semantics from prior search results available in the system's transaction history, is developed using Factor Analysis. Another algorithm, a short-term learner (STL) that captures user's image perceptions based on image features and user's feedbacks in the on-going transaction, is developed based on Linear Discriminant Analysis. Then, a mechanism is introduced to integrate these two algorithms to one retrieval procedure. Finally, a retrieval strategy that includes learning and searching phases is defined for arranging images in the output of the system.

The developed relevance feedback methodology proved to reduce the effect of human subjectivity in providing feedbacks for complex images. Retrieval algorithms were applied to images with different degrees of complexity. LTL is efficient in extracting the semantics of

complex images that have a history in the system. STL is suitable for query and images that can be effectively represented by their image features. Therefore, the performance of the system in retrieving images with visual and conceptual complexities was improved when both algorithms were applied simultaneously. Finally, the strategy of retrieval phases demonstrated promising results when the query complexity increases.

CHAPTER 1

INTRODUCTION

Due to the advances in digital photography, storage capacity and networks speed, storing large amounts of high quality images has been made possible. Digital images are used in a wide range of applications such as medical, virtual museums, military and security purposes, and personal photo albums. However, users have difficulties in organizing and searching large numbers of images in databases, as the current commercial database systems are designed for text data and not well suited for digital images. Therefore, an efficient way for image retrieval is desired.

In order to respond to this need, researchers have tried extending Information Retrieval (IR) techniques used in text retrieval to the area of image retrieval. In this approach, a set of keywords are assigned to each image. Then, IR techniques such as term frequency (tf) and inverse document frequency (idf) are used to estimate the weights of the keywords associated with the document based on the keywords in the search query and the documents (images) with smaller distances to the query are returned to the user [105]. However, there are significant limitations to this approach. First, the approach is not scalable since each object needs to be manually annotated with keywords and/or textual descriptions, making it impractical for large data sets. Second, due to the subjectivity of the human annotator, the annotations may not be consistent or complete which negatively effects retrieval performance. Furthermore, it may be infeasible to describe visual content (e.g., shape of an object) simply using words.

To overcome the above problems, researchers applied advances in image processing, database management, and information retrieval to the area of image retrieval and introduced Content-Based Image Retrieval (CBIR) in the 1990's. In CBIR systems, image processing

techniques are used to extract visual features such as color, texture and shape from images. Therefore, images are represented as a vector of extracted visual features instead of just pure textual annotations. An object model is defined to represent images based on visual features. A user formulates a query by providing examples of images similar to the ones s/he wishes to retrieve. The system uses a query model to convert the image into an internal representation of query, based on features extracted from input images. A retrieval model performs image retrieval by computing similarities between images in object and the query representations, and the results are ranked based on the computed similarity values. Overall similarity (distance) between an object and the image query is computed as a weighted summation of similarities (distances) over the feature set. The object, query, and retrieval models together define a CBIR model [86].

The retrieval model may include an image indexing or clustering module, which expedites searching in large image databases. Due to the high dimensionality of feature vectors, image clustering is challenging. Traditional database techniques are basically designed for low dimension indices and are not efficient with high dimension indices. Another significant issue in image clustering is that images with similar semantics may not fall in one cluster as image clustering is performed based on image low-level features. Many approaches have been proposed to reduce the gap between high-level image semantics and low-level image features and improve the clusters by applying image segmentation techniques on region-based features [94] and clustering image segments instead of original images [107].

Since all image low-level features cannot capture high-level semantic concepts, most retrieval methods have tried to find an optimum set of feature weights to model the user's perception based on image features (feature weighting). Some CBIR systems ask the user to set the feature weights [13]; however, there are several shortcomings to such approaches. Users may

find it difficult to express their query appropriately in terms of the provided features since they do not initially have a clear idea of the information needed. Furthermore, there may be a mismatch between the users' perception of the visual properties and the feature representations that are actually used for retrieval.

Relevance feedback approaches have been successfully applied in the information retrieval area [104]. In such approaches, the user needs to provide the retrieval system with positive examples, negative examples or both. In a CBIR system, positive examples are images that are similar to the images the user is looking for, and negative examples are those that are not similar to user's query. In each retrieval iteration, the system uses relevance feedback data to modify feature weights in order to create a more accurate query model.

However, most feedback approaches use the feedbacks only in the current query session and do not have a learning mechanism to memorize the feedbacks provided previously to reuse them in favor of future image retrievals [75]. Recommendation systems, an emerging technology in e-commerce, store the feedbacks from all the users to help them in choosing products they are likely to be interested in. Such systems have also been applied in web browsing to help users find web pages they are interested in. Considering images as the products (or web pages) in a recommendation system, the techniques used in such systems can be applied to image retrieval systems to improve the quality of retrieval. Recent image retrieval approaches have been proposed based on long-term learning from previous feedbacks as well as short-term learning from feedbacks in the current query session [128]. Different approaches have been proposed based on Collaborative Filtering [131] - a technique used in recommendation systems, Support Vector Machines (SVM) [16, 89, 91] - a learning and classification method, machine learning methods [122], and probabilistic methods [85].

1.1 PROBLEM STATEMENT

A significant problem in Content-based Image Retrieval (CBIR) systems is the gap between high-level semantics in human minds and low-level features computable by machines. This dissertation proposes new methods and features to be applied to CBIR systems to reduce the gap between image semantics and image features, and improve the image retrieval performance.

The proposed method not only refines the query by using the relevance feedback data in the current query session, but also learns the image semantics from relevance feedback data in previous queries. In a model-based approach, the proposed method defines a model to create the image semantics in an image database and find the relationship between images and semantic classes. In the retrieval process, the system finds the similarity of the current query session to the semantic classes in the database and returns highly ranked images in those classes.

Figure 1.1 Example of a semantically rich image

In most of the CBIR systems, relevance feedback is provided in the form of positive and negative examples. However, when images or query concepts are semantically rich, it is not convenient for the users to transfer the degrees of relevancy they have in their minds through binary feedbacks to the system; therefore, the quality of the system input is reduced and learning performance plunges. For example, the image shown in Figure 1.1 is semantically rich because it

is related to many concepts such as a city, river, boat, mountain, etc. In the proposed method, users have the flexibility of labeling the retrieved images with a score between 0 and 1.

Moreover, related works in the area of learning image semantics from previous relevance feedback data can assign binary memberships to the images; however, the proposed method has the ability to assign different degrees of relevancy to each image for the semantic classes.

1.2 OBJECTIVES

Given the stated image retrieval problem and related clustering and retrieval issues, the objective of this dissertation is to develop and evaluate a new image retrieval method for an image database with the following characteristics:

- Images in the database belong to many semantic classes with different levels of relevancy
- The retrieval process is based on learning image semantics
- Learning is based on the relevance feedback data from current and prior users
- Image features are used in computing image similarities

To achieve the above objectives, three areas are identified in a CBIR system as input, process, and output, and the following steps are taken in this dissertation:

- Evaluate feedback-providing methods to find the optimum scheme that minimizes the variance between feedbacks provided by different users, or a single user in different sessions, for a specific query concept
- Develop and evaluate an image retrieval algorithm that learns from transactions history and applies image features in computing image and query similarities
- Propose retrieval strategies to reduce the number of feedback iterations when the number of semantic classes is high
- Develop a CBIR system in VB.NET to evaluate the proposed method

- The proposed method is compared to SVM-based approaches

1.3 ORGANIZATION OF THE DISSERTATION

In Chapter 2, content-based image retrieval systems are introduced and previous works in the area of CBIR systems are reviewed. The developed methods are explained in Chapter 3 in detail and the results of experiments are presented in Chapter 4. Conclusion and future work are discussed in Chapter 5.

CHAPTER 2

RELATED WORK

In this section, content-based image retrieval systems are introduced and their characteristics are studied. Then, recent approaches and methodologies in the area of CBIR systems are discussed. Two methods, fuzzy clustering and latent semantic indexing, were the preliminary approaches in this dissertation work; therefore, these methods are presented and discussed in more detail.

2.1 CONTENT-BASED IMAGE RETRIEVAL SYSTEMS

Image retrieval approaches were designed based on the information retrieval techniques applied to the retrieval of text documents. In such approaches, a set of keywords are assigned to each image, and information retrieval methods are used to cluster and retrieve images. On the other hand, Content-Based Image Retrieval (CBIR) systems, introduced in the 1990's, apply the image processing techniques to extract visual features such as color, texture, and shape from images. In CBIR systems, images are represented by a vector of image features instead of a set of keyword.

In CBIR systems, it is well known that high-level user perceptions can not be captured by low-level image features [93, 108]. Therefore, region-based retrieval systems [107] were introduced that attempt to overcome the deficiencies of feature-based image retrieval by representing images at the object-level. A region-based retrieval system [13] applies image segmentation to decompose an image into regions, which correspond to objects if the decomposition is ideal.

Region-based retrieval systems segment images into regions, and retrieves images based on the similarity between regions. Relevance feedback [86, 93, 94, 106, 123] is another approach

to reduce the gap between high-level image concepts and low-level image features by involving the user's perception of images in the retrieval process. This approach gradually refines the original image query based on the feedbacks the user provides on the retrieved images in each iteration.

Another challenge in CBIR systems is multi-dimensional indexing [130]. The visual image features for CBIR systems are high-dimensional numerical data. It is difficult to manage these data with traditional database systems because these systems are designed for text data and low-dimensional numerical data. Therefore, many researchers have proposed architectures for indexing high-dimensional data in CBIR systems [31].

2.1.1 Visual Features

In a CBIR system, it is very challenging to find a set of features that can model the user's perception of images in the database. There has been significant image processing research to find specific image features to detect 'face' images in a database [61], or tumors [73] and X-ray images [1] in medical images. In general, low level image features are based on color, texture and shape because they are most understandable by the users and can be represented effectively by a computer.

Color is probably the most important feature that users can specify when they create image queries. In addition, proper color measures can be reliable even in the presence of changes in illumination, view angle, and scale. There are several methods applied in image retrieval using color features. The histogram intersection method [14] and its successors have performed well for large databases even with the changes of viewpoint. Usually, histograms are not computationally complex but they are sensitive to different lighting conditions. Improvements can be obtained by storing illumination-independent color features [14]. A color-constancy

algorithm creates the derivative of the logarithm of the original image before the histogram intersection. This way, the ratio of neighboring pixels' values stays constant even though illumination is changed. Moment-based color distribution features are proposed to be matched more robustly than color histograms [14]. Color sets [14] can be an efficient alternative to color histograms for representation of color information by applying a color indexing algorithm that uses the back-projection of binary color sets to extract color regions from images. This technique provides both an automated extraction of regions and representation of color content. It overcomes some of the problems with color histogram techniques such as high-dimensional feature vectors, spatial localizations, indexing and distance computation.

Typical texture measures used in image retrieval systems are coarseness, contrast and directionality. Coarseness measures the scale of the texture (pebbles versus boulders), contrast describes its vividness, and directionality describes whether it has a favored direction (like grass) or not (like a smooth object). Some papers use texture orientation in searching a database of vacation photos for likely "city/suburb" shots [57]. Good texture discrimination is not all needed in image retrieval but more important is the perceptual similarity of textures.

Most shape features used by CBIR systems are circularity, eccentricity, major axis orientation and algebraic moment [15]. Sometimes differences between objects of the same type are due to changes in viewing geometry or they are due to physical deformation. One object, for example, can be a stretched, bent, tapered or dented version of the other. To describe these deformations, therefore, it is reasonable to model the physics by which real objects deform, and then to use that information to guide the matching process. In general, most CBIR systems using shape-based similarity assume that objects are simple, for example they are composed of only one homogeneous part.

2.1.2 Review of Existing CBIR Systems

The most widely known image retrieval system is IBM's QBIC [42] (Query by Image Content) system. In QBIC, the user is allowed to specify certain characteristics of the image they want to find. The results are returned in descending order score of textual relevance to the query. Recent versions of QBIC contain simple automated region segmentation functionality. In other previous systems, color histograms have been used and proved to be helpful, although the use of such global features as a point of query has provided little information about how that color is distributed spatially about the image.

Simplicity [118] incorporates the properties of all the segmented regions so that information about an image can be fully used. To segment an image, the systems partitions the image into blocks and extracts a feature vector for each block. The k-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one region in the segmented image. Six features are used for segmentation. Three of them are color components (LUV color space), and the other three represent energy in high frequency bands of the wavelet transform. A significance credit is assigned to the regions to be used in distance function. The significant factor can be uniform (all regions are equally important), based on the area percentage, or location of the region.

Blobword [13] is another CBIR system that is based on segmenting the image into regions and querying the image database using features of those regions instead of basing the query on global properties. Blobworld recognizes images as collections of objects that are in a spatial relationship to one another. Using the Expectation-Maximization algorithm to estimate the parameters of this model, the resulting pixel-cluster memberships provide a segmentation of the image. Once the image is segmented, features of the different segments are produced, such as

color and texture. While querying, the user is allowed to access the segments directly to determine which features of the image are important to his/her query. When results are returned, the user also sees the Blobworld representation of the image, which is used to refine the user's query.

In VisualSeek [103], each image is decomposed into regions of equally dominant colors. For each region, feature properties and spatial properties are retained for the subsequent queries. A query consists of finding the images that contain the most similar arrangements of similar regions. The color region extraction uses the back-projection technique [103]. To start a query, the user sketches a number of regions, positions them on a grid, and selects a color for each region. To find the matches of a query image with a single region, queries on color set, region absolute location, area and spatial extent are first done independently. The results of these queries are intersected and from the obtained candidate set, the best matching images are taken by minimizing a total distance given by the weighted sum of the four distances mentioned. If the query image consists of a number of regions, in absolute or relative location, then for each region positioned in absolute location, a query like that described above is made, and for regions positioned by relative location individual queries on all attributes except location are performed. For the intersection of all this query results, the relative spatial relations specified by the user are evaluated using 2D string representation

MARS is the pioneer of CBIR systems in implementing relevance feedback techniques. Queries in MARS [86, 93] can be a combination of low-level features (color, texture, shape) and textual descriptions. There is a tree associated with each query. In a query tree, the leaves represent the feature vectors (the terms of the boolean expression defining the query) while the internal nodes correspond to boolean operators or more complex terms indicating a query by

object. The tree is evaluated bottom-up, each internal node receives from each child a list of ranked images and combines these lists according to the weights on the parent-child links. In MARS, color is represented by a 2D histogram over the HS coordinates of the HSV space [14] and the similarity distance between two color histograms is computed by histogram intersection. Texture is represented by two histograms, one measuring the coarseness and the other one the directionality of the image, and one scalar defining the contrast. In order to extract the color/texture layout, the image is divided into 5 × 5 sub images and for each sub image, features are extracted. The object in an image is segmented out in two phases. First, a k-means clustering method in the color-texture space is applied, then the detected regions are grouped by an attraction based method. A number of attractor regions are defined and each region is associated with the attractor that has the largest attraction to it. The attraction between two regions, i and j, is defined as $F_{ij}=M_i\ M_j\ /\ d_{ij}^2$, where M_i, M_j are the sizes of the two regions and d_{ij} is the Euclidean distance between the two regions in the spatial-color-texture space. The Euclidean distance between the vector representations is used to compute the texture similarity between two sub images. A weighted sum of the 5 × 5 color/texture similarities is used to compute the color/texture layout distance between two images. The shape of the boundary of the extracted object is represented by means of Fourier Descriptors. The similarity between two textures of the whole image is determined by a weighted sum of the Euclidean distance between contrasts and the histogram intersection distances of the other two components. The user can also choose a set of desired features from a list when querying the system.

2.1.3 Relevance Feedback

In relevance feedback, human and computer interact to convert high-level queries to models based on low-level features. Relevance feedback is a powerful technique used in

traditional text-based information retrieval systems. In some CBIR systems [13], users are asked to provide the system, as a part of the query, with some extra information such as the level of importance for each feature, or suggesting a set of features to be used in image retrieval. It seems to be an efficient way to help the user modeling his query; however, different users (or the same user at different instances) may have a different perception of the notion of similarity between image properties. Moreover, it may not even be feasible to express the information need of a user exactly as a weighted combination of features of a single query image.

These approaches fix the image similarities and query representation, which makes the system very rigid. To overcome the above mentioned difficulties, researchers proposed Query Refinement framework in [86] that utilizes feedback from users to support:

- Query Modification allows users to refine the query representation. A user may start from a query object that approximately capture his information need. In each iteration of feedback, the system modifies the representation of the query to a more suitable representation.
- Query weighting changes the relative weights of different features in the query representation. The re-weighting mechanism allows the system to learn the user's interpretation of similarity/distance function.

Query Modification can be achieved using either of two approaches: query expansion and query point movement. In the query point movement approach, a query is represented by a single point in a feature space and refinement process attempts to move that point toward the direction where relevant points were located. A query point movement approach has been presented in MARS [86] and MindReader [51]. On the other hand, query expansion does not assume that a query is represented as a point in a multidimensional space. Instead, it modifies the query by selectively adding new relevant objects to the query representation. Experimental evaluation in

MARS [93] shows that query expansion outperforms query point movement in retrieval effectiveness. Another advantage of query expansion over query point movement is that query expansion can be coupled with existing information systems without requiring any modification being made to them. Such a coupling may be desirable if data collections contain mixed media objects (e.g., web pages containing both text and images) and we wish to exploit existing text information retrieval system to support the text part of the content based query. On the other hand, integrating query point movement to a retrieval system will require a modification to its internal query representation which may not be allowed.

Relevance feedback mechanism is the process of automatically adjusting an existing query using the information fed back by the user about the previously retrieved objects such that the adjusted query is a better approximation to the users' information need. Under the assumption that low-level features can capture high-level concepts, the relevance feedback techniques try to establish the link between high-level concepts and low-level features from the user's feedback. Furthermore, the burden of specifying the feature weights is removed from the user. The user only needs to mark which images s/he thinks are relevance to the query.

The weights in the query object are dynamically updated to model the high-level concepts and perception subjectivity. In the most of image retrieval systems, relevance feedback data is in the form of positive examples, negative examples or both. Studies shows using only positive example lead to more improvement than only negative examples. However, best improvement in retrieval is obtained by using positive and negative examples together [63, 73].

2.2 SHORT-TERM AND LONG-TERM LEARNING

In relevance feedback-based approaches, a CBIR system learns from feedbacks provided by the user. Learning in CBIR systems is categorized as short-term learning and long-term

learning in the literature [43]. In short-term learning, only the feedbacks for the current search session are used in the learning algorithm, and image features are the primary source of data. The main challenge in this approach is to find the best combination of image features that presents the user's query. Such optimum set of features can include features that capture similarities between positive images, or features that discriminate positive examples from negative ones. Therefore, feature weighting, discriminant analysis, SVM, and instant learning methods are widely used in short-term learning. On the other hand, long-term learning approaches utilize the feedbacks collected during prior search transactions. Accumulated feedbacks are stored in a search history matrix. A search history matrix, denoted by $H_{N.M}$, stores the labels provided by the user for image x_i, i=1,…,M in transaction t_k, k=1,..,N. A transaction is the set of feedbacks collected form a user during relevance feedback iterations of a search session. It is assumed that the user does not change the query image she has in her mind during the relevance feedback iterations. Therefore, each transaction corresponds to a semantic and can be represented by labeled images in a L-dimensional space where L is the number of images labeled in the transaction. Table 2.1 shows a search history with four transactions and six images as X1 to X6.

Table 2.1 An example of a search history

t	X1	X2	X3	X4	X5	X6
Transaction 1	+	+	–	–	–	–
Transaction 2	+	+	–	–	–	–
Transaction 3	–	–	+	+	–	–
Transaction 4	–	–	–	–	+	+

The first step in a long-term learning approach is detecting the number of semantic classes, which is the number of concepts presented in a search history matrix, and creating the semantic space by defining each semantic class. Then, each image should be assigned to its corresponding semantic class. For example, the history matrix shown in Table 2.1 presents three

semantic classes. As usually the size of search history matrix is large, statistical models and approaches such as principal component analysis and latent semantic analysis are popular in long-term learning approaches.

2.3 FEATURE WEIGHTING

In CBIR systems, the distance between two images is computed as a weighted summary of their feature distances:

$$Dist(i_1, i_2) = \sum_{j=1}^{m} w_j . d(f_{i_1,j}, f_{i_2,j})$$

where, w_j is the weight for feature j and d is a distance function. Popular distance functions are Manhattan, Euclidean, and Cosine distances, discussed in [55].

Relevance feedback data are used to modify the query representation in order to capture user perception by updating feature weights. Updated feature weights modify the pair-wise image distances; therefore, level of similarity between the query and images in the database are changed for the next retrieval iteration. On the other hand, many researchers believe that assigning too many weights to the features may not help to build a reliable model. Therefore, Feature Selection approaches [34, 62, 80, 87, 116] have been proposed for image retrieval systems. Feature Selection can be considered as a special case of feature weighting, where the weights of a subset of features is one, and for the others is zero. Feature selection approaches are mainly based on dimension reduction techniques such as Singular Value Decomposition method (Section 2.6.3.2).

2.3.1 Variance-Based Methods

In variance-based feature weighting methods, positive examples are mapped to feature space and the variance of data along each feature is computed. The main idea is that features with less variance are more important because they have the ability to specify a feature value for

relevant images. For example, the first feature in Figure 2.1 is more important than the second one. As only positive examples are used in this method, it can be regarded as a one-class problem.

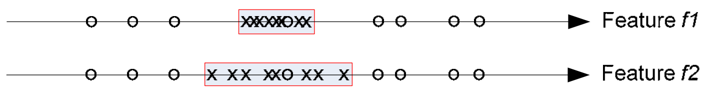


Figure 2.1 Comparison of two features with different variances

Therefore, after computing variances on the set of positive examples in each feature dimension, features with less variance have more weight. Computed weights are usually normalized to sum up to one.

$$w_j = \frac{1}{\sigma_j}$$

The above metric is computed based on only positive examples. There is another method [123], a two-class problem, which modifies the above equation to also use the data from negative examples. These methods penalize feature with misclassification. A misclassification is defined as having negative examples in a class of positive examples. The more misclassification, the smaller the weight is.

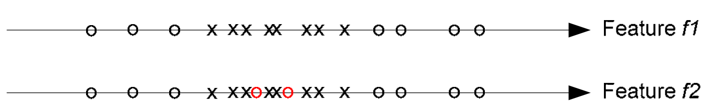


Figure 2.2 Comparison of two features with different number of misclassifications

For example, the first feature in Figure 2.2 is more important than the second one due to the lower number of misclassifications although they have equal variances. To combine variance and misclassification criteria, the following metric is proposed in [55]:

$$\delta_j = 1 - \frac{n(U \cap R)}{n(U)}$$

where U is the set of negative examples, R the range of positive examples, $U \cap R$ is the set of negative examples which fall in the range of positive examples, and n(U) is the cardinality of set U. The metric δ_j simply shows how much positive and negative examples are mixed. The weight of a feature is computed by:

$$w_j = \frac{\delta_j}{\sigma_j}$$

If $\delta_j = 1$, positive and negative examples can be separated along feature j and the two-class problem converts to a one-class problem.

2.3.2 Entropy

In information theory, entropy is defined as a theoretical lower bound on the number of bits necessary to encode information [71]. Information entropy is measured as:

$$Entropy(S) = -\sum_{i=1}^{L} p_i \log p_i$$

where p_i is the relative frequency of class *i* in S (a priori probability). Entropy has a value of zero when all the patterns belong only to one class, and has a value of one when all classes are in equal number. In [129], entropy is used to reduce the dimensions of color histograms in a CBIR system based on the idea that the entropy of an image measures the information content of the image. Therefore, image entropy is introduced as a visual feature that is computed based on image color histograms, and a entropy-based similarity function is formulated.

2.4 DISCRIMINANT ANALYSIS

The objective of discriminant analysis is to find the most discriminant features of data (x_i) in the original high-dimensional space, and map data points to a projected low-dimensional space

in a way that discriminant features are preserved. Linear discriminant analysis (LDA) is a popular method in CBIR area. LDA tries to find the transformation matrix *W* that maximizes the separation between different classes while minimizing within-class scatters in the new subspace. It can be mathematically formulated as:

$$S_b = \sum_{i=1}^{C} p_i (m_i - m_G)(m_i - m_G)^T$$

$$S_w = \sum_{i=1}^{C} \sum_{x_j \in Class_i} (x_j - m_i)(x_j - m_i)^T$$

$$W_{LDA} = \arg\max_{W} \frac{|W^T S_b W|}{|W^T S_w W|}$$

where S_b is called between-class scatter matrix, S_w is within-class scatter matrix, C is total number of classes, and P_i is the prior probability of class i which is sometimes simply the number of data points in class i. The mean of class i is represented by m_i, and m_G is the global average of all data points. The optimum W is obtained by solving the following generalized maximum eigenvalue problem:

$$S_b W = \lambda S_w W$$

There is an issue in computing LDA. To solve the above equation, the inverse of S_w should be obtained. However, when the rank of S_w is less than the number of dimensions, it is singular and has no inverse. In such situations, a common approach called Regularization is used to make S_w a full rank matrix by adding small quantities to its diagonal elements. Another approaches are projecting feature vectors into a subspace of only a few of its principal components (PCA) or applying a null-space.

When there are only two classes, the process is known as Fisher Discriminant Analysis (FDA). A significant problem with FDA is its assumption that negative examples are drawn from

the same distribution, which is not usually true in the case of image data. Another choice is Multiple Discriminant Analysis (MDA) that considers each negative example as a different class and creates a (N_N+1)-class discriminant analysis problem where N_N is the number of negative examples. Again, this assumption may not be true and some of negative examples do belong to the same distribution. Biased Discriminant Analysis (BDA) [131] keeps negative examples away from positive examples, and clusters only positive examples.

$$S_{NP} = \sum_{i \in Negative} (x_i - m_p)(m_i - m_p)^T$$
$$S_P = \sum_{i \in Positive} (x_i - m_p)(m_i - m_p)^T$$
$$W_{BDA} = \arg\max_W \frac{|W^T S_{NP} W|}{|W^T S_P W|}$$

It assumes that "all positive examples are alike; each negative example is negative in its own way" [131]. This means that all positive examples should be located closely in the same area in the feature space. However, semantically similar images may not be close to each other in the feature space, especially when their relations are defined based on high levels of semantic concepts. Discriminant analysis can be expressed as a combination of informative and discriminative learning with compactness and discrimination factors respectively. Compactness factor is related to minimizing within-class variations. BDA compacts only positive examples while LDA compacts both positive and negative points. Discrimination is maximizing between-classes variations, and can be done by keeping negative examples away from the mean of positive examples or vice versa. LDA applies both strategies by maximizing class means from the global average (m_G) as it assumes there is the same distribution for all data points. BDA keeps only negative examples away from positive ones by maximizing the total distances of negative examples from the mean of positive examples (m_P).

Empirical experiments with synthesized data showes that when the number of positive examples (N_P) is much higher than the number of negative examples (N_N), compacting negative examples, and discriminating negative examples from positive examples is the most efficient strategy. On the other hand, when $N_N >> N_P$, it would be better to compact positive examples and keep them away from the mean of negative points. The reason is when the number of positive examples is much higher, it would be a heavy burden to compact them or discriminate them from negative examples. It would be the same for negative examples when their number is much higher than positive points.

2.5 SUPPORT VECTOR MACHINES (SVM)

Support vector machines are a core machine learning technology. They have been successfully applied to tasks such as handwritten digit recognition [115], object recognition [77], and text classification [54]. In the area of image retrieval, SVMs have been used for feature weighting [16, 89, 91]. SVMs are basically used for binary classification. In the simplest form, SVMs are hyper-plains that separate the training data $\{x_1, \ldots, x_n\}$ in a data space by a maximal margin rule (see Figure 2.3).

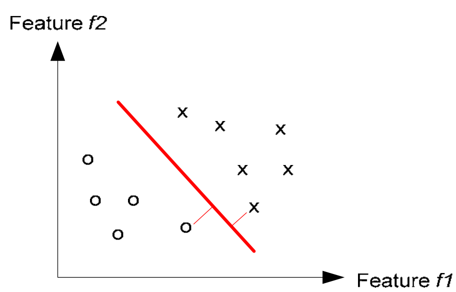


Figure 2.3 SVM classification: supports and margins

All vectors lying on one side of the hyper-plain are labeled as +1, and all vectors lying on the other side are labeled as -1. The training instances that lie closest to the hyper-plain on each side of it are called support vectors, and a margin is defined as the minimum distance of support

vectors from the hyper-plain. Therefore, the best hyper-plain is the one that maximizes the margins in the data space. SVMs project the original training data in the input space to a higher dimensional feature space via a kernel operator K. Data points (x_i) are presented as $\Phi(x_i)$ in feature space and define a set of classifiers as $D(x_i) = \mathbf{w}.\Phi(x_i) + w_0$ where $\mathbf{w}$ is the vector of dimension weights in the feature space. The classifier $D(x_i)$ classifies data point x_i as +1 or -1 according to the following relations:

$$\mathbf{w}.\Phi(x_i) + w_0 \geq +1; \quad \text{if } y_i = +1$$
$$\mathbf{w}.\Phi(x_i) + w_0 \leq -1; \quad \text{if } y_i = -1$$

As an example, consider an Exclusive-OR (XOR) operation on two data binary data points. As it is shown in Figure 2.4, there is no linear classifier in the input space to separate the two classes (1 and -1).

Table 2.2 SVM solution for Exclusive-OR problem

Sample	Input Space	Feature Space Z						Output
i	(x_1, x_2)	1	x_1	x_2	x_1x_2	x_1^2	x_2^2	Y
1	(1, 1)	1	1	1	1	1	1	1
2	(1, -1)	1	1	-1	-1	1	1	-1
3	(-1, -1)	1	-1	-1	1	1	1	1
4	(1, 1)	1	-1	1	-1	1	1	-1

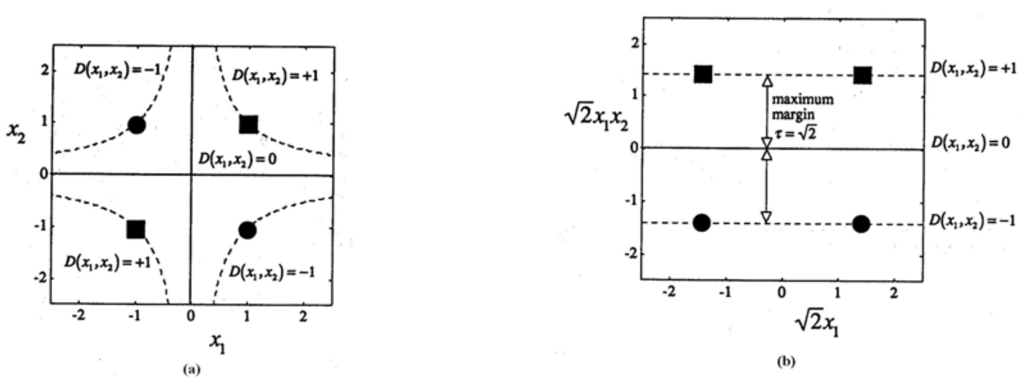


Figure 2.4 (a) Input space, and (b) Feature space for XOR problem [115]

Using a kernel function [22], a higher dimension feature space Z is created in which a hyper-plain can be found to classify data points with a weight vector **w** = (0, 0, 0, $1/\sqrt{2}$, 0, 0) as it is shown in Table 2.2 that x_1x_2 is equal to output Y. Kernel functions can also be used for non-linear classifiers [22].

In image retrieval by relevance feedback, SVMs can be applied to the image features space. Data points are images which are labeled as positive (+1) or negative (-1). The task of SVMs is to create a hyper-plain to separate all images in the database to two group of relevant (+1) and irrelevant (-1) images. During the relevance feedback process, an SVM is constructed in each dimension of the feature space and the generalization error is computed and features with smaller generalization error are assigned larger weights. Generalization error measures how good a classifier can classify training data. In another SVM method [89], weights are assigned to each types of feature rather than each dimension of the features so that only a few weights need to be estimated which may have less risk in relevance feedback problems with high dimensionality on the features and small size of training samples.

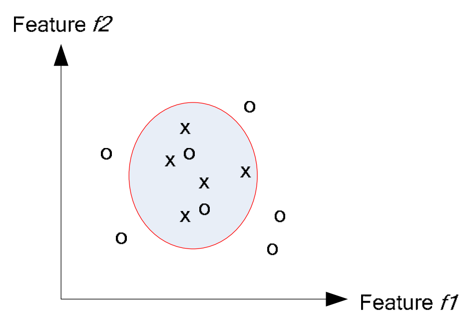


Figure 2.5 One-class SVM

In relevance feedback problems, positive examples can be assumed to belong to one class. However, negative examples are different from the query in many different ways and may not belong to one class. Therefore, a one-class SVM method [19] is proposed which tries to put

the positive examples to one class. Again, larger weights are assigned to features with small generalization errors.

Probabilistic models have also been applied in image retrieval by relevance feedback to find the probability that the user selects each image. Bayesian models [25, 106] are widely used to solve such probabilistic models. In a learning approach, a Discrimination version of Expectation-Maximization (D-EM) algorithm is proposed to use data from relevance feedback to cluster images [49, 122]. In image retrieval by relevance feedback, users label only a small ratio of images in the database and EM algorithm has been proven to be suitable for problems with small size of labeled data. The algorithm iterates in two steps until no specific improvement is achieved. In the first step, cluster centers are estimated based on the labeled data, and in the second step, unlabeled images are labeled using the cluster centers computed in the first step. The relevance feedback problem has been also studied as an optimization problem [50, 92]. It is a query point movement approach to construct a point as the new query for next relevance feedback iteration such that it minimizes the distances of currently labeled images from the current query. Lagrange multipliers are used to solve a minimization problem [92] in the form of:

$$\text{Min } J = \mathbf{s}.\mathbf{d}$$

$$\text{subject to: } \sum_{i=1}^{N} \frac{1}{u_i} = 1$$

where **s** is the set of scores for N labeled image by the user. The distance of the query from image *i* in feature space is $\mathbf{g_i}$, and $d = [d_1, \ldots, d_N]$ is the weighted distance ($d_i = \mathbf{u}^*\mathbf{g_i}$) of the query from all labeled images, where **u** is the vector of feature weights.

2.6 RECOMMENDATION SYSTEMS

The main idea behind the recommendation systems is that similar users are likely to have similar tastes. The task of a recommendation system is to measure the similarities between users

and suggests to them the "favorites" of users who are similar to them. Users are similar if they have same opinion about a set of items. User profiling techniques are used to gather data about users' opinions and tastes. Recommendation systems have been widely used in e-commerce. Recently, by the invention of mobile devices, recommendation systems have been also applied in personalizing web sites and adaptive user interfaces. The large e-commerce web sites offer million of products for sale. Choosing among so many options is challenging for the customers [97]. A recommendation system in an e-commerce web site receives information from a customer about which products s/he is interested in, and recommends products that are likely to fit his needs. Today, recommendation systems are deployed on hundreds of different sites serving millions of customers, such as Amazon.com, and eBay.com.

Personalized web sites [5, 6, 7, 66, 72, 82] have absorbed attention especially with the invention of mobile devices. Mobile devices require different web browsing technologies due to their bandwidth and screen limitations. Web site personalization provides personalized web sites in order to answer various needs of different users and mobile devices according to such limitations. A web site personalizer is an intermediary between the web site and the visitor and may be located on the web server, on the visitor's device, or at a proxy server in between.

Web site personalization can be divided into two categories: Personalizing Navigation [5, 7, 66, 82] and Personalizing Contents [6]. In navigation personalization, the goal is predicting the user's web page destination and shortening browsing time by skipping some intermediate pages, providing links to the pages which are probably the user's destination page, or in the ideal case, showing the destination page. Usually the performance measure in this case is the number of pages the user has to browse before reaching his destination page. In content personalization, the system tries to highlight the contents of the pages that are predicted to be the user's point of

interest and omit those parts in a page that the user may not be interested. Personalization models use the user's profile [5, 6], current user's visited pages [66, 82] and the structure of the web site [5] to predict user's interests.

Adaptive user interfaces can be defined as interfaces which automatically are customized for users. Currently, in many desktop applications the user can customize the toolbar, select which menus to be visible, or create macros for custom functionalities. Adaptive user interfaces trace the user's behavior to estimate his needs and automatically apply such changes [119].

2.6.1 Recommendation Systems Techniques

Various statistical and knowledge discovery techniques have been proposed and applied in recommendation systems. All the techniques are based on users' profiles to make the recommendations. Static profiling is the process of analyzing a user's static and predictable characteristics. Such information usually comes from users themselves, e.g. registration or survey forms. Through static profiling the system knows what kind of information the user is generally interested as soon as the user has supplied the information. There are several problems with static profiling. First, the profile is static, and is only valid for a certain period until the user changes his interests. Hence, a static profile degrades in quality over time. In addition, the input is based on the individual's interest, prone to users' subjectivity and may not accurately reflect an objective view that can infer the interests of other users with similar interests. Dynamic profiling is the process of analyzing a user's activities or actions to determine user's interests [84]. In e-commerce, dynamic profiles are created based on the users' prior purchases or product browsing. In web site personalization, the visited web sites [66], web site navigations [6], or favorite list [5] can be used for a user's dynamic profile. Similarly, in user interface personalization [119], the user behavior provides the main data for user profile. User behavior

can be studied by monitoring how often a control is used, what parameters a user usually enter, what views of an application a user often selects, etc. After preparing user profiles, a recommendation system applies information filtering methods [83, 95] to the data from profiles to find the similarities between the users. Two kinds of approaches for information filtering have been presented in the literature: Content-based Filtering and Collaborative Filtering.

2.6.1.1 Content-Based Filtering (Memory-Base): Content-based filtering is a memory-based profiling approach that compares the contents of items associated with a user profile and selects those documents whose contents best match the contents of another user profile using some similarity measures. For example, two persons who have specified that they are interested in pictures of Grand Canyon are assumed to be similar. The limitations of this method are:

- Content Limitation: Content-based methods can only be applied to a few kinds of content, such as text and image, and the extracted features can only capture certain aspects of the content.
- Over-Specialization: Content-based recommendation system provides recommendations based on user profiles. Therefore, users have no chance of exploring new items that are not similar to those items included in their profiles.

2.6.1.2 Collaborative Filtering (Model-Base): On the other hand, collaborative filtering is a model-based profiling approach that organizes users with similar interest into peer groups, thus items considered interesting by peers are recommended to other members of that group. As this approach relies heavily on user clusters, its effectiveness highly depends on how well the clustering of profiles correlates the users. CF-based recommendation systems suffer from [100]:

- Sparsity: Due to large number of items, the profile matrix is sparse as users are usually reluctant to rate the items. Therefore, the system may not have enough information about some

users and cannot provide recommendations for then, or the generated recommendations may not accurate.

- Scalability: Collaborative filtering methods use algorithms based on nearest-neighbors concept [110] to find similar users to a specific user. The time complexity of executing nearest-neighbor algorithms grows linearly with the number of items and the number of users, so the efficiency in recommendation systems with large-scale applications decreases.
- Synonymy: Since contents of the items are completely ignored, latent association between items is not considered for recommendations. Thus, as long as new items are not rated, they are not recommended; hence, false negatives are introduced.

Due to the various applications of collaborative filtering methods in recommendation systems, these methods are reviewed in more detail in the next section.

2.6.2 Collaborative Filtering Methods

The main purpose of collaborative filtering is finding the similarity of two users, clustering similar users to a group, and recommending the favorite items of one of the users to the other one [9]. Many machine learning methods such as Bayesian networks [11], clustering [47, 114], and rule-based [101] methods are used in collaborative filtering approaches in order to cluster similar users in a group.

2.6.2.1 Pearson Correlation Coefficient: The correlation between two users, *user a* and *user u*, is computed by the Pearson correlation coefficient:

$$P_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a) * (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a)^2 * \sum_{i=1}^{m}(r_{u,i} - \bar{r}_u)^2}}$$

where $r_{a,i}$ is the rating of item i given by user a. If item j is not rated by user a, the system estimates this rating ($p_{a,j}$) based on other users ratings for item j and their similarities to user a.

$$p_{a,j} = \bar{r}_a + \frac{\sum_{u=1}^{n} (r_{u,j} - \bar{r}_u) * p_{a,u}}{\sum_{u=1}^{n} p_{a,u}}$$

To come up with the synonymy problem in collaborative filtering, Content- Collaborative Filtering methods [20, 69, 80] use a content-based predictor to enhance users similarities, and then provide recommendations through collaborative filtering. Such methods modifies Pearson correlation coefficient by entering extra weighting factors related to the number of items co-rated by two users. Therefore, the similarity of two users who rated more items can be captured more accurately. In order to improve collaborative filtering methods in large database, RecTree [17] algorithm is proposed.

2.6.2.2 Singular Value Decomposition (SVD): The collaborative filtering matrix is usually very large and sparse. It has the items in the columns and users in the rows. In order to cluster the matrix and also reduce noise, dimension reduction methods can be applied to this matrix. Singular Value Decomposition technique (SVD) [22], a dimension reduction method, is very popular in the context of collaborative filtering. SVD method characterizes the correlational structure among large sets of objects is via Eigenfactor Analysis [22].

In SVD dimension reduction, the idea is that the important structure of high dimensional data lies along the axis of maximum variance. Thus, the covariance of the data is computed. Using this covariance matrix, the eigenvalues and eigenvectors are computed. To project the original data to a d dimensions matrix, the d largest eigenvalues are selected, and the corresponding eigenvectors provide the desired projection.

If P is an m × n matrix and n < m, then A can be written using so-called singular value decomposition of the form: $A = U\Sigma V^T$. Here, U is an m × n matrix and V is n × n square matrix, both of which have orthogonal columns so that $U^TU = V^TV = I$, and Σ is a diagonal matrix. Then, U is the system of eigenvectors of A and D has the square roots of the eigen values along its diagonal. As large eigen values correspond to dominant correlations, only k dimensions related to the k-largest eigen values can be selected.

2.6.2.3 Association Rules: Association rules discover the co-occurrence of two sub sets of items in transactions. As an example, "if user *u* buys item *x* and item *y*, then s/he also buys item *z*" is a typical rule in e-commerce domain. Apriori [39], DHP [79], and FP-Tree [41] are some of the well-known algorithms for finding association rules in databases. Here the basic concept of association rules is explained.

Let collection of items be denoted by I. A transaction $T \subseteq I$ is defined to be a subset of items that are put in the same class by the user. A class can be the set of purchased items. An association rule between two sets of items X and Y, such that $X, Y \subseteq I$ and $X \cap Y = \phi$, states that the presence of items in set X in transaction T indicates a strong likelihood that items from set Y also appear in T. This association rule can be denoted by X→Y.

The quality of association rules is commonly evaluated by looking at their **support** and **confidence** metrics. The support of a rule measures the occurrence frequency of the pattern in the rule. For a rule X→Y, support is the number of transactions containing X and Y divided by the total number of transactions. Confidence measures the strength of implication. For a rule X→Y, confidence is number of transactions containing both X and Y divided by the number of transactions containing X. A rule with high level of confidence provides an accurate prediction, as it shows that two items usually appear together if one of them appears. Low support in a rule

shows that the co-occurrence of two items, in general, is infrequent as support is computed over the dataset.

2.6.3 Application of Recommendation Systems in Image Retrieval

Recommendation systems can provide the user with a list of recommendations (i.e. products to purchase, or links to web pages) or automatically apply the recommendations (i.e. going to a web page or creating an adapted interface). In the domain of image retrieval, the recommendation systems can be used to recommend an image to the user based on the images the user selected so far and the similarities of the user with other users. While most relevance feedback approaches do not have a learning mechanism to memorize the feedbacks conducted previously to reuse them in future queries, recommendation systems analyze data in both current and old query sessions. In the following, recent learning approaches in image retrieval based on prior feedbacks are explained.

In a method based on hypergraphs [29], the relevance feedback data in image retrieval transactions are collected, a hypergraph is used to represent images correlationship and the semantic clusters are obtained by hypergraph partitioning. A hypergraph is an extension of a graph in the sense that each hyperedge can connect more than two vertices. So, each vertex represents an image, and positive examples in each transaction connect to each other and create a hyperedge. To perform partitioning, a multilevel hypergraph partitioning algorithm, HMETIS [10] is used. In the beginning, HMETIS partitions the hypergraph into two parts such that the weight of the hyperedges that are cut by the partitioning is minimized. Each of these two parts can be further bisected recursively, until each partition is highly connected.

Two metrics, fitness and connectivity, are used to measure the quality of the partitions. Fitness measures the ratio of edge weights that are within the partition and those involving any

vertex of this partition. Connectivity shows that the vertex has many edges connecting good proportion of the vertices in the partition. In another approach [132], collaborative filtering method is applied to relevance feedback logs to find the most similar feedback patterns in the past to the current feedbacks. This approach also uses the relevance feedback, so a relevance feedback log consists of a string of 0 and 1's. The similarity function used is based on the Edit Distance [113] metric, which is originally a string matching method. The following are some basic concepts about edit distance:

For any character strings A, B, the distance between A and B is defined as the minimum cost of transformation from A to B through some character insertions, deletions, and replacements necessary to make two strings equal. Let g be the cost function for edit operation. Then, for an operation sequence S:

$$g(\mathrm{S}) = \sum_{i=1}^{m} g(s_i)$$

The Edit Distance between string A and B, denoted by δ(A,B), is defined as the minimum cost of the operation sequence transforming A to B: δ(A,B) = min g(S). The proposed method in [132] is a memory-based method and in each retrieval process, the current relevance feedback log, in the form of a string of zeros and ones, is compared to all previous logs using the Edit Distance function.

The concepts of short term learning, based on the feedback during the current query session, and long term learning, based on feedback over many query sessions, are discussed in Section 2.2. A semantic space that includes the semantic classes is considered for a database. The semantic space is created based on the previous query results. For this purpose, a semantic matrix is used which has the images in the columns and semantics in the row. If an image belongs to a semantic class, the corresponding value in the semantic matrix is 1, and otherwise, it

is 0. Therefore, an image can belong to many semantic classes. In order to find the semantic matrix, a results matrix is used. Each row in results matrix has the relevance feedback values for one query session. In order to find number of rows in semantic matrix, it is assumed that the number of semantic classes is equal to number of total images. The Singular Value Decomposition (SVD) technique is applied to the semantic and results matrices to reduce the dimension of the semantic matrix. The model works based on positive examples, so irrelevant images and not reviewed images fall in to one group, and have the value of zero in the results matrix. In short term learning, a set of weights are assigned to semantic classes and in each iteration, the mistakes (when the classifier labels an image irrelevant but the user labels it relevant) are used to modify those weights and help the classifier to improve.

2.7 CLUSTERING

In this section, the concept of clustering and some data clustering approaches are explained. Some popular clustering algorithms are introduced and Fuzzy clustering is described in detail. The problem of missing values in data processing and related techniques in handling such problems are introduced and discussed.

The task of a clustering algorithm is to partition a data set into subgroups such that those in each particular group are more similar to each other (inter-similarities) than to those of other groups (intra-dissimilarities). A clustering algorithm can be agglomerative [4] or divisive [24]. An agglomerative approach begins with each data point as a cluster, and successively merges clusters together until a stopping criterion is satisfied. A divisive method begins with all data points in a single cluster and performs splitting until a stopping criterion is met. Stopping criteria can be defined by validation rules [123]. A validation rule measures some characteristics of created clusters such as compactness and separateness. Compactness measures how close are the

data point to each other in a cluster, and separateness measure how far clusters are located from each other.

In another aspect, a clustering algorithm can be Hard [3] or Fuzzy [8]. A hard clustering algorithm assigns each data point to a single cluster during its operation and in its output. A fuzzy clustering method assigns degrees of membership in several clusters to each data point. A fuzzy clustering can be converted to a hard clustering by assigning each pattern to the cluster with the largest measure of membership.

2.7.1 Clustering Algorithms

In the following, two main approaches in clustering, hierarchical and partitional clustering are introduced and the basic required steps for clustering a data set are explained.

2.7.1.1 Hierarchical Clustering: A hierarchical algorithm yields a nested grouping of data points, and similarity levels at which groupings change [59]. Most hierarchical clustering algorithms are variants of the single-link, complete-link, and minimum-variance [31] algorithms. These algorithms differ in the way they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of data points drawn from the two clusters (one pattern from the first cluster, the other from the second).

In the complete-link algorithm, the distance between two clusters is the maximum of all pair-wise distances between patterns in the two clusters. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria. The required steps in hierarchical agglomerative clustering algorithm are:

Step 1. Compute the proximity matrix containing the distance between each pair of data points. Treat each data point as a cluster.

Step 2. Find the most similar pair of clusters using the proximity matrix. Merge these two clusters into one cluster. Update the proximity matrix to reflect this merge operation.

Step 3. If maximum number of clusters is reached or the maximum value in proximity matrix is less than a threshold, stop. Otherwise, go to Step 2.

Based on the way the proximity matrix is updated in Step 2, a variety of agglomerative algorithms can be designed. Hierarchical divisive algorithms start with a single cluster that includes all given objects, and keep splitting the clusters based on some criterion to obtain a partition of singleton clusters.

2.7.1.2 Partitional Algorithms: A partitional clustering algorithm obtains a single partition of the data instead of a clustering structure. Partitional methods have advantages in applications involving large data sets for which the construction of a hierarchical structure is computationally prohibitive. A problem accompanying the use of a partitional algorithm is the choice of the number of desired output clusters. Thus, partitional techniques usually produce clusters by optimizing a criterion function. In practice, the algorithm is typically run multiple times with different starting states, and the best configuration obtained from all of the runs is used as the output clustering.

The most intuitive and frequently used criterion function in partitional clustering techniques is the squared error criterion, which tends to work well with isolated and compact clusters. The squared error for clustering a data set containing *k* clusters is:

$$e^2 = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (x_i^{(j)} - c_j)^2$$

where, $x_i^{(j)}$ is the i^{th} data point in j^{th} cluster and c_j is the centroid of cluster j with the size of n_j.

The k-means is the simplest and most commonly used algorithm employing a squared error criterion. It starts with a random initial partition and keeps reassigning data points to

clusters based on the similarity between the data point and the cluster centers until a convergence criterion is met. The k-means algorithm is popular because it is easy to implement, and its time complexity is O(n), where *n* is the number of data points. A major problem with this algorithm is that it is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen. Required steps in k-mean clustering algorithm are:

Step 1. Choose *k* cluster centers to coincide with *k* randomly-chosen data points or *k* randomly defined points inside the space containing the data set.

Step 2. Assign each data set to the closest cluster center.

Step 3. Recompute the cluster centers using the current cluster memberships.

Step 4. If a convergence criterion is not met, go to Step 2.

A typical convergence criteria can be no (or minimal) reassignment of data points to new cluster centers, or minimal decrease in squared error. A variation of k-mean algorithm [32] permits splitting and merging of the resulting clusters. Typically, a cluster is split when its variance is above a pre-specified threshold, and two clusters are merged when the distance between their centroids is below another pre-specified threshold [126]. Using this variant, it is possible to obtain the optimal partition starting from any arbitrary initial partition, provided proper threshold values are specified. The dynamic clustering algorithm [12] permits representations other than the centroid for each cluster, such as maximum-likelihood. The regularized Mahalanobis distance is used in [68] to obtain hyper-ellipsoidal clusters.

2.7.1.3 Nearest Neighbor Clustering: Nearest neighbor distances can be used in clustering procedures. In an iterative procedure [110], data points are assigned to the cluster of

its nearest labeled neighbor data point, provided the distance to that labeled neighbor is below a threshold. The process continues until all data points are assigned.

2.7.1.4 Artificial Neural Networks for Clustering: Artificial neural networks (ANNs) [68] are motivated by biological neural networks. ANNs have been used extensively for both classification and clustering. ANNs process numerical vectors and so require datasets to be represented by quantitative features only. ANNs learn a set of interconnection weights and act as feature selectors by appropriate selection of weights.

2.7.2 Fuzzy Clustering

Traditional clustering approaches generate partitions, and each data point belongs to one and only one cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each data point with every cluster using a membership function. Fuzzy clustering has been widely used in the area of information retrieval [70], and data mining [76]. In the following, a partitional fuzzy clustering algorithm is given.

Step 1. Select an initial fuzzy partition of the n objects into k clusters by selecting the $n \times k$ membership matrix U. An element u_{ij} of this matrix represents the membership of object x_i in cluster c_j.

Step 2. Using U, find the value of a fuzzy criterion function, e.g., a weighted squared error criterion function, associated with the corresponding partition. One possible fuzzy criterion function is:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij} (x_i - c_j)^2$$

$$\text{Subject to: } \sum_{j=1}^{k} u_{ij} = 1 \text{ ; for i = 1, …, n}$$

Reassign data points to clusters to reduce this criterion function value and recompute U. In order to minimize above function, cluster centers and membership matrix are computed by:

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}.x_i}{\sum_{i=1}^{n} u_{ij}}$$

$$u_{ij} = \frac{1}{\sum_{h=1}^{k} [\frac{x_i - c_j}{x_i - c_h}]^2}$$

Step 3. Repeat Step 2 until entries in U do not change significantly. In fuzzy clustering, each cluster is a fuzzy set of all the patterns. Larger membership values indicate higher confidence in the assignment of the pattern to the cluster. The most popular fuzzy clustering algorithm is the fuzzy *c*-means (FCM) algorithm. Even though it is better than the hard *k*-means algorithm at avoiding local minima, FCM can still converge to local minima of the squared error criterion [8, 70]. A fuzzy c-shell algorithm [27] can be used for detecting circular and elliptical boundaries.

A clustering algorithm is a probabilistic method when it is necessary that the total membership values of an instant in all clusters be equal to 1. In probabilistic clustering, the membership values can only show how a data point is related to the clusters. On the other hand, there is no guarantee that a data point with greater membership in a cluster is closer to the cluster center than another data point with smaller membership in that cluster. To solve this problem, possibilistic approaches [46] are introduced. Such approaches remove the condition regarding summarizing memberships to one. In possibilistic clustering, a set of weights (w_i) are defined for the clusters, and the objective function is:

$$\min P(U,V;w) = \sum_{i=1}^{c}\sum_{k=1}^{n} {u_{ik}}^m {D_{ik}}^2 + \sum_{i=1}^{c} w_i \sum_{k=1}^{n} (1-u_{ik})^m$$

In the above formula, the first part is similar to a probabilistic objective function, where D_{ik} is the distance of a data from center of the cluster. However, the second part acts as a penalty that tries to bring the sum of memberships to one. Similar to Fuzzy clustering method, required equations for computing U, V, and w of each iteration can be found by equalizing the second partial derivations of the above formula to zero.

2.7.3 Missing Values

Missing values can occur in attributes, instances, or both in a dataset; and each case needs its own technique. In processing data with missing values, it is assumed that missing values occur randomly [112]. However, in some cases it cannot be assumed some features are missing randomly. For example, in medical reports some attributes may be left blank because they are inappropriate for some class of illnesses. Another example is missing values due to intentionally unanswered questions, such as income, on questionnaires. Thus, missing values of this kind should be distinguished and treated differently from feature values that are missing randomly. In relevance feedback-based image retrieval problem, only a small ratio of the images are scored by the user and there are several missing values (images with no score) in a image query session.

If the number of missing values is small, instances with missing values can be discarded. Otherwise, there are methods dealing with missing values that can be categorized as pre-replacing and embedded methods.

Pre-replacing methods replace missing values before data processing. Statistics-based approaches such as linear regression [65], mean-mode methods [38], and Hot deck imputation [65] fill in missing values in data by estimated values based on the information available in the dataset. Machine learning approaches such as nearest neighbor estimation [88], neural network [88], Expectation-Maximization [88], and decision tree imputation [90] generate a classifier

based on the available data, classify instances with missing values, and replace the missing value with the centriod of the class that the instance belongs to.

Embedded methods deal with missing values during the data mining process. Case-wise deletion [67], lazy decision tree [33], and dynamic path generation [120] methods fall in this group. Cluster-based algorithms [35] can be used in both pre-clustering and embedded approaches. In cluster-based algorithms, the missing value of an instance is replaced by the center of the cluster the instant belongs to. A fuzzy clustering approach [64] can be used to estimate a missing value of an instance based on multiple clusters as the instance has different memberships in clusters. In a probabilistic approach [112], it is assumed that the data is d-dimensional with normal distribution and the missing attribute of an instance is replaced with one in the nearest neighbor of the instance.

2.7.3.1 Fuzzy Clustering with Missing Values: There are many fuzzy clustering methods to estimate missing values; however, clustering methods more of interest in CBIR systems than those design only for the purpose of estimating missing values. Methods based on Principal Component Analysis (PCA) are used to cluster data with missing values [48]. This method simultaneously applies the Fuzzy clustering method with the local principal components. Local principal components are extracted by using eigenvectors of the data matrix, and are used as the basis vectors of the cluster centers. In another method, Fuzzy clustering algorithm is modified to handle the problem of missing values [96]. In this method, a missing attribute in a pattern is substituted by its average over the complete patterns. However, there is no sense to replace a missing attribute in a pattern with its average over other patterns as long as the similarity of the pattern with missing attribute to other patterns is proven.

2.8 SUMMARY

During past years, researchers have tried to improve the performance of CBIR systems by introducing relevance feedback based approaches in order to capture users perception of the image. Recently, long-term learning was introduced to CBIR area based on the idea of recommendation systems to classify images based on multi-users perceptions instead of only one. However, there was no research found in the literature on studying feedback methodologies in CBIR systems, while it seems essential to study how users evaluate image relevancy to their queries, or transferring their perceptions to the retrieval system.

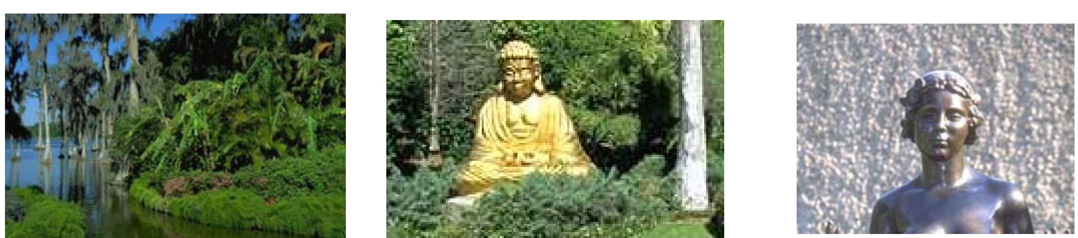
Most current approaches assume that an image has binary memberships in semantic classes, which is a noticeable limitation when images are semantically rich. Due to the high-dimension problem in long-term learning approaches, many dimension reduction methods are introduced; however, such methods are developed based on the binary membership assumption for images. This dissertation work focuses on developing retrieval methods that utilize an efficient feedback methodology to learn from different users, and applies a learning model that considers different memberships for each image; therefore, multi-concept queries can be modeled more accurately and semantically rich images, such as Figure 1.1, can be processed more effectively.

CHAPTER 3

METHODOLOGY

Most existing image retrieval methods assume that images have binary memberships in semantic classes. However, images may belong to many classes with different degrees of relevance – which may vary due to the user subjectivity. For example, in Figure 3.1, most users would consider images (a) as “forest” and (c) as “statue”. But, what about image (b)? It is related to forest and statue; however, not as strong as images (a) to forest or image (c) to statue.

Similarly, the majority of retrieval models are based on binary (hard) feedbacks. However, soft feedbacks - a score between 0 an 1 to show the degree of relevance of the image to the query - provide more flexibility for users, especially when the query or images are semantically rich. Furthermore, the experiments described in Section 4.2 demonstrate that soft labels reduce the variance of feedbacks provided by different users for a query, as well as feedbacks entered by the same user in different sessions. The method in this dissertation is based on the application of soft labels.



(a) (b) (c)

Figure 3.1 Degrees of relevance to semantic classes

An image retrieval system was developed that learns image semantics from search history (long-term learning) and image features (short-term learning). According to the current CBIR systems shortcomings discussed in Chapter 2, the characteristics of the developed system are as the following:

- Images belong to many semantic classes with different levels of relevancy.
- The retrieval process is based on learning image semantics.
- Learning is based on the relevance feedback data from current and prior users.
- Image features are used in computing image similarities.

In the preliminary steps of the retrieval system design, a missing value estimation mechanism, using fuzzy clustering framework, was developed to model multi-class images and solve the sparsity problem of the search history matrix. To improve the retrieval performance, probabilistic latent semantic analysis, a statistical based approach, is applied and a mixture model was developed to merge image feature and search history data sources. Preliminary approaches and related results are presented in Sections 3.1 and 3.2.

To design the retrieval system, a long-term learning algorithm was designed, which is presented in Section 3.3. The main challenge in short-term learning methods is the fact that semantically similar images may not be located close together in the image feature space. For example, images (b) and (c) in Figure 3.1 are semantically similar as both of them are related to the concept of “statue”. However, in the image feature space, they are not located close to each other as their colors features are quite different. Therefore, a discriminant projection is proposed for short-term learning algorithm in Section 3.4 to map disjoint clusters of relevant images in the feature space to close data points in a new subspace. Finally, it is shown in Section 3.5 how these two retrieval algorithms, short-term and long-term, jointly work together in a CBIR system.

In Section 3.6, retrieval strategies are introduced based on the concepts of “most positive” and “most ambiguous” images. Most Positive images are those with high similarity to the query concept. Most Ambiguous images are generally semantically rich and belong to many semantic classes. In the developed method, ambiguous images are used to summarize the

concepts embedded in the database by displaying multiple concepts with a single image; therefore, the required number of images for capturing user's query concept is reduced.

3.1 A FUZZY CLUSTERING METHOD FOR IMAGE RETRIEVAL

This section presents a summary of [101] that was developed as the first approach in this dissertation work. As mentioned before, one of the limitations in the current image retrieval systems is assigning images to different semantic classes with binary memberships. In [101], Fuzzy clustering is applied to the search history matrix to create a semantic space without the above limitation. In this work, each transaction can belong to one or many semantic classes with different memberships. For example, a transaction searching for a view of downtown can belong to outdoor, buildings, people or city semantic classes. Due to the sparsity problem in transactions matrix, missing values (unlabeled images in a transaction) are first filled and then the matrix is clustered to create the semantic space. Then, the score of images in transactions, and the memberships of transactions in semantic classes are used to determine the membership of images in the semantic classes.

3.1.1 Missing Values

In this method, missing values are filled with an estimated score. The estimated score of image i in transaction j is called $estScore_{ij}$ which uses the labeled data in transaction j to estimate the score of unlabeled data.

The set of scored images in transaction j is used for estimating the missing values in that transaction and is defined as $\mathbf{R_j}$ = {image p | $Scor_{pj}$ is entered by the user}. S_{ipj} is an estimated Score of unlabeled image i in transaction j according to its similarities to image p. The variable S_{ipj} is equal to $Similarity(i,p) * Score_{pj}$ if and only if $p \in \mathbf{R_j}$; otherwise, it is not defined. *Similarity (i, p)* is calculated by (Eq. 3.1), using the following normalized image features matrix:

$$Similarity(i,p) = 1 - \sum_{f \in F} w_f.(x_{i,f} - x_{p,f})^2 \qquad \text{(Eq. 3.1)}$$

where **F** is the set of image features, w_f is the feature weights and $x_{i,f}$ is the normalized feature f value of image i. After calculating the image similarities and S_{ipj}, (Eq. 3.2) is used to estimated the score for image i in transaction j:

$$estScore_{ij} = \frac{\sum_{p \in R_j} S_{ipj}}{|R_j|} \qquad \text{(Eq. 3.2)}$$

Now, the missing values are filled with their estimates. That is, for transaction j, the score of image i is $Score_{ij}$ if image i is scored by the user; otherwise, the estimated score is used instead:

$$Score_{ij} = \begin{cases} Score_{ij} & ; i \in R_j \\ estScore_{ij} & ; o.w. \end{cases} \qquad \text{(Eq. 3.3)}$$

As (Eq. 3.2) shows, the average of scores estimated by each labeled image (S_{ipj}) is used to estimate the missing score of an image. A small variance of S_{ipj} shows that the estimated scores are close based on different labeled images. Moreover, when there are more labeled images considered in the estimation, there is more confidence about the estimation. Therefore, the error of estimation is presented by:

$$error\_of\_estimation_{ij} = \frac{\sigma_{ij}}{\sqrt{|R_j|}} \qquad \text{(Eq. 3.4)}$$

where σ_{ij} is the standard deviation of S_{ipj} and $|R_j|$ is the cardinality of $\mathbf{R_j}$. The *error_of_estmation*$_{ij}$ measures the error of estimating the score of image i in transaction j. If there are equal scores for two images in a transaction, one entered by a user and the other one estimated by (Eq. 3.2), it is essential to find a way to differentiate these two values as they must have different affects on similarity measures. The error of estimation helps to do so.

Definition 1- Data space Φ is defined as one interval Fuzzy value with symmetric triangle membership. Data point p in the Φ space is shown as p = (α, β) where α is the center and β is the offset (Figure 3.2).

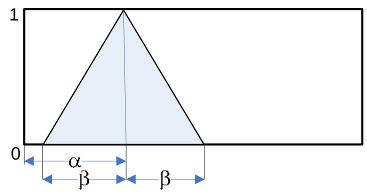


Figure 3.2 Data point p (α, β) in Φ space

In order to associate the error of estimation in the data point, the score values are transformed to Φ space where α is considered as the most probable value for the data point (e.g. score value) and β is the *error_of_estimation*. In the other word, β is a measure of uncertainty of data point and the bigger the β, the more uncertain data point is. The is no estimation for labeled data, and the error of estimation for such data is zero.

$$\alpha = Score_{ij} \qquad \text{(Eq. 3.5)}$$
$$\beta = error\_of\_estimation_{ij}$$

3.1.2 Creating Semantic Space

In order to create the semantic space, the transactions matrix is clustered and each cluster represents a semantic class. A semantic class includes a set of transactions that are related to a semantic concept. Each class has a set of features, which is the set of the images in the database, to be used in computing the distances between classes.

The distance between two classes is basically measured by the total differences of the cluster features (e.g. images). Because any one transaction can point to more than one semantic class, a Fuzzy clustering algorithm is used to assign different membership values to transactions.

After creating the semantic classes from the transaction matrix, all the images in the database are classified into these classes.

Fuzzy c-mean (FCM) is basically a Fuzzy version of the well known k-mean clustering algorithm (Section 2.7.2). FCM seeks to minimize the following objective function:

$$J(U,C) = \sum_{k=1}^{C} \sum_{j=1}^{m} u_{j,k}.d(T_j, c_k) \qquad \text{(Eq. 3.6)}$$

Subject to:

$$\sum_{k=1}^{C} u_{j,k} = 1 \ , \ \forall j \qquad \text{(Eq. 3.7)}$$

where $u_{j,k}$ is the membership of transaction j in cluster k and $d(T_j, C_k)$ is the distance of transaction j from the center of cluster C_k. FCM assumes that the number of clusters is known. There are algorithms that extend FCM to handle problems with adaptive numbers of clusters by applying methods such as cluster merging using compactness-separation validity measures [124, 125]. The Competitive Agglomeration algorithm (Section 2.7.1) has the advantage of both hierarchical and partitional clustering. In hierarchical clustering the number of clusters need not be known in advance. In partitional clustering, each cluster is represented by its center and the sum of distances of data from cluster centers is used as objective function to be minimized. The agglomeration process starts with each sample data as a cluster and ends with an optimal number of clusters.

In this work, the Competitive Agglomeration algorithm is modified to be able to cluster the transactions matrix with data points in the Φ space. A new distance function is introduced in the following section to be used in the CA algorithm. The outputs of the CA algorithm are the optimal number of classes in the semantic space and $m_{j,k}$, the transactions membership values in each class.

3.1.3 Distance Function

In this section, a distance function is introduced to measure the distance of a data point from a cluster center. Data points have an uncertainty of β and are defined in Φ space. However, cluster centers have no uncertainty and are real.

3.1.3.1 Minimum and Maximum Distances: Two types of distance functions are introduced to measure the distance of a data point from a cluster center. In general, the minimum and maximum distances between two sets are computed by:

$$D(A,B) = \min\{d(x,y) \mid x \in A, y \in B\} \qquad \text{(Eq. 3.8)}$$

$$D(A,B) = \max\{d(x,y) \mid x \in A, y \in B\} \qquad \text{(Eq. 3.9)}$$

An γ-cut of a fuzzy number A is an interval number A_γ that contains all the values of real numbers that have a membership grade in A greater than or equal to the specified value of α.

$$A_\gamma = [a_1, a_2] = \{x \in A \mid \mu_A(x) \geq \alpha\} \qquad \text{(Eq. 3.10)}$$

Thus, by taking an α-cut of a fuzzy number, one can process the operations on fuzzy numbers via the interval operations. It is interesting to note that the set of all γ-cuts of any triangular fuzzy number is a family of nested intervals. The level set of A is the set of all levels γ∈[0,1] that represent distinct γ-cuts of the given fuzzy number. Therefore, the minimum and maximum distances of a data point *P* in Φ space, where $P_\gamma = [p_1,p_2]$, from a cluster center *C* is computed by:

$$\| P_\gamma - C \|_1 = \begin{cases} 0 & , p_1 \leq C \leq p_2 \\ \min(|C - p_1|, |C - p_2|) & , o.w. \end{cases} \qquad \text{(Eq. 3.11)}$$

$$\| P_\gamma - C \|_1 = \max(|C - p_1|, |C - p_2|) \qquad \text{(Eq. 3.12)}$$

In order to find the distance of the data point from a cluster center, the integration of the distances for different values of γ is used. In the case of minimum distance, from (Eq. 3.11):

$$\int_0^1 \| P_\gamma - C \|_1 \, d\gamma = \begin{cases} \dfrac{\int_{x=\alpha-\beta}^{\alpha} (x-C)dx}{\beta} & ,C \le \alpha - \beta \\ \dfrac{\int_{x=C}^{\alpha} (x-C)dx}{\beta} & ,\alpha - \beta \le C \le \alpha + \beta \\ \dfrac{\int_{x=\alpha}^{\alpha+\beta} (C-x)dx}{\beta} & ,\alpha + \beta \le C \end{cases}$$

Therefore:

$$\int_0^1 \| P_\alpha - C \|_1 \, d\gamma = \begin{cases} (\alpha-\beta) - C + \dfrac{\beta}{2} & ,C \le \alpha - \beta \\ \dfrac{(\alpha - C)^2}{2\beta} & ,\alpha - \beta \le C \le \alpha + \beta \\ C - (\alpha+\beta) + \dfrac{\beta}{2} & ,\alpha + \beta \le C \end{cases} \qquad \text{(Eq. 3.13)}$$

In the case of maximum distance, from (Eq. 3.12):

$$\int_0^1 \| P_\gamma - C \|_1 \, d\gamma = \begin{cases} \dfrac{\int_{x=\alpha}^{\alpha+\beta} (x-C)dx}{\beta} & ,C \le \alpha \\ \dfrac{\int_{x=\alpha-\beta}^{\alpha} (C-x)dx}{\beta} & ,\alpha \le C \end{cases}$$

Therefore:

$$\int_0^1 \| P_\gamma - C \|_1 \, d\gamma = \begin{cases} \alpha - C + \dfrac{\beta}{2} & ,C \le \alpha \\ C - \alpha + \dfrac{\beta}{2} & ,\alpha \le C \end{cases} \qquad \text{(Eq. 3.14)}$$

Similar to FCM algorithm, the partial derivatives are found in order to compute the memberships and in each iteration. Setting $\frac{\partial}{\partial U} J(U,C) = 0$ gives:

$$u_{ij} = \frac{1}{\sum_{h=1}^{k} [\frac{D(P_i - C_j)}{D(P_i - C_h)}]^2}$$

which is the same as regular FCM algorithm. However, setting $\frac{\partial}{\partial C} J(U,C) = 0$ gives:

$$\sum_{k=1}^{n} u_{ik} \cdot \frac{\partial}{\partial C} D(P_k, C_i) = 0$$

which needs more work to be solved. In order to solve the above equation, first the derivative of the distance function are computed by (Eq. 3.13) for minimum and (Eq. 3.14) for maximum distance functions.

$$\frac{\partial}{\partial C} D(P_k, C_i) = \begin{cases} -1 & , C \le \alpha - \beta \\ \frac{C - \alpha}{\beta} & , \alpha - \beta \le C \le \alpha + \beta \\ 1 & , \alpha + \beta \le C \end{cases} \quad \text{(Eq. 3.15)}$$

$$\frac{\partial}{\partial C} D(P_k, C_i) = \begin{cases} -1 & , C \le \alpha \\ 1 & , \alpha \le C \end{cases} \quad \text{(Eq. 3.16)}$$

As the differentials are monotone increasing and change the sign over the defined interval, a simple linear search for C_i is possible to satisfy:

$$\sum_{k=1}^{n} u_{ik} \cdot \frac{\partial}{\partial C} D(P_k, C_i) = 0$$

There are numerical methods to find the root of a function. Bisection method is based on the fact that a function will change sign when it passes through zero. To improve the slow convergence of the bisection method, the secant method assumes that the function is approximately linear in the local region of interest and uses the zero-crossing of the line

connecting the limits of the interval as the new reference point. The Newton-Raphson method finds the slope (the tangent line) of the function at the current point and uses the zero of the tangent line as the next reference point. The process is repeated until the root is found. To find a root of *f(x)* = 0 with the initial guess x_0, the following iteration is used:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

(Eq. 3.15) and (Eq. 3.16) show piecewise linear differentials. Thus, the data points are ordered, and when $\frac{\partial}{\partial C} D(P_k, C_i) > 0$, cluster center is obtained.

Algorithm 3.1: Finding Clusters with minimum distance function

Inputs: Ordered data points (X_j)
Output: Cluster Centers (C_i)

```
M=0
N = -∑_{k=1}^{n} u_ik
While j ≤ 2n
    If Xj = α − β
        If β = 0 then
            N = N + uik
            If N > 0 then ci = Xj
        Else
            M = M + uik / βk
    If Xj = α + β
        If β = 0 then
            N = N + uik
            If N > 0 then ci = Xj
        Else
            M = M + uik / βk
        N = N + M(Xj+1 − Xj)
    j = j + 1
    If N > 0 then ci = Xj − N / M
End While
```

Figure 3.3. Find cluster centers with minimum distance

In the case of minimum distance function, $\alpha - \beta$ and $\alpha + \beta$ are the critical points where the function changes the slop. Therefore, all (α_k - β_k) and (α_k + β_k)'s are ordered as X_1, X_2, …, X_{2n}. Algorithm 3.1 finds the cluster center, where *N* shows the value of the function.

As the algorithm starts from the smallest value of X_j, (Eq. 3.15) shows that the differential has the value of (-1) and the value of the function is initialized to the negative of membership summation. The slope of the function is measured by $M = \frac{\partial^2}{\partial C^2} D(P_k, C_i)$ and initialized to zero.

```
Algorithm 3.2: Finding Clusters with maximum distance function
Inputs: Ordered data points (Xj)
Output: Cluster Centers (Ci)
1:    M=0
2:    N = -Σ_{k=1}^{n} u_ik
3:    While j ≤ n
4:          N = N + u_ik
5:          If N > 0 then ci = Xj
6:    End While
```

Figure 3.4. Find cluster centers with maximum distance

In the case of maximum distance, the only critical point is α. Therefore, all α_k's are ordered as $X_1, X_2, \ldots, X_n$. The following algorithm finds the cluster center.

3.1.4 Merging Clusters

In FCM algorithm, the number of clusters should be specified. However, the number of semantic classes is unknown in an image retrieval problem. Therefore, a large number is assigned as the maximum number of clusters, and the transactions matrix is clustered. Then, compactness and separateness measures are used to find "weak" clusters, omit them, and reduce the number of clusters. The minimum distance of the cluster center *k* from other cluster centers is computed to measure the separateness of a cluster:

$$SP_k = \min_{k' \neq k} \{D(C_k, C_{k'})\} \quad \text{(Eq. 3.17)}$$

Compactness of cluster *k* is measured by:

$$CP_k = \frac{\sum_{j=1}^{S} m_{jk}}{\sum_{j=1}^{S} m_{jk}.D(T_j, C_k)} \quad \text{(Eq. 3.18)}$$

where m_{jk} is the membership of transaction T_j in cluster C_k. A weak cluster has a compactness measure less than μ_{CP} - $3\sigma_{CP}$ or a separateness measure less than μ_{CP} - $3\sigma_{CP}$. The clustering algorithm iteratively runs until there are no weak clusters left.

3.1.5 Image Clustering

When the transactions are clustered, the semantic space is created and each cluster is regarded as a semantic class. In order to cluster the images, it is necessary to find the membership of images in each of the semantic classes. The memberships of transactions ($m_{j,k}$) are used to find $\mu_{i,k}$, the membership of image *i* in cluster *k* (semantic class *k*).

$$\mu_{i,k} = \sum_{j=1}^{S} m_{j,k} * Score_{i,j} \quad \text{(Eq. 3.19)}$$

Fuzzy c-mean clustering algorithm is classified as a probabilistic algorithm because the membership values of a data point sum to one, which is a weakness associated with probabilistic algorithms (see Section 2.7.2). However, in the proposed method, this problem does not exit any more after memberships of the images are calculated by (Eq. 3.19).

As the total score for an image in transactions matrix is not necessarily one, the total membership values for an image in semantic clusters may not be equal to one. As there are usually images with high relevancy to many semantic classes rather than only a few classes, the total membership value in the semantic space for such images is more than other images.

$$\sum_{k=1}^{C} \mu_{i,k} \neq 1, \forall i \quad \text{(Eq. 3.20)}$$

3.1.6 Adding a New Image to the Database

Adding a new item to the item list is considered as a significant problem in most recommendation systems as there is no search history for a newly added item, and it does not appear in other searches. In the proposed image retrieval system, when a new item is added to the database, its features are extracted. Using (Eq. 3.1), the similarity of the new image to all other images in the database is calculated. Based on these similarities, the membership of the new item to the semantic classes is estimated by:

$$\mu_{i,k} = \sum_{j \in I} m_{j,k} * Similarity(i, j) \qquad \text{(Eq. 3.21)}$$

Thus, the new image is counted in the retrieval process, and after some transactions there would be some feedbacks available for it so that the image can participate in the clustering to update its memberships.

3.1.7 Retrieval Process

This section explains how the semantic space and image clusters can be used for image retrieval, and what happens upon the arrival of a new query. For this purpose, the set of images labeled by the user in the current search session is defined as **L**, and in relevance feedback iteration *t*, the recently labeled (scored) images are added to the previous list:

$$L_{t+1} = L_t \cup \{i \mid i \in I, Score(t+1)_i \neq \phi\} \qquad \text{(Eq. 3.22)}$$

where $Score(t+1)_i$ is the score of image i in iteration t+1. An image query in the image retrieval system includes a set of images. When an image query is presented to the system, the distance of the query from each cluster of the images is calculated. This distance is used to find $\mu_{q,k}$, the membership of the query in each cluster:

$$D(q_t, k) = \frac{\sum_{i \in L_t} (Score_i - C_{k,i})^2}{|L_t|} \quad \text{(Eq. 3.23)}$$

$Score_i$ is the score of image i entered by the user in the current transaction (a transaction may include many feedback iterations) and $C_{k,i}$ is the center of cluster k along the dimension i. As both variables are real, the average of Manhattan distances is used to compute the distance of the current query from semantic classes. The membership of the query q_t in cluster k is calculated by:

$$\mu_{q_t,k} = \frac{1}{D(q_t, k)} \quad \text{(Eq. 3.24)}$$

Then, $\mu_{q,k}$'s are normalized to sum the memberships to 1. A set of unlabeled data is defined a as **U** = **I** – **L**, where **I** is the set of all images in the database and **L** is the set of images labeled in the current query session. Finally, the set of unlabeled images (**U**) is ranked by (Eq. 3.25), and the top-r images are returned to the user, where r is the size of the relevance feedback window (number of images displayed to the user in each iteration).

$$rank\_image_i = \frac{\sum_{k=1}^{C} |\mu_{q,k} - \mu_{i,k}|}{C} \quad \text{(Eq. 3.25)}$$

If the user is satisfied, the process ends; otherwise, the user gives labels (scores) to the returned images and sends back the query for the next relevance feedback iteration. When the transaction terminates (the user either is satisfied or aborts), the user is asked should the current transaction be added to transaction logs. The recently added search transactions are behaved as an individual cluster before the offline clustering. During the long-term learning, these transactions participate in the offline clustering and may modify the semantic space or cluster centers.

The proposed method is tested on a database of 1000 images, and 6 semantic classes. Data was collected based on single user search history. There was 30 transactions with 100 images randomly selected from the database. Collected data was divided to training and test sets. The algorithm successfully identified the number of semantic classes and clustered images to the semantic classes with an average clustering error of 5% for six semantic classes. However, the algorithm was not computationally efficient, and experiments showed that when the size of the search history is large, due to high number of images or transactions, the algorithm is unable to assign correct memberships to the transactions, and therefore, to the images. In the case of large search history size, all transactions are assigned to semantic classes with equal membership values. Therefore, it proved that instance-based methods, such as fuzzy clustering, are inefficient in learning semantic concepts in high dimensional spaces, which is also cited in [43].

On the other hand, statistical approaches study correlations in high volume data sets and seem to be more practical in analyzing search history data. In the next section, a probabilistic approach that is used in information retrieval [45] is applied to the field of image retrieval.

3.2 PROBABILISTIC LATENT SEMANTIC ANALYSIS

A second approach applied in this dissertation work that is demonstrated in [102]. In this section, a summary of [102] is presented, and it is explained how PLSA benefits a long-term learning approach in image retrieval problems to find the hidden semantic classes from the search history.

Latent Semantic Analysis (LSA) maps a data set $\mathbf{H}_{N.M}$ with N rows and M columns to a space of reduced dimensionality $\mathbf{H}'_{N.K}$, called latent semantic space. The mapping is computed by decomposing the search history matrix with SVD (section 2.6.2.2), $\mathbf{H}=\mathbf{U}\Sigma\mathbf{V}^t$, where $\mathbf{U}$ and $\mathbf{V}$

are orthogonal matrices, and the diagonal matrix Σ contains the singular values of $\mathbf{H}$. The LSA approximation of $\mathbf{H}$ is computed by selecting k largest singular values in Σ as $\mathbf{H}_{\mathrm{Nk}}=\mathbf{U}\Sigma_{(\mathrm{k})}\mathbf{V}^{\mathrm{t}}$.

Probabilistic Latent Semantic Analysis (PLAS) [45] provides a probabilistic structure for discovering the latent variables. The core of PLSA is a statistical model called aspect model. PLSA associates a set of hidden variables z_h with observations in the co-occurrence data. PLSA is explained in detailed with the application in image retrieval in the following section.

Now, consider Table 2.1, explained in Section 2.2, as a sample search history. In a probabilistic framework, the search history can be presented by a joint probability of $P(t,x)$. In fact, a paired observation corresponds to score of an image in a search session, which is provided by a user. PLSA associates a set of hidden variables z_h with observations in the co-occurrence data; therefore, each semantic class in the semantic space can be assigned a prior probability of $P(z_h) = \Sigma_t \Sigma_x P(z_h|t,x).P(t,x)$. The probabilistic latent factor model can be described as the following generative model:

- Select a session t_k with probability $P(t_k)$
- Pick a latent class z_h with probability $P(z_h|t_k)$
- Generate an image x_i with probability $P(x_i| z_h)$.

Therefore, the joint probability of a pair of observed data (t_k, x_i) can be obtained, while the latent factor z_h is discarded:

$$P(t_k, x_i) = P(t_k).P(x_i \mid t_k) \qquad \text{(Eq. 3.26)}$$

where,

$$P(x_i \mid t_k) = \sum_{h=1}^{H} P(x_i \mid z_h).P(z_h \mid t_k) \qquad \text{(Eq. 3.27)}$$

This formulation helps to model the assumption of multi-class images, where z_h is a hidden semantic and an image belongs to different semantic classes with probability of $P(x_i| z_h)$, and each transaction can be related to many semantic classes with probability of $P(t_k| z_h)$. Applying Bayes rule, it is straightforward to obtain the joint probability of a paired observation by:

$$P(t_k, x_i) = \sum_{h=1}^{H} P(z_h).P(t_k \mid z_h).P(x_i \mid z_h) \quad \text{(Eq. 3.28)}$$

$P(z_h)$, $P(t_k| z_h)$ and $P(x_i| z_h)$ need to be estimated to find the joint probability of a pair of observations in (Eq. 3.28). This can be achieved by maximizing the following log-likelihood function:

$$L = \sum_k \sum_i s_{ki}.\log(P(t_k, x_i) \quad \text{(Eq. 3.29)}$$

where s_{ki} is the score provided by the user for image x_i in transaction t_k. The standard procedure for maximum likelihood estimation is the Expectation Maximization (EM) algorithm. EM alternates two steps: Expectation (E) step computes the posterior probabilities for the latent variable z_h, and maximization (M) step updates parameters based on the probabilities computed in E-step. EM algorithm starts with some initial values for $P(z_h)$, $P(t_k| z_h)$ and $P(x_i| z_h)$. In E-step:

$$P(z_h \mid t_k, x_i) = \frac{P(z_h).P(t_k \mid z_h).P(x_i \mid z_h)}{\sum_{h'=1}^{H} P(z_{h'}).P(t_k \mid z_{h'}).P(x_i \mid z_{h'})} \quad \text{(Eq. 3.30)}$$

which is the probability that an image x_i in transaction t_k is explained by the factor corresponding to z_h. By applying Lagrange multipliers on

$$\begin{aligned} \sum_k \sum_h P(t_k \mid z_h) - 1 = 0 \\ \sum_i \sum_h P(x_i \mid z_h) - 1 = 0 \end{aligned} \quad \text{(Eq. 3.31)}$$

constraints (see [45] for more details), one can obtain the M-step equations as:

$$P(z_h) = \frac{\sum_{i,k} s_{ki}.P(z_h \mid t_k, x_i)}{\sum_{i,k} s_{ki}}$$

$$P(x_i \mid z_h) = \frac{\sum_{k} s_{ki}.P(z_h \mid t_k, x_i)}{\sum_{k,i'} s_{ki'}.P(t_k, x_{i'})} \qquad \text{(Eq. 3.32)}$$

$$P(t_k \mid z_h) = \frac{\sum_{i} s_{ki}.P(z_h \mid t_k, x_i)}{\sum_{k',i} s_{k'i}.P(t_{k'}, x_i)}$$

Iterating the above computations for expectation and maximization steps approaches a local maximum of the log-likelihood in (Eq. 3.29).

3.2.1 Integrating Image Features

As it is shown above, PLSA associates a set of hidden variables with observations in the co-occurrence data. Therefore, PLSA can also be applied to image features matrix, ***F,*** to find the joint probability of a paired observation in this co-occurrence data. Similar to (Eq. 3.28):

$$P(f_j, x_i) = \sum_{h=1}^{H} P(z_h).P(f_j \mid z_h).P(x_i \mid z_h) \qquad \text{(Eq. 3.33)}$$

Based on the shared component $P(x_i| \, z_h)$, (Eq. 2.28) and (Eq. 2.33) are combined by modifying the log- likelihood function as:

$$L = \alpha \sum_k \sum_i s_{ki}.\log(P(t_k, x_i)) + (1-\alpha) \sum_i \sum_j f_{ij}.\log(P(x_i, f_j)) \qquad \text{(Eq. 3.34)}$$

where α adjust the relative weight of two observations in search history data and image features data. Similar to (Eq. 2.30), P($z_h|f_j,x_i$) can be computed, and applying the constraint

$$\sum_j \sum_h P(f_j \mid z_h) - 1 = 0 \qquad \text{(Eq. 3.35)}$$

in M-step computations, there is the following modification:

$$P(z_h) = \alpha \frac{\sum_{i,k} s_{ki}.P(z_h \mid t_k, x_i)}{\sum_{i,k} s_{ki}} + (1-\alpha) \frac{\sum_{i,j} f_{ij}.P(z_h \mid x_i, f_j)}{\sum_{i,j} f_{ij}} \qquad \text{(Eq. 3.36)}$$

and this additional probability to be used in (Eq. 3.32):

$$P(f_j \mid z_h) = \frac{\sum_i f_{ij}.P(z_h \mid f_j, x_i)}{\sum_{j',i} f_{ij}.P(f_{j'}, x_i)} \quad \text{(Eq. 3.37)}$$

PLSA offers an efficient probabilistic structure, which can be used to discover detailed relations between transactions, images, and semantic classes. For example, a hidden semantic class, z_h, can be represented by a set of images. An image can be an ideal symbol for z_h if it has a strong relation to z_h, and z_h has a strong to that image. The second condition, excludes common images in the database to be the symbol of a class. Therefore, a set of images, x_{syb}, is found that satisfies:

$$P(x_{syb} \mid z_h).P(z_h \mid x_{syb}) \geq \mu_x \quad \text{(Eq. 3.38a)}$$

where μ_x is a predefined threshold. Therefore, common images in the database can identified as non-symbol images with high probability of $P(x_i| z_h)$ in a semantic class. Similarly, image features that efficiently represent a semantic space can be identified by:

$$P(f_{syb} \mid z_h).P(z_h \mid f_{syb}) \geq \mu_f \quad \text{(Eq. 3.38b)}$$

where μ_f is a predefined threshold for detecting image features with a discrimination power between semantic classes.

3.2.2 Mixture Decomposition

To illustrate the relation of PLSA to LSA, PLSA model can be presented in matrix notation. Defining matrices, $\mathbf{U}=P(t_k|z_h)$, $\mathbf{\Sigma}=\text{diag}(P(z_h))$ and $\mathbf{V}=P(x_i|z_h)$, the joint probability of the model is obtained by $\mathbf{P}=\mathbf{U\Sigma V}^t$. Similar to LSA, semantic factors with highest $P(z_h)$ can be used to estimate the joint probabilities in a reduced dimension space. Despite these similarities, there is a fundamental difference in the function used for determining the optimal decomposition between LSA and PLSA. LSA applies a L2-norm on an implicit additive Gaussian assumption.

PLSA relies on the likelihood function of the multidimensional sampling and aim at an explicit maximization of the predictive power.

3.2.3 Image Retrieval Based on PLSA

In image retrieval, the knowledge obtained from PLSA during image searching and feedback iterations is used. The image query is usually initiated by an example, or by labeling a set of images randomly returned by the system. In the case of query example, image features are extracted and $P(z_h|f_j)$ is used to find the relation between the current search session, t_c, and semantic classes in the database. Otherwise, a set of image symbols, x_{syb}, are displayed to the user to be scored.

As the relevance feedback continues, t_c accumulates all scores provided by the user for the images returned by the system. In each iteration, first, the semantic class that the current user is looking for is determined by $P(z_h|t_c)$, and second, images that are related to desired semantic classes are returned to the user. This task can be achieved through the following EM steps:

$$P(z_h \mid t_c, x_i) = \frac{P(x_i \mid z_h).P(z_h \mid t_c)}{\sum_{h'=1}^{H} P(x_i \mid z_{h'}).P(z_{h'} \mid t_c)} \qquad \text{(Eq. 3.39a)}$$

$$P(z_h \mid t_c) = \frac{\sum_{i \in L} s_{ci} P(z_h \mid t_c, x_i)}{\sum_{h'} \sum_{i \in L} s_{ci} P(z_{h'} \mid t_c, x_i)} \qquad \text{(Eq. 3.39b)}$$

After obtaining $P(z_h|t_c)$, the joint probability of current transaction and image x_i is:

$$P(t_c, x_i) = \sum_{h=1}^{H} P(x_i \mid z_h).P(z_h \mid t_c) \qquad \text{(Eq. 3.40)}$$

Images with the highest $P(t_c,x_i)$ are returned to the user, and the feedback iterations continue.

PLSA experiments with the dataset used in fuzzy clustering approach (D1000) demonstrated similar results as FCM; however, experiments with a data set of 2000 images

(D2000) and 6 semantic classes showed a significant improvement over FCM (Table 3.1). Experiments with different number of transactions showed that the system detects different set of image symbols, which are considered as cluster centers, when the distribution of transactions or images change in the search history matrix.

Table 3.1 FCM and PLSA clustering errors

	People		Vehicle		Building		Statue		Boat		City	
	FCM	**PLSA**	**FCM**	**PLSA**	**FCM**	**PLSA**	**FCM**	**PLSA**	**FCM**	**PLSA**	**FCM**	**PLSA**
D1000	6%	6%	7%	6%	4%	5%	5%	4%	4%	4%	5%	5%
D2000	28%	15%	59%	27%	31%	14%	34%	20%	38%	19%	66%	26%

The main advantage of PLSA is the possibility of obtaining the statistical structure of search history and image features datasets and finding the relations between images. The disadvantage of the proposed PLSA method is the cost of computation, mainly due to EM calculations. Therefore, a critical parameter in the proposed PLSA approach is the stopping criteria in EM iterations. EM algorithm is applied to each transaction in the search history, and each image in the image features data. An optimum selection of stopping criteria is needed for EM to reduce the computation time, while achieving high accuracy.

The comparison of FCM and PLSA results demonstrated that statistical based approaches are promising in learning image semantics from search history. In PLSA the number is semantic classes should be given, but the presented FCM approach detects the number of classes. However, this feature of FCM proved to be ineffective when the size of search history matrix is large. Experiments with PLSA showed that the classes in the semantic space created by the algorithm may not necessarily match precise concepts recognized by the user; however, the generated semantic space is used as a transformation to map users perceptions of images to the image categorization in the system.

3.3 LONG-TERM LEARNING METHOD

It is well known that the performance of Content-Based Image Retrieval (CBIR) systems is primarily limited by the gap between low-level features and high-level semantic concepts. In order to reduce this gap, long-term learning strategies have attracted attention as a means to improve the retrieval process [44] after the introduction of the relevance feedback to the CBIR systems. The basic idea is that when a set of images are highly scored in many search transactions, those images may be related to each other in some way corresponding to a “hidden” semantic concept [43]. Statistical approaches are promising in detecting the relationships between images and defining semantic space corresponding a search history.

In the method, the recommendation systems concept is merged with content-based image retrieval methods. Two main modeling approaches in recommendation systems are memory-based and model-based (Section 2.6). In memory-based approaches, a new query is simply compared to all available transactions in the history, and highly scored items in the most similar transactions are recommended to the user.

On the other hand, in model-based approaches, the search history is used to build a model during an off-line process. Afterward, only the model is used for query processing. The benefits of model-based approaches are reduced on-line processing time and higher accuracy. The experiment results achieved in Sections 2.6 and 2.7 also suggested that statistical modeling, the dominant method in model-based approaches, is preferable to instance-learners such as clustering, the primary method for memory-based approaches. All queries sent to the image database and their corresponding relevance feedback scores are stored in a search history matrix H. Each row of the search history matrix is a transaction, storing the user scores for each image during relevance feedback iterations. A semantic space, by definition, shows the relations

between images and semantic classes. To find a semantic space, $Z_{r.n}$, based on a given search history, H is required to be decomposed as:

$$H_{t.n} = G_{t.r} \cdot Z_{r.n} \qquad \text{(Eq. 3.41)}$$

where, t is the number of transactions, n is the number of images in the search history, and r is the number of semantic classes, which is required to be estimated by the system. The proposed long-term learning algorithm applies a Factor Analysis model to a given search history matrix H to find G and Z. In the following section, the basics of Factor Analysis are explained.

Suppose that t continuous variables $x_1, \ldots, x_t$ have been observed on each of n sample individuals, and a model is needed to explain the resulting association among these t variables by means of r latent variables $z_1, \ldots, z_r$. The assumption is that x_i's are conditionally uncorrelated, given the values of all z_j. Therefore, the factor analysis model is defined as:

$$\begin{gathered}
x_1 = \mu_1 + \gamma_{11}z_1 + \gamma_{12}z_2 + \ldots + \gamma_{1r}z_r + e_1 \\
x_2 = \mu_2 + \gamma_{21}z_1 + \gamma_{22}z_2 + \ldots + \gamma_{2r}z_r + e_2 \\
\vdots \\
x_t = \mu_t + \gamma_{t1}z_1 + \gamma_{t1}z_2 + \ldots + \gamma_{tr}z_r + e_t
\end{gathered}$$

where, μ_i and γ_{ij} are constants, while z_j and e_i are random variables (i=1, …, t; j=1, …, r). The minimal set of assumptions about these random variables to ensure uncorrelated variables assumption is that e_i are uncorrelated with each other and with z_i. It can be shown that the above model is convertible to the following matrix format [60]:

$$X = \mu + \Gamma Z + E \qquad \text{(Eq. 3.42)}$$

where, is Γ is a ($t.r$) matrix of γ_{ij} constants, Z is a set of random vectors with mean zero and dispersion matrix I. Error matrix E is a set of random vectors with mean zero and dispersion

matrix Φ=diag(Φ^2_1, ..., Φ^2_t), and μ is a set of constants, representing the mean of the vectors in X. In Equation (2), μ can be omitted if X is mean-centered.

Above model is similar to the multivariate regression model; however, the main distinguishable characteristic is that Z is also unknown in the Factor Analysis model. It is possible to estimate parameters in Factor Analysis method by embedding the multivariate regression model with an Expectation-Maximization (EM) iterative scheme; however, EM iterations, as it was seen in Section 2.10, would be extremely computationally extensive. Therefore, another option is selected based on Principal Component Analysis.

Assume Principal Component Analysis (PCA) is used to create a new set of variables y_i based on the linear combination of the observed data x_i (mean-centered) in such a way as to maximize successively the variance of y_i. If λ_i is the ith largest eigenvalue of the dispersion matrix of x'=(x_1,…,x_t), and α'_i=(α_{i1},…, α_{it}) is the corresponding eigenvector, then the principal components are given by:

$$\begin{aligned} y_1 &= \alpha_{11}x_1 + \ldots + \alpha_{1t}x_t \\ y_2 &= \alpha_{21}x_1 + \ldots + \alpha_{1t}x_t \\ &\vdots \\ y_t &= \alpha_{t1}x_1 + \ldots + \alpha_{tt}x_t \end{aligned}$$

and var(y_i) = λ_i (i=1,…t). Now, since the matrix (α_{ij}) is orthogonal, the above transformation can be inverted to:

$$\begin{aligned} x_1 &= \alpha_{11}y_1 + \alpha_{21}y_2 + \ldots + \alpha_{t1}y_t \\ x_2 &= \alpha_{12}y_1 + \alpha_{22}y_2 + \ldots + \alpha_{t2}y_t \\ &\vdots \\ x_t &= \alpha_{1t}y_1 + \alpha_{2t}y_2 + \ldots + \alpha_{tt}y_t \end{aligned}$$

Consequently, if the (*t*-*r*) components with smallest variance are treated as noise and set equal to η_i for i[th] components:

$$x_1 = \alpha_{11}\sqrt{\lambda_1}\frac{y_1}{\sqrt{\lambda_1}} + \alpha_{21}\sqrt{\lambda_2}\frac{y_2}{\sqrt{\lambda_2}} + \ldots + \alpha_{r1}\sqrt{\lambda_r}\frac{y_r}{\sqrt{\lambda_r}} + \eta_1$$
$$x_2 = \alpha_{12}\sqrt{\lambda_1}\frac{y_1}{\sqrt{\lambda_1}} + \alpha_{22}\sqrt{\lambda_2}\frac{y_2}{\sqrt{\lambda_2}} + \ldots + \alpha_{r2}\sqrt{\lambda_r}\frac{y_r}{\sqrt{\lambda_r}} + \eta_2$$
$$\vdots$$
$$x_t = \alpha_{1t}\sqrt{\lambda_1}\frac{y_1}{\sqrt{\lambda_1}} + \alpha_{2t}\sqrt{\lambda_2}\frac{y_2}{\sqrt{\lambda_2}} + \ldots + \alpha_{rt}\sqrt{\lambda_r}\frac{y_r}{\sqrt{\lambda_r}} + \eta_r$$

Replacing $\gamma_{ij} = \alpha_{ji}\sqrt{\lambda_j}$, and $z_i = \frac{y_i}{\sqrt{\lambda_i}}$:

$$\begin{aligned} x_1 &= \gamma_{11}z_1 + \gamma_{12}z_2 + \ldots + \gamma_{1r}z_r + \eta_1 \\ x_2 &= \gamma_{21}z_1 + \gamma_{22}z_2 + \ldots + \gamma_{2r}z_r + \eta_2 \\ &\vdots \\ x_t &= \gamma_{t1}z_1 + \gamma_{t2}z_2 + \ldots + \gamma_{tr}z_r + \eta_t \end{aligned} \qquad \text{(Eq. 3.43)}$$

which is equal to (Eq. 3.42). In Factor Analysis, the loading matrix Γ is usually rotated to obtain a new matrix that assigns only few high loads to each variable, keeping the other loadings small. The varimax algorithm [60] is an orthogonal rotation method that maximizes the variance of the squared loadings in each column of the loading matrix, so that each variable presents high loading for fewer factors. Rotation may reveal hidden patterns and favor data interpretation. After linear coefficients in Γ are determined and the matrix is rotated, factor scores for each data point in the sample is computed to transfer the data to a lower dimension [60]. The comparison of (Eq. 3.41) and (Eq. 3.42) suggests that observed variables x_i to be considered as transactions in the history matrix, sample data of size *n* as images, and latent variables z_i as semantic classes. Therefore, the following steps decompose the search history matrix H to G and Z matrices:

Step 1. Make the search history matrix H mean centered.

Step 2. Find eigenvalues (λ_i) and eigenvectors (α_{ij}) of H.

Step 3. Project H to the new PCA space to find y_i.

Step 4. Use the following formula to find the elements of matrix G:

$$\gamma_{ij} = \alpha_{ji}\sqrt{\lambda_j}$$

Step 5. Rotate the loading matrix G using varimax, and compute the factor scores Z.

3.3.1 Image Retrieval with Long-term Learning Algorithm

In an offline process, the semantic space Z is computed based on the available search history H. During the image retrieval process, the related semantic classes to the query are detected based on the feedbacks provided by the user and the relations between the labeled images and semantic classes in Z. Then, unlabeled images with associations to the highly related classes are returned to the user. Algorithm 3.3 summarizes the process.

```
Algorithm 3.3: Long-term Retrieval
Inputs: Semantic space (Z), Feedbacks in current session (S)
Output: Set of similar images (R)
   % find the relation of the query to the semantic space
for each i in the Labeled image set
      Q = Q + Z(i).s(i)
end for

   %find related images in the highly correlated classes
for each r in semantic classes
      M = M + Q(r).Z
end for

   %return highly related images
R = Top-n (M)
```

Figure 3.5 Long-term learning algorithm

In the above formulation, the assumption of multi-class images with different degrees of relevance is satisfied by creating matrix Z that shows the correlation of each image to each semantic class. In addition, multi-class queries are supported by computing the relation of the query to each semantic class. This flexible model lets the user to define complex queries defined

across multiple classes, and improves the retrieval of semantically rich images that are related to multiple semantic classes.

3.4 SHORT-TERM LEARNING METHOD

As it is shown in Figure 3.1, semantically similar images may not be located closely in the image features space, and create separated clusters. In this section, a discriminant projection is introduced to map disjoint clusters of relevant images in the feature space to close data points in a new subspace.

Due to the advantage of the soft labels in concept learning (see Section 4.2), the method was developed for relevance feedback with soft labels. The algorithm is an extension of the idea of Biased Discriminant Analysis (BDA), studied in Section 2.4. BDA assumes that “all positive examples are alike; each negative example is negative in its own way” [131]. The proposed method has the same assumption regarding irrelevant images; however, assumes that relevant images may also be located in different subclasses in the feature space.

A main drawback of applying linear discriminant methods, including BDA, to image retrieval problems is the assumption of linear relationship between variables while there is no guarantee that image features of relevant images (or irrelevant images) fit in a linear model. Therefore, disjoint subclass of relevant images are modeled as a local neighborhood around each image in the developed method, the assumption of linearity is applied only to neighborhoods. The process includes discrimination and compactness phases [131]. In the discrimination phase, irrelevant images are separated from relevant images based on their semantic distances. The semantic distance between two images, sd_{ij}, is computed by the difference of their scores assigned by the user in a transaction:

$$sd_{ij} = | s_i - s_j | \qquad \text{(Eq. 3.44)}$$

In the compactness phase, the relevant images are detected to map them to close locations in the new subspace. For this purpose, points that are not only located closely in the feature space, but also related semantically are detected. As mentioned, relevant images may lay in disjoint subclasses in the feature space; therefore, it is assumed that the neighborhood of a relevant image is related to one of the relevant classes in the feature space.

Two images are considered similar if they are semantically similar (based on the user's feedbacks), and they are in the same neighborhood in the original image feature space. To define a neighborhood, the pair-wise Euclidean distances between images (d_{ij}, $i \neq j$) are computed in the feature space. Image similarity, which is based on semantic similarity and image feature similarity, between two images is noted by D_{ij}.

As feedback scores are between zero and one, d_{ij}'s are normalized to be between zero and one, and keep the balance between the influence of the semantic differences and feature distances between two images.

$$\begin{aligned} d_{ij} &\leftarrow Normalize(d_{ij}) \\ D_{ij} &= (s_i * s_j) - d_{ij} \\ D_{ij} &\leftarrow Normalize(D_{ij}) \end{aligned} \qquad \text{(Eq. 3.45)}$$

Then, D_{ij} is normalized and computed for all images by:

$$D_{ij} = \begin{cases} D_{ij} & ;i \neq j \\ 1 & ;i = j \end{cases} \qquad \text{(Eq. 3.46)}$$

Finally, the total similarity between two images, TS_{ij}, is defined based on their similarities in a neighborhood defined by a threshold of ε. (Eq. 3.47) computes the similarity between only images located in a close neighborhood in the feature space, and assigns a similarity value of

zero to the images outside a neighborhood. This property is used in (Eq. 3.48) to find the mean of the image features in a neighborhood.

$$TS_{ij} = \begin{cases} D_{ij} & ; D_{ij} \geq \varepsilon \\ 0 & ; o.w. \end{cases} \tag{Eq. 3.47}$$

When the total similarity between two images are computed, the mean of image features, m_i, for the neighborhood of image i is computed by:

$$m_i = \sum_j x_j . TS_{ij} \tag{Eq. 3.48}$$

If two images i and j are not located in a neighborhood, TS_{ij} is zero and image feature values of image j is not considered in computing image feature values for the neighborhood of image i. According to the linear discriminant analysis model (Section 2.4), soft discriminant analysis (SDA) can be formulated as:

$$\begin{aligned} S_{NP} &= \sum\nolimits_{i,j} (m_i - m_j)(m_i - m_j)^T . sd_{ij} \\ S_P &= \sum\nolimits_{i,j} (x_j - m_i)(x_j - m_i)^T . TS_{ij} \\ W_{SDA} &= \arg\max_W \frac{|W^T S_{NP} W|}{|W^T S_P W|} \end{aligned} \tag{Eq. 3.49}$$

where S_{NP} is the distance of negative examples from the center of positive examples, S_p is the distance of images from the center on positive images, and the optimum W is obtained by solving the following generalized maximum eigenvalue problem:

$$S_{NP} W = \lambda S_P W \tag{Eq. 3.50}$$

It can be shown that BDA is a special case of SDA when the data is labeled binary, and positive examples are visually similar. The later assumption forces the neighborhood threshold to extend and cover all data points. Therefore, neighborhood similarities (d_{ij}) are omitted, and the image similarities, D_{ij}, are computed based on only semantic similarity.

If x_i is a positive example, $s_i*s_j=1$ and $TS_{ij} = D_{ij} = 1/N_P$ from (Eq. 3.45) and (Eq. 3.47) for all positive examples $j \neq i$, and 0 otherwise; where N_P is the number of positive examples. Thus, $m_i = m_p$ according to (Eq. 3.8), where m_p is the mean of image features for positive examples. On the other hand, if x_i is a negative example, $TS_{ij} = D_{ij} = 0$ for all images, and 1 for $i=j$. Thus, $m_i = x_i$. In Table 3.1, X_i and X_j represent two sets of images. According to the definition of BDA, S_{NP} is computed for the distance of negative example from positive example. As Table 3.1 shows, when both negative and positives examples are available (cases 2 and 3), $sd_{ij} = 1$, and S_{NP} is computed in (Eq. 3.49). In other cases, sd_{ij} is zero and S_{NP} is not computed. Similarly, S_P is computed only for positive examples in BDA.

Table 3.2 Relation between SDA and BDA

Case	$\boldsymbol{X_i}$	$\boldsymbol{X_j}$	$\boldsymbol{sd_{ij}}$	$\boldsymbol{TS_{ij}}$	$\boldsymbol{m_i}$	$\boldsymbol{m_j}$
1	+	+	0	1	m_P	m_P
2	+	–	1	0	m_P	X_j
3	–	+	1	0	X_i	m_P
4	–	–	0	0	X_i	X_j

Table 3.2 shows that TS_{ij} is non-zero only when both sets are positive (case 1), and zero otherwise. Therefore, S_P is computed in (Eq. 3.9) only if both examples are positive. Therefore, SDA is the general model of BDA for images with soft labels, and relevant images located in disjoint clusters in the feature space.

3.4.1 Image Retrieval with Short-term Learning Algorithm

In order to find similar images to the current query, the distance between each unlabeled image x_u and all scored images in the current transaction t_c are computed. Distance computation is performed in the new subspace, which is updated after each feedback iteration. The estimated score for x_u can be computed by (Eq. 3.51).

$$Y \leftarrow W_{SDA}.x$$
$$estScore(u) = \sum_{i \in L_c} \frac{s_{ci}}{DistY_{iu}} \qquad \text{(Eq. 3.51)}$$

where Y is the projected value of a data point x to the new space, L_c is the set of labeled images in transaction t_c, s_{ci} is the score of image x_i in transaction t_c, and $DistY_{iu}$ is the Euclidean distance between x_i and x_u in the projected space. The advantage of this short-term learning method is that a semantic space is constructed for each transaction and image semantic relations are extracted based on each individual search session. Algorithm 3.4 summarizes the process of finding similar images after each iteration.

Algorithm 3.4: Short-term learning: Finding similar images after each feedback iteration

Inputs: Feedbacks in current iteration (**S**), Image features (**F**)
Output: Set of similar images (**R**)

```
Compute W_SDA
for each unlabeled image x_u in t_c do
    estScore(u) = 0
    for each labeled image x_i in t_c do
        DistY(i,u)=Distance(Y_i,Y_u,W_SDA)
        estScore(u)=estScore(u) + S(i)/DistY(i,u)
    end for
end for
R = Top-n (estScore(u))
```

Figure 3.6 Short-term learning algorithm

At the end, images with highest estimated score are returned as the algorithm output.

3.5 IMAGE RETRIEVAL

In previous sections, long-term and short-term image retrieval algorithms were explained. Primary experiments showed that the developed retrieval algorithms demonstrate different performances for different type of queries.

For example, the top row images in Figure 3.7 are related to Forest class. As they are visually similar, they can be represented by a set distinguishable image features, and short-term learning is effective for retrieving images related to the Forest concept. On the other hand,

images in the second row, representing the concept of Car, cannot be presented by a set of distinct features due to high variation of image features, especially color features. Therefore, short-term learning is not effective and long-term learning is required to find similar images based on prior search results. Usually, images fall between these two extremes, and a combination of both short-term and long-term learnings is required.

Figure 3.7 Examples of short-term and long-term effectiveness

The developed system is equipped with a mechanism to trace the images returned to the user by each retrieval algorithm, compare the user's feedbacks, and measure the effect of each retrieval algorithm on the system performance.

The output of Algorithm 3.3 is called R1. There is an importance factor α associated with images in R1, which is computed based on the cosine distance between the image, x_i, and all images labeled in the current transaction, t_c, in the semantic space. The weights are normalized to have $\Sigma\ \alpha = 1$.

$$\alpha_i = \sum_{j \in t_c} Cos(x_i, x_j) \qquad \text{(Eq. 3.52)}$$

Similarly, the output of Algorithm 3.4 is called R2, with weights of β, which are computed by (Eq. 3.53) and normalized to have $\Sigma\ \beta = 1$. The difference between unlabeled image x_i and labeled image $x_{i'}$ from current transaction t_c is computed in feature space by $dy_{ii'}$.

$$\beta_i = \sum\nolimits_{i' \in L} estScore(t_c, x_i) / dy_{ii'} \qquad \text{(Eq. 3.53)}$$

The effect of long-term learning algorithm on system performance is measured by λ that is updated after each iteration by (Eq. 3.54). The total importance factor of an image in current iteration is computed by $\theta_i = \lambda\alpha_i + (1-\lambda)\beta_i$. Images with highest θ values are returned to the user.

$$\lambda_{c+1} = \left(\sum\nolimits_{i \in R1} \frac{s_i}{|R1|}\right) / \left(\sum\nolimits_{i \in R1} \frac{s_i}{|R1|} + \sum\nolimits_{i \in R2} \frac{s_i}{|R2|}\right) \qquad \text{(Eq. 3.54)}$$

In (Eq. 3.54), λ_{c+1} is the effect factor of long-term retrieval algorithm, and s_i is the score assigned by the user for image *i*. λ_0 is initialized to 0.5 for the first iteration. The retrieval algorithm for the developed system is summarized in Figure 3.8.

Algorithm 3.5: Image Retrieval Algorithm

Inputs: Semantic space (**Z**), Feedbacks in the current session (**S**), Image features (**F**)

Output: A set of relevant images (**R**)

1: Update λ (33)

2: (R1, α) = Set of relevant images returned by the long-term retrieval algorithm (Algorithm 3.1)

3: (R2, β) = Set of relevant images returned by the short-term retrieval algorithm (Algorithm 3.2)

4: Compute importance factors for θ for each image

5: Balance importance factors (33)

6: R = Images with highest importance factors of θ

Figure 3.8 Image retrieval algorithm

3.6 RETRIEVAL PHASES

In keyword-based queries such as Google, the user enters keywords related to the topic that she is looking for. After the results are returned by the system, she may change keywords:

she drops or adds some keywords to direct the search algorithm towards topics of her interest. In a CBIR system; however, the query is created by labeling images. Unlike keyword search, the user is limited to the images on the screen to build up her query. In the developed system, the user is provided with a mechanism to construct her query based on not only concepts displayed on the screen, but also other concepts available in the database. For this purpose, the idea of "most positive" and "most ambiguous" images [16] are used.

Most Positive (MP) images are those determined by the system to be similar to the query image. Therefore, they are also similar to each other, presenting a narrow set of concepts. If the retrieval system returns only MP images, the user has a limited set of concepts to build the query. On the other hand, Most Ambiguous images (MA) are semantically rich and present multiple concepts; therefore, more concepts can be presented to the user by MA images. The combination of soft feedbacks option and ambiguous images, provide the user with the flexibility of creating the query using fewer images.

(a)

(b)

(c)

Figure 3.9 Application of ambiguous images (semantically rich images)

An example is presented in Figure 3.9. Instead of showing one image related to Sea and one for City concepts, a more ambiguous image can be used to combine the concepts of the two images. Therefore, more concepts can be displayed on one page to the user. Using soft labels, the user interested in Sea, provides image (c) with some degree of relevancy. In the next iteration, the system returns images (a) and (b) only if image (c) has a non-zero label to clarify which image interested the user. On the other hand, if image (c) is assigned a zero score, the system learns that the user is interested in neither sea nor city concepts.

Image retrieval in the developed system includes two phase. The first phase is query learning, which is based on the idea of presenting concepts available in the database to the user to help her create her query. MA images are used in this phase because they carry more concepts within a single image; therefore, the number of images required to be displayed on the screen is minimized. According to the definition of the most ambiguous images, a MA factor is computed for an image by (Eq. 3.55) as the total of relationships in the semantic classes, where r is the number of semantic classes. Images are sorted discerningly based on their MA factors when the system searches for MA images.

$$MAfactor_i = \sum_{c=1}^{r} Z_{c,i} \qquad \text{(Eq. 3.55)}$$

When the user recognizes that the system has correctly captured her query concept, she informs the system to finish the learning phase and start the search phase. In search phase, MP images are computed based on their similarities to the query concept, and returned to the user. A weakness of the current CBIR system interfaces is the limited interaction between user and system. The user finds out about the structure of the concepts learnt by the system only after she sends her feedbacks to the system and the results are returned during many iterations. A new feature was introduced in the developed system to reduce the gap between the concepts learnt by

the system for an image, and the user's perception of that image. When users click on an image displayed on the system interface, the top-4 similar images to that image, which are computed based on long-term learning, are shown to the user. Therefore, the user gains more information about the image and its relation with unlabeled images in the database.

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, experiments related to the feedback methodologies, long-term and short-term learning algorithms, the developed CBIR system in VB.NET, and application of retrieval phases are presented. In Section 4.1, two feedbacks methodologies are proposed, and their performances are compared to binary feedbacks, which are the popular method in CBIR systems. The developed long-term and short-term learning algorithms are tested and compared to related methodologies in the literature in Sections 4.3 and 4.4. The two algorithms are put together and a CBIR system is developed in VB.NET. Section 4.5 discusses the features of this system. Finally, the application of retrieval phases is tested in Section 4.6.

4.1 STUDYING RELEVANCE FEEDBACK METHODOLOGIES

In this section, the performance of various feedback methodologies are studied. A feedback methodology is the method used by the system to collect users' feedbacks. An efficient feedback method collects more information from a user in fewer iterations, and acts as a constructive tool for the user to transfer the image query she has in her mind to the system. As multiple users participate to train the proposed system, it is important to apply a feedback methodology that minimizes the subjectivity between users.

Two factors are used to measure the quality of feedbacks: the variance of feedbacks provided by different users for a specific query, and the variance of feedbacks provided by a user in two sessions. During the experiment, participants describe the concept that they have in their minds for each query, and explain their feedbacks criteria to the researcher. At the end of experiments, participants were asked to indicate their preferred feedback methods and explain the reasons of their selections.

4.1.1 Binary and Scored Feedbacks

A user experiment was performed to compare binary feedbacks to scored feedbacks based on reducing user subjectivity, and differences in users behaviors and users preferences.

Two types of image searches are introduced in [25] as target and category search. In target search, the user searches for a specific picture. The user can exactly explain the target image in object and some image features levels. In category search, the user is looking for images from a certain category. Searching for a picture of a specific car is an example of target search while searching for pictures of automobiles is a category search. In the experiments, participants were instructed to perform category searches.

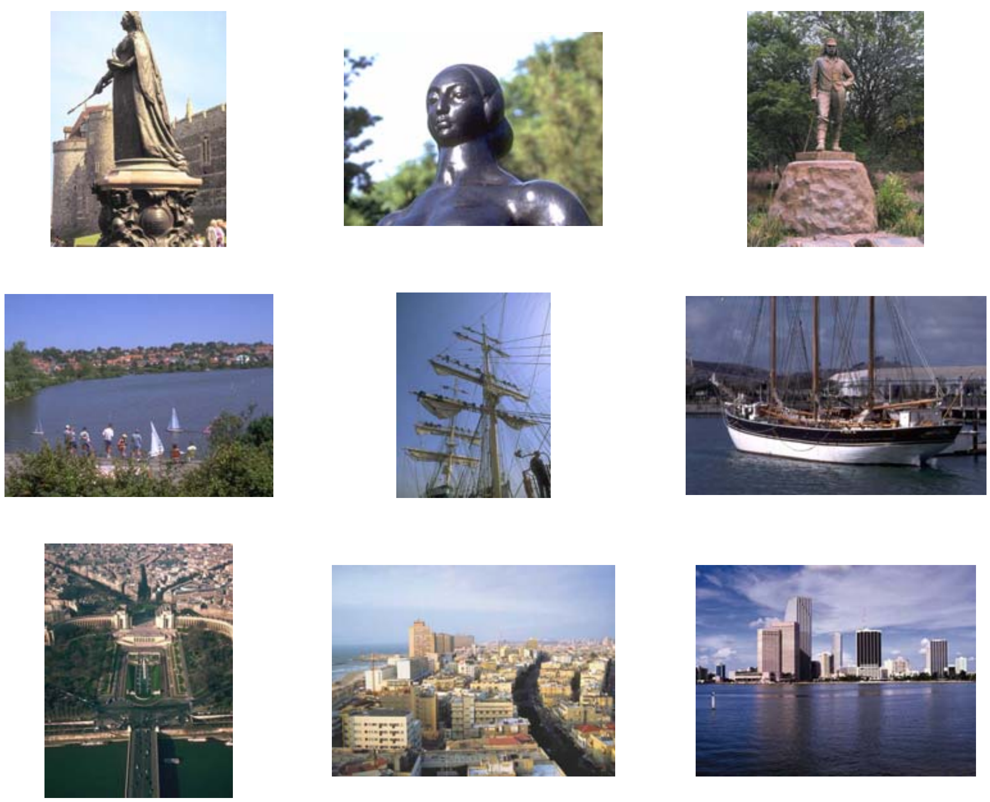


Figure 4.1 Image query generators: each row is used for a query

It is assumed that a CBIR user has a picture in her mind, which is referred to as a mental image [127], and searches for images from the same category as the mental image category.

Therefore, an image query, the user's mental image, is known to nobody but the user. To generate image queries for this experiment, a subject is provided with keywords or images to help her with creating a mental image. In the first case, a set of keywords are displayed to the subject for each query. Then, the users explain to the researcher what types of images come to their minds, and describe the specifications of their mental images. Three keyword queries were used: "people faces" (Q1), "memorial building" (Q2), and "transportation vehicle" (Q3) in this experiment. Although there is no guarantee that generated queries are exactly similar, they can be classified based on their query generators. That is, all user feedbacks provided for a query generator, for example "statue" are considered to be in a class for further analysis.

In the second case, three similar images were shown to the subject for each query; users explain their image perceptions, and their selection or scoring criteria. Figure 4.1 shows the image query generators. Searching for images related to "statues" (Q4), "sailing" (Q5), and "city skylines" (Q6) were the image-based queries in this experiment. Thus, there were 6 queries, and 6 image sets with a size of 100 images for each query to be labeled. Images of a set are randomly displayed to the subjects on five pages with 20 images on each page.

Participants provide or modify their scores for images on a page, and then click on a "submit" button to go to the next page. They can not change their scores/selections after they submit a page. Each subject provided binary and scored feedbacks for each of 6 queries. Therefore, there were a total of 12 tasks which were randomly presented to a subject in a session.

4.1.1.1 Interface: The user interface for collecting feedbacks in this experiment in includes two areas. On the left side, the query generators (images or keywords) are shown, and images are displayed in the other area. In binary feedback tasks, the user clicks the image to

selects it, and can unselect it by clicking the image for a second time (Figure 4.2). Selected images are distinguished by a frame appeared around the image.

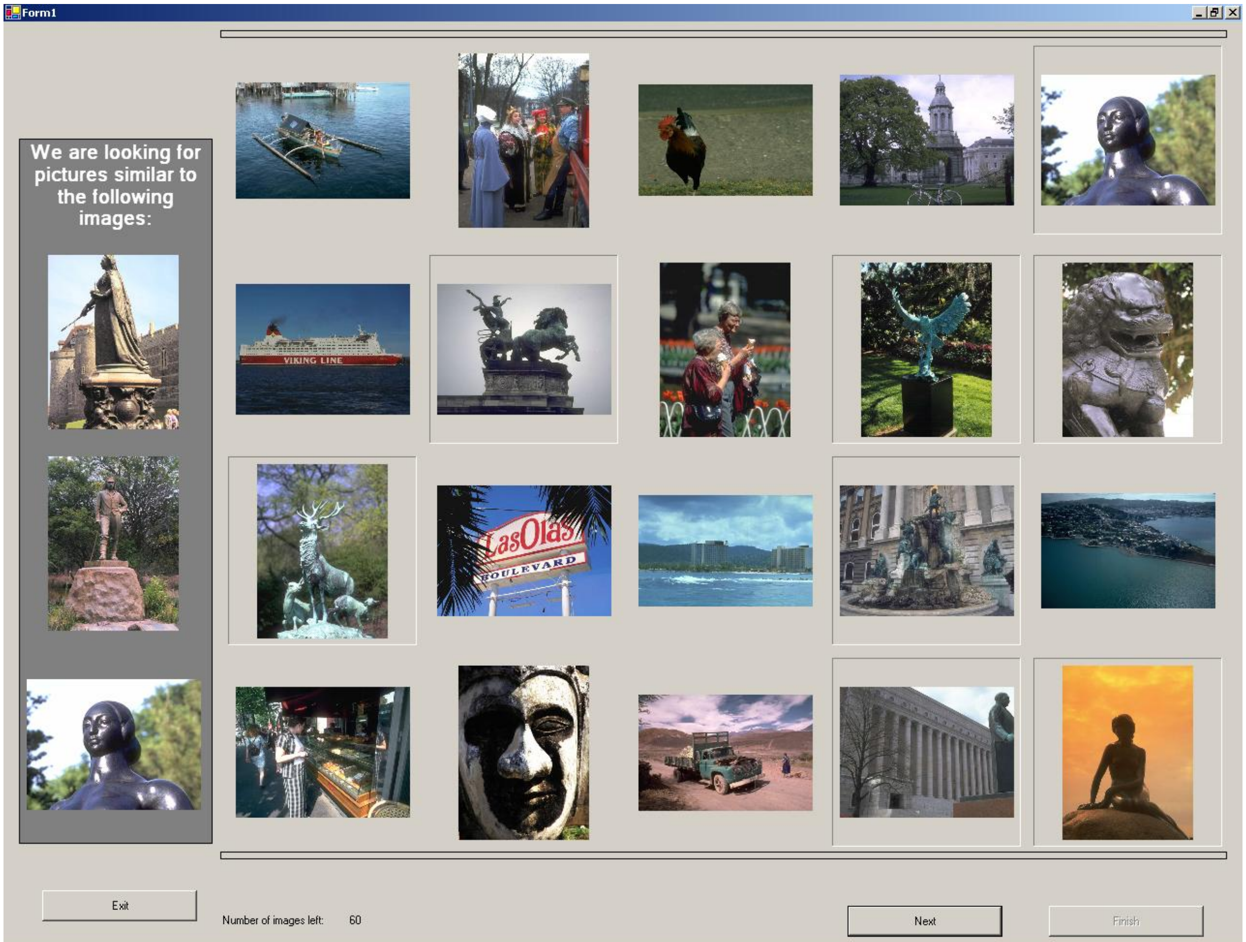


Figure 4.2 User interface for binary feedbacks

There is a slide bar under each image in scored feedback tasks and the user moves the bar to assign a score to the image (Figure 4.3). Bars are initially located on the left side of the bar zero scores. Users can indicate a perfect match by dragging the slide bar to its most right extreme. It was decided to remove any labels or other score indicators from the slide bars because it was found that they are confusing for many users in the pre-experiment tests.

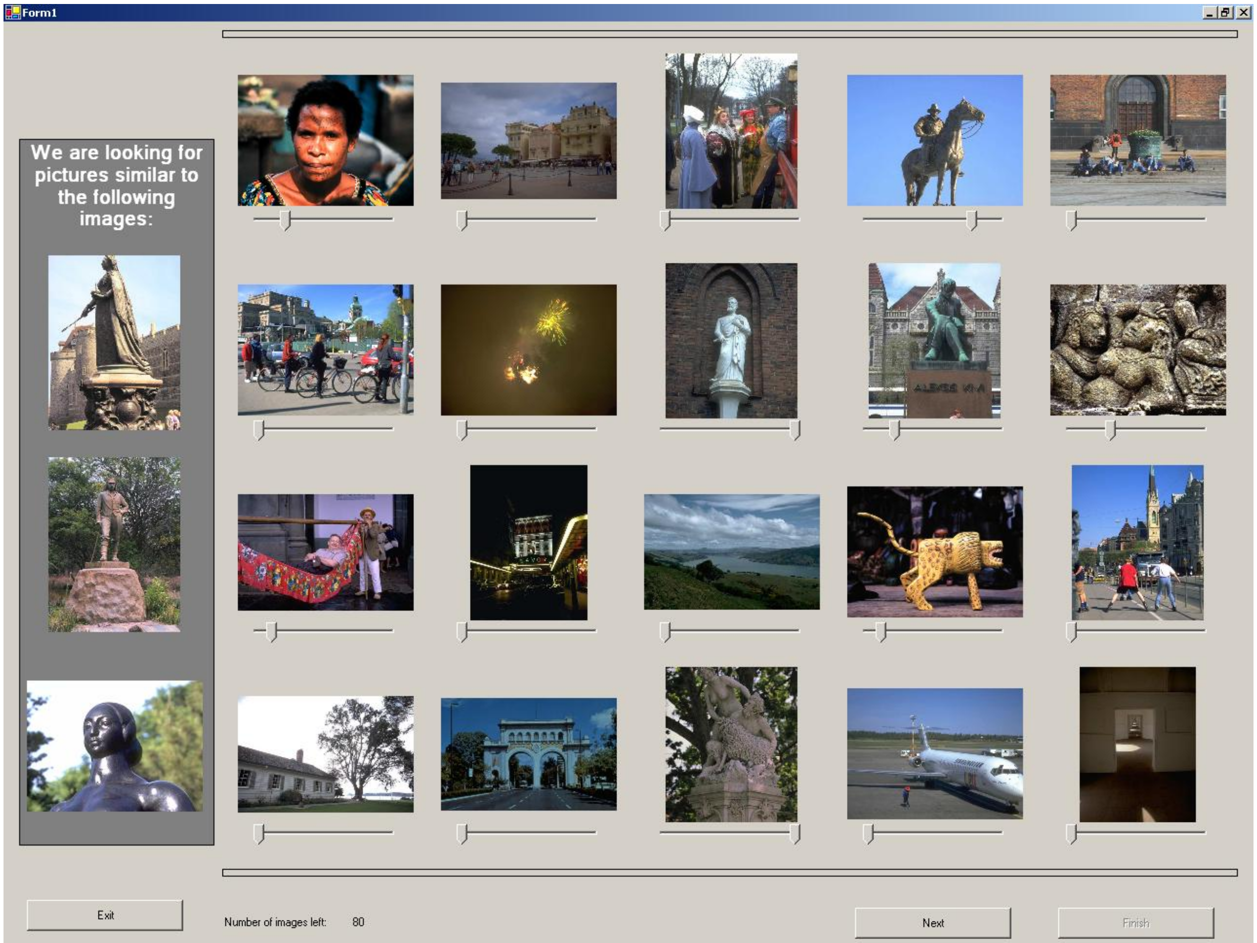


Figure 4.3 User interface for scored feedbacks

4.1.1.2 Participants: In this experiment, 22 engineering students (16 males and 6 females) with average age of 21 participated. The number of graduate and undergraduate students was 5 and 17 respectively. Subjects performed the experiment in two sessions with a gap of about 10 days in between.

4.1.1.3 Studying Feedbacks Provided by Different Subjects: As the image retrieval system is to be trained and tested by multiple subjects, it is important to use a feedback methodology that reduces the factor of subjectivity in image selection. Due to different image perceptions, users do not usually assign similar labels to an image for the similar query. Therefore, an efficient feedback methodology is required to minimize the variance between

labels provided by different users because the performance of the learning process is affected by the variance of training data [115].

To compare binary and scored feedbacks in more detail, analysis of variance (ANOVA) [60] is used to study the effect of different factors in a relevance feedback method. In ANOVA, a factor is an independent treatment variable whose values (levels) are set by the experimenter. Therefore, there are four factors in the relevance feedback experiment including subjects (22 levels), sessions (2 levels), queries (6 levels), and images (100 levels). As different image sets were used for each query, a three-way (factor) ANOVA was used for each query. The null hypothesis for ANOVA is that there is no difference for the population means of different levels of each factor in the model. The alternative hypothesis is that the means are not equal. ANOVA tends to reject the null hypothesis because if the distribution of only single level is different from others within a factor, null hypothesis for that factor is rejected, implying that the population means of the levels are not equal in that factor.

Each null hypothesis is associated with a p-value in an ANOVA model. If any p-value is near zero, this casts doubt on the associated null hypothesis. For example, a sufficiently small p-value for a null hypothesis of factor X suggests that at least one X-sample mean is significantly different from the other X-sample means; that is, there is a main effect due to factor X. It is common to declare a result significant if the p-value is less than 0.05 or 0.01. In N-way ANOVA, the interaction between factors can also be studied in addition to the effect of each factor. A small p-value for the interaction of two factors rejects the null hypothesis and implies that those factors are independent.

Query	Binary Feedbacks						Scored Feedbacks					
	Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F	Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
1	X1	3881.9	16	242.618	70.43	0	X1	2919.2	16	182.448	118.7	0
	X2	8.5	1	8.5	2.47	0.1164	X2	23.4	1	23.389	15.22	0.0001
	X3	37857	99	382.394	111	0	X3	30119.2	99	304.234	197.94	0
	X1*X2	308	16	19.25	5.59	0	X1*X2	187.5	16	11.718	7.62	0
	X1*X3	15000.5	1584	9.47	2.75	0	X1*X3	7634.2	1584	4.82	3.14	0
	X2*X3	276.8	99	2.796	0.81	0.9095	X2*X3	158.5	99	1.601	1.04	0.3735
	Error	5456.7	1584	3.445			Error	2434.6	1584	1.537		
	Total	62789.4	3399				Total	43476.6	3399			
2	X1	3135.5	16	195.967	31.14	0	X1	3932.4	16	245.773	88.55	0
	X2	2.4	1	2.382	0.38	0.5385	X2	29.7	1	29.742	10.72	0.0011
	X3	26632.3	99	269.013	42.74	0	X3	17143.2	99	173.163	62.39	0
	X1*X2	519.1	16	32.445	5.16	0	X1*X2	219	16	13.687	4.93	0
	X1*X3	22682.2	1584	14.32	2.28	0	X1*X3	13817.8	1584	8.723	3.14	0
	X2*X3	759.4	99	7.671	1.22	0.076	X2*X3	212.7	99	2.148	0.77	0.9496
	Error	9969.1	1584	6.294			Error	4396.6	1584	2.776		
	Total	63700	3399				Total	39751.4	3399			
3	X1	702.8	16	43.923	15.35	0	X1	1629.9	16	101.868	53.38	0
	X2	0.7	1	0.735	0.26	0.6123	X2	27.9	1	27.901	14.62	0.0001
	X3	65292.1	99	659.516	230.46	0	X3	51039.1	99	515.546	270.14	0
	X1*X2	178.8	16	11.173	3.9	0	X1*X2	122.5	16	7.654	4.01	0
	X1*X3	8338.4	1584	5.264	1.84	0	X1*X3	7506.8	1584	4.739	2.48	0
	X2*X3	337.5	99	3.409	1.19	0.1024	X2*X3	195.6	99	1.976	1.04	0.3891
	Error	4533	1584	2.862			Error	3023	1584	1.908		
	Total	79383.3	3399				Total	63544.7	3399			
4	X1	2096.9	16	131.059	60.44	0	X1	2254.3	16	140.895	109.13	0
	X2	1.1	1	1.059	0.49	0.4848	X2	80.8	1	80.758	62.55	0
	X3	55503.6	99	560.643	258.55	0	X3	41950.4	99	423.742	328.2	0
	X1*X2	82.9	16	5.184	2.39	0.0015	X1*X2	139.6	16	8.723	6.76	0
	X1*X3	10338.4	1584	6.527	3.01	0	X1*X3	7304.5	1584	4.611	3.57	0
	X2*X3	181.3	99	1.831	0.84	0.8602	X2*X3	215.5	99	2.177	1.69	0
	Error	3434.7	1584	2.168			Error	2045.1	1584	1.291		
	Total	71638.9	3399				Total	53990.3	3399			
5	X1	1456.1	16	91.004	26.73	0	X1	1889	16	118.061	59.83	0
	X2	0.7	1	0.735	0.22	0.6422	X2	41.6	1	41.581	21.07	0
	X3	54727.6	99	552.804	162.4	0	X3	39110.6	99	395.056	200.21	0
	X1*X2	278.8	16	17.423	5.12	0	X1*X2	295.1	16	18.445	9.35	0
	X1*X3	11496.9	1584	7.258	2.13	0	X1*X3	7191	1584	4.54	2.3	0
	X2*X3	378.7	99	3.825	1.12	0.197	X2*X3	232.8	99	2.351	1.19	0.102
	Error	5391.8	1584	3.404			Error	3125.5	1584	1.973		
	Total	73730.6	3399				Total	51885.5	3399			
6	X1	3347.5	16	209.217	34.63	0	X1	2692.3	16	168.271	68.27	0
	X2	375.6	1	375.559	62.16	0	X2	271.1	1	271.059	109.97	0
	X3	46478	99	469.474	77.7	0	X3	37156.1	99	375.314	152.27	0
	X1*X2	511.9	16	31.996	5.3	0	X1*X2	277.9	16	17.369	7.05	0
	X1*X3	18252.5	1584	11.523	1.91	0	X1*X3	9306.1	1584	5.875	2.38	0
	X2*X3	692.1	99	6.991	1.16	0.1446	X2*X3	318.7	99	3.219	1.31	0.0263
	Error	9570.4	1584	6.042			Error	3904.3	1584	2.465		
	Total	79228	3399				Total	53926.6	3399			

Figure 4.4 ANOVA tables for binary and scored feedbacks

ANOVA tables for each query are displayed in Figure 4.4. Subjects, sessions, and images are shown by X1, X2, and X3. Interactions between factors are shown by X1*X2, X1*X3, and X2*X3. Comparing ANOVA tables demonstrates the following results:

1. The effect of subjects is significant on both binary and scored feedbacks (p-value is zero). It implies that different subjects, as expected, provided different scores.

2. The image factor also has a p-value of zero and, as expected, implies that different scores are assigned to the images in the experiment. Comparing F-values for image factor (X3) in binary and scored feedbacks suggests that scored feedbacks differentiated images more effectively. The distribution of scores (Section 4.1.2.3) explains this fact more precisely.

3. It seems that the effect of sessions is significant only when scored feedbacks are used because this factor has low p-values for scored and large p-values for binary feedbacks. However, the high p-value of the interaction between sessions and images shows the relation between these two factors, which is resulted from the existence of images with low-level of complexity or "easy-to-label" in each query. Such images are not semantically rich and can be precisely labeled as relevant or irrelevant; therefore, they are labeled similarly in different sessions by binary feedbacks, and lead to lower between-session variances. The distribution of easy-to-label images is discussed in Section 4.1.2.3.

Similarly, between-subject variances are affected by the distribution of images with low level of complexity. However, the number of subjects (unlike number of sessions) is high in the experiment and the interaction between subject and image is not significant.

The main interest of the experiment is to evaluate the performance of feedback methods for semantically rich images and queries; therefore, images with low-levels of complexity are detected and dropped off the analysis. In each query, the first 50 images with lowest variances are considered "easy-to-label".

To compare between-subjects variances in binary and scored feedbacks, a null hypothesis is set up as H_0: The variance between subjects in scored feedbacks is equal to or higher than

binary feedbacks. H_0 was tested for each query and session; therefore, each test included a sample size of 1100. F-tests with significant levels of 5% and 10% were used to compare the variances. Test results are reported in Table 4.1 for the first and second sessions.

Table 4.1 Analysis of between-subjects variances

Query	**Q1**		**Q2**		**Q3**		**Q4**		**Q5**		**Q6**	
Session	1	2	1	2	1	2	1	2	1	2	1	2
F stat	0.72	0.66	0.64	0.62	0.82	0.79	0.80	0.71	0.76	0.67	0.71	0.66
H_0	Reject	Reject	Reject	Reject	Reject	Reject	Reject	Reject	Reject	Reject	Reject	Reject

In above tests, degree of freedom is 1099 for both binary and scored feedbacks, $F_{2.5\%}$=0.88, $F_{5\%}$=0.90, and $F_{10\%}$=0.92. H_0 is rejected for all queries with significant levels of 5% and 10%; therefore, it can be stated that scored feedbacks demonstrated lower between-subjects variances relative to binary feedbacks. Comparing results from Table 4.1 and ANOVA tables shows that when images are semantically rich, scored feedbacks help users to provide scores that are more accurate than binary feedbacks. On the other hand, scored feedbacks create "noise" in users input when images are not complex. Figure 4.5 shows the average of between-subjects variances in both sessions for scored feedbacks and binary.

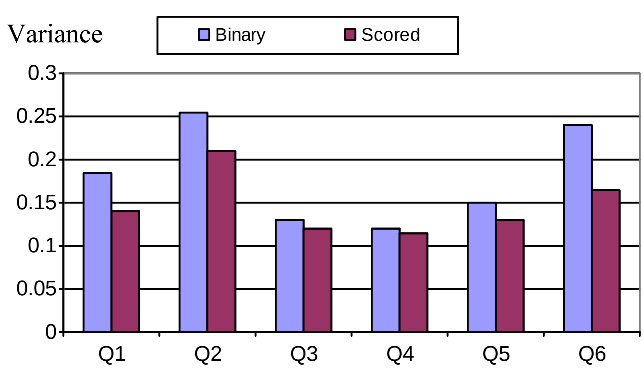


Figure 4.5 Average of between-subjects variances

4.1.1.4 Studying Feedbacks Provided in Different Sessions: In the same way that users may assign different labels to an image for a query, a particular user may assign different labels to an image in different search session. Therefore, an efficient feedback methodology should also minimize the between-sessions variances.

H_0 was set as the between-session variances for scored feedbacks are equal to or larger than binary feedbacks. F-test was used to compare the variances. Results show that there is no significant difference between scored and binary feedbacks regarding between-session variances for transportation, memorial building, and statue queries. Matching the results with level of complexity for each query (Section 4.1.2.6) indicates that when query concept is simple (transportation and statue) there is no significant difference between binary and scored feedbacks. Moreover, when the query concept is so complex (memorial building) that makes users confused, they provide feedbacks in a random manner; therefore, there is no significant difference between feedback methodologies.

Table 4.2 Analysis of between-sessions variances

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**
F stat	0.75	0.94	0.96	1.12	0.88	0.73
H0	Rejected	Accepted	Accepted	Accepted	Rejected	Rejected

Figure 4.6 shows that when binary feedbacks were used, the variance between subjects achieved a stable state after eight users provided their feedbacks. On the other hand, adding more than five people to the experiments with scored feedbacks did not change the variance. Therefore, it can be concluded that fewer subjects are needed to label images for learning concept from scored feedbacks than binary feedbacks.

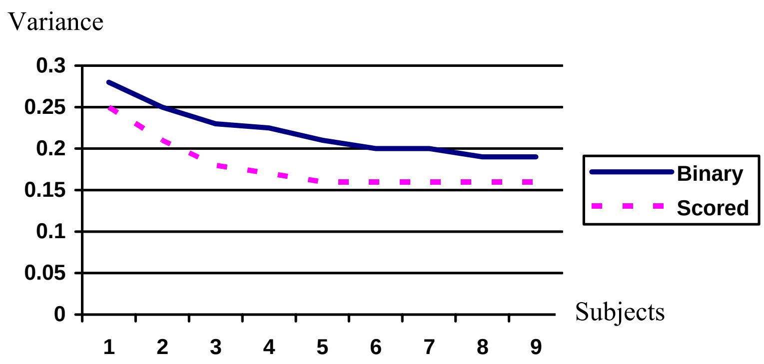


Figure 4.6 Stability of feedbacks

Another factor to compare binary and scored feedbacks is the number of principal components that can represent the search history matrix. In principal component analysis, the percentage of original data that is explained by the top-k important principal components can be estimated by:

$$p = \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{n} \lambda_i \qquad \text{(Eq. 4.1)}$$

where λ_i is the i^{th} largest eigenvalue of the data matrix, and n is total number of principal components. The transactions history for this experiment includes 587 images. Using Principal Component Analysis, the 6 most important components, which is equal to the number of semantic classes for the experiment, represent 63% of data when binary feedbacks are used, and 89% of data when scored feedbacks are used. Therefore, transactions matrix can be reduced by replacing 587 images by 6 components at the cost of loosing only 11% of data when scored feedbacks are used – but 37% in the case of binary feedbacks.

4.1.2 Qualitative Observations

Qualitative observations are related to user behaviors studied during the experiment, including scoring strategies, feedback strategies, and required time to provide feedbacks. Results presented in this section are qualitative studies that helped user-system interaction improvement.

User behavior study helps researchers to understand how users interact with an image retrieval system. Therefore, researchers can design the system in such a way to improve the human and system interactions, collect more valuable data, and advance the retrieval performance in a CBIR system.

4.1.2.1 Scoring Strategy: A scoring strategy refers to the mechanism that a user scores images in a scored relevance feedback scheme. Users consider many factors in their scoring strategies, which may have different levels of importance for different users or queries. Two main set of factors are distinguished in this experiment through observations and interviewing subjects: inter-image and intra-image scoring factors.

Table 4.3 Number of participants for each scoring strategy

	Q1	Q2	Q3	Q4	Q5	Q6
Only Inter-image criteria	20	20	20	15	18	17
Inter-image and Intra-image criteria	2	2	2	7	4	5

Inter-image scoring factors are object size, importance of the object, and image features. Object size is simply the portion of the image assigned to an object. Importance of an object in an image expresses how much the image is related to that object. Sometimes, the largest object is the most important one in an image; however, relatively common objects such as “sky” have low importance though they may occupy a large portion of the image. In some queries, users look for an object with a precise color, texture, or shape specifications.

The main intra-image scoring factor is adjacency of images as users usually compare an image with other images displayed on the same page. Therefore, an image displayed along with some absolutely irrelevant images, may be assigned a different score (or label) from the case that it is displayed on a page with some more relevant images. Furthermore, the score (or label) of an image assigned in early feedback iterations may be different form later iterations in a query

session. As the retrieval process improves during relevance feedback iterations, more relevant images are returned to the user in later iterations. These two issues of relevance feedback have not been discussed in the literature. Image set domain is another factor. If users browse the images in the database and familiarize with the type and theme of images available in the database, their feedbacks may be different from the case they see images for the first time while providing feedbacks.

4.1.2.2 Users Feedback Strategies: Some users strategies are introduced in [26] for binary feedbacks. A feedback strategy explains how a user interacts with the system while providing feedbacks. An "annoyed" user randomly selects a subset of retrieved images and correctly labels them. A "cooperative" user correctly labels all relevant images. A "minimalist" user correctly labels only a few relevant images. An "optimistic" user selects all relevant images, and images with some degree of relevance (ambiguous images) as relevant. Finally, "tired" users make mistakes in their selections.

In scored feedbacks approach, similar feedback strategies were observed that can be explained based on users scoring strategies. "Annoyed" users may skip some pictures, or finish the scoring job if they feel they have labeled enough images. "Minimalist" users usually under-rate relevant images, and do not assign any scores to the most of the images. On the other hand, "optimistic" users over-rate relevant images, try to find a clue in the images related to the target concept and provide most of the images with at least a minimum score. Finally, "tired" users do not follow their own scoring criteria properly.

Another type of feedback strategies, which can be called "simple" users, was observed in scored feedbacks. A scored relevance feedback scheme is converted to binary for "simple" users, that is, images with a score higher than a threshold are considered relevant, and irrelevant

otherwise. In the experiments, a user is considered “simple” if s/he provides more than 90 binary feedbacks in a scored feedbacks task. To detect “annoyed” or “tired” users, ten images were selected by the researcher as perfect matches in each query. Users who missed labeling them are considered as “annoyed/tired”. The average (μ) and standard deviation (σ) of scores provided by users in each query were computed. Users with an average higher than ($\mu+\sigma$) are considered “optimist”, and users with an average lower than (μ - σ) are considered “minimalist”. In Table 4.4, the statistics for different feedback strategies are reported for binary (B) and scored (S) feedbacks in the second session of the experiment.

It was observed that a certain group of subjects acted as “minimalist” and “optimist” in all query sessions. On the other hand, “annoyed/tired” and “Simple” users acted differently in different queries. Feedback strategies are mostly related to the users personal characteristics; however, some associations between users feedback strategies and the complexity of the queries were observed in the experiment. When the query was simple, users usually behaved “cooperative”. When the query was complex, users were “annoyed”. However, they might change their strategies during a feedback session.

Table 4.4 Number of observed feedback strategies in each query

	Q1		Q2		Q3		Q4		Q5		Q6	
	B	S	B	S	B	S	B	S	B	S	B	S
Annoyed/Tired	1	2	0	0	2	2	3	2	1	1	0	1
Minimalist	0	0	2	1	1	0	3	3	3	2	1	1
Optimist	2	2	1	0	3	2	2	3	2	3	3	3
Simple	-	4	-	5	-	5	-	4	-	6	-	5

Intra-image scoring strategy factors such as the distribution of relevant, irrelevant, and ambiguous images in a feedback session can be important in determining a user’s feedback

strategy. It was observed that users are usually "minimalist" if they see too many ambiguous images, especially if there are some perfect matches. On the other hand, if there are few ambiguous images along with some irrelevant images, they were "optimistic".

4.1.2.3 Distribution of Feedbacks: Figure 4.7 shows the distribution of assigned scores in the scored feedbacks in the experiment as 45% irrelevant, 18% relevant, and 37% with different degrees of relevancy. In binary feedback, subjects labeled 68% of images as irrelevant and 32% as relevant. Therefore, 23% of images labeled as irrelevant had some degrees of relevancy, and 14% of images labeled relevant were not perfect matches. It can be stated that binary feedbacks are noisier than scored feedbacks because images with a score of 1 in scored format are perfect matches, and images with a score of zero are absolutely irrelevant. On the other hand, there are no differences between images labeled as relevant in binary format though they may be relevant with different levels of relevancy.

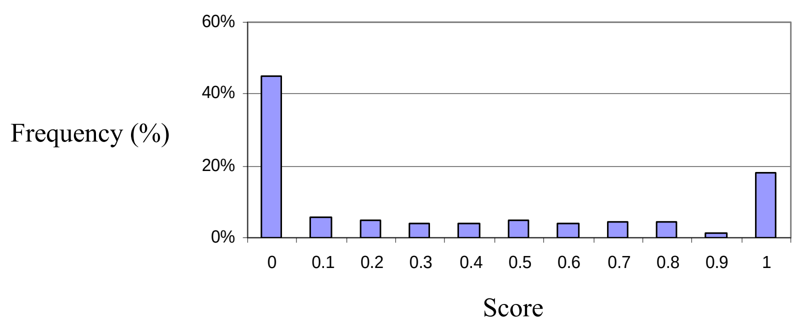


Figure 4.7 Distribution of assigned scored in the experiment

It was observed that when subjects were confused with providing a score for an image, they found it more comfortable to assign a score of 0.5 to the image. A number of subjects hesitated to provide a full score for images with a perfect match, and reserved full scores to differentiate those images from more relevant images they might see in following pages. Therefore, scores above 0.9 may represent a perfect match too.

4.1.2.4 Selection Threshold: Comparing scored to binary feedbacks for an image, a selection threshold is defined as the score that a user considers as a threshold for her binary selections. Intuitively, the selection score is 50%, that is, images with a score above 0.5 in scored feedback should be selected as relevant in binary feedback scheme.

Scored feedbacks were converted to binary with a threshold of 0.5, and a two-pair t-test was used to examine if converted scored feedbacks are equal to binary feedbacks for each image. The null hypothesis is set to "equal means of binary and converted scored feedbacks", and it is tested for each subject. Figure 4.8 shows the number of tests (out of 22) in which the null hypothesis was accepted in each query (α = 5%). Results show that for average of 13 subjects, binary feedbacks are equal to scored feedbacks with a cut-off of 0.5. Moreover, only five subjects were in all six queries and two sessions.

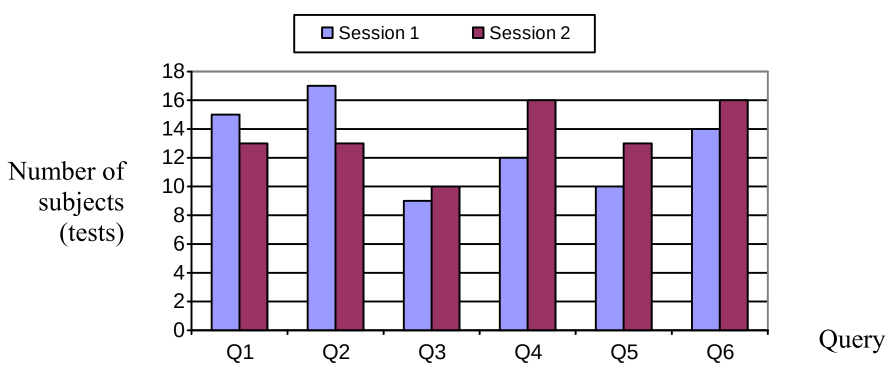


Figure 4.8 Number of tests with accepted null hypothesis in each query

4.1.2.5 Time: The software used in the experiment was set to store times that subjects spent on the experiment; however, subjects were told that time is not an issue to prevent any changes in their performances caused by pressure of time considerations. To study the differences in required time for providing feedbacks, the following null hypothesis are test.

- Q_0: The average of time in the first session is equal for scored and binary feedbacks.

- L_0: The average of time in the second session is equal for scored and binary feedbacks.

Table 4.5 Experiment times (seconds)

	Binary Feedbacks				Scored Feedback				T stat	
	Session 1		Session 2		Session 1		Session 2			
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Q_0	L_0
Q1	96	42	90	20	156	76	140	40	-4.29	-4.11
Q2	122	49	99	41	222	117	165	48	-5.05	-4.68
Q3	102	27	100	18	191	56	160	43	-4.88	-4.54
Q4	88	21	83	20	156	31	147	38	-4.52	-4.26
Q5	107	32	97	15	163	38	152	30	-4.47	-4.05
Q6	123	33	112	23	176	47	155	38	-4.36	-3.90
Average	106		96		177		153			

Above null hypothesis were rejected with a t-test (α = 10%) for all queries. In the tests, degree of freedom was 21, and $t_{2.5\%}$=1.98, $t_{5\%}$=1.66, and $t_{10\%}$=1.28. The average and standard deviation of time spent by subjects for labeling 100 images in each query, feedback method, and session are shown in Table 4.5. Therefore, it can be stated that participants finished the second session in shorter time, and scored feedbacks are more time demanding than binary feedbacks (as Q_0 and L_0 were rejected). However, results show that the required time for providing scored feedbacks for 100 images is only about 70 seconds in the first session, and 56 seconds in the second session longer than binary feedbacks. Therefore, scored feedbacks can not have any negative influence on subjects' performances caused by their tardiness from the longer experiment times in compare to binary feedbacks. The average time for labeling 100 images decreased 8% (binary feedbacks) and 12% (scored feedbacks) from the first to second session.

4.1.2.6 Interviewing Subjects: When subjects finished a session, including providing binary and scored feedbacks for six queries, they were interviewed by the researcher to study the qualitative factors of the experiment. Participants were asked to rank queries based on how

clearly they could create a mental image (subject's definition of the query) for that query, and how certain they were about their mental image during their searches. It is assumed that when subjects can hardly make a clear mental image or their mental image changes during the search, the query concept is complex.

Figure 4.9 shows percentage of participants indicating the level of difficulty for each query. The most difficult query selected by the subjects is "memorial building". Participants were trying to describe old and magnificent buildings for this concept but they could not come up with a clear image description.

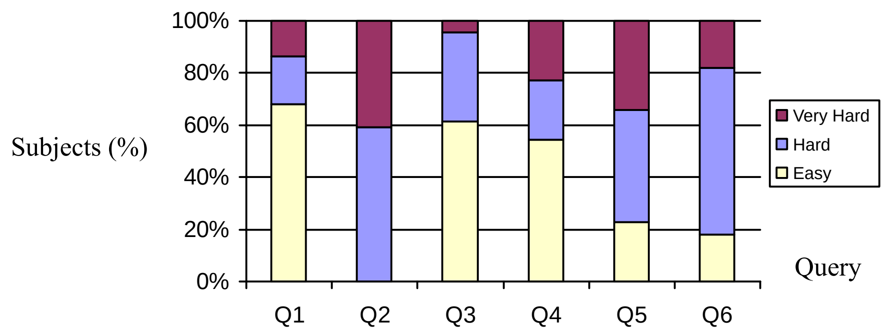


Figure 4.9 Level (percentage) of difficulty for each query

Seven subjects out of 22 stated that they were more comfortable with keyword-based queries as they could create and describe their own images. On the other hand, 15 subjects preferred image-based queries as they were provided with some examples at the start of their experiments. Based on Figure 4.9, the queries are sorted from simple to complex as transportation vehicle (Q3), statues (Q4), people faces (Q1), city skylines (Q6), recreational sailing (Q5), and memorial buildings (Q2). Table 4.2 shows that participants needed 50% more time to label images for a complex query (Q2) than a simple query (Q4).

In the first session, 10 subjects out of 22 preferred binary to scored feedbacks because they found it tedious to slide the scoring bar for 600 images. However, they believe that scored

feedback is a great help when they were looking for more abstract concepts, and 8 of them prefer a scored relevance feedback where there are less than 150 images to be scored, which is about 8 relevance feedback iterations in a CBIR system. Although preferring scored feedbacks, many participants pointed out the main drawback of the scored feedbacks as the difficulty in providing an accurate score when the image is not either relevant or irrelevant, but related with some degree of relevance. In the second session, only five subjects preferred the binary feedbacks while four of them feel more comfortable with scored feedbacks when the number of images is less (about 150 images).

4.1.3 Scored and Formulated Feedbacks

Results of comparing binary to scored feedbacks showed that scored feedbacks reduce the subjectivity by decreasing the variance between different subject, and different session. However, most of the participants in the experiment indicated a main drawback of scored feedbacks as the difficulty of deciding on a score to assign. Based on the results obtained through users behavioral studies and scoring strategies, another feedback methodology, formulated feedbacks, is proposed to be tested in this section.

A formulated feedback is introduced through an example: A user is interested in images related to buildings next to ocean (image (a) in Figure 4.10). Therefore, the query is decomposed to two elements: Building and Ocean, which as stored in set called Q. Similarly, image *i*, returned by the system is decomposed to its elements, which are stored in a set called P_i. If n(Q) shows the number of elements in Q, and $Q \cap P_i$ is the intersection of P and Q sets, a formulated feedback for image *i* is computed by:

$$\frac{n(Q \cap P_i)}{\max \{n(Q), n(P_i)\}} \qquad \text{(Eq. 4.2)}$$

Image (a): Query Image

Image (b) Score = 0.66

Image (c) Score = 1.0

Image (d) Score = 0.5

Image (e) Score = 0.5

Figure 4.10 Formulated feedbacks

Therefore, the formulated feedback for image (b) is 0.66 because it is related to building, ocean, and mountain. Image (c) is a perfect match and is assigned 1, and images (d) and (e) get scores of 0.5 as they are partially related.

Formulated feedback suggests users a methodology in assigning soft labels to the images in a CBIR system; however, it does not guarantee equal scores to be provided by different users. It helps users to have a consistent scoring strategy while comparing images with a query image.

To test the performance of formulated feedbacks, an experiment was set up similar to the previous one with only one session. Image queries, shown in Figure 4.11, and keywords of “vacational beach” (Q1), “mountain and beach view” (Q2), “downtowns located by the ocean”

(Q3), "rural road" (Q4), and "ancient statues" (Q5) were jointly used for each query. Participants in this experiment were three senior undergraduate and two graduate engineering students.

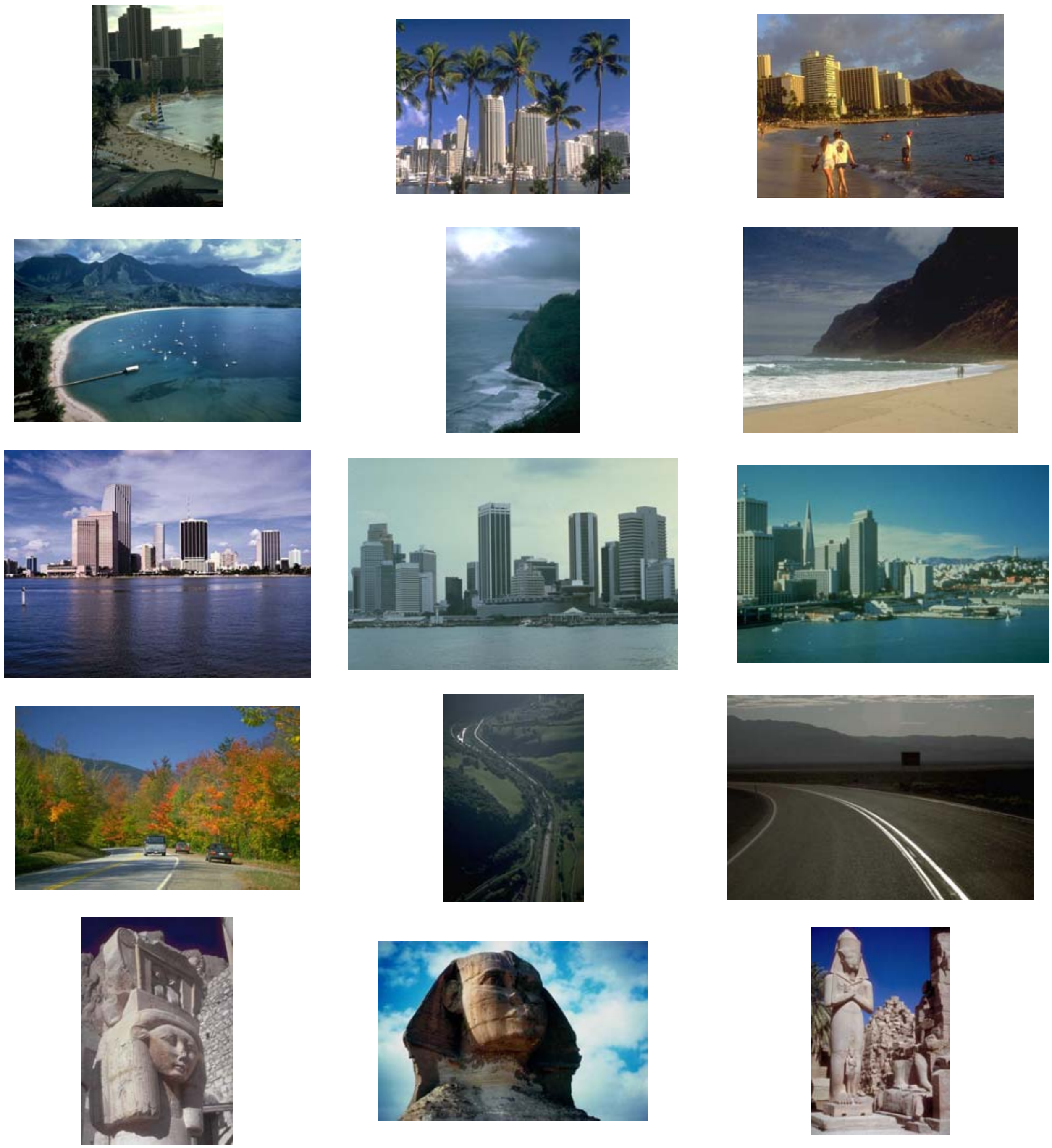


Figure 4.11 Image query generators: each row is used for a query

First, an introduction to the experiment procedure and a warm up example were presented to the participants. Then, they provided scored feedbacks for all six queries. When they finished,

formulated feedbacks method was explained to them through an example, and they went through the same queries to provide formulated feedbacks. During the experiment, participants assigned concepts to Q and P sets of (Eq. 4.2), and orally explained how they computed the scores.

In the same way as previous experiment, formulated (Fr) and scored (S) feedbacks are compared based on variance, stability, and required principal components. The null hypothesis of H_0: Between-subject variances are higher for formulated feedbacks than scored feedbacks than scored feedbacks was tested for each query with a sample size of 100 images. F-tests were used to compare the level of difference between variances. Degree of freedom is 499 for formulated and scored feedbacks, $F_{2.5\%}$=0.83, $F_{5\%}$=0.86, and $F_{10\%}$=0.89. Results in Table 4.6 shows than H_0 is rejected for all queries, and it can be claimed that formulated feedbacks demonstrated a lower between-subjects variance than scored feedbacks.

Table 4.6 Analysis of between-subjects variances

	Q1		**Q2**		**Q3**		**Q4**		**Q5**	
	Fr	**S**	**Fr**	**S**	**Fr**	**S**	**Fr**	**S**	**Fr**	**S**
Average of Variances	0.10	0.17	0.09	0.14	0.10	0.16	0.14	0.18	0.11	0.16
F stat	0.62		0.74		0.66		0.80		0.70	
H0 (5% and 10%)	Rejected		Rejected		Rejected		Rejected		Rejected	

Figure 4.12 shows that when users were instructed how to provide feedbacks in a formulated format, an average of only three people were enough to collect data for a query because the variance remains constant. This characteristic of formulated feedbacks is used in the next section to generate test datasets for complex (multi-concept) queries.

The transactions history matrix for this experiment includes 386 images. Using Principal Component Analysis in (Eq. 4.1), the eight most important components represent 92% of data

when either scored or formulated feedbacks were used. Therefore, both methods provided similar level of efficiency regarding the dimension reduction process of search history matrix.

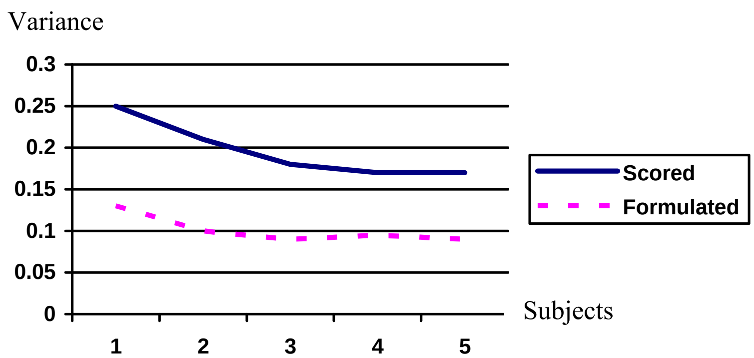


Figure 4.12 Stability of feedbacks

4.2 IMAGES

Complexity in image databases may refer to query complexity or image complexity. Query complexity is defined based on the distribution of target images in the database according to the query image; on the other hand, image complexity is defined based on the individual characteristics of an image. Query complexity factors are introduced in [36] as sparsity, isolation, and diversity of images in the database.

Image complexity can be defined based on image features (visual) complexity that refers to the variations in image features. For example, a picture of a forest during the fall season is visually more complex than a picture of a clear sky. There are many different metrics proposed to measure the visual complexity of an image [30].

Three levels of image complexity are introduced in [30] as visual (Level 1), logical (Level 2), and abstract (Level 3) complexities. Image features play a more significant role in Level 1 and 2 than Level 3 queries. At Level 2, object recognition and inference about the image content are required. Level 3 query processing requires detailed image understanding and reasoning about the objects in the picture and their relations.

Table 4.7 Image categories used for creating transactions by users
User feedbacks are collected for only highlighted concepts

1	Ancient	11	Castle	21	Flower	31	Italy	41	Paris
2	Animal	12	Caves	22	Food	32	Kids	42	People
3	Arizona	13	Cityscape	23	Forest	33	Latin America	43	Recreation
4	Asian	14	Construction	24	Gem	34	Lighthouse	44	Religion
5	Automobile	15	Desert	25	Golf	35	Market	45	Road
6	Ballet	16	Dolphins	26	Horserace	36	Memorial	46	Rock
7	Beach	17	Downtown	27	Household	37	Military	47	Rodeo
8	Boat	18	Earth	28	India	38	Mountain	48	Sea
9	Building	19	Entertainment	29	Industry	39	New Orleans	49	Sky
10	Carnival	20	Farm	30	Insects	40	New York	50	Statue

An image set of 5411 images is collected from Corel Images for the experiments of the developed CBIR system. Images are selected in such a way that roughly fall to one or many classes of Table 4.7. There are examples in the following sections that show that selected images represent all three levels of complexity. Image features vary significantly between images (Level 1), there are multi-class images (Level 2), and there are images related to abstract concepts (Level 3) associated with locations such as New York, New Orleans, etc.

4.2.1 Image Features

As the image collection consists of general images with different topics, texture features are not proper in finding image similarities (Section 2.1.1). Therefore, only color features including 64-bin RGB, 32-bin HSV, and 32-bin YIQ histograms are extracted as the image.

4.2.2 Data Collection

Data collection includes training and test data in the form of transactions. A transaction is a set of feedbacks corresponding to a specific query. Most research experiments for CBIR systems use images with low-level of complexity, and apply automatic relevance feedback based on a binary pre-categorization. In the following experiments; however, real users participated to

provide feedbacks for images with high level of complexity and generate data for the experiments.

To collect data, a query was presented to the subjects using an image and a keyword together, and the user provided formulated feedbacks (Section 4.1.3) for a set of 100 images displayed on the screen through the same user interface utilized in scored feedbacks experiments. The image set for a query includes 20 images selected by the researcher, and 80 images randomly selected by the system from a set of images pre-categorized by Corel Images. Queries were non-abstract single-class concepts, which are highlighted in Table 4.7. A total of 80 transactions were collected including three to four transactions related to queries with higher level of complexity such as Ancient, Memorial or Religion, and one or two transactions related to less complex queries such as Beach or Golf.

There are other types of classes in Table 4.7, such as Animal or Flower, that low user subjectivity is expected in labeling images in those categories because those images are only related to one single concept. Therefore, automatic feedbacks in binary format were used to create transactions for queries related to such classes. In addition, the pre-categorization of Corel Images is used to create binary feedbacks for queries such as Arizona, Paris, or Latin America (20 transactions). The transactions collected during previous experiments of testing binary, scored, and formulated feedbacks are also added to the above data collection (264 transactions from the first experiment, and 50 from the second one). Therefore, 414 transactions are available in the dataset. The sources of collected data are summarized in Table 4.8.

A total of 42 students with the average age of 23 participated in different phases of data collection. There were 28 undergraduate students, and 14 graduates; 38 Engineering and

Computer Science students, two Chemistry, and two Business majors. Participants were familiar with computers, and used internet search engines at least once in a week for the last two years.

Table 4.8 Source of data collected from users

Source of data collection	Number of transactions	Type of feedbacks
Experiment in Section 4.1.1	132	Binary
Experiment in Section 4.1.1	132	Scored
Experiment in Section 4.1.3	25	Scored
Experiment in Section 4.1.3	25	Formulated
Additional user feedbacks	80	Formulated
Pre-categorized images	20	Binary

4.2.3 Performance Criteria

In information retrieval systems, including image retrieval systems, two criteria of precision and recall are widely used. Precision is the ratio of relevant images returned by the system to the total number of retrieved images. Recall is the percentage of relevant images returned by the system to the total number of available relevant images. As the feedbacks are provided in scored format in the next experiments, the following formula is used to compute precision:

$$\frac{\Sigma score_i\text{, where i is the image retrieved by the system}}{\text{Total number of retrieved images}}$$

In the experiments, the feedback size, which is the number of images displayed to the user in each iteration, is set to 20 images, and the precision is measured at the end of the 5^{th} iteration because the number of labeled images in each transaction of the test data is 100.

4.2.4 Training and Test Dataset

The system was trained by transactions of one-concept queries. A balanced set of transactions was selected as training set that includes three transactions for each single concept.

The performance of the retrieval system is measured for different levels of query complexity, which is defined based on the number of concepts included in the query. Thus, one-concept, two-concept, and three-concept queries are used to test the system. Test data is available for one-concept queries using the feedbacks in the collected data. For two-concept query, the only available user data is five queries with a total of 50 transactions from the experiment in Section 4.1.3, and there are no users' feedbacks available for three-concept queries. Therefore, it was decided to synthesize feedbacks for multi-concept queries. Users' feedbacks for single-concept queries and formulated feedbacks method are used for synthesizing feedbacks. In the following, it is explained how a synthesized feedback is generated.

Figure 4.13 An example of a two-concept query

Assume that the system is running a two-concept query of farm and vehicle, and a feedback is needed to be synthesized for above image. If the user's scores for above image are noted by Score(farm) and Score(vehicle) when queried for farm and vehicle, a synthesized feedback for this two-concept query is assigned to the above image by:

$$\text{Synthesized\_Score}(\text{farm} + \text{vehicle}) = \max \{1, \text{Score}(\text{farm}) + \text{Score}(\text{vehicle})\}$$

The preference is to collect Score(farm) and Score(vehicle) from the same user's feedbacks; however, there is a high chance that the image is not rated for both single-concept queries of farm and vehicle by the same user. In that case, Score(farm) and Score(vehicle) are collected from different users' feedbacks.

4.3 LONG-TERM LEARNING METHOD

The long-term learning and retrieval algorithm (LTL) is tested in this section, and compared to Support Vector Machines (SVM, Section 2.5), a widely used method in image retrieval systems. A balanced set of transactions is used to train the system. Such a dataset contains three transactions for each concept in Table 4.7. The results are categorized for one-concept (Table 4.9), two-concept (Table 4.10), and three-concept (Table 4.11) queries.

Table 4.9 Precision for long-term learning algorithm (LTL) and SVM with one-concept queries

	Concept	LTL	SVM		Concept	LTL	SVM		Concept	LTL	SVM
1	Ancient	0.43	0.33	**18**	Earth	0.65	0.77	**35**	Market	0.79	0.75
2	Animal	0.60	0.68	**19**	Entertainment	0.63	0.58	**36**	Memorial	0.62	0.46
3	Arizona	0.67	0.65	**20**	Farm	0.81	0.76	**37**	Military	0.72	0.73
4	Asian	0.55	0.51	**21**	Flower	0.80	0.83	**38**	Mountain	0.74	0.58
5	Automobile	0.65	0.53	**22**	Food	0.61	0.83	**39**	New Orleans	0.76	0.79
6	Ballet	0.60	0.62	**23**	Forest	0.63	0.55	**40**	New York	0.72	0.78
7	Beach	0.78	0.70	**24**	Gem	0.71	0.77	**41**	Paris	0.53	0.56
8	Boat	0.72	0.68	**25**	Golf	0.81	0.76	**42**	People	0.71	0.66
9	Building	0.76	0.63	**26**	Horserace	0.81	0.82	**43**	Recreation	0.69	0.63
10	Carnival	0.79	0.72	**27**	Household	0.81	0.78	**44**	Religion	0.75	0.52
11	Castle	0.63	0.65	**28**	India	0.44	0.51	**45**	Road	0.58	0.60
12	Caves	0.60	0.73	**29**	Industry	0.81	0.76	**46**	Rock	0.75	0.76
13	Cityscape	0.72	0.62	**30**	Insects	0.81	0.76	**47**	Rodeo	0.70	0.72
14	Construction	0.75	0.69	**31**	Italy	0.58	0.55	**48**	Sea	0.66	0.46
15	Desert	0.66	0.67	**32**	Kids	0.66	0.72	**49**	Sky	0.58	0.64
16	Dolphins	0.81	0.86	**33**	Latin America	0.66	0.59	**50**	Statue	0.76	0.53
17	Downtown	0.68	0.53	**34**	Lighthouse	0.81	0.85				

SVM is designed to work with binary labels. In the dataset, few transactions with binary labels are available; therefore, the scored feedbacks are required to be converted to binary when the system is trained with SVM. It seems that 0.5 is a reasonable cut-off threshold, that is

feedbacks with a score larger than 0.5 to be set to one, and those less than 0.5 to zero. As the number of positive examples is important in training a SVM classifier [115], it was decided to set the cut-off to 0.4 to include more positive examples in the training set. In transaction with users binary feedbacks, the original users feedbacks were used.

Moreover, support vector machines are binary classifiers. In the experiment, a set of binary SVM's are trained for each concept based on the method introduced in [115] for multiple classification. A SVM classifier for a concept *c*, classifies transactions as "related to concept *c*" or "not related to concept *c*". To classify a test transaction, the concept with lowest SVM error was selected.

Table 4.10 Precision for long-term learning algorithm (LTL) and SVM with two-concept queries

	Concept 1	Concept 2	LTL	SVM		Concept 1	Concept 2	LTL	SVM
1	Ancient	Statue	0.53	0.45	**17**	Building	Paris	0.56	0.43
2	Asian	Building	0.56	0.48	**18**	Building	Religion	0.55	0.49
3	Arizona	Mountain	0.63	0.56	**19**	Construction	People	0.48	0.44
4	Automobile	Farm	0.71	0.63	**20**	Forest	Road	0.46	0.47
5	Automobile	Road	0.52	0.55	**21**	India	People	0.43	0.42
6	Automobile	City	0.48	0.41	**22**	India	Religion	0.56	0.47
7	Automobile	Construction	0.59	0.42	**23**	Italy	Statue	0.54	0.42
8	Beach	Mountain	0.57	0.46	**24**	Italy	Religion	0.61	0.42
9	Beach	Building	0.54	0.43	**25**	Latin America	Religion	0.60	0.41
10	Beach	People	0.46	0.38	**26**	Market	People	0.70	0.57
11	Boat	Military	0.51	0.52	**27**	Memorial	Building	0.62	0.46
12	Building	India	0.52	0.38	**28**	Memorial	Paris	0.54	0.42
13	Building	Italy	0.55	0.39	**29**	Mountain	City	0.61	0.41
14	Building	Latin	0.48	0.32	**30**	Mountain	Forest	0.59	0.56
15	Building	New Orleans	0.45	0.30	**31**	People	Recreation	0.48	0.47
16	Building	New York	0.53	0.38	**32**	Paris	Statue	0.56	0.44

A critical step in LTL is the choice of number of principal components in the Factor Analysis model. In principal component analysis, the percentage of original data that is explained by the top-k important principal components can be estimated by:

$$p = \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{n} \lambda_i$$

where λ_i is the i^{th} largest eigenvalue of the data matrix, and n is total number of principal components. Therefore, a criterion can be selected to limit the number of useful principal components to the k largest eigenvalues. In the experiment, a criteria was set to stop adding k when adding another principal component leads to a less than 10% increase in p. The algorithm selected 47 principal components.

Results show that image complexity and query complexity play significant role on the performance of SVM. When the query has low-level of complexity, i.e. one-concept query, SVM performs slightly better than LTL because each single concept has its own SVM classifier, which is built based on one-concept queries. If the concept is related to non-complex images, such as cave, food, or earth, SVM performs better than LTL because binary feedbacks are suitable for such type of images, and SVM are originally designed for classifying data with binary labels.

The retrieval performances of multi-concept queries are computed for queries that correspond to at least 50 images in the database, and some of the results are shown in Tables 4.10 and 4.11. In two-concept query classifications, concepts related to the two SVMs with lowest error are selected. Similarly, the three concepts, corresponding three SVMs with lowest errors, were selected for three-concept classification.

Comparing LTL and SVM results for complex queries, i.e. two-concept and three-concept queries, shows that LTL is more efficient in capturing high-level queries. When the query is complex, relevant images are spread out in the multi-dimension space of an SVM

classifier, which is trained for a single concept; therefore, the classifier can not build strong support vectors. On the other hand, the approach of semantic space in LTL reveals the underlining structure of concepts by the means of the principal components of the search history matrix; and the relation between semantic classes (concepts) and images are computed by the scores matrix in the Factor Analysis model. Therefore, the model trained based on single-concept queries can be used for classifying multi-concept queries.

Table 4.11 Precision for long-term learning algorithm (LTL) and SVM with three-concept queries

	Concept 1	Concept 2	Concept 3	LTL	SVM
1	Asian	Building	Forest	0.23	0.20
2	India	Religion	Statue	0.34	0.26
3	Italy	Religion	Statue	0.35	0.25
4	Building	Italy	Religion	0.36	0.21
5	Forest	Latin	Mountain	0.28	0.21

Presented results in this section are based on the training an image retrieval system with a balanced set of transactions. However, this assumption is not valid in real world problems. To study the performance of the retrieval algorithm under a generalized condition, the system was trained with all the available transactions. A large portion of the transactions (264 out of 414) is related to a limited number of semantic classes (boat, statue, memorial, people, automobile, city);therefore, the system detected only 35 semantic classes because the number of classes is computed based on the principal components, which are related to the distribution of images in the history matrix. Results showed that the precision of the system decreased 22% in one-class queries, 26% in two-class queries, and 34% in three-class queries. Therefore, the performance of the long-term learning algorithm is sensitive to the distribution of the images and concepts in the database because it applies a statistical framework to create the semantic space.

4.4 SHORT-TERM LEARNING

In this section, short-term learning algorithm, which is based on the image features and relevance feedbacks of the current transaction, is compared to Biased Linear Discriminant Analysis (BDA, Section 2.3). Precision is used as the performance criteria, and the test dataset includes all the transactions.

For each transaction, all images with the score of one are considered relevant, and one image is randomly selected from relevant images as the image query. Image features are extracted for the image query, and the short-term learning (STL) is applied to find the similar images to the query. If the score for a returned image is not available in the transaction, which is not unlikely, the score of the image is read from another similar transaction. Results are reported for the precision at the end of 5^{th} iteration.

In the short-term learning algorithm, a neighborhood parameter needs to be set (Section 3.2). As image features are normalized and the total similarity between two images, TS_{ij}, has a range of zero to one, the neighborhood parameter ε changes between zero and one. If $\varepsilon = 1$, the neighborhood of a data point includes all images in the space and the short-term algorithm converts to a BDA model with non-binary labels. If $\varepsilon = 0$, a data point has no similar points in the space but itself; therefore, it would not be possible to compute similarities between images. In the experiments, ε is set to 0.5.

Tables 4.12 and 4.13 show the results for the short-term learning (STL). As images in the database present a wide range of concepts, including abstract concepts, with different degrees of visual complexity, few categories can be represented only by their image features. Therefore, queries with poor results (less than 20%) are not shown in the tables. Results for three-concept

queries based on image features were not promising and not reported. For one-class queries, STL and BDA performed very similar, so only STL results are reported in Table 4.12.

Table 4.12 Precision for short-term learning (STL) algorithm with one-concept queries

	Concept	STL		Concept	STL
1	Arizona	0.35	**7**	Earth	0.83
2	Ballet	0.42	**8**	Flower	0.58
3	Beach	0.43	**9**	Forest	0.73
4	Cave	0.72	**10**	Gem	0.65
5	Desert	0.46	**11**	Rock	0.55
6	Dolphin	0.64	**12**	Sky	0.83

STL is a special case of BDA (Section 3.4); therefore, STL is converted to BDA when the query has low visual complexity, such as one-class non-abstract queries reported in Table 4.12. In two-concept queries, relevant images may create disjoint clusters in the feature space. BDA considers a global average of image features; however, the average does not necessarily represent a relevant feature value for the query. On the other hand, STL computes the average of image features only for the neighborhoods (disjoint clusters); therefore, it preserves the range the relevant features values in each neighborhood.

Table 4.13 Precision for short-term learning algorithm with two-concept queries

	Concept 1	Concept 2	STL	BDA
1	Beach	Forest	0.46	0.42
2	Beach	Mountain	0.44	0.45
3	Boat	Military	0.38	0.26
4	Forest	Mountain	0.42	0.40

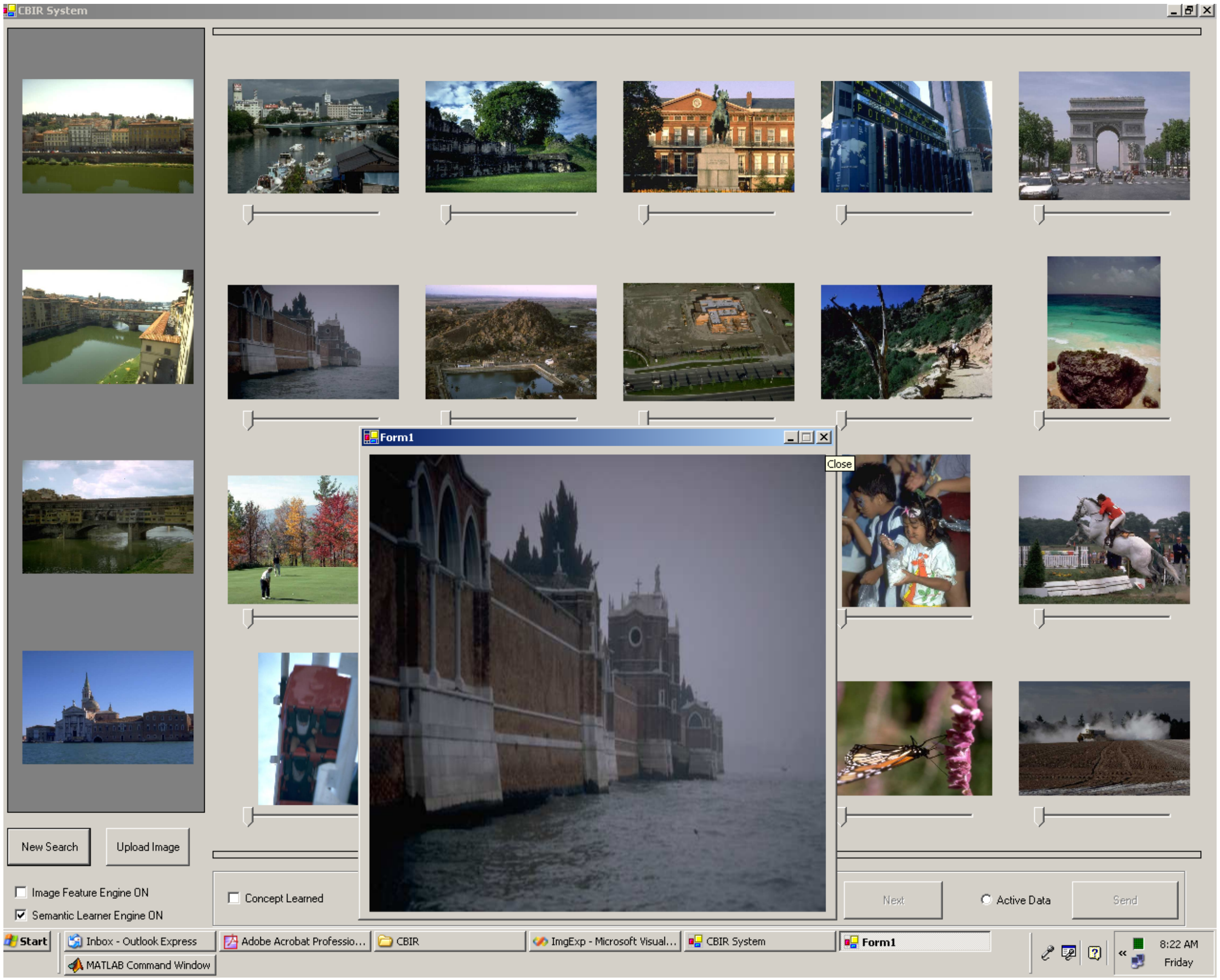


Figure 4.14 Image retrieval interface

4.5 IMAGE RETRIEVAL SYSTEM

Figure 4.14 shows the interface of the designed CBIR system. To start, the user has the option of New Search to brows images in the database and builds up her query, or Upload Image to upload her own image query. The user can activate/deactivate short-term and long-term learning algorithms by selecting Image Feature or Semantic Learner engines. The weight of long-term learning, λ, is initialized to zero, one, or 0.5, when only Image Feature engine is on, only Semantic Learner engine is on, or both engines are on. λ is being updated during the retrieval procedure by (Eq. 3.54).

Figure 4.15 shows the algorithm flowchart for the image retrieval system with relevance feedback mechanism. In each iteration, the system collects the index of scored images (*RFedImages*) along with the assigned scores (*RF*s). The user can set the variable *concept*, which is set to False by default, to True to make the system finish learning phase and start retrieval phase. The mechanism of retrieval phases (Section 3.5) is studied in more detail in Section 4.6.

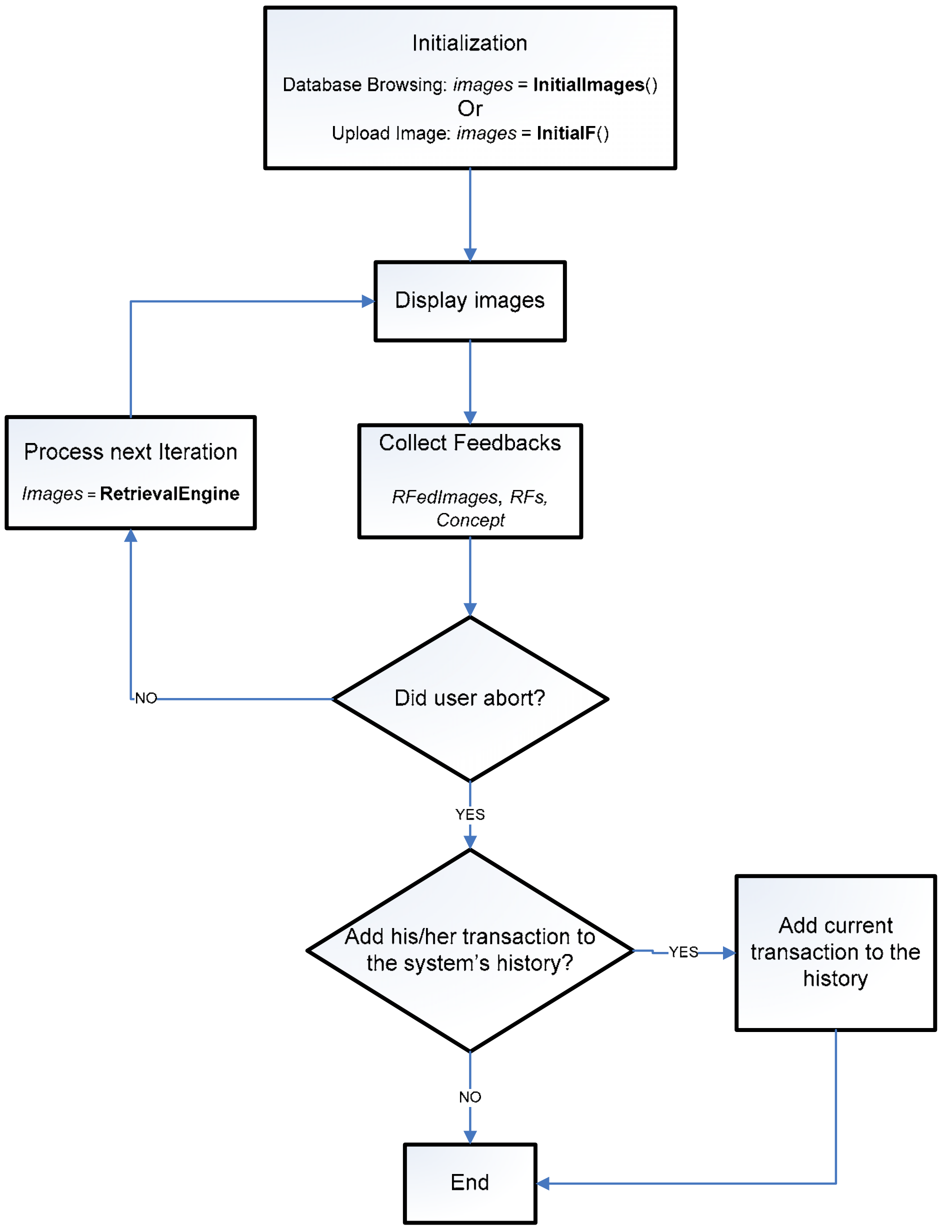


Figure 4.15 Image retrieval system with relevance feedback

4.5.1 Image Initialization with Uploading an Image

If the user uploads her own query image, Image Feature Engine is automatically activated. Image features are extracted for the uploaded image. Similar images, found based on image features similarities (Eq. 3.8), are returned to the user if only Image Features engine is on. If both search engines are on, similar images are sorted by their levels of ambiguity (Eq. 3.15), and then returned to the user.

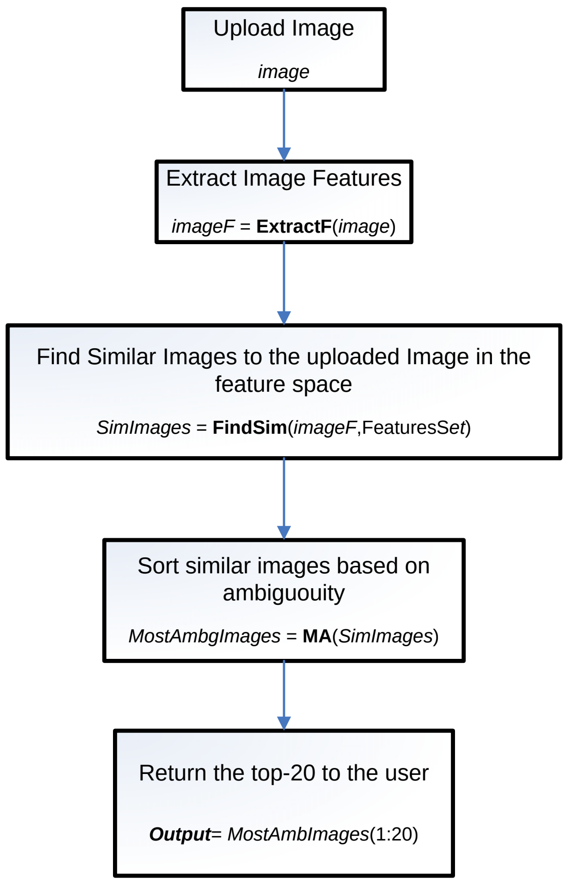


Figure 4.16 Initialization: short-term learning

4.5.2 Image Initialization without Uploading an Image

The InititialImages() function creates an initial set of images if the user chooses to brows the images. First, all images in the database are sorted based on their levels of ambiguity (Eq. 3.15). Then, the system starts returning the most ambiguous images. To avoid returning repeated

concepts to the user, and display a wide range of concepts on the screen, all images similar to a displayed ambiguous image are omitted from the list.

The algorithm starts with a large neighborhood radius to detect and delete similar images from the list. The algorithm reduces the radius to leave more images in the list if the total of returned image is less than the feedbacks size. The user can click on the image to better view an image and understand the relation of that image to the other images in the database before providing any feedbacks. Once clicked, the image is displayed in a larger view on the screen, and highly related images, based on the long-term learning, are presented on the left side of the screen.

Table 4.14 Precision for long-term (LTL) and short-term (STL) learning algorithm with one-concept queries

	Concept	λ	LTL+STL		Concept	λ	LTL+STL
1	Arizona	1.0	0.69	**7**	Earth	1.0	0.75
2	Ballet	1.0	0.60	**8**	Flower	1.0	0.80
3	Beach	0.75	0.80	**9**	Forest	0.8	0.74
4	Cave	1.0	0.62	**10**	Gem	1.0	0.71
5	Desert	0.65	0.68	**11**	Rock	1.0	0.78
6	Dolphin	1.0	0.81	**12**	Sky	0.3	0.85

To test the system with both long-term and short-tem algorithms working jointly together, an experiment was set up similar to previous experiments. In an off-line process, the long-term algorithm created the semantic space and found the relations of each labeled image in the search history to the detected semantic classes. For each test data (transaction), an image with score of one is selected as a perfect match for the query concept in that transaction. Both search engines are considered on, and the initialization process, explained in Section 4.5.1 and by Figure 4.16, returns the first set of images to the user. Relevancy scores for the returned images

are read from the same transaction; however, if there is no score available for an image, the score is read from a transaction with the same query concept.

Long-term and short-term retrieval algorithms are run to find relevant images for the next iteration. Final output of the system in an iteration is computed based on the updated importance factor long-term learning (λ), and short-term learning algorithms ($1-\lambda$). Performance of the system is computed at the end of the 5th iteration. Results for one-concept and two-concept queries are reported in Tables 4.14 and 4.15.

It was revealed in the previous experiment (STL) that short-term algorithm demonstrated poor performance on retrieving many concepts because it was not able to capture those concepts in the database by the application of image features. Therefore, LTL+STL is expected to improve the retrieval performance of query concepts with promising results of STL.

Table 4.15 Precision for long-term (LTL) and short-term (STL) learning algorithm with two-concept queries

	Concept 1	Concept 2	λ	LTL+STL
1	Beach	Forest	0.40	0.66
2	Beach	Mountain	0.85	0.65
3	Boat	Military	1.0	0.57
4	Forest	Mountain	0.85	0.63

Results show that running both learning algorithms together improves the retrieval performance in many categories. The high importance weight of long-term learning algorithm (λ) in Table 4.14 shows that the system uses the background knowledge in final iterations.

To study the behavior of λ in different iterations, the averages of λ and precision are computed for one-class queries shown in Table 4.14. The comparison of the system performance for LTL, and LTL+STL shows that adding the image feature learning algorithm to the long-term

learning mechanism helps the system to learn the query concept faster (Table 4.16). However, after the query concept is captured, the system relies on the background knowledge to find similar images to the query concept.

Table 4.16 Precision (P) for long-term and short-term learning algorithm applied to one-class queries

Iteration	**1**		**2**		**3**		**4**		**5**	
Average	λ	**P**	λ	**P**	λ	**P**	λ	**P**	λ	**P**
LTL	1	0.8	1	0.28	1	0.42	1	0.53	1	0.63
LTL+STL	0.50	0.12	0.55	0.40	0.62	0.56	0.78	0.68	0.86	0.73

4.6 APPLICATION OF RETRIEVAL PHASES

In this section, the effect of retrieval phases on the image retrieval performance is studied. Figure 4.17 shows the structure of the developed system with retrieval phases. After a user provides feedbacks, set of labeled images (*RFedImages*), scores (*RF*s), and variable *concept*, which controls the start/end of retrieval phases, are sent to the retrieval system. The system runs STL and LTL algorithms, updates weight factor λ, and adjusts the set of similar images for LST+STL.

The variable *concept* is set to True when the user indicates that the query concept is learnt (Section 3.4); therefore, the system returns all similar images found by LTL+STL. Otherwise, the output of the system includes three query centers and their importance factors (Eq. 3.51), seven similar images from LTL+STS output, sorted based on their ambiguity (Eq. 3.55), and ten Most Ambiguous (MA) images (Eq. 3.16). If the user assigns no scores to the images in the current transaction, the system initializes a new set of images by InitialeImages() function.

Two sets of tests are preformed, assuming the user finishes learning phase at the end of the second and third iterations. The results are shown in Figure 4.18.

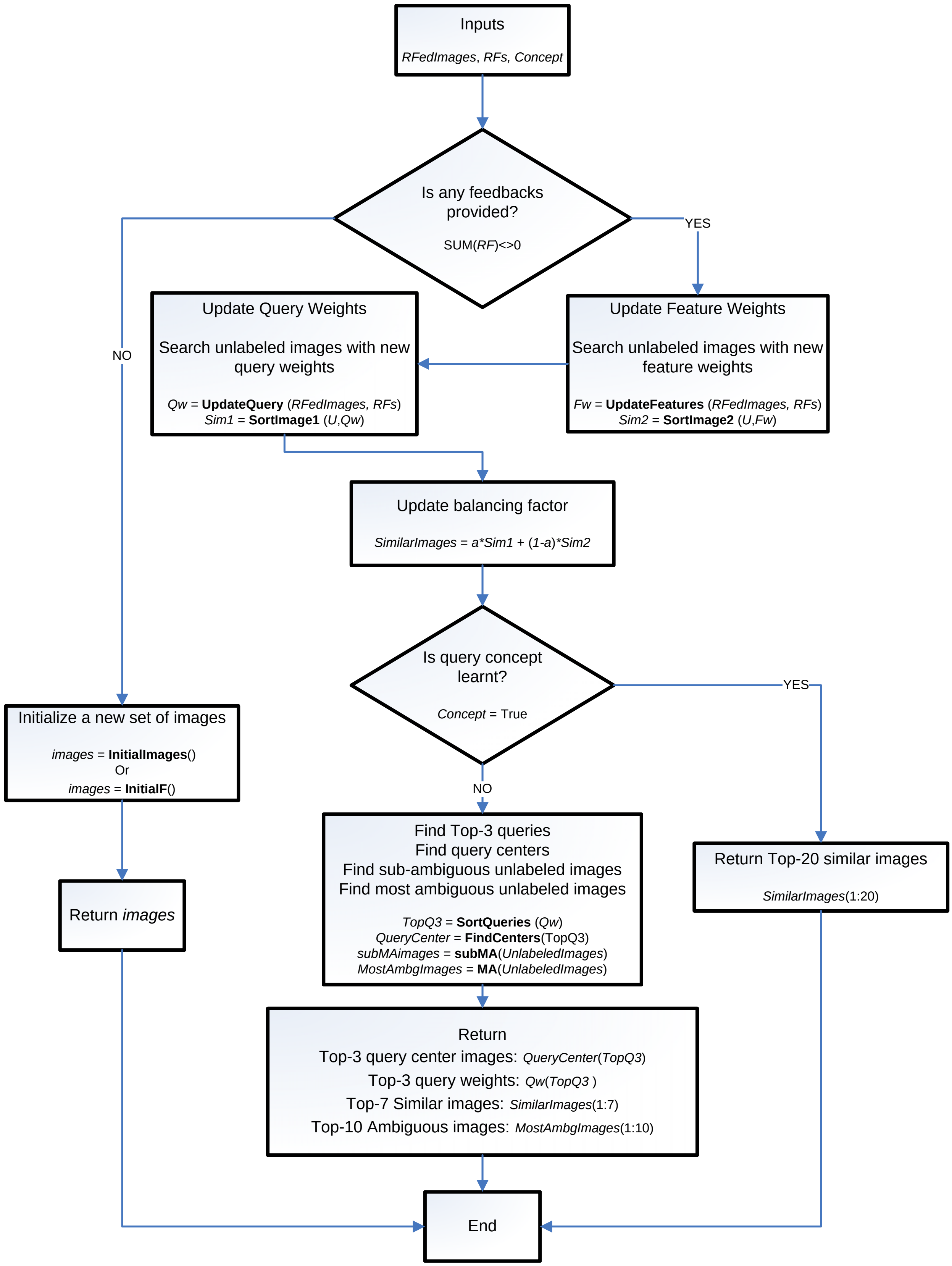


Figure 4.17 Image retrieval system with strategy of retrieval phases

As it is shown, the performance was reduced in all levels of query complexity when a learning phase is included in the retrieval process. However, when the query complexity increases from single to multiple concepts, the retrieval performance shows a trend of improvement with the learning phase applied.

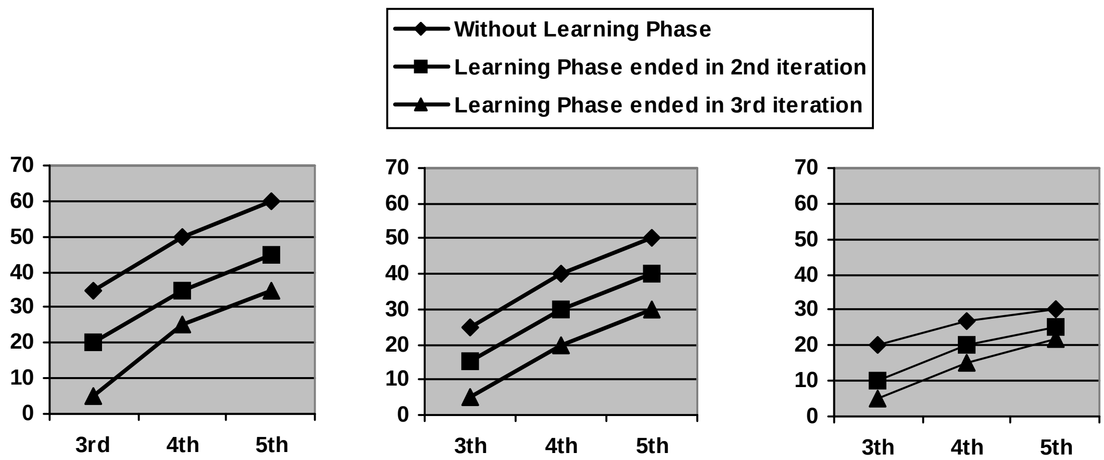


Figure 4.18 Image retrieval performance with retrieval phases
Query complexity: (a) One-class (b) Two-class, (c) Three-class

In the case of one-class queries, there is a high chance that a relevant image shows up in the first iterations, especially when the number of concepts in the database is low relative to the feedback size (number of images returned to the user in each iteration). Therefore, when the system continues the learning phase until the second or third iterations, MA (Most Ambiguous) images keep showing up although they may not be relevant to the query. This procedure is demonstrated in Figure 4.17, where variable *concept* has a False value. On the other hand, when the queries are more complex (multi-class queries), there is more need for MA images to display more concepts to the user and help her properly build her query. Therefore, continuing of learning phase, in which MA images are returns, improves query modeling and retrieval performance.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This dissertation work concentrated on three areas in a CBIR system: input (feedbacks and users behaviors), process (learning and retrieval algorithm), and output (strategy of retrieval phases). The shortcomings of the existing image retrieval systems in each of the above areas were detected, a solution was proposed, and test results were analyzed. In this chapter, a summary of this study is reported, and recommendations for the future work are presented.

5.1 USERS FEEDBACKS

Relevance feedbacks in CBIR systems are used to integrate users' perceptions of images to the retrieval mechanism. Therefore, it is essential to provide users with efficient tools that able them to accurately transfer the image queries they have in their minds to the system.

Query complexity and image complexity highly influence the performance of CBIR systems. In this work, the complexity of a query/image is defined based on the number of concepts that appear in the query/image. Experiments proved that binary feedbacks are inefficient in capturing concepts for complex images as they showed high between-subjects and between-sessions variances (Section 4.1). Moreover, results suggested that more subjects/data are required to train an image retrieval system by binary feedbacks. On the other hand, scored feedbacks provided lower variance; thus, fewer subjects/data can be used to train the system.

Binary feedback also proved to be a weak tool for labeling complex images. Experiments showed that binary feedbacks are noisy in comparison with scored feedbacks because they do not discriminate between images with different levels of relevancy to the query concept.

Although most of the users preferred scored to binary feedbacks for labeling complex images, they pointed out their confusions in assigning accurate scores. Therefore, a feedback

methodology was proposed to guide users in providing scored feedbacks. Scored feedbacks in the new methodology are called formulated feedbacks, which are computed based on the number of concepts appeared in the query and image. Formulated feedback does not guarantee that different subjects provide similar scores for an image; however, it suggests all users the same scoring criterion. Results showed that formulated feedbacks improved the quality of users inputs.

Another drawback of many image retrieval systems is the inefficient user and system interaction. Users usually have a vague idea about how their feedbacks affect the learning and retrieval mechanism of the system. In the developed system, the user has the option of clicking on an image to view the top-4 related images found by the system. Therefore, the user finds out about the relations between images she is labeling and unlabeled images in the database before sending her feedbacks.

5.2 LEARNING AND RETRIEVAL ALGORITHMS

The heart of a CBIR system is the learning and retrieval engine. In this dissertation, two algorithms were developed for an image retrieval system with multi-class images and non-binary feedbacks to support multi-concept queries. An algorithm was designed based on the long-term approach to build a model based on data available from prior search results. The second algorithm was designed to capture a user's query perception based on the image features and scored images. Each algorithm was tested and compared to similar approaches in the literature. Finally, both algorithms were put together to improve the performance of the developed CBIR system.

The long-term learning algorithm was designed based on the semantic space concept. A semantic space represents the relations between images and semantic classes. A factor analysis model was used to find the semantic space from the search history matrix. To solve the model,

principal component analysis was used. The loading matrix in the model shows the relations between transactions and semantic classes, and the scores matrix represents the relationship between semantic classes and images. Scores matrix is computed after the loading matrix is rotated based on varimax criterion.

Different subsets of the available transactions were used to train and test the system to study the performance of the algorithm. The algorithm outperformed support vector machines method that is widely used in CBIR systems (Section 4.3). The performance of the developed algorithm was more significant for complex queries. The structure of the semantic space, that finds the relations between images and semantic classes, is the main advantage in retrieving complex queries.

The second algorithm was designed to find similar images to the query based on their image feature similarities. A linear discriminant model was developed to map data points (images) to a new space in such a way that semantically similar images be located close together, far from irrelevant images. The semantic similarities between images are measured by user's feedbacks. The method is called short-term image learning and retrieval because it utilizes the feedbacks from only the on-going transaction.

Semantically similar images are mapped to close locations by Linear Discriminant Analysis method; however, when similar images are located in distant points in the image features space, LDA does not perform well. When images have Level 1 complexity (visual), semantically similar images may not be located close to each other in the feature space. Therefore, the developed algorithm builds a neighborhood around images that have non-zero scores, and maps only similar images in the neighborhood.

The performance of the developed short-term learning algorithm was compared to the BDA method (Section 4.4). Results were similar for one-concept queries; however, when the complexity of the query increased, the developed algorithm outperformed BDA. As the database included images with high levels of complexity, the performance of the short-term learning algorithm was acceptable on only some queries. As expected, a feature-based image retrieval algorithm was inefficient in retrieving queries related to abstract concept such as Asian. The developed short-term learning algorithm improved the performance of the system on queries with low-level of complexity when jointly activated with the long-term learning algorithm.

5.3 STRATEGY OF RETRIEVAL PHASES

The last area of focus was the strategy of the retrieval phases. In image retrieval systems with relevance feedback, it is important to create an accurate query model in the first retrieval iterations. If images are not labeled precisely, the system assigns loose weights to the query concepts, and more iterations would be required to revise those weights. In current image retrieval methods, when the user provides feedbacks, the system concentrates only on concepts with high weights, ignoring other concepts. Therefore, only highly rated concepts show up in the next iteration. In this way, the user is limited to a narrow set of concepts that are repeatedly displayed on the screen, and the user's ability in defining a precise query model is negatively influenced.

In this study, two phases, learning and retrieval, are defined for an image retrieval process of a CBIR system with relevance feedbacks. In query learning phase, the system allocates separated areas on the screen to Most Positive and Most Ambiguous images. Most positive images are displayed to the user to show her the results of query learning at the current stage. Most Ambiguous images are displayed to expose the user to different concepts available in

the database and help her define the query accurately. In each iteration, the user looks at the Most Positive images and when she recognizes that the system has captured the query concept, she informs the system to switch to the retrieval phase. In retrieval phase, the system only returns Most Positive images.

The effect of applying retrieval strategy to a CBIR system was studied for queries with different levels of complexity (Section 4.6). Experiment results did not show any improvement on applying retrieval strategies; however, comparing the results for one-concept, two-concept, and three-concept queries showed that when the query complexity increases, the performance drops more sharply without the retrieval strategy. Therefore, it seems that the strategy of retrieval phases is helpful for complex queries. As there are few cases of queries with more than three concepts in real world problems, this hypothesis needs to be investigated for a database with a larger number of semantic classes.

5.4 RECOMMENDATIONS FOR FUTURE WORK

The relations between users and CBIR systems can be studied further in Human Factors and Human Computer Interaction areas. Incentive approaches may encourage users to provide more accurate feedbacks. Tracing the user's eye movement may reveal hot spots on the screen, so Most Ambiguous images can be displayed in those areas as such images need more attention. Adoptive interfaces observe user's actions and revise the user interface. Such mechanisms can be applied to CBIR systems to find proper areas on the screen to display Most Positive or Most Ambiguous images. In general, a desired user interface in CBIR systems efficiently interacts with the user, provides user with some information about the progress of on-going processes, and encourages the user to provide accurate feedbacks.

The developed long-term learning and retrieval algorithm demonstrated high precision, and outperformed the widely used method of support vector machines. However, some issues were found during the experiment. Experiments showed that the distribution of transactions used in the training affects the performance of the algorithm. Therefore, a mechanism is needed to create a balanced training dataset that includes equal number of transactions for each concept in the search history. Another approach is to find the optimum number of principal components through statistical methods for the case of a search history matrix with unbalanced transactions.

The nature of search history matrix in CBIR systems is similar to high volume data that is collected in recommendations systems, adaptive interfaces, and web mining technologies; therefore, it is suggested to review recent research in these areas to develop new long-term image learning and retrieval algorithms.

The neighborhood approach in short-term learning algorithm creates disjoint clusters in the image features space. The algorithm does not consider mapping disjoint clusters to close locations in the new space. Therefore, it is appealing to develop a mapping method for disjoint clusters that include data with non-binary labels. Furthermore, a method is required to find the optimum size of neighborhood.

The short-term learning algorithm applies a linear transformation; however, the assumption of linear relations in image features space might not be reasonable. It is suggested to apply and test non-linear transforms to image features data. For example, kernel functions can be used to add non-linearity to a linear model.

Recently, wavelet and Fourier transforms have been applied to large datasets to extract the underlying characteristics of the data. Statistical modeling, data mining methods, and machine learning algorithms are widely applied to the area of image retrieval. A promising

method should support the main two assumptions of the system developed in this dissertation, i.e. multi-class images and queries, and non-binary feedbacks. Moreover, a candidate method should be feasible to be applied to high volume data and high dimensional search history and image features matrixes.

In the developed system, the user is required to indicate to the system to finish the learning phase and start the retrieval phase. It is suggested to develop a mechanism to enable the system to gradually shift from the learning phase to retrieval phase. In that case, the number of Most Ambiguous images, which are usually not related to the query concept, decreases gradually and it would be remarkable to investigate how the performance of the system changes.

In this dissertation work, retrieval model was built based on two sources of data: search history and image features. However, there are usually more data sources available to be added to an image retrieval system. Keywords are very popular by both users and researchers. Image annotation has many limitations, but there are efficient information retrieval and text mining algorithms to search images by keywords. Metadata associated with digital images is another source of data. Recommendation systems are usually equipped with a user profiling system. The same approach can be applied to a CBIR system to retrieve only images that are related to the user's profile.

REFERENCES

[1] Abidi, B.; Mitckes, M.; Abidi, M.;Liang, J. “Grayscale Enhancement Techniques for X-ray Images of Carry-on Luggage”, Proceeding of the 6th International Conference on Quality Control by Artificial Visaion, Vol. 5132, 2003, 579-591.

[2] Akrivas, G., Stamou, G., Kollias, S. “Semantic Association of Multimedia Document Descriptions through Fuzzy Relational Algebra and Fuzzy Reasoning”, IEEE Transactions on Systems, Man, and Cybernetics-Part A, Vol. 34(2), 2004, 190-196.

[3] Al-Sultan, K.; Khan, M. “Computational Experience on Four Algorithms for the Hard Clustering Problem”, Pattern Recognition Letters, Vol. 17(3), 1996, 295-308.

[4] Amadsun, M.; King, R. “Low-level Segmentation of Multi-spectral Images via Agglomerative Clustering of Uniform Neighborhoods”, Pattern Recognition, Vol. 21(3), 1988, 261–268.

[5] Anderson, C.; Domingos, P.; Weld, D. “Adaptive Web Navigation for Wireless Devices”, Proceeding of the 17th International Joint Conference on Artificial Intelligence, 2001.

[6] Anderson, C.; Domingos, P.; Weld, D. “Personalizing Web Sites for Mobile Users”, Proceeding of the 10th International World Wide Web Conference, 2001.

[7] Anderson, C.; Domingos, P.; Weld, D. “Relational Markov Models and their Application to Adaptive Web Navigation”, Proceeding of the 8th ACM Conference on Knowledge Discovery and Data Mining, 2002.

[8] Bezdek, J. Pattern Recognition With Fuzzy Objective Function Algorithms. Plenum Press, 1981, New York, NY.

[9] Billsus, D.; Pazzani, M. “Learning Collaborative Information Filters”, Proceedings of the International Conference on Machine Learning, 1998.

[10] Boley D.; Gini, M.; Gross, R.; Han. H.; Hastings, K.; Karypis, G.; Kumar, V.; Mobasher, B.; Moore, J. “Partitioning-Based Clustering for Web Documents Categorization”, Decision Support Systems, Vol. 27, 1999, 329-341.

[11] Breese, J.; Heckerman, D.; Kadie, C. “Empirical Analysis of Predictive Algorithms for Collaborative Filtering”, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998, 43-52.

[12] Can, F. “Incremental Clustering for Dynamic Information Processing”, ACM Transactions on Information Systems, Vol. 1(2), 1993, 143–164.

[13] Carson, C.; Thomas, M.; Belongie, S.; Hellerstein, J.; Malik, J. "Blobworld: A System for Region-Based Image Indexing Retrieval", Proceeding of the 3rd International Conference on Visual Information Systems, 1999, 509-516.

[14] Castelli, V.; Bergman, L. Image Databases: Search and Retrieval of Digital Imaginary, "Color for Image Retrieval" by Smith J., Wiley, 2002, 285-311.

[15] Castelli, V.; Bergman, L. Image Databases: Search and Retrieval of Digital Imaginary, "Shape Representation for Image Retrieval" by Kimia B., Wiley, 2002, 345-358.

[16] Chang, E.; Tong, S. "SVM Active – Support Vector Machine Active Learning for Image Retrieval", Proceedings of the 9th ACM international conference on Multimedia, 2001, 107-118.

[17] Chee, S.; Wang, K. "RecTree: An Efficient Collaborative Filtering Method", Proceeding of International Conference Data Warehouse and Knowledge Discovery, 2001.

[18] Chen, Y; Wang, J. "A Region-Based Fuzzy Feature Matching Approach to Content-Based Image Retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24(9), 2002, 1252-1267.

[19] Chen, Y.; Zhou, X.; Houng, T. "One-Class SVM for Learning in Image Retrieval", Proceeding of IEEE Conference Computer Vision and Pattern Recognition, 2001.

[20] Chen. Y.; Shahabi, C. "Yoda, An Adaptive Soft Classification Model: Content-Based Similarity Queries and Beyond", Multimedia Systems, Springer-Verlag, 2003.

[21] Cheng, S., Chou, T., Yang, C., Chang, H. "A Semantic Learning for Content-Based Images Retrieval Using Analytical Hierarchy Process", Expert Systems with Applications, Vol. 28, 2005, 495-505.

[22] Cherkassky, V. Learning from Data: Concepts, theory, and methods, John Wiley, 1998.

[23] Choi, J., Cho, M., Park, S., Kim, P. "Concept-Based Image Retrieval Using the New Semantic Similarity Measurement", Lecture Notes in Computer Science, Vol. 2667, 2003, 79-88.

[24] Choudhury, S.; Murty, M. "A Divisive Scheme for Constructing Minimal Spanning Trees in Coordinate Space", Pattern Recognition Letters, Vol. 11(6), 1990, 385–389.

[25] Cox, I.; Miller, M.; Minka, T.; Papathomas, T.; Yianilos, P. "The Bayesian Image Retrieval System, PicHunter: Theroy, Implementation, and Psychophysical Experiments", IEEE Transactions on Image Processing, Vol. 9(1), 2000.

[26] Crucianu, M., Tarel, J., Ferecatu, M. "A Comparison of User Strategies in Image Retrieval with Relevance Feedback", Proceedings of International Workshop on Audio-Visual Content and Information Visualization in Digital Libraries, 2005, 121-130.

[27] Dave, R. "Generalized Fuzzy C-shells Clustering and Detection of Circular and Elliptic Boundaries", Pattern Recognition, Vol. 25, 1992, 713–722.

[28] Dong, A. and Bhanu, B. "Active Concept Learning in Image Databases", IEEE Transaction on Systems, Man, and Cybernetics-Part B, Vol. 35(1), 2005, 450-466.

[29] Duan, L.; Chen, Y.; Gao, W. "Learning Semantic Cluster for Image Retrieval Using Association Rule Hyper-graph Partitioning", Proceeding of the 4th International Conference on Information, Communications & Signal Processing , and the 4th IEEE Pacific-Rim Conference On Multimedia, 2003.

[30] Eakins, J. "Towards intelligent image retrieval", Pattern Recognition, Vol. 35, 2002, 3-14.

[31] Eddy, W.; Mockus, A.; Oue, S. "Approximate Single Linkage Cluster Analysis of Large Data Sets in High-Dimensional Spaces", Computational Statistics and Data Analysis, Vol. 23(1), 1996, 29-43.

[32] Faber, V. "Clustering and the Continuous k-means Algorithm", Los Alamos Science, Vol. 22, 1994, 138–144.

[33] Friedman, J.; Khavi, R. "Lazy Decision Tree", Proceeding of the 13th National Conference on Artificial Intelligence, 1996, 717-724.

[34] Frigui, H.; Boujemaa, N.; Lim, S. "Unsupervised Clustering and Feature Discrimination with Application to Image Database Categorization", Joint 9th International Fuzzy Systems Association World Congress and 20th North American Fuzzy Information Processing Society Conference, 2001.

[35] Fujikawa, Y.; Ho, T. "Cluster-based Algorithms for Dealing with Missing Values", Lecture Notes in Computer Science, Vol. 2336, 2002, 549-554.

[36] Goh, K., Chang, E., Lai, W. "Multimodal Concept-Dependent Active Learning for Image Retrieval", Proceedings of the 12th ACM Multimedia Conference, 2004, 564-571.

[37] Gong, Y.; Proietti, G.; Faloutsos, C. "Image Indexing and Retrieval Based on Human Perception Color Clustering", Conference on Computer Vision and Pattern Recognition, 1998.

[38] Han, J.; Kamber, M. Data Mining: Concepts and Techniques, Moran Kaufman, 2001.

[39] Han, J.; Koperski, K.; Stefanovic, N. "GeoMiner: A System Prototype for Spatial Data Mining", Proceeding of ACM International Conference on Management of Data, 1997, 553-556.

[40] Han, J., Ngan, K., Li, M., Zhang, H. "A Memory Learning Framework for Effective Image Retrieval", IEEE Transactions on Image Processing, Vol. 14(4), 2005, 511-524.

[41] Han, J.; Pei, J.; Yin, Y. "Mining Frequent Patterns without Candidate Generation", Proceeding of ACM International Conference on Management of Data, 2000.

[42] Hanifer, J.; Sawhney, H.; Equitz, W.; Flinker, M.; Niblack, W. "Efficient Color Histogram Indexing for Quadratic From Distance", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17(7), 1995, 729-736.

[43] He, X.; Ma, W.; King, O.; Li, M.; Zhang, H. "Learning and Inferring a Semantic Space from User's Relevance Feedback for Image Retrieval", Proceeding of the 10th ACM Conference on Multimedia, 2002.

[44] Hoi, C.; Lyu, M. "A Novel Log-based Relevance Fedback Technique in Content-based Image Retrieval", Proceeding of the 12th ACM Conference on Multimedia, 2004, 24-31.

[45] Hofmann, T. "Unsupervised Learning by Probabilistic Latent Semantic Analysis", Machine Learning, 42, 2001, 177-196.

[46] Honda, K. "Introduction to Clustering Techniques", International Joint Conference on Neural Networks, 2004.

[47] Honda, K.; Sugiura, N.; Ichihashi, H.; Araki, S. "Collaborative Filtering Principal Component Analysis and Fuzzy Clustering", Lecture Notes in Artificial Intelligence, Vol. 2198, 2001, 394-402.

[48] Honda, K.; Yamakawa, A.; Kanada, A.; Ichihashi, H. "An Application of Fuzzy C-Mean Clustering to PCA-Like Method for Missing Values Estimation", Proceeding of the 16th International Conference on Production Research, 2001.

[49] Houng, T. Zhou, X.; Nakazato, M.; Wu, Y.; Cohen, I. "Learning in Content-Based Image Retrieval", Proceeding of the 2nd International Conference on Development and Learning, 2002.

[50] Houng, T.; Zhou, X. "Image Retrieval with Relevance Feedback: From Heuristic Weight Adjustment to Optimal Learning Methods", Proceeding of International Conference on Image Processing, 2001.

[51] Ishikawa, Y.; Subramanya, R.; Faloutsos, C. “MindReader: Query Databases Through Multiple Examples”, Proceeding of the 2nd Very Large Databases Conference, 1998.

[52] Jin, X, Zhou, Y., Mobasher, B. “Web Usage Mining Based on Probabilistic Latent Semantic Analysis”, Proceeding of ACM-KDD Conference, 2004, 197-205.

[53] Jing, F., Li, M., Zhang, H., Zhang, B. “A Unified Framework for Image Retrieval Using Keyword and Visual Features”, IEEE Transactions on Image Processing, Vol. 14(7), 2005, 979-989.

[54] Joachime, T. “Text Categorization with Support Vector Machines”, Proceeding of the European Conference on Machine Learning, 1998.

[55] John, G.; Kohavi, R.; Fleger, K. “Irrelevant Feature and the Subset Selection Problem”, Proceeding of the 11th International Conference on Machine Learning, 1994, 121-129.

[56] Koskela, A. and Laaksonen J. “Using Long-Term Learning to Improve Efficiency of Content-Based Image Retrieval”, Proceedings of the 3rd International Workshop on Pattern Recognition in Information Systems, 2003, 72-79.

[57] Koutaki, G.; Uchimura, K. “Automatic Road Extraction Based on Cross Detection in Suburb”, The International Society for Optical Engineering Conference, 2003.

[58] Krishnapuram, R., Medasani, S., Jung, S., Choi, Y, Balasubramaniam, R. “Content-Based Image Retrieval Based on Fuzzy Approach”, IEEE Transactions on Knowledge and Data Engineering, Vol. 16(10), 2004, 1185-1198.

[59] Krishnamachari, S.; Mottaleb, M. “Hierarchical Clustering Algorithm for Fast Image Retrieval”, IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases Vol. 2, 1999.

[60] Krzanowski, W. Principles of Multivariate Analysis : A User's Perspective, Oxford University Press, 2000.

[61] Lanzarotti, R.; Campadelli, P.; Borghese, N. “Automatic Features Detection for Overlapping Face Images on their 3D Range Models”, Proceeding of the 11th International Conference on Image Analysis and Processing, 2001.

[62] Law, M.; Jain, A. “Feature Selection in Mixture-Based Clustering”, Neural Information Processing Systems Conference, 2002.

[63] Lelescu, A.; Welfson, O.; Xu, B. “Approximate Retrieval from Multimedia Databases using Relevance Feedback”, Symposium on String Processing and Information Retrieval, 1999.

[64] Li, D.; Deogun, J.; Spaulding, W.; Shuart, B. "Towards Missing Data Imputation: A Study of Fuzzy k-means Clustering Method", Lecture Notes in Computer Science, Vol. 3066, 2004, 573-579.

[65] Little, R.; Rubin, D. Statistical Analysis with Missing Data, Wiley, New York, 1987.

[66] Liu, F.; Yu, C.; Meng, W. "Personalized Web Search by Mapping User Queries to Categories", Proceedings of the 11th International Conference on Information and Knowledge Management, 2002, 558-565.

[67] Liu, W.; White, A.; Thompson, S.; Bramer, M. "Techniques for Dealing with Missing Values in Classification", Lecture Notes, Vol. 1280, 1997, 527-536.

[68] Mao, J.; Jain, A. "A Self-Organizing Network for Hyper-ellipsoidal Clustering (HEC)", IEEE Transactions on Neural Networks, Vol. 7, 1996, 16-29.

[69] Melville, P.; Mooney, R.; Nagarajan, R. "Content-Boosted Collaborative Filtering for Improved Recommendations", Proceeding of 18th National Conference on Artificial Intelligence, 2002.

[70] Mendes, M.; Sacks, L. "Evaluating Fuzzy Clustering for Relevance-Based Information Access", The IEEE International Conference on Fuzzy Systems, 2003.

[71] Mitra, S.; Acharya, T. Data Mining: Multimedia, Soft Computing, and Bioinformatics, Wiley, 2003.

[72] Mobahser, B.; Dai, H.; Luo, T.; Nakagawa, M. "Effective Personalization Based on Association Rule Discovery from Web Usage Data", Proceeding of the 3rd International Workshop on Web Information and Data Management, 2001.

[73] Mojsilovic, A.; Gomes, J. "Semantic Based Categorization, Browsing And Retrieval in Medical Image Databases", Proceeding of the International Conference on Image Processing, 2002, 145, 148.

[74] Muller, H.; Muller, W.; Marchand, S.; Pun, T. "Strategies for Positive and Negative Relevance Feedback in Image Retrieval", International Conference on Pattern Recognition, 2000, 5043-5048.

[75] Muller, H.; Pun, T.; Squire, D. "Learning from User Behavior in Image Retrieval: Application of the Market Basket Analysis", International Journal of Computer Vision, Vol.56, 2004, 65-77.

[76] Nasraoui, O.; Frigui, H.; Joshi, A.; Krishnapuram, R. "Mining Web Access Logs Using Relational Competitive Fuzzy Clustering", Proceeding of the 8th International Fuzzy Systems Association World Congress, 1999.

[77] Papageorgiou, C.; Oren, M.; Poggio, T. “A General Framework for Object Detection”, Proceeding of International Conf on Computer Vision, 1998.

[78] Papathomas, T., Conway, T., Cox, I., Ghosn, J., Miller, M., Minka, T., Yianilos, P. “Psychophysical Studies of the Performance of an Image Database Retrieval System”, International Society of Optical Engineers, Vol. 3299, 1998, 591-602.

[79] Park, J.; Chen, M.; Yu, P. “An Effective Hash-Based Algorithm for Mining Association Rules”, Proceeding of ACM Conference on Management of Data, 1995, 175-186.

[80] Pauwels, E.; Caenen, G. “Regression Models for Relevance Feedback and Feature Selection in Content-Based Image Retrieval”, Proceeding of INRIA Multimedia Content-based Indexing and Retrieval Conference, 2001.

[81] Pennock, D.; Horvitz, E.; Lawrence, S.; Giles, C. “Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model Based Approach”, Proceeding of the 6th Conference Uncertainty in Artificial Intelligence, 2000.

[82] Perkowitz, M.; Etzioni, O. “Adaptive Web Sites,: Conceptual Cluster Mining”, Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the World Wide Web, 1999.

[83] Phatak, D.; Mulvaney, R. “Clustering for Personalized Mobile Web Usage”, Proceedings of the IEEE International Conference on Fuzzy Systems, 2002, 705-710.

[84] Poo, D.; Chen, B.; Goh, J. “A Hybrid Approach for User Profiling”, Proceeding of the 36th International Conference on System Sciences, 2003.

[85] Popescul, A.; Unger, L.; Pennock, D.; Lawrence, S. “Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments”, Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, 2001.

[86] Porkaew, K.; Mehrotra, S.; Ortega, M. "Query Reformulation for Content-based Multimedia Retrieval in MARS", Proceedings of the IEEE International Conference on Multimedia Computing and Systems,1999.

[87] Prasad, V.; Faheema, A.; Rakshit, S. “Feature Selection in Example-Based Image Retrieval Systems”, Proceeding of Indian Conference on Vision Graphics and Image Processing, 2002.

[88] Pyle, D. Data Preparation for Data Mining, Morgan Kaufman, 1999.

[89] Qian, F.; Li, M.; Zhang, J.; Zhang, B. "SVM-Based Feature Weighting Method for Image Retrieval", Proceeding of the 4th International Workshop On Multimedia Information Retrieval, 2002.

[90] Quinlan, J. "Induction of Decision Trees", Machine Learning, Vol.1, 1986, 81-106.

[91] Refecatu, M.; Crucianu, M.; Boujemaa, N. "Retrieval of Difficult Image Classes Using SVM-Based Relevance Feedback", Proceeding of 12th ACM Workshop on Multimedia Information Retrieval, 2004, 23-30.

[92] Rui, Y.; Huang, T. "Optimizing Learning in Image Retrieval", IEEE Conference on Computer Vision and Pattern Recognition, 2000, 236-243.

[93] Rui, Y.; Huang, T.; Mehrotra, S. "Content-based Image Retrieval with Relevance Feedback in MARS", Proceeding of IEEE International Conference on Image Processing, 1997.

[94] Rui, Y.; Huang, T.; Ortega, M.; Mehrotra, S. "Relevance Feedback : A Power Tool in Interactive Content-Based Image Retrieval", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8(5), 1998, 644-655.

[95] Sanghai, S.; Domingos, P.; Weld, D. "Dynamic Probabilistic Relational Models", Proceeding of the 8th International Joint Conference on Artificial Intelligence, 2003.

[96] Sarkar, M.; Leong, T. "Fuzzy K-means Clustering with Missing Values", American Medical Informatics Association Symposium, 2001, 588-592.

[97] Sarwar, B.; Karypis, G.; Konstan, J.; Reidl, J. "Analysis of Recommendation Algorithms for E-Commerce", ACM Conference on Electronic Commerce, 2000.

[98] Schafer, J.; Konstan, J.; Riedl, J. "Recommender Systems in E-Commerce", Proceedings of ACM Conference on E-Commerce, 1999.

[99] Schmitt, I., Schulz, N., Herstel, T. "WS-QBE: A QBE-Like Query Language for Complex Multimedia Queries", Proceedings of the 11th International Multimedia Modeling Conference, 2005, 222-229.

[100] Shahabi, C.; Chen, Y. "Web Information Personalization: Challenges and Approaches", Proceeding of the 3rd Workshop on Databases in Networked Information Systems, 2003.

[101] Shah-hosseini, A.; Knapp, G. "Learning Image Semantics From Users Relevance Feedback", Proceedings of the 12th ACM International Conference on Multimedia, 2004, 452-455.

[102] Shah-hosseini, A.; Knapp, G. "Semantic Image Retrieval Based on Probabilistic Latent Semantic Analysis", Proceedings of the 14th ACM International Conference on Multimedia, 2004, 452-455.

[103] Smith, J.; Chang, S-F. "VisualSEEK: A Fully Automated Content-Based Image Query System", Proceedings of the 4th ACM International Multimedia Conference, 1996.

[104] Soltan G.; Buckley, C. "Improving Retrieval Performance by Relevance Feedback", Journal of the American Society for Information Retrieval", Vol. 41(4), 1990, 288-297.

[105] Sparck Jones, K.; Willett, P. Readings in Information Retrieval, Morgan Kaufmann, 1997.

[106] Su, Z.; Zhang, H.; Ma, S. "Using Bayesian Classifier in Relevance Feedback of Image Retrieval", Proceeding of the 12th IEEE International Conference on Tools with Artificial Intelligence, 2000.

[107] Sumengen, B.; Manjunath, B.; Kenney, S. "Image Segmentation Using Multi-Region Stability and Edge Strength", Proceedings of IEEE International Conference on Image Processing, 2003.

[108] Sun, J.; Sun, Z.; Zhou, R.; Wang, H. "A Semantic-Based Image Retrieval System: VISENGINE", Proceeding of the First International Conference on Machine Learning and Cybernetics, 2002.

[109] Tesic, J., Newsam S., Manjunath, B. "Mining Image Datasets Using Perceptual Association Rules", Proceedings of SIAM 6th Workshop on Mining Scientific and Engineering Datasets, 2003.

[110] Tesic, J.; Manjunath, B. "Nearest Neighbor Search for Relevance Feedback", Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.

[111] Tieu K.; Viola P. "Boosting Image Retrieval", IEEE Conference on Computer Vision and Pattern Recognition, 2000.

[112] Timm, H. Doring, C. Kruse, R. "Differentiated Treatment of Missing Values in Fuzzy Clustering", Lecture Notes in Computer Science, Vol. 2715, 2003, 354-361.

[113] Ukkonen, E. "Algorithms for Approximate String Matching", Information and Control, Vol. 64, 1985, 11-118.

[114] Ungar, L.; Foster, D. "Clustering Methods for Collaborative Filtering", Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence, 1998.

[115] Vapnik, V. Statistical Learning Theory, Wiley, 1998.

[116] Vasconcelos, N. “Feature Selection by Maximum Marginal Diversity”, Neural Information Processing Systems Conference, 2002.

[117] Vasconcelos, N. and Vasconcelos. M. “Scalable Discriminant Feature Selection for Image Retrieval and Recognition”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, 770-775.

[118] Wang, J.; Li, J.; Wiederhold, G. “SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23(9), 2001, 947-963.

[119] Weld, D.; Anderson, C.; Domingos, P.; Etzioni, O. “Automatically Personalizing User Interfaces”, Proceeding of 18th International Joint Conference on Artificial Intelligence, 2003.

[120] White, A. “Probabilistic Induction by Dynamic Path Generation in Virtual Trees”, Research and Development in Expert Systems, Cambridge University Press, 1987.

[121] Wu, K. and Yap, K. “A Perceptual Subjectivity Notion in Interactive Content-Based Image Retrieval Systems”, Intelligent Multimedia Processing with Soft Computing, Springer, 2005.

[122] Wu, Y.; Tian, Q.; Huang, T. “Discriminant-EM Algorithm with Application to Image Retrieval”, IEEE Conference on Computer Vision and Pattern Recognition, 2000.

[123] Wu, Y.; Zhang, A. “A Feature Re-weighting Approach for Retrieval Relevance Feedback in Image Retrieval”, Proceeding of IEEE International Conference on Image Processing, 2002.

[124] Xie, X.; Beni, G. “A Validity Measure for Fuzzy Clustering”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13(8), 1991, 841-847.

[125] Xie, Y.; Raghavan, V.; Zho, X. “3M Algorithm: Finding an Optimal Fuzzy Cluster Scheme for Proximity Data”, Proceeding of IEEE World Congress on Computational Intelligence, 2002.

[126] Xiong, X.; Tan, K. “Similarity-Driven Cluster Merging Method for Unsupervised Fuzzy Clustering”, Proceedings of the 20th Conference on Uncertainty in artificial intelligence, 2004, 611-618.

[127] Yagar, R. and Frederick P. “A Framework for Linguistic Relevance Feedback in Content-Based Image Retrieval Using Fuzzy Logic”, Information Sciences, Vol. 173, 2005, 337-352.

[128] Yoshizawa, T.; Scweitzer, H. “Long-Term Learning of Semantic Grouping from Relevance Feedback”, Proceeding of the 12th ACM Conference on Multimedia, 2004, 165-172.

[129] Zachary, J. and Iyengar, S. “Information Theoretic Similarity Measure for Content Based Image Retrieval”, Journal of the American Society for Information Sciences and Technology, Vol. 52(10), 2001, 856-867.

[130] Zhang, D.; Dong, Y. “Semantic, Hierarchical, Online Clustering of Web Search Results”, Proceedings of the 6th Asia Pacific Web Conference, 2004.

[131] Zhou X. and Huang, T. “Small Sample Learning during Multimedia Retrieval using BiasMap”, IEEE Conference in Computer Vision and Pattern Recognition, 2001, 11-17

[132] Zhou, X.; Zhang, Q.; Zhang, L.; Liu, L.; Shi, B. “An Image Retrieval Method Based on Collaborative Filtering”, Lecture Notes in Computer Science, Vol. 2690, 2003, 1024-1031.

APPENDIX A

Approved Application for Exemption from Institutional Oversight

IRB #: 3103 LSU Proposal #:______________ Revised: 04/15/2005

LSU INSTITUTIONAL REVIEW BOARD (IRB) for
HUMAN RESEARCH SUBJECT PROTECTION

578-8692 FAX 6792
Office:203 B-1 David Boyd Hall

APPLICATION FOR EXEMPTION FROM INSTITUTIONAL OVERSIGHT

Study exempted by
Louisiana State University
Institutional Review Board
203 B-1 David Boyd Hall
225-578-8692
Robert C. Mathews, Chair

Unless they are qualified as meeting the specific criteria for exemption from Institutional Review Board (IRB) oversight, ALL LSU research/projects using living humans as subjects, or samples or data obtained from humans, directly or indirectly, with or without their consent, must be approved or exempted in advance by the LSU IRB. This Form helps the PI determine if a project may be exempted, and is used to request an exemption.

Instructions: Complete this form.

Exemption Applicant: If it appears that your study qualifies for exemption send:

(A) Two copies of this completed form,
(B) a brief project description (adequate to evaluate risks to subjects and to explain your responses to Parts A & B),
(C) copies of all instruments to be used. If this proposal is part of a grant proposal include a copy of the proposal and all recruitment material.
(D) the consent form that you will use in the study. **A Waiver of Written Informed Consent is attached and must be completed only if you do not intend to have a signed consent form.**

to: ONE screening committee member (listed at the end of this form)in the most closely related department/discipline or to IRB office.

If exemption seems likely, submit it. If not, submit regular IRB application. Help is available from Dr. Robert Mathews, 578-8692, irb@lsu.edu or any screening committee member.

Principal Investigator ___Amin Shah-Hosseini________________________________ **Student?**__Yes______Y/N

Ph: __225-229-7639______________ **E-mail**__ashahh1@lsu.edu_____ **Dept/Unit** _Industrial Engineering________

If Student, name supervising professor __Dr. Gerry Knapp________________________ Ph:_225-578-5374______
Mailing Address_3128 CEBA Industrial Engineering, LSU________________________ Ph_______________
Project Title _Learning Image Semantics rom Users Relevance Feedbacks________________________________

Agency expected to fund project__N/A____
Subject pool (e.g. Psychology Students)____
Circle any **"vulnerable populations" to be used**: (children <18; the mentally impaired, pregnant women, the aged, other). Projects with incarcerated persons cannot be exempted.
I certify my responses are accurate and complete. If the project scope or design is later changed I will resubmit for review. I will obtain written approval from the Authorized Representative of all non-LSU institutions in which the study is conducted.

PI Signature ______________________ **Date** 9/22/05 (no per signatures)

===

Screening Committee Action: Exempted √ Not Exempted ____ Category/Paragraph ______

Reviewer Mathews Signature ______________________ Date 9/24/05

Part A: DETERMINATION OF "RESEARCH" and POTENTIAL FOR RISK

This section determines whether the project meets the Department of Health and Human Services (HSS) definition of research involving human subjects, and if not, whether it nevertheless presents more than "minimal risk" to human subjects

VITA

Amin Shah-hosseini was born in Iran on June 14, 1978. He received a Bachelor of Science degree in industrial engineering at Sharif University of Technology in August 2000. Upon graduation, Amin started his graduate studies while working full-time as a Business Analyst/Scheduler in a project management contractor in the fields of power plant construction and energy industry. Amin received his Master of Science degree in industrial engineering in August 2002, and enrolled in the engineering science Ph.D. program with information technology engineering concentration at Louisiana State University in January 2003. Interested in describing behaviors, making predictions, and optimizing processes through data-driven solutions, Amin chose industrial engineering as his major, and computer science and business administration as his minor fields in the interdisciplinary program of engineering science. The degree of Doctor of Philosophy will be conferred at the August 2007 commencement.