An Effective Region-Based Image Retrieval Framework*

Feng Jing
State Key Lab of Intelligent Technology and Systems
Beijing 100084, China
+86-10-62782266
Scenery_JF@hotmail.com

Mingjing Li, Hong-Jiang Zhang
Microsoft Research Asia
49 Zhichun Road
Beijing 100080, China
+86-10-62617711
{mjli,hjzhang}@microsoft.com

Bo Zhang
State Key Lab of Intelligent Technology and Systems
Beijing 100084, China
+86-10-62782266
dcszb@mail.tsinghua.edu.cn

ABSTRACT

We present a region-based image retrieval framework that integrates efficient region-based representation in terms of storage and retrieval and effective on-line learning capability. The framework consists of methods for image segmentation and grouping, indexing using modified inverted file, relevance feedback, and continuous learning. By exploiting a vector quantization method, a compact region-based image representation is achieved. Based on this representation, an indexing scheme similar to the inverted file technology is proposed. In addition, it supports relevance feedback based on the vector model with a weighting scheme. A continuous learning strategy is also proposed to enable the system to self improve. Experimental results on a database of 10,000 general-purposed images demonstrate the efficiency and effectiveness of the proposed framework.

Keywords

Region-based image retrieval, relevance feedback, inverted file, continuous learning.

1. INTRODUCTION

It is well known that the performance of content-based image retrieval (CBIR) systems is mainly limited by the gap between low-level features and high-level semantic concepts. In order to reduce this gap, two approaches have been widely used: region-based features to represent the focus of the user's perceptions of image content [4, 8, 9, 14, 20, 21, 24, 30, 37, 43] and relevance feedback (RF) to learn the user's intentions [2, 13, 19, 22, 29, 33, 35, 40, 42].

Contrasting to traditional approaches [12, 25, 26, 32], which compute global features of images, the region-based methods extract features of the segmented regions and perform similarity comparisons at the granularity of region. The main objective of using region features is to enhance the ability of capturing as well as representing the focus of user's perceptions of image content.

When designing a practical region-based image retrieval system, at least the following issues should be addressed:

1. How to compare two images, i.e. the definition of the image similarity measure.
2. How to make it scalable?
3. How to make it interactive, i.e. the strategy of relevance feedback.

For the first issue, a straightforward solution adopted by most early systems [4, 8, 20, 21, 30, 40] is to use individual region-to-region similarity. When using such systems, the users are forced to select a limited number of regions from the query image in order to start a query session. As discussed in [37], due to the uncontrolled nature of the images available, automatically and precisely extracting image objects is still beyond the reach of the state-of-the-art in computer vision. Therefore, the above systems tend to partition one object into several regions with none of them being representative for the object. Consequently, it is often difficult for users to determine which regions should be used for retrieval.

To provide users a simpler querying interface and reduce the influence of inaccurate segmentation, several image-to-image similarity measures that combine information from all of the regions have been proposed [9, 14, 24, 31, 37]. Such systems only require the users to assign a query image, and therefore relieve the users from puzzling decisions. For example, the SIMPLIcity system [37] uses an integrated region matching as its image similarity measure. By allowing many-to-many relationship of the regions, the approach is robust to inaccurate segmentation.

To solve the second issue, many efforts have been made along either or both of the two directions: saving time and saving space. For the former, the efforts can be classified into two categories: one is using traditional tree structures [8, 24, 34], such as R*-tree [3] and M-tree [5], the other is using statistical clustering [20, 36]. For the latter, not much attention has been paid. One effort is the system proposed in [43], which is based on vector quantization. Few works have addressed both of the facets. One successful example is the VisualSEEk system [30], which uses compact color set representation of regional color features and efficient indexing techniques for color information, region sizes, absolute and relative spatial locations.

* This work was performed at Microsoft Research Asia

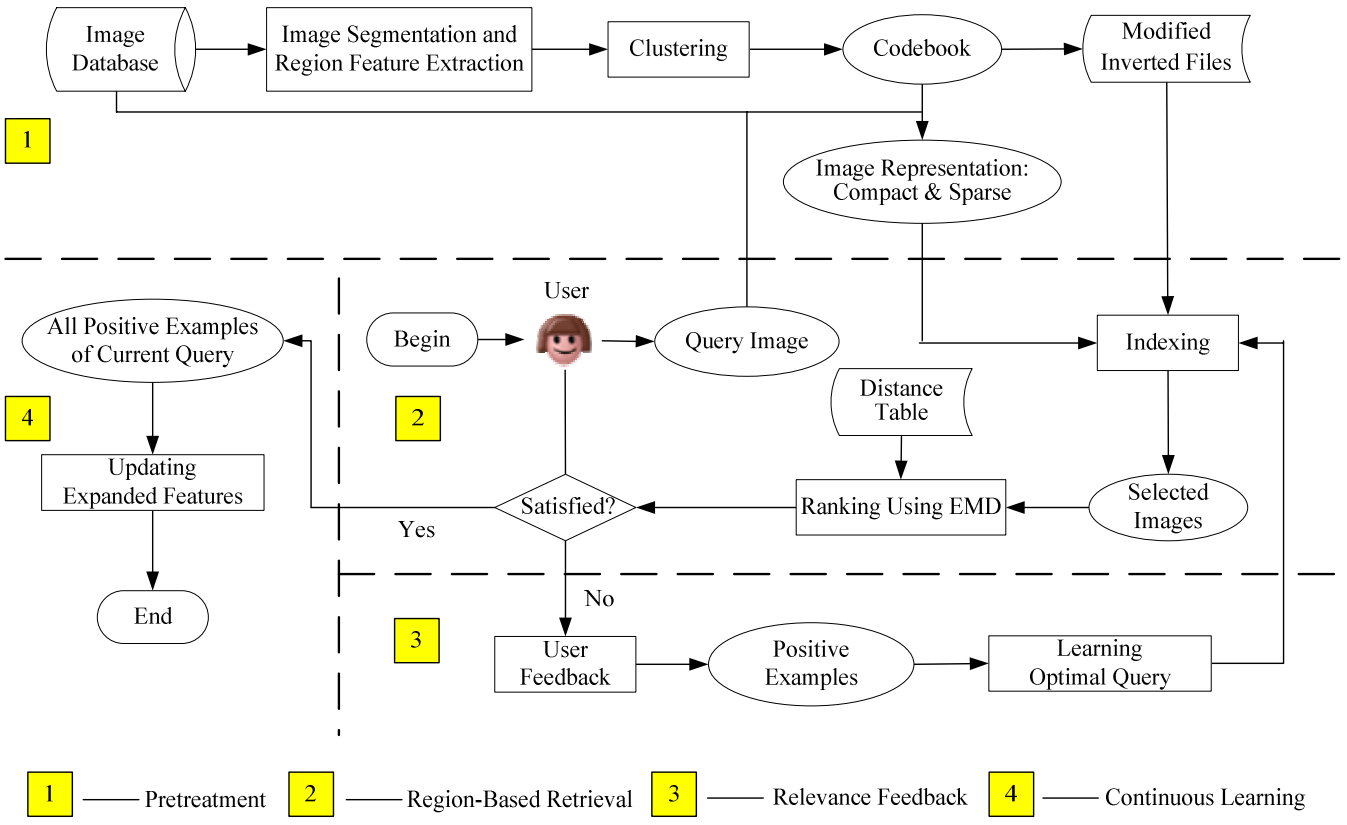

Figure 1. Overview of the proposed framework

Although relevance feedback has shown its great potential in image retrieval systems that use global feature representations, it has seldom been introduced to the region-based retrieval systems. Minka and Picard performed a pioneering work in this area by proposing the FourEyes system [21, 22]. FourEyes contains three stages: grouping generation, grouping weighting and grouping collection. In the generation stage, it produces many plausible groupings including both with-in image groupings and across-image groupings induced by different models. The collection stage is guided by a rich example-based interaction with the user. The weighting stage adapts the collection stage's search space across uses, so that in later interactions, good groupings are found given few examples from the user. In spite of its many good characteristics, FourEyes system has two disadvantages. One is the use of region-to-region similarity measure, and the other is the re-clustering of all the features when a new image is added. Thus it is not very scalable. Other efforts in this direction include the IDQS system developed by Wool et al [40], which can be considered a region-based retrieval method using classification as its feedback scheme.

In this paper, we propose a novel framework to address the above three issues. The image similarity measure we adopt is the Earth Mover's Distance (EMD) [28], which has a rigorous probabilistic interpretation. To be scalable, a new region-based image representation is utilized, which saves much storage as well as facilitates the indexing scheme based on a modified inverted file strategy. Also, relevance feedback is considered and performed in a way similar to the classical Rocchio's algorithm [27]. By following the *self-organization principle* [22], our framework supports continuous learning, which enables it to self improve. An overview of the proposed framework is shown in Figure 1.

The organization of the paper is as follows: Section 2 describes the representation of images based on image segmentation and region clustering. The searching and indexing schemes are presented in Section 3. The learning strategies including both the short-term learning, i.e. relevance feedback, and continuous learning are described in Section 4 and Section 5, respectively. In Section 6, we provide experimental results that evaluate all aspects of the framework under several different situations. Section 7 concludes with a discussion of the future work.

2. IMAGE REPRESENTATION

In the proposed framework, images in a database are first segmented into homogeneous regions, and similar regions from all images are clustered into a small number of groups. Based on region clusters, low-level features of regions are quantized such that images can be represented in a way similar to the vector space model in text information retrieval [1]. This compact representation not only reduces the storage space, but also facilitates the fast indexing for content-based image retrieval.

2.1 Image Segmentation

The segmentation method we utilized is proposed in [17]. It can be considered an improvement over the JSEG [6] algorithm.

First, a criterion for homogeneity of a certain pattern is proposed. Applying this criterion to local windows in an image results in the

"H-image". The high and low values of an H-image correspond to possible region boundaries and region interiors, respectively. Then, a region growing method is used to segment the image based on the H-image. Finally, visually similar regions are merged to avoid over-segmentation. More details can be found in [17].

2.2 Region Properties

We use two properties to describe a region: the low-level feature extracted from the region and its importance weight. Both of the properties are used in the similarity measure between two images. For the former, many kinds of low-level features may be used. In the current implementation, we use color moment, which is shown to be robust and effective [32]. We extract the first two moments from each channel of CIE-LUV color space. Since our objective is to test the indexing scheme and the learning strategy rather than to evaluate features, the feature we use is not as sophisticated as those used in other region-based image retrieval systems [20, 30]. For the latter, the percentage of region area is used temporarily. More satisfactory weighting methods are discussed in Section 7. The only requirement is that the sum of importance weights for an image should be equal to 1.

2.3 Region Clustering

Considering that there exist many regions from different images that are very similar in terms of the features, some compression technique like vector quantization (VQ) can be used to reduce the storage space without sacrificing much accuracy.

More specifically, the Generalized Lloyd Algorithm (GLA) [7], an iterative clustering algorithm, is used as the quantizer. The features of regions from all images in the database are used as training data, and the codebook size is determined experimentally by balancing the efficiency and the accuracy. For each codeword, several properties are calculated and stored. One property is the mean of all the data belonging to it, i.e. the center, while other properties are covered in Section 3.2.

For each region of an image in the database, the cluster that it belongs to is identified and the corresponding index in the codebook is stored, while the original feature of this region is discarded. For the region of a new image, the closest entry in the codebook is found and the corresponding index is used to replace its feature.

A similar approach is described in [43], where two typical VQ methods are combined and exploited to generate a codebook, based on which images are encoded and retrieval is conducted. Our proposed approach differs from [43] in two aspects. First, the basic elements used to generate the codebook are regions, whereas in [43] are blocks. Since regions are more homogeneous than blocks and there are only 4 or 5 regions on average contrasting to tens of thousands of blocks in each image, more representative training data is used in our codebook design. As a result, our codebook can generalize better than the one used in [43]. Second, instead of trying to minimize the image reconstruction error, our goal is to maintain the retrieval accuracy.

2.4 Compact and Sparse Representations

Based on the codebook, now each image can be represented in two ways. One is a set of regions with each region described by an importance weight and an index of the corresponding codeword. Besides this compact representation, a sparse but uniform representation is also utilized. In this representation, an image is a vector with each dimension corresponding to a codeword.

More formally, the compact representation $\vec{I}_C$ of an image I that contains n regions $\{R_1,...,R_n\}$ is a region set $\vec{I}_C = \{(CI_{R1}, W_{R1}),...,(CI_{Rn}, W_{Rn})\}$, where CI_{Ri} and W_{Ri} are the codeword index and importance weight of R_i. The sum of W_{Ri} s should be 1. The sparse representation $\vec{I}_S$ of I is vector $\vec{I}_S = (w_1, w_2, ...w_N)$ where N is the number of the codewords. For a codeword $C_i, 1 \le i \le N$, if there exists a region R_j of I that corresponds to it, then $w_i = W_{Rj}$, otherwise, $w_i = 0$. This representation is sparse, for an image usually contains very few regions comparing to the number of the codewords. Since $\sum_{j=1}^{n} W_{Rj} = 1$, $\sum_{i=1}^{N} w_i = 1$.

The compact representation is used in the actual storage and the retrieval process (Section 3), while the sparse representation is used in relevance feedback (Section 4) and continuous learning (Section 5).

3. REGION-BASED RETRIEVAL

In image retrieval, a novel indexing scheme is used to quickly filter out candidate images before the similarity between the query and an image in the database is calculated, thus the retrieval speed is improved significantly.

3.1 Image Similarity Measure

Based on the compact image representation, the distance between two images is measured using the Earth Mover's Distance (EMD) [28]. The EMD measures the minimal cost that must be paid to transform one distribution into another. It is based on the transportation problem [11] and can be solved efficiently by linear optimization algorithms that take advantage of its special structure.

Considering that EMD matches perceptual similarity well and can operate on variable-length representations of the distributions, it is suitable for region-based image similarity measure. In this special case, the ground distance is an equally weighted Euclidean distance between the features of two codewords. Since the Euclidean distance is a metric and the total weight of each signature is constrained to be 1, the distance is a true metric according to [28]. EMD incorporates the properties of all the segmented regions so that information about an image can be fully utilized. By allowing many-to-many relationship of the regions to be valid, EMD is robust to inaccurate segmentation.

To make the computation of EMD faster, a distance table can be constructed optionally, which records the distance between all individual codewords. The distance between two codewords is the Euclidean distance between their representative features. With this distance table, the calculation of the ground distance is simplified to search in the look up table. As a result, the time of computing EMD is reduced. Indeed the distance table cost some

storage space, but this can be offset by the time saved, especially when the dimension of the region feature is high.

Alternatively, integrated region matching (IRM) [37] can be used as the image similarity measure, which is a simplified version of EMD. Instead of solving a liner programming problem directly, integrated region matching uses a greedy algorithm to achieve an approximation. As a result, integrated region matching is not a metric, which makes most of the traditional indexing techniques impossible.

3.2 Indexing Using Modified Inverted File

As discussed in Section 1, many traditional tree structures such as R-tree [10], R*-tree [3], M-tree [5], and SR-tree [18], have been introduced to index region-based image retrieval systems. Unfortunately, the speed and accuracy of these algorithms degrade in high dimensional spaces [3, 18, 38], which is referred to as the curse of dimensionality. For example, the performance of R*-trees degrades by a factor of 12 as the number of dimensions increases from 5 to 10 [38].

On the other hand, the inverted file as the most common indexing structure has been widely used in the information retrieval for its simplicity and effectiveness [1, 39]. The text retrieval community has developed techniques for building and searching inverted files very efficiently [39]. The key realization is that in such systems both queries and stored objects are sparse: they have only a small subset of all possible features. Search is thus restricted to the subspace spanned by the query.

The idea of the inverted file has recently been introduced to the domain of image retrieval [23]. In [23], more than 80,000 features are available to the system with each image having about 1,000 ones. The mapping from features to images is stored in an inverted file based on which three strategies were proposed to effectively decrease the response time. Although not clearly claimed, the indexing scheme of [36] is a rough implementation of the inverted file strategy.

Our indexing scheme adopts the inverted file strategy but with some modifications. Since EMD takes the distances between different codewords into consideration, which are usually not large enough to be neglected, we do not use the inverted files in the common way as in [36]. Instead, we first expand the codewords of the regions contained in the query image to some degree, which means that the regions are assumed to correspond to more than one codeword. The codewords used to expand the query are the ones that are most similar to the original codeword of the region. The number of the expanded codewords to a query region is determined by the weight of the region since the regions with larger weights play more important roles in the computation of image similarity. For a region with weight w, we expand $\lfloor w \cdot k \rfloor$ codewords to it, where k is a natural number and is set to be 10 currently. Since $0 < w \le 1$, the maximum number of the codewords that will be expanded is k. Therefore, in our implementation, an inverted file contains an entry for a codeword which consists of not only a list of the images that have a region corresponding to the codeword, but also k most similar codewords, except itself, sorted by their similarity to it. After the expansion, only the images that appear in one of the inverted files of any of the expanded codewords are investigated by the EMD to answer the query, while other images are skipped. In this way, we can reduce the response time significantly while maintaining the performance of the system.

Although the current image similarity measure, i.e. EMD, is a metric and the dimension of present features, i.e. color moment, is low, which makes some distance-based indexing techniques possible, e.g. the M-tree [5], we index the database in our own way. The reason is that our indexing strategy is independent of the similarity measure and the feature dimension. This independency makes it generalize well to future developments, such as using more complicated similarity measures that may be non-metric and more sophisticated features that may be of high dimensionality.

4. RELEVANCE FEEDBACK

As images are represented using a vector model, traditional relevance feedback methods, such as Rocchio's algorithm, can be utilized in the proposed framework.

4.1 Rocchio's algorithm

Rocchio's algorithm [27] for relevance feedback and query expansion was developed in the mid-1960's and has, over the years, proven to be one of the best relevance feedback algorithms in the field of information retrieval. Rocchio's algorithm was proposed in the framework of the vector space model [1]. When documents are to be ranked for a query, an *ideal query* should rank all the relevant documents above all non-relevant documents. However, such a query might just not exit, or even if it does exist for the training documents, it might be over-fitting and not generalize well to new documents. Therefore, Rocchio's approach lowers the aims and forms a query that maximizes the difference between the average score of a relevant document and the average score of a non-relevant document. Rocchio called this an *optimal query*, which is defined as:

$$\vec{Q}_{opt} = \frac{1}{R}\sum_{d \in \mathrm{Re}\,l} \vec{d} - \frac{1}{N-R}\sum_{d \notin \mathrm{Re}\,l} \vec{d} \qquad (1)$$

where $\vec{d}$ denotes the weighted term vector of document d, $R = |\mathrm{Re}\,l|$ is the number of relevant articles, and N is the total number of articles in the collection.

4.2 Weighted Query Expansion

Based on the sparse image representation, i.e. the vector representation, a straightforward way of integrating Rocchio's algorithm into our framework is to define the optimal query as the mean of the feature vector of all the positive ones (neither negative nor non-positive examples are considered in the current framework). The objective is to make the optimal query close to all the positive examples. From the user's point of view, it is unnecessary to rank the current positive examples in the next iteration. Instead, they can be directly arranged to be on top of ranking list in spite of their similarities to the query. Since we are released from the constraints that the optimal query should be close to all the positive examples, we can move it towards some direction to make it generalize better.

At each interaction, the newly added positive examples might have more potential in finding the user-intended but undetected images,

because they reflect the user's query concept more precisely. Therefore, they should have more contributions to the optimal query. A similar idea can be found in [35], which introduced a decaying factor to reduce the effect of previous positive examples. Our previous work in [16] has also reflected this bias.

More specifically, assume that there are n positive examples $I_1, \ldots, I_n$, with $I_1, \ldots, I_m$ being the prior ones and $I_{m+1}, \ldots, I_n$ being the new ones. For the sake of simplicity, the initial query is treated as a positive example. Let $\vec{I}_k$ denotes the vector representation of a positive example I_k, that is, $\vec{I}_k = (w_{k,1}, \ldots, w_{k,N})$. Let the optimal image be I_{opt} with the feature vector being $\vec{I}_{opt} = (w_{opt,1}, \ldots, w_{opt,N})$. Let α (≥ 0) be a factor that controls the importance of the prior positive examples. The smaller α is, the lower the importance of the prior positive regions will be.

The optimal query is defined to be:

$$\vec{I}_{opt} = \beta(\alpha \sum_{k=1}^{m} \vec{I}_k + \sum_{k=m+1}^{n} \vec{I}_k) \quad (2)$$

where β serves as a normalization factor to make $\vec{I}_{opt}$ satisfy the constraint: $\sum_{i=1}^{N} w_{opt,i} = 1$. Since $\alpha \sum_{k=1}^{m} \sum_{i=1}^{N} w_{k,i} + \sum_{k=m+1}^{n} \sum_{i=1}^{N} w_{k,i} = \alpha m + n - m = m(\alpha - 1) + n$, $\beta = \frac{1}{m(\alpha-1)+n}$. Considering that the smaller the number of the prior positive examples, the more important they should be, we adaptively choose the value of α. In fact, α is set to be $\frac{1}{m}$ (for m = 0, i.e. the first iteration, α does not play a part) and $\beta = \frac{1}{n-m+1}$.

In summary, the feedback operates as follows. At the first iteration, the positive examples are equally considered to form the optimal query. During the following iterations, the current optimal query acts only as one positive example, and is combined with the newly labeled positive examples in the same way as the first iteration, while all the prior ones are ignored. In this way, the importance of prior positive examples gradually decays and the importance of the newly added ones is emphasized accordingly.

4.3 Region Pruning

Most of the current works on relevance feedback do not address the indexing issue and assume an exhaustive search of the database during each iteration. Such computations can become prohibitive as the database size increases. One exception is adaptive nearest neighbor search [41], which updates a relatively small number of nearest neighbors intelligently and efficiently from one iteration to the next without searching the whole dataset repeatedly. Since the underlying distance measure used in [41] is weighted Euclidean, the technique can not be directly integrated into our framework. On the other hand, the optimal query can also be represented with a coarse representation, which makes the indexing scheme proposed in Section 3.2 possible. As more positive examples marked by the user, the optimal query will contain more codewords, which will makes the indexing ineffective, for more images need to be compared with the query using EMD. To alleviate this inefficiency, a simple pruning technique is utilized. The codewords with the corresponding weights smaller than a threshold ξ are not considered in the indexing stage. ξ is set to be 0.02 in our experiments.

5. CONTINUOUS LEARNING

A relevance feedback process is inherently incremental: except for the first round, the system always learns from both previous feedbacks from the user(s), and the new information from the current round. However, most of the current feedback methods share the drawback of *forgetting* user preferences across multiple query sessions, thus requiring the feedback loop to restart for every new query. Not only is this proceeding frustrating from the user's point of view but it also constitutes a significant waste of system resources.

The first work in CBIR that embodies the idea of continuous learning was done by Minka [21]. As stated in his master's thesis [22]: "Continuous learning attempts speedup by learning continuously, across problem, not just within them. While the individual tasks assigned to the learner may be supervised, continuous learning is an unsupervised process. The learner is not explicitly told which aspects of a problem will carry over to the next. Therefore a continuous learner must be self-organizing. This gives a hint about how to construct one, by following the *self-organization principle*: after solving a problem, make a change that will allow the system to perform better on the same problem again." Besides the work of Minka, there are also some attempts that try to implement the idea of continuous learning [2, 15, 19, 33].

5.1 Updating Learned Feature

Following the *self-organization principle*, we perform the continuous learning in the following way. First, another image feature is utilized, which is called the *learned feature*. The learned feature of an image is in the same form as the original feature, i.e. can also be represented with the compact and sparse representations. It is initialized to be the same as the original feature and is updated when the corresponding image is used as a query or labeled as a positive example in a search session.

More specifically, let the query be I_Q with the original and learned feature being $\vec{I}_Q^{\,O}$ and $\vec{I}_Q^{\,L}$ respectively. Assume that there are totally n positive examples $I_1, \ldots, I_n$ marked by the user during the whole query session. Denote the original and learned feature of I_i, $1 \leq i \leq n$, as $\vec{I}_i^{\,O}$ and $\vec{I}_i^{\,L}$. $\vec{I}_Q^{\,L}$ and $\vec{I}_i^{\,L}$s are updated as follows:

$$\vec{I}_{Qnew}^{\,L} = \frac{1}{n+1}(\vec{I}_{Qold}^{\,L} + \sum_{i=1}^{n} \vec{I}_i^{\,O}) \quad (3)$$

$$\vec{I}_{inew}^{\,L} = \frac{1}{2}(\vec{I}_{iold}^{\,L} + \vec{I}_Q^{\,O}) \quad (4)$$

The intuition why the query and the positive examples should update differently is the following. The positive examples are

similar to the query in their own but may be different ways. For example, when the user is searching for images of fruit using a query image that contains both apples and oranges, the positive example may contain either some apples or some oranges or even other fruits. So what we can make sure is that all the positive examples are similar to the query and vice versa, but not that the positive examples are similar to each other. This is why the query is updated by all the positive examples, while the positive examples are updated only by the query. Alternative strategies, e.g. updating the query in the same way as the positive examples, degrade performance in our experiments.

5.2 Using Learned Feature

Unlike the strategy used in [33] where the stored query parameters are only used in the initial query stage, in our framework, they are also used in the relevance feedback stage. More precisely, (2) is modified to be:

$$\vec{I}_{opt} = \beta(\alpha\sum_{k=1}^{m}\vec{I}_k^L + \sum_{k=m+1}^{n}\vec{I}_k^L) \quad (5)$$

where $\vec{I}_k^L$ is the learned feature of I_k.

With this modification, the precision of new queries can still be improved as long as the learned features of relevant images have been updated in previous relevance feedback. On the contrary, in [33], the stored parameters will be helpless to the new queries.

6. EXPERIMENTS

We tested our framework with a general-purpose image database of about 10,000 images from COREL. 1,000 images were randomly chosen from totally 79 categories as the query set. Denote the query set as $QS = \{I_1,...,I_{1000}\}$. Unless otherwise noted, the default results of the experiments are averages of the 1,000 queries.

To determine the size of the codebook, different numbers of clusters have been selected and evaluated. Considering that the value of *k*, i.e. the maximum number of the expanded codewords (see Section 3.2) is related with *N*, i.e. the size of the codebook, we did not use the indexing scheme in this evaluation. A retrieved image is considered a match if it belongs to the same category of the query image. The average precisions within the top 20 (30, 50) images (P(20), P(30), P(50)) are shown in Figure 2. As shown in it, the general trend is that the larger the codebook size, the higher the retrieval accuracy. However, larger codebook size means bigger image feature vector, which will cost more computations in the learning stages. Also, larger codebook will lead to more storage. Therefore, we use 400 as the number of the clusters, which corresponds to the first turning point in Figure 1. Since there are totally 43,009 regions of all 10,000 images in the database, each cluster on average contains 100 regions.

To test the efficiency of our indexing scheme, we compare the retrieval time and accuracy of our framework with and without indexing schemes. The results are shown in Table 1 and Figure 3, respectively. IF denotes indexing using traditional inverted files, while MIF denotes our modified inverted file strategy described in Section 3.2. In the sequential search that is denoted by SEQ, all the images in the database are compared with the query using EMD. As shown in Table1, MIF and IF are four and five times faster than SEQ respectively. Contrasting to IF, which degrades the performance to the extent that can not be ignored, MIF is almost the same as SEQ.

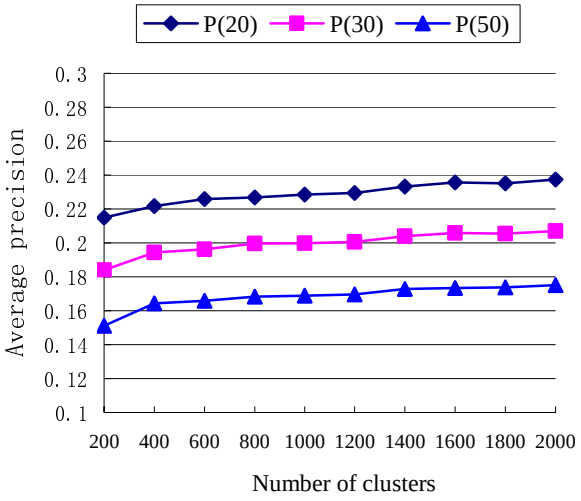

Figure 2. Average precisions for different sizes of the codebook

Table 1. Retrieval time of our framework with and without indexing schemes.

	SEQ	IF	MIF
Search Time (ms)	351	70	91

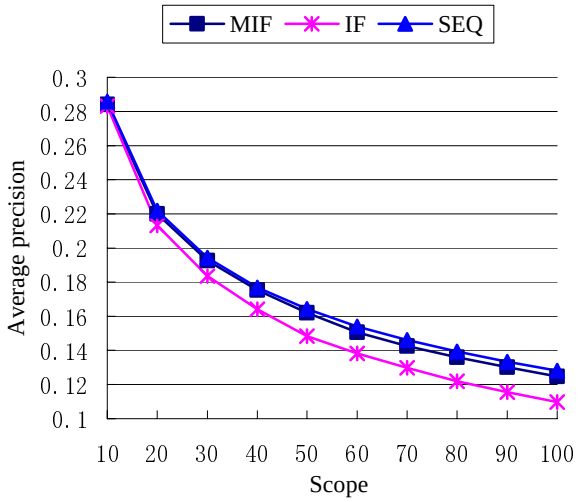

Figure 3. Accuracy comparison of two indexing schemes and sequential search. IF denotes the inverted file strategy, and MIF denotes the proposed modified inverted file strategy. SEQ denotes the sequential search.

To show the effectiveness of our learning algorithms, we simulated users' feedback as follows. For a query image, 5 iterations of user-and-system interaction were carried out. At each iteration, the system examined the top 30 images that are most similar to the optimal query, excluding those positive examples

labeled in previous iterations. Images from the same category as the initial query image were used as new positive examples. At next iteration, all positive images were placed in top ranks directly, while others were compared with the optimal query again and ranked according to their distance values. As stated in Section 4.2, we put more emphasis on latest positive examples. To see the effect of this bias, we rewrite formula (2) to be:

$$\vec{I}_{opt} = \beta(\gamma \sum_{k=1}^{m} \vec{I}_k + (1-\gamma) \sum_{k=m+1}^{n} \vec{I}_k) \quad (6)$$

where $0 \le \gamma < 1$ and plays a similar role as α. We let γ be a constant and gradually increase its value from 0 to 0.9 with each value of γ corresponds to an algorithm. The accuracy of the algorithms after 2-5 iterations of feedback is shown in Figure 4. Since there are no prior positive examples after one iteration, the performance of the algorithms will be the same and therefore not compared. The accuracy here and also in the rest of the paper means the average precision within top 50 images, i.e. average P(50). From the figure, we can see that the accuracy drops consistently as the value of γ increases. The result suggests that the more emphasis we put on the latest positive examples the better retrieval result we will get. Figure 5 demonstrates the accuracy comparison between the proposed weighting scheme, i.e. adaptively choosing α to reflect the bias on latest positive examples, and an equally weighting scheme, i.e. $\alpha = 1$. As expected, the proposed strategy is clearly better than the other one.

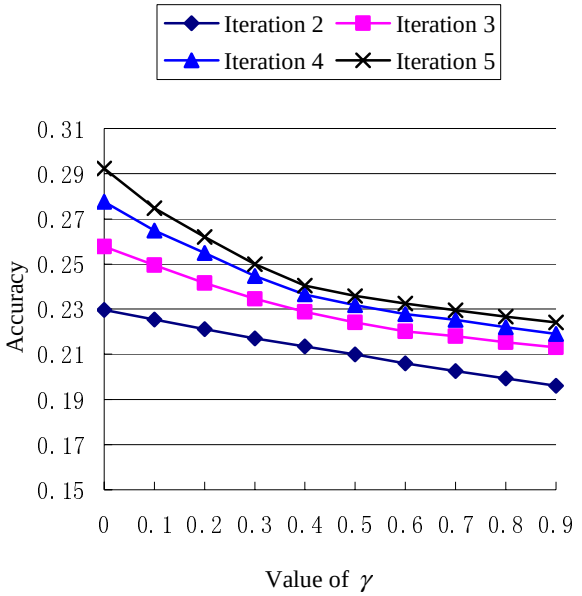

Figure 4. The accuracy of the algorithms after 2-5 iterations of feedback with different γ values

Considering that it is difficult to design a fair comparison with existing region-based image retrieval systems that use relevance feedback, such as the FourEyes [21] system whose propose is annotation and the IDQS [40] system that depends on manually defined queries, we compared our feedback method with the one proposed by Rui Yong in [29]. It is a combination and improvement of its early version and MindReader [13], and reflects the performance of some state-of-the-art relevance feedback algorithms based on only positive examples. Since [29] requires global features, two representative ones are used. One is the color moments (CM) computed in the same way as in Section 2.2 and the other is the correlogram (CG) [12]. For the latter, we consider the RGB color space with quantization into 64 colors. The distance set D = {1, 3, 5, 7} is used for computing the autocorrelograms, which results in a feature vector of 256 dimensions. Since correlogram properly incorporates spatial information into color histogram, it has been proven to be one of the most effective features in CBIR. To show the effectiveness of our short-term learning algorithm, the continuous learning stage is skipped, that is, only the original features are used in the computation of the optimal query. The results are shown in Figure 6. Before any feedback, our algorithm is worse than Rui's method using CG, but a little better than that using CM. After five iterations of feedback, our algorithm boosts its accuracy (12%) more than Rui's method using both CG (10%) and CM (8%), which means that our algorithm has more potential.

To evaluate the continuous learning stage, the following simulation was performed.

Step 1: 10,000 random query and feedback sessions were carried out using continuous learning scheme.

Step 2: Based on the original query set QS. Two new query subsets are formed: $QS_U = \{I_i \mid \vec{I}_i^L \neq \vec{I}_i^O, 1 \le i \le 1000\}$ and $QS_O = \{I_i \mid \vec{I}_i^L = \vec{I}_i^O, 1 \le i \le 1000\}$, where $\vec{I}_i^O$ and $\vec{I}_i^L$ are the original and learned feature of I_i. The queries in QS_U are the ones that have been updated, while those in QS_O are the ones that have not been updated.

Step 3: Evaluate the performance of our framework with (CL) and without (NCL) continuous learning stage on three query set: QS_U, QS_O and QS.

The accuracy comparison of CL and NCL on two query subsets is shown in Figure 7. As shown in the Figure, for the queries that have been updated, 5% improvement of the accuracy could be gained without any interaction, and consistently better performance could be achieved if feedback was provided. On the other hand, for the queries that have not been updated, CL performs the same as NCL before any feedback, but if feedback exists, CL yields better performance after one iteration and onwards. The performance of CL on the whole query set is also evaluated and compared with NCL and Rui's method. The results are shown in Figure 6. It can be seen from the figure that CL is much better than NCL and Rui's method using CM. Compared with Rui's method using CG, although CL is slightly worse after one interaction, it consistently yields better performance after two interactions and the accuracy after 5 interactions is higher than that of [29] by 4%. The results of the simulation shows that the continuous learning stage makes our framework starting from a higher level and boosts the retrieval performance in a faster way.

7. CONCLUSIONS AND FUTURE WORK

The main contributions of this work are in identifying and building various components required by a working system for

efficient and effective retrieval of images. Based on a VQ scheme and a modified inverted file strategy, two image representations are proposed with one being compact and the other being sparse but uniform. The compact representation facilitates both the storage and indexing stages, which makes our framework efficient and scalable. On the other hand, the uniform representation enables a simple but effective on-line learning algorithm. By accumulating the training across sessions with the users, the system built on the framework improves over time and can solve similar problems better the next time. Tested on large-scale image databases, the framework has demonstrated high efficiency, accuracy, and scalability.

We are currently investigating the extension of our proposed framework in the following aspects. (1) Integrating more sophisticated features including both visual and textual ones [20, 30]. (2) Allowing parallel access to features [39]. The inverted file facilitates parallel access to features as there is no need for writing access or synchronization. As a result, it allows the search load to be balanced on different computers, and to have smaller inverted files to access. (3) Using negative examples. Without a doubt, negative examples can provide valuable discriminative information. The question is how to model and utilize them in the right way [42]. (4) Developing more reasonable region weighting schemes. We have developed a region-weighting scheme that learns the weighting of regions based on relevance feedback. It has shown its effectiveness both with [15] and without [14] relevance feedback when there are enough training examples, i.e. user inspects and marks enough images (e.g. top 50). We have implemented it in the current framework and found that when only few examples are available, the algorithm does not work very well. The reason we think is that it is a probabilistic method, which will become meaningless given too few training examples. Although no way has been found to thoroughly resolve the problem, the idea of continuous learning, we believe, will be helpful.

8. ACKNOWLEDGMENTS

We would like to thank Yossi Rubner for his source code of EMD [28].

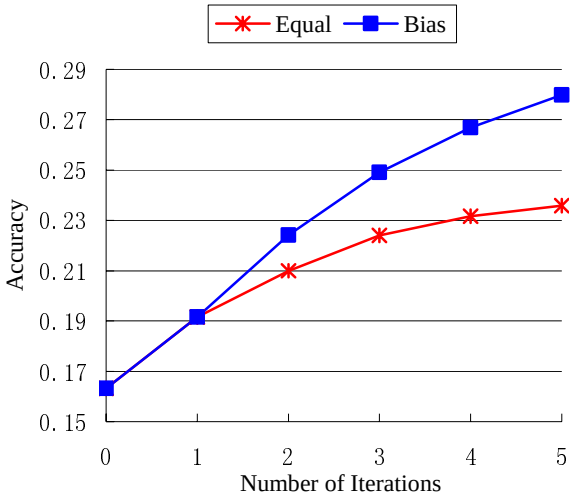

Figure 5. Accuracy comparison of the weighting schemes with (bias) and without (Equal) bias on latest positive examples

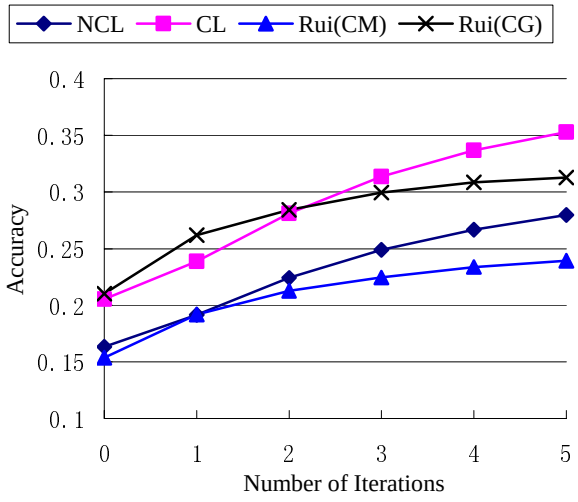

Figure 6. Accuracy comparison of our learning algorithm and Rui Yong's one

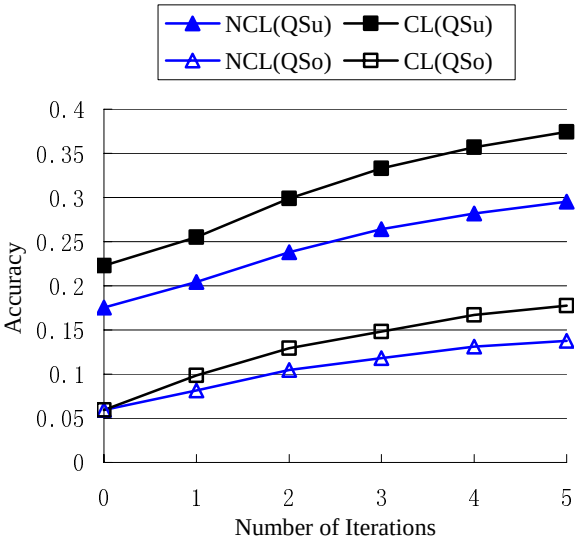

Figure 7. Accuracy comparison of our learning scheme with and without continuous learning for two kinds of queries: queries having been updated and queries having not been updated

9. REFERENCES

[1] Baeza-Yates, R., and Ribeiro-Neto, B., "Modern Information Retrieval". Addison-Wesley, June 1999.

[2] Bartolini, I., Ciaccia, P., and Waas, F., "FeedbackBypass: A New Approach to Interactive Similarity Query Processing", 27th International Conference on Very Large Data Bases, Roma, Italy, 2001.

[3] Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B., "The R*-tree: An efficient and robust access method for points and rectangles," Proc. ACM SIGMOD, pp. 322-331, Atlantic City, NJ, 23-25 May 1990.

[4] Carson, C. et al, “Blobworld: a system for region-based image indexing and retrieval,” Third Int. Conf. On Visual Information Systems, 1999.

[5] Ciaccia, P., Patella, M., Zezula, P., “M-tree: An efficient access method for similarity search in metric spaces,” Proc. Int. Conf. on Very Large Databases, Athens, Greece, 1997.

[6] Deng, Y., Manjunath, B. S. and Shin, H., “Color Image Segmentation”, in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '99, Fort Collins, CO, vol.2, pp.446-51, June 1999.

[7] Gersho, A., and Gray, R. M., “Vector Quantization and Signal Compression”, Kluwer Academic Publishers, 1992.

[8] Gong, Y., Zhang, H.J., Chuan, H. C., and Sakauchi, M., “An Image Database System with Content Capturing and Fast Image Indexing Abilities”. In Proceedings of IEEE International Conference on Multimedia Computing and Systems, pages 121-130, Boston, May 1994.

[9] Greenspan, H., Dvir, G. and Rubner, Y., “Region Correspondence for Image Matching via EMD Flow”, IEEE workshop on Content-based Access of Image and Video Libraries, June 2000.

[10] Guttman, A., “R-trees: A dynamic index structure for spatial searching,” Proc. ACM SIGMOD, pp. 47-57, Boston, MA, June 1984.

[11] Hitchcock, F. L., “The distribution of a product from several sources to numerous localities”. J. Math. Phys., 20:224–230, 1941.

[12] Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R.. “Image indexing using color correlograms”. In Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec., pages 762--768, 1997.

[13] Ishikawa, Y., Subramanya, R. and Faloutsos, C., “Mindreader: Query databases through multiple examples,” in Proc. of the 24th VLDB conference, (New York), 1998.

[14] Jing, F., Zhang, B., Lin, F.Z., Ma, W.Y., Zhang, H.J., “A Novel Region-Based Image Retrieval Method Using Relevance Feedback”, Proc. 3rd ACM Intl Workshop on Multimedia Information Retrieval (MIR), 2001.

[15] Jing, F., Li, M., Zhang, H.J., Zhang, B., “Learning Region Weighting from Relevance Feedback in Image Retrieval”, Proc. the 27th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2002.

[16] Jing, F., Li, M., Zhang, H.J., Zhang, B., “Region - based relevance feedback in image retrieval”, Proc. IEEE International Symposium on Circuits and Systems (ISCAS), 2002.

[17] Jing, F., Li, M., Zhang, H.J., Zhang, B., “Unsupervised Image Segmentation Using Local Homogeneity Analysis”, submitted to IEEE International Conference on Multimedia & Expo (ICME) 2002.

[18] Katayama, N., Satoh, S., “The SR-tree: An index structure for high-dimensional nearest neighbor queries,” Proc. ACM SIGMOD pp. 369-380, Tucson, AZ, 1997.

[19] Lee, C., Ma, W.Y., and Zhang, H.J., "Information Embedding Based on user's relevance Feedback for Image Retrieval," Proc. of SPIE Photonics East, 1998.

[20] Ma, W.Y., and Manjunath, B.S., “NETRA: A toolbox for navigating large image databases”, in Proc. IEEE International Conference on Image Processing, Santa Barbara, California, Vol. I, pp. 568-571, Oct 1997.

[21] Minka, T.P., Picard, R.W., “Interactive Learning Using A Society of Models", Pattern Recognition, vol. 30, no. 4, pp. 565-581, April 1997.

[22] Minka, T.P.. “An image database browser that learns from user interaction”. Master's thesis, MIT Media Laboratory, 20 Ames St., Cambridge, MA 02139, 1996.

[23] Muller, H., Squire, D. M., Muller, W., and Pun, T., “Efficient access methods for content-based image retrieval with inverted files,” in Panchanathan et al. 23 (SPIE Symposium on Voice, Video and Data Communications).

[24] Natsev, A., Rastogi, R., and Shim, K., “WALRUS: A similarity retrieval algorithm for image databases”, Proc. ACM SIGMOD Int. Conf. on Management of Data, 1999.

[25] Niblack, W. et al. “The QBIC project: querying images by content using color, texture, and shape”, in Proc. SPIE, vol. 1908, pp. 173-187, San Jose, February 1993.

[26] Pentland, A., Picard, R., and Sclaroff, S., “Photobook: Content-based Manipulation of Image Databases.” In SPIE Storage and Retrieval for Image and Video Databases II, number 2185, Feb. 1994, San Jose, CA.

[27] Rocchio, J. J., “Relevance feedback in information retrieval”. In The SMART Retrieval System-- Experiments in Automatic Document Processing, pp. 313-323, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc.

[28] Rubner, Y., Tomasi, C., and Guibas, L., “A Metric for Distributions with Applications to Image Databases.” Proceedings of the 1998 IEEE International Conference on Computer Vision, January 1998.

[29] Rui, Y., and Huang, T.S., “Optimizing Learning in Image Retrieval”, Proceeding of IEEE int. Conf. On Computer Vision and Pattern Recognition, Jun. 2000.

[30] Smith, J.R., and Chang, S.-F., “VisualSEEk: a fully automated content-based image query system”, in Proc. ACM Multimedia, Boston, MA, Nov. 1996.

[31] Smith, J. R., and Li, C.-S., “Image Classification and Querying Using Composite Region Templates,” Computer Vision and Image Understanding, Vol. 75, No. 1/2 (1999), pp. 165-174.

[32] Stricker, M., and Orengo, M., “Similarity of Color Images”, in Storage and Retrieval for Image and Video Databases, Proc. SPIE 2420, pp 381-392, 1995.

[33] Su, Z., Li, S., and Zhang, H.-J., “Extraction of Feature Subspaces for Content-Based Retrieval Using Relevance Feedback”. Proc. ACM International Multimedia Conference (MM '01), Ottawa, Canada, October 2001.

[34] Thomas, M., Carson, C., and Hellerstein, J., “Creating a Customized Access Method for Blobworld”. In Proc. of the 16th Int. Conf. on Data Engineering, San Diego, USA, page 82, 2000.

[35] Vasconcelos, N., and Lippman, A., “Learning from user

feedback in image retrieval system", in Proc. of NIPS'99, Denver, Colorado, 1999.

[36] Wang, J.Z., Du, Y.P., "Scalable Integrated Region-Based Image Retrieval Using IRM And Statistical Clustering", Proc. ACM and IEEE Joint Conference on Digital Libraries, Roanoke, VA, ACM, June 2001.

[37] Wang, J. Z., Li, J., and Wiederhold, G., "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, 2001.

[38] White, D.A., and Jain, R., "Algorithms and strategies for similarity retrieval," Storage and Retrieval in Image, and Video Databases, vol. 2,060, pp. 62-72, 1996

[39] Witten, I.H., Moffat, A., and Bell, T.C., "Managing gigabytes: compressing and indexing documents and images", Van Nostrand Reinhold, 115 Fifth Avenue, New York, NY 10003, USA, 1994.

[40] Wood, M.E., Campbell, N.W., and Thomas, B.T., "Iterative refinement by relevance feedback in content based digital image retrieval," in Proceedings of The Fifth ACM International Multimedia Conference (ACM Multimedia 98), pp. 13--20, (Bristol, UK), September 1998.

[41] Wu, P., Manjunath, B.S., "Adaptive Nearest Neighbor Search for Relevance Feedback in Large Image Datasets", Proc. ACM International Multimedia Conference (MM '01), Ottawa, Canada, October 2001.

[42] Zhou, X. S., and Huang, T. S., "Comparing Discriminate Transformations and SVM for Learning during Multimedia Retrieval", ACM Multimedia2001, Sept. 30-Oct 5, 2001, Ottawa, Ontario, Canada, 2001.

[43] Zhu, L., "Keyblock: an approach for content-based image retrieval". ACM Multimedia2000, 157-166.