A Noise-Resilient Collaborative Learning Approach to Content-Based Image Retrieval

Xiaojun Qi,[1,*] Samuel Barrett,[2] Ran Chang[1]
[1]*Department of Computer Science, Utah State University, Logan, UT 84322-4205*
[2]*Department of Computer Science, The University of Texas at Austin, Austin, TX 78712*

We propose to combine short-term block-based fuzzy support vector machine (FSVM) learning and long-term dynamic semantic clustering (DSC) learning to bridge the semantic gap in content-based image retrieval. The short-term learning addresses the small sample problem by incorporating additional image blocks to enlarge the training set. Specifically, it applies the nearest neighbor mechanism to choose additional similar blocks. A fuzzy metric is computed to measure the fidelity of the actual class information of the additional blocks. The FSVM is finally applied on the enlarged training set to learn a more accurate decision boundary for classifying images. The long-term learning addresses the large storage problem by building dynamic semantic clusters to remember the semantics learned during all query sessions. Specifically, it applies a cluster-image weighting algorithm to find the images most semantically related to the query. It then applies a DSC technique to adaptively learn and update the semantic categories. Our extensive experimental results demonstrate that the proposed short-term, long-term, and collaborative learning methods outperform their peer methods when the erroneous feedback resulting from the inherent subjectivity of judging relevance, user laziness, or maliciousness is involved. The collaborative learning system achieves better retrieval precision and requires significantly less storage space than its peers.

1. INTRODUCTION AND RELATED WORK

With the rapidly growing number of digital images on the Internet and in digital libraries, the need for large image database management and effective image retrieval tools has been growing. Content-based image retrieval (CBIR) techniques are viable solutions to finding desired images from multimedia databases. However, the semantic gap between low-level visual features and high-level semantic meanings remains a challenging issue to be solved. Humans bridge this gap using their knowledge about the world. However, computer vision techniques have been

*Author to whom all correspondence should be addressed; e-mail: Xiaojun.Qi@usu.edu.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 26, 1153–1175 (2011)

View this article online at wileyonlinelibrary.com. • DOI 10.1002/int.20503

struggling to bridge this gap ever since the advent of the computer vision. This paper focuses on bridging the semantic gap using a novel collaborative learning approach.

Present CBIR techniques can be classified into four categories[1−3]: global feature based,[4−7] region level feature based,[8−16] object level feature based,[10,16−21] and relevance feedback (RF) based.[7,18,22−25] Among these, RF-based techniques have been widely used to bridge the semantic gap by learning the user's query concept (i.e., query targets). These techniques first solicit the user's relevance judgments of retrieved images at each feedback iteration. They then refine retrieval results at the next iteration by applying short-term and/or long-term learning techniques to previous judgment information. A query session is finished when the user is satisfied with the retrieval results.

Short-term learning techniques aim to find out which images are relevant to the user's query over the course of a single query session. Query updating and statistical learning techniques are two common short-term learning techniques. By using the user's subjectively labeled information, query updating techniques improve the representation of the query itself, while statistical learning techniques improve the classification boundary between relevant and irrelevant images or predict the relevance of unlabeled images which are attainable during the training stage. Examples of query updating techniques include query reweighing,[26] query shifting,[27] and query expansion.[28] Examples of statistical learning techniques include inductive learning and transductive learning. Specifically, inductive learning techniques, e.g., decision tree learning,[29] Bayesian learning,[30−32] support vector machine (SVM) learning,[33] fuzzy SVM (FSVM) learning,[34] and boosting,[35] create various classifiers which separate the relevant (i.e., positive) and irrelevant (i.e., negative) images and generalize well on unlabeled images. The transductive learning technique, e.g., manifold-ranking-based learning,[7] uses each unlabeled image as a vertex in a weighted graph to propagate the ranking score of labeled images. However, all these short-term learning techniques are limited in their usefulness. Query updating techniques exclusively use low-level features to update the query concept and therefore cannot capture the semantic meaning of an image and cannot achieve satisfactory retrieval results. Statistical learning techniques cannot achieve good and reliable classification/prediction results due to the small amount of user-labeled training images. In addition, the short-term learning techniques cannot remember the user's feedback after a query session and therefore cannot utilize the user's feedback in future retrievals.

To overcome the above shortcomings, long-term learning techniques have been proposed to discover the relationships among images over the course of multiple queries. They use historical retrieval experiences over many search sessions to estimate the semantic relationships of images. State-of-the-art long-term learning techniques include the statistical correlation technique,[36] the semantic space-based technique,[37] the log-based technique,[38] and the memory learning technique.[39] For example, the statistical correlation technique uses a triangular matrix to store semantic correlation collected from the statistics of users' feedback information. Other three techniques use a square matrix to store measurements of the accumulated semantic correlation between images. The memory learning technique further forms a knowledge model to learn hidden semantic relations. As the size of the database

increases, the size of the matrix increases as well to store memorized feedback information. The matrix may be sparse if all the queries fall into a few semantic categories. This sparsity may deteriorate the learning performance for a large-scale database. In addition, erroneous feedback, resulting from inherent subjectivity of judging relevance, user laziness, or maliciousness, may also lead to store incorrect semantic information and degrade the retrieval accuracy.

To address the limitations of current CBIR systems, we propose a RF-based, noise resilient collaborative learning approach to bridging the semantic gap. Our system effectively uses the RF as a tool to learn semantic clusters (SCs) of the image database. The proposed method applies a short-term block-based FSVM learning technique to find a more accurate decision boundary to classify images as relevant or irrelevant to the query at each feedback iteration. It also applies a cluster-image weighting algorithm to find the images most semantically related to the query image. The retrieval results from these two techniques are then combined to improve the retrieval accuracy. At the end of each query session, a long-term dynamic semantic clustering (DSC) technique is employed to assign user-labeled images into appropriate SCs to remember semantic relationships among these images. These SCs efficiently store accumulated users' semantic relevance information and significantly aid the RF task by incorporating the stored semantic knowledge. Our system can scale well to a large image database since any database only contains a relatively small number of semantic categories compared to the number of database images. Here, we summarize the major functionality and contributions of the proposed system as follows:

- To address the problem of a small number of samples, our short-term learning technique chooses additional blocks of images to enlarge the training set and learn the user's query concept from a more accurate low-level visual perspective. This learning technique is the first attempt to use block-based FSVM to address the small sample problem in the RF learning.
- To address the large storage problem, our long-term learning technique uses dynamic SCs to efficiently store the accumulated semantic relationships among images. This learning technique is the first attempt to use DSC to compactly store the historical retrieval experiences over many search sessions.

The rest of the paper is organized as follows. Section 2 describes our proposed collaborative image retrieval framework by incorporating both short-term and long-term learning techniques. Section 3 presents the RF-based short-term block-based FSVM learning technique. Section 4 presents the RF-based long-term DSC technique. Section 5 demonstrates the effectiveness of our proposed system and shows the extensive experimental results of comparing our system with peer systems. Section 6 gives concluding remarks and presents future research directions.

2. THE PROPOSED COLLABORATIVE IMAGE RETRIEVAL FRAMEWORK

The algorithmic flow of our collaborative learning framework is as follows: For each database image, two sets of features are extracted offline and saved in

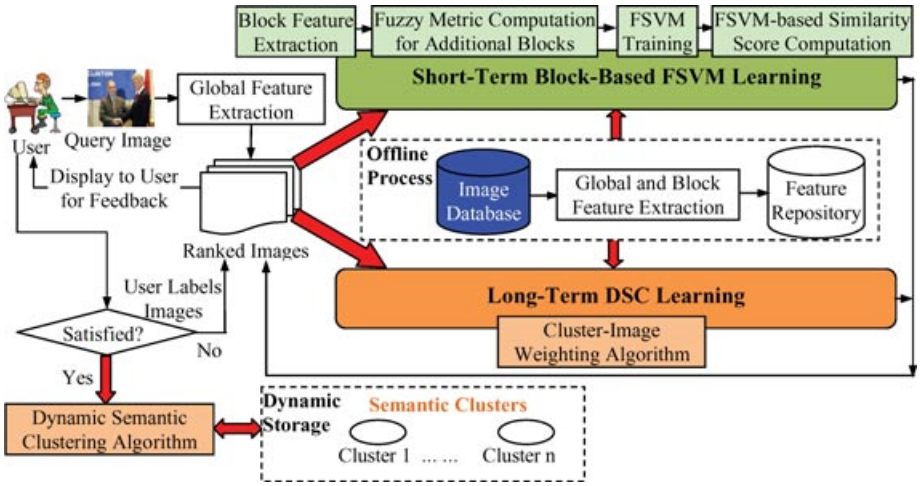

Figure 1. The proposed collaborative image retrieval framework.

the feature repository. One set contains global color and texture features for initial retrieval. The other set contains block-based color, edge, and texture features for RF-based retrieval. When the user supplies a query image q, the system returns the top n images, which are then classified by the user as either relevant or irrelevant to query q. This process continues for a few feedback iterations, or until the user is satisfied with the retrieval results. For each iteration step, the proposed system simultaneously performs short-term block-based FSVM learning and long-term DSC learning to return the top n images. The similarity between query q and an arbitrary image D_i in the database, denoted as $S(q, D_i)$, is defined as

$$S(q, D_i) = w_{\text{short}} \cdot S_{\text{short}}(q, D_i) + w_{\text{long}} \cdot S_{\text{long}}(q, D_i), \tag{1}$$

where $S_{\text{short}}(q, D_i)$ and $S_{\text{long}}(q, D_i)$, respectively, measure the short-term and long-term similarity scores between q and D_i; w_{short} and w_{long}, respectively, are the weights assigned to the short-term and long-term similarity scores with $w_{\text{short}} = w_{\text{long}} = 0.5$ since our experiments show both short-term and long-term learning techniques equally contribute to the final similarity score. Figure 1 provides an overview of our collaborative framework, which consists of short-term block-based FSVM learning and long-term DSC learning.

Short-term learning first splits all labeled training images into five predefined blocks. Next, it applies the k-means algorithm to group relevant blocks (i.e., the blocks from relevant training images) into k clusters and group irrelevant blocks (i.e., the blocks from irrelevant training images) into another k clusters. The clusters of the relevant blocks may correspond to several semantics (e.g., mountain, beach, etc.) related to the user's query concept. The clusters of the irrelevant blocks may correspond to several semantics that do not match the user's query concept. For each cluster, short-term learning chooses the nearest neighboring blocks from the other

unlabeled database images. These added blocks are given the same label as their closest cluster, and the accuracy of their labels is estimated by a fuzzy metric. These blocks and their label accuracy (i.e., weights) are used to enlarge the training set for the FSVM. We choose the FSVM for training since it captures the nature of the data better than the SVM, where the partial or ambiguous membership is a common phenomenon.[34] Finally, we compute $S_{\text{short}}(q, D_i)$ by totaling the directed distance of five predefined blocks of D_i to the classification boundary.

Long-term DSC learning first applies a cluster-image weighting algorithm during each iteration to estimate SCs that the query image represents and to compute $S_{\text{long}}(q, D_i)$. The system then returns the top n images ranked by combining short-term and long-term similarity scores. In our system, a higher similarity score corresponds to a stronger similarity to the query. At the end of the query session, long-term DSC learning treats the set of images that have been labeled during all iteration steps as a new SC. It next computes the occurrence-based similarity and dissimilarity measures between the new SC and each existing SC, and finds all existing SCs whose similarity measure sufficiently outweighs the dissimilarity measure. The new SC is then merged with the SCs found in the previous step by incorporating its semantic information into the corresponding existing clusters. The merged clusters are further compared with the other clusters for any additional merging possibilities to ensure that all the SCs are distinct. The resulting distinct SCs store the memorized feedback information in a compact manner, thus facilitating the cluster-image weighting algorithm to learn the semantic content of the images.

3. SHORT-TERM BLOCK-BASED FSVM LEARNING

The short-term learning framework contains four components: feature extraction, fuzzy metric computation for additional blocks, FSVM training, and FSVM-based similarity score computation. In the following, we explain these four components in detail.

3.1. Feature Extraction: Global and Block Feature Extraction

We use the expanded MPEG-7 150-bin edge histogram descriptor (EHD) and the 64-bin ($8 \times 2 \times 4$) HSV-based scaled color descriptor (SCD)[40] to extract global low-level features for each image. The sum of the normalized weighted Euclidean distance of EHD and the normalized Euclidean distance of SCD between q and D_i is used to measure $S_{\text{short}}(q, D_i)$.

For the following RF sessions, we use block-based local features to represent each image from a different perspective. In our system, we divide an image into five predefined non-overlapping blocks, whose layout is shown in Figure 2. This blocking scheme has shown to be efficient and effective in our prior image annotation system.[41] Even though this blocking system may divide an object into different blocks or put multiple objects into one block, our proposed collaborative learning will partially resolve this issue by seamlessly utilizing the fuzzy metric, FSVM, the cluster-image weighting algorithm, and the DSC algorithm.

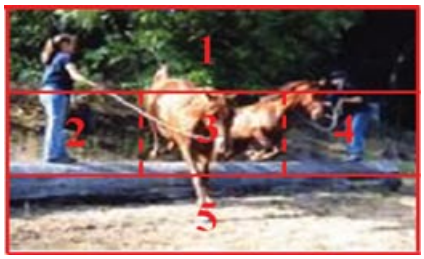

Figure 2. Illustration of the layout of image blocks.

We compute a set of compact low-level features to represent each block. These features were used in our prior CBIR system[42] and were proven to be effective. They consist of 9-D color, 18-D edge, and 9-D texture features. Specifically, we use the first three moments in HSV color space to represent color features. We use the edge direction histogram to represent the edge features in the grayscale image. We use the entropy of each of nine detail subbands to represent texture features in the grayscale image. These local features complement the global features to represent an image from different perspectives and therefore achieve better low-level feature-based retrieval results.

3.2. Fuzzy Metric Computation for Additional Blocks

At each RF session, the user labels each returned image as either relevant or irrelevant to the query image. To address the problem of a small number of images returned at each RF session, we search for additional block-based local features to expand the training set to more accurately learn the user's query concepts from the visual perspective. Expanding the training set also reduces the algorithm's sensitivity to noise in the images. We then assign each additional block a label based on its closeness to relevant and irrelevant clusters. However, these labels may not be in accordance with the user's query concept. Therefore, we predict the accuracy of the labels and assign fuzzy weights to the labels by measuring a fuzzy metric from the following two perspectives: (1) The distance ratio between the additional block to the nearest cluster center of the same label and to the nearest cluster center of the opposite label. If an additional block is closer to the cluster center of the same label and farther to the cluster center of the opposite label, its assigned label is more likely to be correct. (2) The classification results of the SVM and the distance to the classification boundary of the SVM. If the classification result is consistent with the current label, a larger distance to the classification boundary means that the assigned label is more likely to be correct. If the classification result is inconsistent with the current label, a larger distance to the classification boundary means that the assigned label is less likely to be correct. These two measures of the label accuracy computed from the distance ratio and active SVM learning are incorporated to measure the accuracy of the predicted label (i.e., pseudo label). That is, they provide relative

1. The user labels each of the returned top n images as either relevant or irrelevant to query q.
2. Retrieve five block-based local features for each labeled image, where each of five blocks retains the same label (either relevant or irrelevant) as its parent image.
3. Apply the k-means algorithm to cluster all relevant blocks, where k is empirically determined to be eight as in[43]. All irrelevant blocks are similarly grouped into eight clusters by applying the k-means algorithm.
4. For each of the eight relevant clusters, choose the five nearest blocks from the remaining database images to expand the positive training set. The negative training set is expanded similarly.
5. Assign each of the additional unlabeled blocks (i.e., 40 relevant blocks and 40 irrelevant blocks) the same label as its associated cluster.
6. Train an initial SVM classifier using all block-based local features of the user labeled n images.
7. Employ a fuzzy metric to evaluate the relevance of each pseudo-labeled block x_p (i.e., the additional block with a predicted label as determined in Step 5).
   - 7.1. Compute *SameD,* the distance of x_p to the nearest cluster center of the same label.
   - 7.2. Compute *OppD,* the distance of x_p to the nearest cluster center of the opposite label.
   - 7.3. If (*SameD* < *OppD*), set $w_1(x_p)$ to be $e^{-(\alpha_1 r(x_p))}$. Otherwise, set $w_1(x_p)$ to be 0. Here, $w_1(x_p)$ is a measure of the accuracy of the label of x_p, $r(x_p)$ is the ratio between *SameD* and *OppD*, and α_1 is a scaling factor and is empirically set to be 1 in our system.
   - 7.4. If the pseudo-label is the same as the label determined by the SVM, set $w_2(x_p)$ to be $1/(1+e^{-\alpha_2 y})$. Otherwise, set $w_2(x_p)$ to be $1/(1+e^{\alpha_2 y})$. Here, $w_2(x_p)$ is a measure of how well the pseudo-label agrees with the label determined by the SVM, y is the distance from x_p to the SVM boundary, and α_2 is a scaling factor and is set to be 1 in our system.
   - 7.5. Compute the final predicted weight of x_p as $g(x_p) = w_1(x_p) \times w_2(x_p)$.

Figure 3. Algorithmic summary of fuzzy metric computation for additional blocks.

correct fuzzy information on the additional chosen image blocks. The algorithmic overview of the fuzzy metric computation for additional blocks is summarized in Figure 3.

3.3. FSVM Training

The FSVM is trained using the local features of the blocks of all returned images and additional blocks, their labels (1 or -1), and their membership values. In an FSVM, each training sample is associated with a fuzzy membership value μ_i ranging from 0 to 1. This value reflects the confidence degree of the class label of the training data. The higher the value, the more confident the class label. We assign a membership of 1 to blocks of user-labeled images, and assign a membership of $g(x_p)$ to each additional pseudo-labeled block x_p. The FSVM solves the following optimization problem for m training data of the form (x_i, y_i, μ_i), where x_i represents the ith local feature, y_i represents the label (i.e., -1 or 1) of the ith local feature, and μ_i represents the fuzzy membership value of the ith local feature:

$$\min_{\omega,b,\xi} \left(\frac{1}{2}\omega^T\omega + C\sum_{i=1}^{m} \mu_i \xi_i \right), \tag{2}$$

$$\text{subject to } y_i(\omega^T \phi(x_i) + b) > 1 - \xi_i, \quad \xi_i > 0, i = 1, 2, \ldots, m.$$

Here, C is the penalty parameter of the error term, w is the coefficient vector, b is a constant, and ξ_i is a slack variable for handling nonseparable training data. The error term ξ_i is scaled by fuzzy membership μ_i. As a result, these membership values weigh the soft penalty term in the cost function of SVM. That is, training samples with larger membership values have more impact on the training than those with smaller values. The nonlinear FSVMs with the Gaussian radial basis function (RBF) kernel are used in our system due to their excellent results compared with other kernels.[44] This RBF kernel is defined as

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) = \exp(\gamma \|x_i - x_j\|^2), \quad \gamma > 0. \tag{3}$$

Two FSVMs related parameters C and γ are predetermined by applying the three-fold cross-validation and grid-search algorithms[45] on exponentially growing sequences of C and γ on several sets of prelabeled training images. In our grid-search algorithm, we used the balance error (i.e., the ratio of the number of false positives to the number of negatives plus the ratio of the number of false negatives to the number of positives) to decide the best pair of C and γ. This balance error addresses the unbalanced data issues. The pair that gives the minimum cross-validation error is selected as the optimal parameters and is used in our CBIR system.

3.4. FSVM-Based Similarity Score Computation

The sum of the directed distance from each block of an image D_i to the trained FSVM boundary is computed. Here, the directed distance is a distance with a positive or negative sign where the positive distance falls on the positive side of the trained boundary and the negative distance falls on the negative side of the trained boundary. The normalized total directed distance of each image D_i is used to measure $S_{\text{short}}(q, D_i)$ for selecting top images during the RF sessions.

3.5. Complexity Analysis of Short-Term Learning

The complexity of computing global features is $O(g)$ with g being the dimension of global features (i.e., $g = 214$). The complexity of computing block features is $O(d)$ with d being the dimension of local features (i.e., $d = 36$). The complexity of computing fuzzy metric is $O(KNd)$, where K is the number of clusters (i.e., $K = 16$) and N is the total number of image blocks (i.e., $N = 125$). The complexity of training FSVM is the same as training SVM since the dual problem of FSVM is a quadratic programming problem, similar to that of SVM. So, the complexity of the FSVM classifier is $O(N_{sv}^3 + NN_{sv}^2 + dNN_{sv})$,[46] where N_{sv} is the number of SVs (i.e, $N_{\text{sv}} \ll N = 125$ in most cases). Once trained, the classification step of FSVM involves only simple calculation, which is cost efficient.

The algorithm was implemented using Matlab 7.6.0.324 (R2008a) on a Pentium IV Quad CPU at 2.66GHz PC running Windows XP operating system. On average, it takes 0.11 and 0.49 seconds to compute global and block features for an image, respectively. For the 12,000-image database, the average retrieval time per query using short-term learning is 1.3876 seconds for each iterative RF step. This

computational time can be reduced to around 0.0694 seconds if the algorithm is implemented in C language.

4. LONG-TERM DYNAMIC SEMANTIC CLUSTERING LEARNING

During the image retrieval process, our system dynamically constructs SCs based on users' RF, where each SC corresponds to a high-level semantic category. The construction of SCs is mainly based on a valid assumption[36] that if two images are jointly labeled as positive examples in a search session, it is likely that they contain similar semantic content and belong to the same semantic categories. The higher the number of RF sessions in which the two images are labeled as positive examples, the higher the semantic similarity between them. Three intuitive observations also guide this construction: (1) The semantic relationship among images is complicated and therefore it is impractical to store all possible semantic relationships, which may require a lot of storage space for a large-scale image database. (2) An image normally has several semantics (i.e., contains several interesting objects) and therefore belongs to several semantic categories. (3) Humans tend to classify objects into semantic categories and remember how well each object belongs to each category.[47] In our proposed long-term learning framework, we first apply the cluster-image weighting algorithm during each RF iteration step to estimate the relationship of an image to each SC and evaluate the semantic similarity of two images. We then apply the DSC algorithm at the end of each query session to identify and merge the SCs that represent the same semantic concept. This algorithm increases the semantic information provided by the clusters and reduces the storage space for remembering the historical feedback experiences. In the following, we explain these two algorithms in detail.

4.1. DSC Algorithm

The basic flow of our DSC algorithm is as follows:

(1) Initially set the SCs as empty (i.e., no query has been submitted so far.)
(2) Each query session generates two sets of images: a set relevant to the query image (i.e., a positive set ***PosSet***) and a set irrelevant to the query image (i.e., a negative set ***NegSet***). This information is used to create a candidate SC SC_{new}. This cluster is stored as two $N \times 1$ vectors, where N is the total number of images in the database and the index of each cell in the vectors corresponds to the index number of the database images. The first vector, $\boldsymbol{Related}_{\text{new}}$, stores the information related to the images in ***PosSet*** by recording 1s in the cells whose indices correspond to the images in ***PosSet*** and recording 0s in the remaining cells. The second vector, $\boldsymbol{Occurred}_{\text{new}}$, stores the information related to the images in ***PosSet*** and ***NegSet*** by recording 1s in the cells whose indices correspond to the images in ***PosSet*** and ***NegSet***, and recording 0s in the remaining cells. For ease of discussion, we call the cells with nonzero values in both ***Related*** and ***Occurred*** as marked cells or marked images.

(3) For each SC SC_j (including SC_{new} and the existing SCs), compute the relevancy of each marked image $D_{j,i}$ in $\boldsymbol{Related}_j$ of SC_j to SC_j by

$$M(D_{j,i}, SC_j) = \frac{Related(D_{j,i}, SC_j)}{Occurred(D_{j,i}, SC_j)}, \tag{4}$$

where occurred $(D_{j,i}, SC_j)$ is the number of times that image $D_{j,i}$ was returned with other images from cluster SC_j, and related$(D_{j,i}, SC_j)$ is the number of times that image $D_{j,i}$ was labeled as relevant to cluster SC_j.

(4) Estimate the similarity between SC_{new} and each existing SC SC_j by computing their relevancy level (i.e., finding how strongly marked images belong to both clusters) by

$$Sim(SC_{\text{new}}, SC_j) = \max\left(\frac{\sum_{i=1}^{m_1} M(D_{\text{new,i}}, SC_{\text{new}}) * M(D_{\text{new,i}}, SC_j)}{m_1}, \frac{\sum_{i=1}^{n_1} M(D_{j,i}, SC_j) * M(D_{j,i}, SC_{\text{new}})}{n_1}\right), \tag{5}$$

where m_1 is the number of positive images marked in $\boldsymbol{Related}_{\text{new}}$ of SC_{new} and n_1 is the number of positive images marked in $\boldsymbol{Related}_j$ of SC_j.

(5) Estimate the dissimilarity between SC_{new} and each SC_j by computing their irrelevancy level (i.e., finding how strongly marked images that are relevant to one cluster, but irrelevant to the other cluster) by

$$DisSim(SC_{\text{new}}, SC_j) = \max\left(\frac{\sum_{i=1}^{m_1} I(D_{\text{new,i}}, SC_{\text{new}}) * M(D_{\text{new,i}}, SC_j)}{m_1}, \frac{\sum_{i=1}^{n_1} I(D_{j,i}, SC_j) * M(D_{j,i}, SC_{\text{new}})}{n_1}\right), \tag{6}$$

where $I(D_{j,i}, SC_j)$ measures the irrelevancy of each marked image $D_{j,i}$ in $\boldsymbol{Related}_j$ of SC_j to SC_j (i.e., the membership that the image does not belong to the SC) and is computed as

$$I(D_{j,i}, SC_j) = \begin{cases} 1 - M(D_{j,i}, SC_j) & \text{if } M(D_{j,i}, SC_j) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

(6) Compute the difference in similarity and dissimilarity between SC_{new} and each SC_j by

$$Diff(SC_{\text{new}}, SC_j) = Sim(SC_{\text{new}}, SC_j) - DisSim(SC_{\text{new}}, SC_j) \tag{8}$$

(7) Find all existing SC SC_js whose $Diff(SC_{\text{new}}, SC_j)$ is larger than 0.25 (i.e., SC_j and SC_{new} represent the same semantic concept). For each of these clusters, perform the following operations:

(a) Merge SC_j with SC_{new} by summing $\boldsymbol{occurred}_j$ and $\boldsymbol{occurred}_{\text{new}}$ and summing $\boldsymbol{related}_j$ and $\boldsymbol{related}_{\text{new}}$.

(b) Treat this merged cluster as SC_{new} and repeat steps 3–7 to determine whether the enlarged SC overlaps any other clusters.

(8) If no cluster SC_j has $Diff(SC_{new}, SC_j)$ larger than 0.25 (i.e., SC_{new} does not represent the same semantic concept of any existing cluster), add SC_{new} as a new SC.

Here, Equation 8 evaluates the overall similarity between the new cluster and each existing cluster by subtracting their similarity level by their dissimilarity level since any two SCs may share some common semantics and have their own distinct semantics. The higher the overall similarity, the more similar the two clusters, the more likely the two clusters should be merged. The merging threshold of 0.25 in step 7 is empirically determined. A small threshold means more SCs may be considered as similar to the newly constructed SC and merging will occur frequently. A large threshold means fewer SCs may be considered as similar to the newly constructed SC and merging will occur infrequently. We experimented with different thresholds of 0.1, 0.15, 0.2, 0.25, 0.3, and 0.35. The value of 0.25 achieves the best accuracy and the best retrieval time. It also leads to a reasonable number of SCs, which is close to the actual number of semantic categories in the database.

Figure 4 illustrates the basic idea of our proposed DSC algorithm. Let us define each piece of RF as $R^c_{a,b}$, where a indicates iterations for a query session, b indicates the rank of the returned images with 1 being the most similar and m being the least similar to the query, and c indicates the query number with 1 being the first and n being the nth query. This DSC algorithm is mainly responsible for providing the noise-resilient capability since the majority of the feedback is correct and the merging of two SCs is flexible based on the empirically determined threshold. That is, each SC (i.e., before merging and after merging) still stores the relatively correct information.

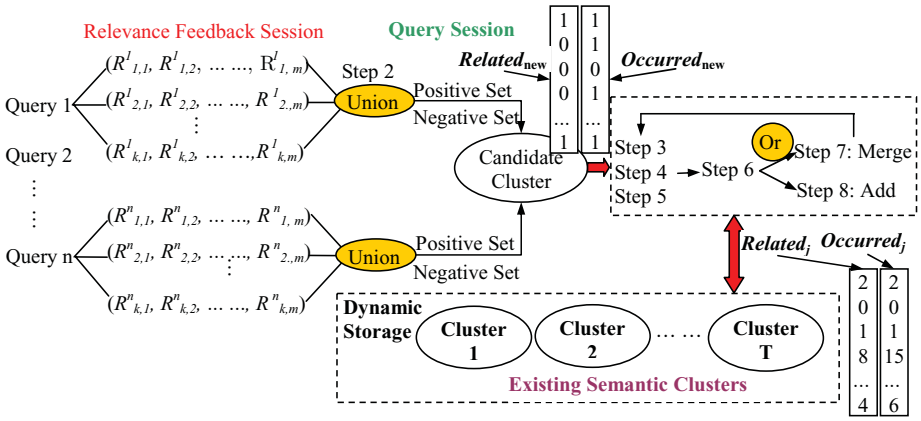

Figure 4. Illustration of DSC algorithm. Note: Each cluster is represented by two vectors, ***Related*** and ***Occurred***.

4.2. Cluster-Image Weighting Algorithm

The cluster-image weighting algorithm is applied to decide which images are most semantically related to query. It incorporates two strategies in its computation: (1) If some relevant images in an existing cluster are marked as relevant to query, it is likely that the other relevant images in the cluster are also relevant. (2) If some relevant images in an existing cluster are marked as irrelevant to query, it is likely that the other relevant images in the cluster are also irrelevant. This enables the system to give high similarity scores to relevant images in clusters the relevant images reside, while giving low similarity scores to relevant images in clusters the irrelevant images reside. At each RF iteration, the returned images are divided into two sets based on the user's labeling: a positive set and a negative set. For ease of discussion, we call the query image as q, the positive set as *Pos,* and the negative set as *Neg*. For each database image D_i and existing SC SC_j, the algorithmic view of this weighting method is as follows:

(1) Compute the semantic similarity between query q and each SC_j by

$$SemSim(q, SC_j) = \max\left(\sum_{D_p \in \text{Pos}} related(D_p, SC_j) - \sum_{D_q \in \text{Neg}} related(D_q, SC_j), 0\right), \tag{9}$$

where the first summation term represents the overall similarity between SC_j and all the images in *Pos*, and the second summation term represents the overall similarity between SC_j and all the images in *Neg*.

(2) Compute the probability for each SC_j to represent the semantic meaning of query q by:

$$SemProb(q, SC_j) = \frac{SemSim(q, SC_j)}{\sum_{i=1}^{T} SemSim(q, SC_i)}, \tag{10}$$

where T is the total number of existing SCs.

(3) Compute the semantic similarity between query q and each database image D_i as

$$S_{\text{long}}(q, D_i) = \frac{1}{T}\sum_{j=1}^{T}\left(SemProb(q, SC_j)^* M(D_i, SC_j)\right), \tag{11}$$

where $M(D_i, SC_j)$ is the relevancy of image D_i to the SC SC_j as defined in Equation 4.

Here, we assign the highest value to the image which is the most semantically similar to the query.

4.3. Complexity Analysis of Long-Term Learning

The complexity of the DSC algorithm is $O(T^2)$, where T is the number of clusters to date. The complexity of the cluster-image weighting algorithm is $O(T \times M)$ with M being the total number of images in the database.

For the 12,000-image database, it takes the DSC algorithm an average of 0.1246 seconds to construct and merge SCs at the end of each query session. It takes the cluster-image weighting algorithm 0.0537 seconds to compute the semantic similarity between the query image and all the database images. For our proposed collaborative system, the average retrieval time per query is 1.5824 seconds for each RF iteration. This retrieval time can be reduced to around 0.0791 seconds if converting from Matlab to C.

5. EXPERIMENTAL RESULTS

We extensively tested our CBIR system on the 6,000-COREL database and the 12,000-image database (i.e., 6,000-COREL images plus 6,000-real-world images). The COREL database contains 60 categories with 100 images per category. Similarly, the 6,000-real-world images contain 60 categories with 100 images per category. We used three websites, e.g., http://www.flickr.com/, http://images.google.com/, and http://picasa.google.com/, to collect 2,000 images for 20 categories, respectively. Specifically, we used the APIs of these three websites to search for 20 distinct key words, respectively. We then downloaded top 200 images for each category and manually picked the most appropriate 100 images.

To evaluate the effectiveness of short-term learning, long-term learning, and collaborative learning, we designed a set of experiments on the benchmark database, the 6,000-COREL database. To facilitate the evaluation process, the CBIR system automatically selects query images and performs the RF process. Specifically, a retrieved image is automatically classified as relevant if it is in the same semantic category as the query. The first set of experiments evaluated the effectiveness of the short-term block-based FSVM learning by incorporating correct feedback. The second set of experiments evaluated the effectiveness of the long-term DSC learning by constructing SCs using different number of queries and incorporating correct feedback. We randomly chose 2%, 5%, and 10% from each category of the image database as queries and performed a query session for each chosen query to construct three types of SCs, respectively. After the initial training, the system was then tested using the remaining 90% of the database images as queries. The third set of experiments evaluated the effectiveness of collaborative learning by, respectively, using three types of SCs and incorporating correct feedback. Another three sets of experiments were performed to incorporate the possible erroneous feedback in the real-world RF processes, wherein erroneous feedback may result from user inherent subjectivity of determining semantic relevance, user laziness in carefully labeling each returned image, or user maliciousness in trying to break the retrieval system. These additional three sets of experiments were evaluated for the effectiveness of short-term learning, long-term learning, and collaborative learning by incorporating

5% random erroneous feedback. To introduce the noise, we let the simulated "user" misclassify some relevant images as irrelevant and some irrelevant images as relevant. In each experiment, we performed four iterations of RF with the top 25 images returned in each iteration using Equation (1). All the algorithms are compared in terms of the retrieval precision, which is defined as the ratio between the number of relevant images returned and the total number of images returned.

To further evaluate the effectiveness of our system, we extensively compared our system with four state-of-the-art systems on the 12,000-image database from two perspectives: (1) An image only belongs to one major semantic category; (2) an image may belong to multiple semantic categories. For each experiment, we randomly chose 10% of the database as queries and performed the corresponding query sessions to construct SCs. We finally compared the retrieval performance of these five systems using the remaining 90% of the database images as queries by incorporating correct RF and 5% erroneous feedback, respectively.

5.1. Effectiveness of Short-Term Learning on the 6,000-COREL Database

We compared the proposed block-based FSVM learning method with five short-term learning systems, which include the global FSVM learning method,[34] the global soft label SVM learning method as used in short-term learning of the log-based technique,[38] the global SVM learning method as used in short-term learning of the memory learning technique,[39] the manifold method,[7] and the block-based SVM learning method, which uses the same blocking scheme as deployed in our system. Figure 5a shows a comparison among the average retrieval precision of these six short-term learning methods without any erroneous feedback. The figure clearly shows that the manifold method achieves the best retrieval accuracy in four iterations when compared to the SVM-based methods. This is mainly due to the use of the stored affinity matrix for propagating the ranking score of labeled images to the unlabeled images. Among the SVM-based methods, our block-based FSVM method achieves the best overall performance at the average retrieval precision of 22.8%, 43.3%, 66.2%, and 78.0% in four iterations. The global soft label SVM method achieves the comparable retrieval accuracy as our block-based FSVM method. The global SVM method achieves the worst overall retrieval accuracy in four iterations. The global FSVM method substantially improves the retrieval accuracy of the global SVM method during the early iterations because expanding the training set may greatly improve the effectiveness of the FSVM when the training set is small. However, in later iterations, increasing the training set is not as necessary as in the early iterations since the training set increases with more RF iterations. Furthermore, the system may mislabel some of the additionally chosen images and result in less accurate classification in later iterations. The block-based SVM method improves the retrieval accuracy more rapidly than the global SVM method as the training set grows in later iterations. It also achieves better retrieval accuracy than the global FSVM method in later iterations. This clearly shows the effectiveness of our proposed fuzzy block-based learning approach since it combines the strengths of the global FSVM and block-based methods.

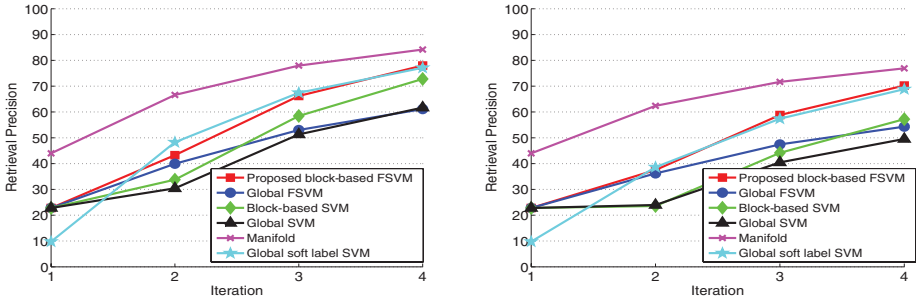

Figure 5. Comparison of six short-term learning systems for the 6,000-COREL database: (a) with correct feedback; (b) with 5% erroneous feedback.

We further evaluated these six systems in the context of erroneous feedback. Figure 5b compares the average retrieval precision of the five SVM-based learning methods and the manifold method at the level of 5% random erroneous feedback. It clearly shows that the mislabeling makes the two non-FSVM methods significantly decrease their retrieval precision in all iterations when compared to their retrieval precision achieved using correctly labeled information. The two FSVM methods, the soft label method, and the manifold method do perform worse with simulated errors, but the decrease in average retrieval precision is relatively small. Therefore, they are more resilient than the non-FSVM methods. Specifically, our block-based FSVM method achieves the average precision of 22.8%, 37.5%, 58.8%, and 70.2% in four iterations.

In summary, our extensive experimental results clearly show that the fuzzy metric computation as used in the block FSVM-based method achieves the following goals: (1) Provide relatively correct fuzzy information on the additional chosen image blocks; and (2) make block FSVM-based classification achieve better retrieval results than other SVM-based classification when both correct and erroneous relevance feedback are involved.

5.2. Effectiveness of Long-Term Learning on the 6,000-COREL Database

We compared the proposed long-term DSC learning method with the log-based learning method as used in long-term learning of the log-based technique,[38] and the semantics-based memory learning method as used in long-term learning of the memory learning technique.[39] Figure 6 shows a comparison between the average retrieval precision of these three long-term learning methods after using a different number of training queries (i.e., 2%, 5%, and 10% from each category of the image database) to build the long-term repositories. The figure also compares the average retrieval precision of these three long-term learning methods in the context of having no erroneous feedback and having a level of 5% erroneous feedback. It clearly demonstrates that the proposed long-term learning method performs a little bit worse with simulated errors, while the other two long-term learning methods

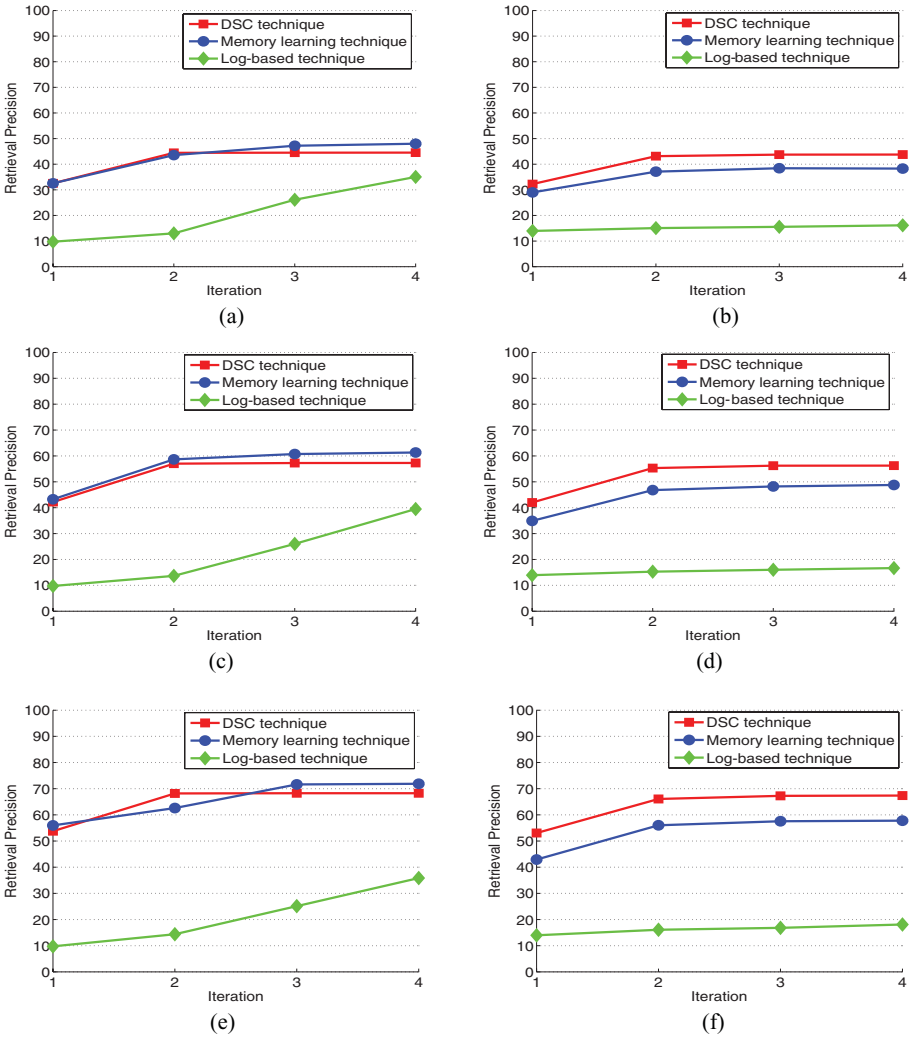

Figure 6. Comparison of three long-term learning systems for the 6,000-COREL database. Using 2% of database images as training data and with (a) correct feedback. (b) Five percent erroneous feedback. Using 5% of database images as training data and with (c) correct feedback. (d) Five percent erroneous feedback. Using 10% of database images as training data and with (e) correct feedback. (f) Five percent erroneous feedback.

suffer from a substantial decrease in average retrieval precision with simulated errors. That is, our long-term learning is more resilient to the erroneous feedback than the other two peers. It consistently achieves the best retrieval accuracy in all iterations when erroneous feedback is involved. Specifically, at the 5% noise level, it achieves the average precision of 32.3%, 43.1%, 43.7%, and 43.8% in four iterations using 2% of the database images as queries to build SCs; it achieves the average precision of 42.0%, 55.3%, 56.2%, and 56.3% in four iterations using 5% of the

database images as queries to build SCs; and it achieves the average precision of 53.1%, 66.1%, 67.3%, and 67.4% in four iterations using 10% of the database images as queries to build SCs. However, the retrieval performance of our DSC method is slightly inferior to the memory learning method when correct RF is involved. It is also important to note that our method achieves these slightly worse results by using significantly less storage space.

It should be noted that the more queries (i.e., training images) used to build the learning repository, the higher the retrieval accuracy for all three long-term learning systems. Their retrieval precision is also almost flat after the systems remember the semantic relationship among images, which is quite normal since no short-term learning is taking place.

5.3. Effectiveness of Collaborative Learning on the 6,000-COREL Database

We compared the proposed collaborative learning system (i.e., DSC + block-based FSVM) with Hoi's log-based system (i.e., log-based + global soft label SVM),[38] Han's memory learning system (i.e., memory learning + global SVM),[39] a variant of the proposed system that incorporates the global SVM method as the short-term learning scheme (i.e. DSC + global SVM), and the manifold system.[7] Figure 7 shows the average retrieval precision of these five CBIR systems with and without a 5% chance of the user mislabeling each returned image after using a different number of training queries (i.e., 2%, 5%, and 10% of the database images) to build the long-term repositories. The figure clearly shows that our proposed system achieves the best retrieval precision in all iterations with both correct and erroneous feedback. Figures 7a–7e demonstrate that our variant system achieves a retrieval performance comparable to the memory learning system when the RF contains no erroneous information. Therefore, we can safely say that our short-term learning facilitates long-term learning to significantly boost the average retrieval precision in all iterations by discovering more relevant images and providing more semantic information. Meanwhile, our long-term DSC learning addresses the possible unbalanced training data issues resulting from early iterations of short-term learning by utilizing the estimated relationships of images learned over many search sessions. It should be noted that our proposed system and its variant used significantly less storage space to achieve better retrieval results than the memory learning system. Figures 7b, 7d, and 7f demonstrate our proposed system and its variant are resilient to erroneous feedback, while the log-based system and the memory learning system significantly drop the retrieval precision in all iterations. This resilience is mainly due to the merging of similar SCs by using effective overlapping measures as summarized in Equations 4–8. This noise resilience feature is important in the real world since it is normal for users to make mistakes due to the inherent subjectivity of determining semantic relevance, user laziness, or maliciousness.

Figure 7 also clearly shows that the size of the training set does affect the precision of all systems, but it does not significantly affect the performance relationship of the four combined short-term and long-term systems. Our proposed system performs the best in all six experiments and its variant performs the second

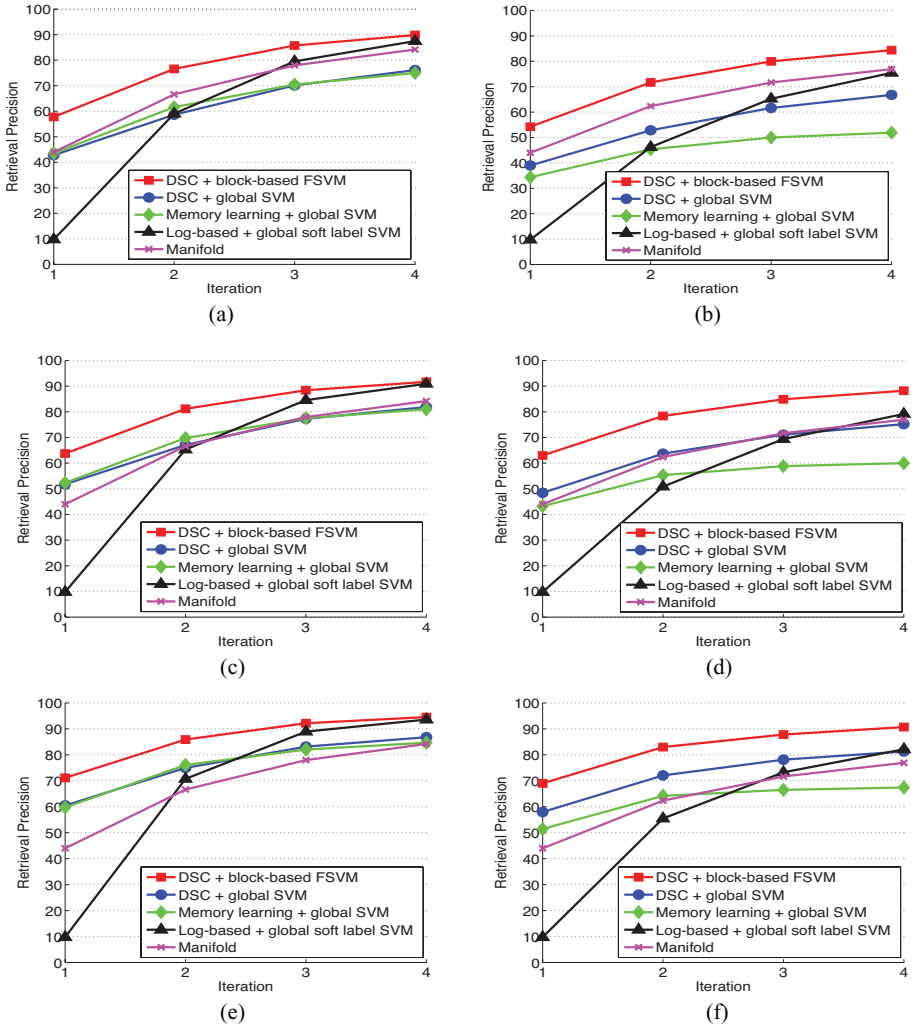

Figure 7. Comparison of five CBIR systems for the 6,000-COREL database. Using 2% of database images as training data and with (a) correct feedback. (b) 5% erroneous feedback. Using 5% of database images as training data and with (c) correct feedback. (d) 5% erroneous feedback. Using 10% of database images as training data and with (e) correct feedback. (f) Five percent erroneous feedback.

best in most experiments. The log-based system generally performs better than our variant system at the third and fourth iterations when the correct feedback is involved. However, it suffers from substantial decrease in the average retrieval precision with simulated errors, while our variant system achieves better accuracy at the first two iterations and comparable accuracy at later iterations. The manifold system only stands out when all four combined short-term and long-term systems use 2% of the database images as training data to build the long-term repository. In

summary, by the fourth iteration without simulated user errors, our system achieved a precision of 94.5% with 600 training queries, 91.7% with 300 training queries, and 89.9% with 120 training queries. By the fourth iteration with simulated 5% user errors, our system achieved a precision of 90.7% with 600 training queries, 88.2% with 300 training queries, and 84.4% with 120 training queries. This shows that a small training set is sufficient for obtaining high retrieval accuracy. This characteristic is attractive in real-world situations with databases of millions of images.

5.4. Effectiveness of Collaborative Learning on the 12,000-Image Database

We further compared the retrieval performance of the above five systems on the 12,000-image database, where an image only belongs to one major semantic category. Figure 8 shows the average retrieval precision of four CBIR systems with and without a 5% chance of the user mislabeling each returned image after using 10% of the 12,000 images to build the long-term repositories. The manifold system cannot run on our computer due to its requirement of several matrices of 12,000 $\times$ 12,000. As a result, the manifold system is not included in the comparison. Figure 8a clearly shows that our system achieves a best retrieval performance comparable to its variant system and the log-based system in later two iterations when no erroneous feedback is involved. However, our system achieves the best retrieval precision and its variant system achieves the second best retrieval precision in all iterations with erroneous feedback, as shown in Figure 8b. In summary, by the fourth iteration without simulated user errors, our system achieved a precision of 75.9% with 1,200 training queries. By the fourth iteration with simulated 5% user errors, our system achieved a precision of 65.7% with 1,200 training queries.

We also compared the retrieval performance of the four systems on the 12,000-image database, where an image may belong to multiple semantic categories. Figure 9 shows the average retrieval precision of four CBIR systems with and

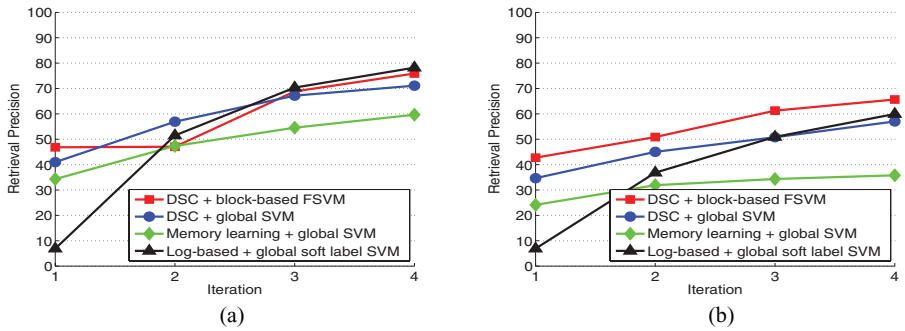

Figure 8. Comparison of four CBIR systems for the 12,000-image database with one semantic meaning. Using 10% of database images as training data and with (a) correct feedback; (b) 5% erroneous feedback.

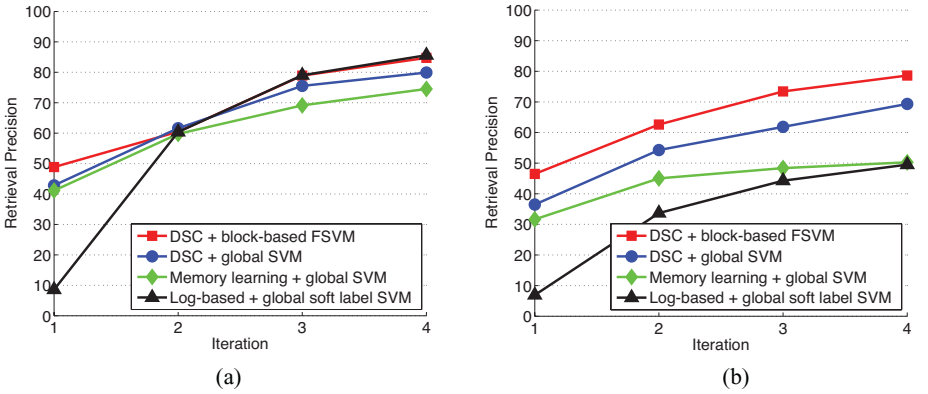

Figure 9. Comparison of four CBIR systems for the 12,000-image database with multiple semantic meanings. Using 10% of database images as training data and with (a) correct feedback; (b) 5% erroneous feedback.

without a 5% chance of the user mislabeling each returned image after using 10% of the 12,000 images to build the long-term repositories. Figure 9a clearly shows that our system achieves a best retrieval performance comparable to the log-based system in later three iterations when no erroneous feedback is involved. However, our system achieves the best retrieval precision and its variant system achieves the second best retrieval precision in all iterations with erroneous feedback, as shown in Figure 9b. The other two peer systems substantially decrease their retrieval precision in all iterations when compared to their retrieval precision achieved using correctly labeled information. In summary, by the fourth iteration without simulated user errors, our system achieved a precision of 84.8% with 1,200 training queries. By the fourth iteration with simulated 5% user errors, our system achieved a precision of 78.6% with 1,200 training queries.

5.5. Storage Effectiveness of Collaborative Learning

Our proposed system uses less storage space to save semantic knowledge learned to date. Specifically, it requires $O(N \times C)$ space where N is the total number of images in the database, C is the number of SCs learned, and $C \ll N$. Based on our experiments on a 6,000-image database, C was approximately 81 for the variant of the proposed system and 68 for the proposed system. These are both close to 60, the optimal number of SCs as determined by humans for the database, as there are 60 categories in the subset of the COREL database used for testing. The value of C for our proposed system was approximately 139 for the 12,000 images. This value is close to the optimal number of SCs as determined by humans for the 12,000-image database. The log-based system, the memory learning system, and other long-term learning systems require $O(N^2)$ space to store the learned knowledge, where N is the total number of images in the database. In our experiments, the proposed system stored semantic information on $129 \times 12{,}000 = 1{,}548{,}000$ relationships, while

the log-based and the memory learning systems stored semantic information on $12{,}000^2 = 144{,}000{,}000$ relationships for the 12,000 images. That is, the proposed system required approximately 1.07% as much storage space as both log-based and memory learning systems. This efficient storage is necessary for real-world situations with databases of millions of images.

6. CONCLUSIONS AND FUTURE WORK

This paper introduces a noise-resilient, collaborative learning approach to image retrieval. It seamlessly incorporates short-term block-based FSVM learning and long-term DSC learning to bridge the semantic gap between low-level visual features and high-level semantic meanings. Specifically, the short-term learning technique chooses additional local features to expand the training set and learns the user's query concepts from the visual perspective. The long-term learning technique uses dynamic SCs to efficiently store the accumulated feedback information in a compact manner and learns the user's query concept from the semantic perspective. The short-term learning technique effectively addresses the issue of small numbers of training images being returned at each RF iteration, while the long-term learning technique reduces the storage requirements for storing long-term feedback information. Our extensive experimental results show that our proposed system achieves a comparable retrieval precision without erroneous feedback and achieves the best retrieval precision with erroneous feedback when compared to the three peer systems, i.e., the log-based system, the memory learning system, and its variant. The proposed system also requires significantly less space and is therefore more capable of scaling to large databases.

We plan to test the proposed technique for its effectiveness and scalability on a larger database by comparing with additional emerged state-of-the-arts systems. Next, we plan to investigate different semantic clustering techniques to determine the relationships among images for better retrieval. We will also obtain a sufficient number of human subject tests to simulate the user's query log information and to see how our system would do with real human feedback. Finally, we plan to research the effect of utilizing the state-of-the-art indexing systems[48] in the simulated online image retrieval environment and plan to explore the potential of applying the proposed technique in the image annotation task by propagating users' annotations to related images.

Acknowledgment

This research was supported by NSF grant IIS-0453552: REU Site Program in Computer Vision and Image Processing.

References

1. Antani S, Kasturi R, Jain R. A survey on the use of pattern recognition methods for abstraction, indexing, and retrieval of images and video. Pattern Recognit 2002;35(4):945–965.

2. Smeulders A, Worring M, Santini S, Gupta A, Ramesh J. Content based image retrieval at the end of the early years. IEEE Trans PAMI 2000;22(12):1349–1380.
3. Zhou XS, Huang TS. Relevance feedback for image retrieval: a comprehensive review. J ACM Multimedia Syst 2003;8(6):536–544 (Special Issues on CBIR).
4. Lin CH, Chen RT, Chan YK. A smart content-based image retrieval system based on color and texture features. Image Vis Compu 2009;27(6):658–665.
5. Gevers T, Smeulders A. Pictoseek: combining color and shape invariant features for image retrieval. IEEE Trans. Image Process 2000;9(1):102–119.
6. Pentland A, Picard RW, Sclaroff S. Photobook: content based manipulation for image databases. Int J Comput Vis 1996;18(3):233–254.
7. He J, Li M, Zhang H, Tong H, Zhang C. Generalized manifold-ranking-based image retrieval. IEEE Trans Image Process 2006;15(10):3170–3177.
8. Brank J. Image categorization based on segmentation and region clustering. In: First Starting AI Res Symp (STAIRS'02), Lyon, France; 2002. pp 145–154.
9. Carson C, Belongie S, Greenspan H, Malik J. Blobworld: image segmentation using expectation maximization and its application to image query. IEEE Trans PAMI 2002;24(8):1026–1038.
10. Chen Y, Wang J. A region-based fuzzy feature matching approach to content-based image retrieval. IEEE Trans PAMI 2002;24(9):1252–1267.
11. Kumar S, Loui AC, Hebert M. An observation constrained generative approach for probabilistic classification of image regions. Image Vis Comput 2003;21:87–89.
12. Ma MY, Manjunath BS. NeTra: a toolbox for navigating large image databases. Multimedia Syst 1999;7(3):184–198.
13. Chan YK, Ho YA, Liu YT, Chen RC. A ROI image retrieval method based on CVAAO. Image Vis Comput 2008;26(11):1540–1549.
14. Rajashekhara SC. Segmentation and region of interest based image retrieval in low depth of field observations. Image Vis Comput 2007;25(11)1709–1724.
15. Smith JR, Li CS. Image classification and querying using composite region templates. Comput Vis Image Underst 1999;75(1–2):165–174.
16. Wang J, Li J, Wiederhold G. SIMPLIcity: semantics-sensitive integrated matching for picture libraries. IEEE Trans PAMI 2001;23(9):947–963.
17. Li J, Wang JZ. Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Trans. PAMI 2003;25(10):1075–1088.
18. Lu Y, Zhang H, Liu W, Hu C. Joint semantics and feature based image retrieval using relevance feedback. IEEE Trans Multimedia 2003;5(3):339–347.
19. Sheikholeslami G, Chang W, Zhang A. SemQuery: semantic clustering and querying on heterogeneous features for visual data. IEEE Trans Knowl Data Eng 2002;14(5):988–1002.
20. Vailaya A, Figueiredo M, Jain A, Zhang H. Image classification for content-based indexing. IEEE Trans Image Process 2001;10(1):117–130.
21. Vasconcelo N. Exploiting group structure to improve retrieval accuracy and speed in image databases. In: IEEE Int Conf Image Process (ICIP'02), Rochester, NY; 2002. pp 980–983.
22. Cox IJ, Miller ML, Minka TP, Papathomas TV, Yianilos PN. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. IEEE Trans Image Process 2000;9(1):20–37.
23. Ortega M, Rui Y, Chakrabarti K, Mehrotra S, Huang TS. Supporting similarity queries in MARS. In: Fifth ACM Int Conf Multimedia (MULTIMEDIA'97), Seattle, WA; 1997. pp 403–413.
24. Rui Y, Huang TS, Ortega M, Mehrotra S. Relevance feedback: a power tool for interactive content based image retrieval. IEEE Trans Circuits Video Technol 1998;8(5):644–655.
25. Smith JR. Integrated spatial and feature image systems: retrieval, compression and analysis, PhD Thesis, Graduate School of Arts and Sciences, Columbia University, New York, February, 1997.
26. Kushki A, Androutsos P, Plataniotis KN, Venetsanopoulos AN. Query feedback for interactive image retrieval. IEEE Trans Circuits Syst Video Technol 2004;14:644–655.

27. Muneesawang P, Guan L. An interactive approach for CBIR using a network of radial basis functions. IEEE Trans Multimedia 2004;6(5):703–716.
28. Widyantoro DH, Yen J. Relevant data expansion for learning concept drift from sparsely labeled data. IEEE Trans Know Data Eng 2005;17(3):401–412.
29. MacArthur SD, Brodley CE, Shyu CR. Relevance feedback decision trees in CBIR. In: Workshop on Content Based Access of Image and Video Libraries (CBAIVL'02), Hilton Head Island, SC; 2000. pp 68–72.
30. Chang E, Goh K, Sychay G, Wu G. Cbsa: content-based soft annotation for multimodal image retrieval using Bayes point machines. IEEE Trans Circuits Syst Video Technol 2003;13(1):26–38.
31. Su Z, Zhang H, Li S, Ma S. Relevance feedback in CBIR: Bayesian framework, feature subspaces, and progressive learning. IEEE Trans Image Process 2003;12(8):924–936.
32. Wilson S, Stefanou G. Bayesian approaches to content based image retrieval. In: Int Workshop/Conf Bayesian Stat Appl, Varanasi, India; 2005. pp 455–465.
33. Tong S, Chang E. Support vector machine active learning for image retrieval. In: Ninth ACM Int Conf Multimedia, Ottawa, Canada; 2001. pp 107–118.
34. Wu K, Yap KH. Fuzzy SVM for content-based image retrieval. IEEE Comput Intell Mag 2006;1(2):10–16.
35. Tieu K, Viola P. Boosting image retrieval. In: IEEE Int Conf Comput Vision Pattern Recognit, Hilton Head Island, SC; 2000. pp 228–235.
36. Li M, Chen Z, Zhang H. Statistical correlation analysis in image retrieval. Pattern Recognit 2002;35:2687–2693.
37. He X, King O, Ma WY, Li M, Zhang H. Learning a semantic space from user's relevance feedback for image retrieval. IEEE Trans Circuits Syst Video Technol 2003;13(1)39–48.
38. Hoi S, Lyu M, Jin R. A unified log-based relevance feedback scheme for image retrieval. IEEE Trans Knowl Data Eng 2006;18(4):509–524.
39. Han J, Ngan KN, Li M, Zhang HJ. A memory learning framework for effective image retrieval. IEEE Trans Image Process 2005;14(4):511–524.
40. Manjunath BS, Salembier P, Sikora T. Introduction to MPEG-7: Multimedia Content Description Interface. Section IV: Visual Descriptors. Chichester, UK: Wiley; 2002. pp 177–280.
41. Qi X, Han Y. Incorporating multiple SVMs for automatic image annotation. Pattern Recognit 2007;40(2):728–741.
42. Qi X, Chang R. Image retrieval using transaction-based and SVM-based learning in relevance feedback sessions. In: Fourth Int Conf Image Anal Recognit (ICIAR'07), Montreal, Canada, 2007. Lecture Notes in Computer Science (LNCS), Vol. 4633, pp 637–649.
43. Carneiro G, Chan AB, Moreno PJ, Vasconcelos N. Supervised learning of semantic classes for image annotation and retrieval. IEEE Trans PAMI 2007;29(3):394–410.
44. Scholkopf B, Sung K, Burges C, Girosi F, Niyogi P, Poggio T, Vapnik V. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE Trans Signal Process 1997;45(11)2758–2765.
45. Hsu C, Chang C, Lin C. A practical guide to support vector classification. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, Taiwan; 2003.
46. Christopher J, Burges C. A tutorial on support vector machines for pattern recognition. Knowl Discov Data Min 1998;2(2):235–244.
47. Pinker S. How the mind works. New York: W. W. Norton & Company; 1997. pp 258.
48. Philbin J, Chum O, Isard M, Sivic J, Zisserman A. Object retrieval with large vocabularies and fast spatial matching. In: Int Conf Comput Vis Pattern Recognit (CVPR'07), Minneapolis, MN; 2007. pp 1–8.