Multimed Tools Appl
DOI 10.1007/s11042-013-1693-4

Complementary relevance feedback-based content-based image retrieval

Zhongmiao Xiao · Xiaojun Qi

Abstract We propose a complementary relevance feedback-based content-based image retrieval (CBIR) system. This system exploits the synergism between short-term and long-term learning techniques to improve the retrieval performance. Specifically, we construct an adaptive semantic repository in long-term learning to store retrieval patterns of historical query sessions. We then extract high-level semantic features from the semantic repository and seamlessly integrate low-level visual features and high-level semantic features in short-term learning to effectively represent the query in a single retrieval session. The high-level semantic features are dynamically updated based on users' query concept and therefore represent the image's semantic concept more accurately. Our extensive experimental results demonstrate that the proposed system outperforms its seven state-of-the-art peer systems in terms of retrieval precision and storage space on a large scale imagery database.

Keywords Content-based image retrieval · Relevance feedback model · Semantic features · Long-term learning

1 Introduction

With the amount of digital photography data growing at an accelerating rate, the development of efficient image retrieval systems to find images of interest in this haystack of data has become an active research area in recent years [19]. Content-based image retrieval (CBIR) has emerged as one of the solutions to overcome the limitations entailed by text-based image retrieval and has evolved significantly since the early 1990s. It allows users to directly submit image examples, object sketches, or other low-level visual information (e.g., color, texture, and shape features) to find images of interest by using image processing and similarity matching techniques to analyze and compare the content of all images in an imagery collection. This technique alleviates the burden for manual annotations required by the text-based image retrieval. However, as the ranking of retrievals is calculated based on

Z. Xiao · X. Qi (✉)
Department of Computer Science, Utah State University, Logan, UT 84322-4205, USA
e-mail: Xiaojun.Qi@usu.edu

Z. Xiao
e-mail: zhongmiao.xiao@aggiemail.usu.edu

Published online: 12 September 2013

selected image features, the retrieval accuracy may be unsatisfactory due to the semantic gap (e.g., the discrepancy between the low-level visual features and high-level semantic concepts). So, the main research objectives are to design effective techniques that assist the user to find images of interest quicker than before and provide the user with a better search experience than before. Two recent comprehensive surveys [7, 13] provide insights into the advances in multimedia information retrieval and, in particular, CBIR. In these articles, relevance feedback is recognized as one of the most promising techniques to bridge the semantic gap between low-level visual features and high-level semantic meanings and improve the retrieval performance.

Relevance feedback (RF) [23] is an interactive search technique, where the CBIR system engages in a dialogue with the user, with the goal to find out the user's query intention, i.e., what the user is looking for. The process treats the retrieval session as repetitive query reformulation operations. It entails presenting retrieved images to the user after submitting query image(s) and soliciting user's relevance judgments of retrieved images over the course of several rounds of interaction. Through successive human-computer interactions, the CBIR system repeatedly modifies the query descriptive information (feature, matching models, metrics or any meta-knowledge) as a response to the users' feedback on retrieved results. Therefore, it learns the query close to its optimal and returns more user-desired images after each round.

Most existing RF techniques use short-term learning or intra-query learning to find out which images are relevant to the user's query in a single query retrieval session. The state of the CBIR system is reset after each query session. Representative short-term learning techniques include query updating (e.g., query reweighting, query shifting, and query expansion) and statistical learning techniques (e.g., inductive learning and transductive learning) [16]. However, short-term learning techniques cannot capture the semantic meaning of an image and therefore cannot achieve satisfactory retrieval results. In addition, they cannot remember the user's feedback after a query session and therefore cannot utilize the user's feedback in future retrievals.

Recently, long-term learning or inter-query learning extends short-term learning by utilizing the information gathered during previous retrieval sessions to improve the retrieval results in future sessions. Specifically, long-term learning infers relationships between images by analyzing the feedback log, which contains the accumulated feedback history collected from multiple query sessions given by users over time. The information in the log is usually aggregated into a semantic, relevance, or affinity matrix, which allows the CBIR system to discover extra knowledge (i.e., the semantic relevance level of a database image to the current query). Representative long-term learning techniques [9] include retrieval pattern-based learning and feature vector model-based learning. Here, we briefly review several long-term techniques that are related to the proposed approach.

He et al. [11] apply singular value decomposition on the feedback log to build a semantic space, which contains relationships between the images and one or more classes. They further apply the dot product operations on the semantic space to find semantically similar images. Similarly, Shah-hosseini and Knapp [18] apply probabilistic latent semantic analysis on the feedback log to create the semantic space for capturing relationship between the images and classes. Hoi et al. [12] apply the statistical correlation on the feedback log to analyze the relationship between the current and past retrieval sessions and provide an SVM with more training examples which have the largest relevance scores to the query images. Han et al. [10] use the feedback log to compute the ratio of the co-positive feedback frequency and the co-feedback frequency for analyzing the relationship among each query session. They further apply the memory learning technique to build a knowledge memory model to store the semantic information and learn semantic relations. Yin et al. [21] introduce virtual features to record all

the concepts discovered from the long-term user feedback and their significance to the image. That is, virtual features store multiple long-term relevant information associated with each image. Specifically, these virtual features are updated based on the user's RF and are adapted to changes in user relevance perception. They are further used to estimate the semantic relevance between images. Xiao et al. [20] propose a long-term cross-session learning scheme for CBIR. Based on the feedback log, they formulate the high-level dynamic semantic features for each database image in terms of retrieval patterns. They then derive the semantic relevance among images and combine the high-level semantic similarity with the low-level visual similarity to find top images that are similar to the query image. Chang et al. [2] create semantic clusters based on the feedback log to group semantically similar images. They then incorporate the reliability scores of each database image into the affinity matrix to construct a weighted manifold structure to discriminately propagate the ranking scores of labeled images. They further expand this weighted manifold structure by incorporating the reliability scores, the fuzzy membership, and the high-level semantic similarity score into the affinity matrix to construct a semantic manifold structure [3]. These learning techniques reduce the semantic gap and achieve impressive retrieval results. However, they usually require a large matrix to store the feedback log to learn the relationships among all images.

Clustering techniques are therefore proposed to learn the hidden concepts (i.e., the semantic categories) in the image database by clustering the feedback. Researchers apply the semantic grouping [22], semi-supervised non-negative matrix factorization-based clustering [4], Chunklet assignment-based clustering [5], cluster affinity search [14], and dynamic semantic clustering techniques [16] on the feedback log to cluster images into one or more semantically homogenous clusters. These clustering techniques effectively reduce the search space. However, the complexity of constructing semantic clusters is high.

In the paper, we propose a complementary RF-based CBIR system, which takes advantages of both short-term and long-term learning techniques and exploits the synergism between them to improve the retrieval performance. This new system combines two learning techniques that are complementary to each other in a hope to eradicate the weakness of individual learning. It also exploits synergism between two learning techniques so a general meta-strategy is created and the conception can be implemented in several variations. In our proposed RF-based CBIR system, the training and retrieval processes have the following advantages over other common RF-based CBIR systems:

- Seamless integration of low-level visual features and high-level semantic features in short-term learning to effectively represent the query in a single retrieval session. The high-level semantic features are dynamically updated based on users' query concept and therefore represent the image's semantic concept more accurately.
- Quick construction of an adaptive semantic repository in long-term learning to store retrieval patterns (i.e., similarity of relevant and irrelevant images) of historical query sessions.
- Efficient merging of the similar semantic concepts to keep the adaptive semantic matrix compact.
- Effective composition of low-level visual and high-level semantic similarity measure to estimate the semantic relevance among images.

The remainder of this paper is organized as follows. Section 2 presents the architecture of the proposed general RF-based CBIRsystem together with the detailed discussion for each important component. Section 3 compares our system with seven state-of-the-art peer systems and four variant systems on various databases. Section 4 draws conclusions and presents future directions.

2 Architecture of the proposed system

Our complementary RF-based CBIR system combines two RF models denoted by Θ and Ω where Θ belongs to the long-term learning method which infers relationships between images by analyzing the accumulated feedback history collected from multiple query sessions given by users over time and Ω belongs to the short-term learning method which modifies the query descriptive information as a response to the user's feedback on the retrieved results for a query image. In other words, it collaborates Θ and Ω together and maximizes synergism between them. The relevant information learned from previous queries provides valuable clues for reformulating the response to future instances of these queries. In this way, our proposed CBIR system allows the short-term learning method Ω to benefit from the long-term learning method Θ. Figure 1 shows the architecture of the proposed system. Let Q denote the query submitted by the user. The CBIR system first applies both Θ and Ω to retrieve a set of v database images that are most similar to Q. Since no knowledge has been learned at the beginning of the retrieval process, semantic features employed by Θ are empty and the system simply applies Ω on low-level visual features to retrieve a set of similar database images using Eq. (2), which we will discuss later in Section 2.1. After this initial retrieval, semantic features are no longer empty and the system applies Ω on updated high-level semantic features and low-level visual features to retrieve a set of v similar database images using Eq. (4), which we will discuss later in Sections 2.2 and 2.4. In the meantime, the system applies Θ to dynamically update semantic features using the RF information collected at the current query session, which we will discuss later in Section 2.3. These retrieved images are presented to the user requesting for his/her judgment on their similarity to Q. That is, relevant or similar images are positively labeled and irrelevant or dissimilar images are negatively labeled. These positively labeled (P) and negatively labeled (N) images are then exploited in two ways to improve the retrieval performance.

- Update the semantic repository employed in Θ to reinforce long-term learning.
- Update query's visual and semantic features employed in Ω to reinforce short-term learning.

Finally, the CBIR system employs both updated Θ and Ω to start the next round of retrieval to obtain a new set of v database images that are most similar to Q. This process is

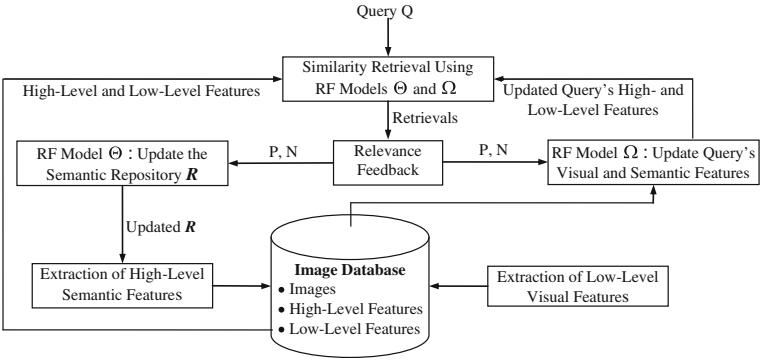

Fig. 1 The proposed complementary RF-based CBIR system

repeated multiple times until the user is satisfied with the retrieval results. The algorithmic view of this process is summarized as follows:

1. Initialize the semantic repository $\boldsymbol{R}$, an 2D matrix whose row number equals to the number of images in the database and whose column number equals to 2 % of the total number images in the database (e.g., the maximum length of the high-level semantic feature for all images in the database), as all 0's.
2. Load in the visual features of all database images from the database of low-level features.
3. For each randomly selected query image Q, perform the following operations:

    3.1. Extract its low-level visual features *qLowFea*.
    3.2. Extract its high-level semantic features *qHighFea*.
    3.3. Load in the semantic features of all database images from the database of high-level features.
    3.4. Apply Eq. (4) described in Section 2.2 to compute the overall similarity between Q and each database image DIm_i.
    3.5. Return top v images to Q based on the sorted overall similarity in the ascending order.
    3.6. While the user is not satisfied with the retrieval results, perform the following operations:

    a) Allow the user to label the relevance of each retrieved image to Q. That is, positively labeled images are considered as relevant retrieved images and non-labeled images are automatically considered by the system as negatively labeled images, which are considered as irrelevant retrieved images.
    b) Apply RF model Θ to update $\boldsymbol{R}$.
    c) Apply RF model Ω to update *qLowFea* and *qHighFea*.
    d) Apply Eq. (4) to compute the overall similarity between Q and each database image DIm_i.
    e) Return top v images to Q based on the sorted overall similarity in the ascending order.

In order to facilitate the learning process, we deliberately ensure that the retrieved images are not returned in the following iterations for a query session. After a sufficient number of query sessions, which equals to 10 % of the number of images in the database, a semantic repository with a decent amount of semantic relationships is constructed. We then perform the online retrieval process in a normal way. That is, we do not enforce the rule such that the retrieved images are not returned in the following iterations. Next, we explain all the components, namely, extraction of low-level visual features, extraction of high-level semantic features, RF model Θ, and RF model Ω, in detail.

2.1 Extraction of low-level visual features

All three important features, e.g., color, edge, and texture features, are utilized to represent each image in the imagery database. We adopt the 100-dimensional low-level features as employed in our prior CBIR system [17] to represent each image. These features include the 64-bin HSV color histogram and a set of 36 compact low-level features, which consist of 9-dimensional color, 18-dimensional edge, and 9-dimensional texture features. Specifically, we compute the first three moments in each HSV color channel to represent color features.

We compute the 18-bin edge direction histogram to represent the edge features in the grayscale image. We compute the entropy of each of nine detail subbands of a 3-level wavelet transform to represent texture features in the grayscale image. These global features were proven to be effective in our prior system and they are easy to compute and complementary to each other.

A normalization technique is further applied to each of 100 feature components to ensure its values fall in the [0, 1] range. We choose the linear scaling to unit range technique [1] as our normalization technique due to its simplicity and effectiveness. This technique first finds the lower bound l and the upper bound u for a feature component x. It then normalizes x by:

$$\widetilde{x} = \frac{x - l}{u - l} \tag{1}$$

The Euclidean distance is used to measure low-level visual feature-based dissimilarity between two images, I_i and I_j, by

$$LowDisSim(I_i, I_j) = \left\| LVF(I_i) - LVF(I_j) \right\|_2 = \sqrt{\sum_{k=1}^{100} \left(LVF(I_i(k)) - LVF(I_j(k)) \right)^2} \tag{2}$$

where $LVF(I_i)$ represents normalized low-level visual features of an image I_i and $LVF(I_i(k))$ represents the k^{th} value of normalized low-level visual features of an image I_i.

2.2 Extraction of high-level semantic features

Each image is also represented by semantic features, which represent the semantic relevance between each positively labeled image and the corresponding query image(s) and the semantic irrelevance between each negatively labeled image and the corresponding query image(s). This semantic relationship is learned by digesting the long-term feedback history from multiple query sessions. Since semantic features are closely related to the high-level semantics of an image, we also call semantic features as high-level semantic features. In our system, we confine the maximal length of the high-level semantic feature for a database image to be 2 % of the total number of images in the database, which is a reasonable and conservative estimate for the maximal number of semantic concepts (e.g., foreground objects or background implicitly marked by the users as a set of relevant images in the RF step) contained in all images in a database. These high-level semantic features are directly constructed from the dynamically reformulated feedback log, which is called semantic repository $\boldsymbol{R}$ in our system and will be explained in Section 2.3. Specifically, each row in $\boldsymbol{R}$ represents semantic features of the image corresponding to the same row. Each column in $\boldsymbol{R}$ represents a possible semantic concept contained in an image. Each value in $\boldsymbol{R}$ represents the relationship between a database image and the corresponding semantic concept. For example, the 1st row in $\boldsymbol{R}$ represents the semantic features of the 1st database image and the 1000th row represents the semantic feature of the 1000th database image. If the 1st column in $\boldsymbol{R}$ represents the semantic concept (e.g., sky and mountain) of the 1st query and the 2nd column in $\boldsymbol{R}$ represents the semantic concept (e.g., flower and waterfall) of the 2nd query, the value in the 1st column of $\boldsymbol{R}$ means the relevance of a database image to the sky and mountain semantic concepts and the value in the 2nd column of $\boldsymbol{R}$ means the relevance of a database image to the flower and waterfall semantic concepts. A larger positive value indicates the database image likely to possess the corresponding semantic concept. A smaller negative value indicates the database image unlikely to possess the corresponding semantic concept.

The dot product of two high-level semantic feature vectors is used to measure high-level semantic feature-based similarity between two images, I_i and I_j, as follows:

$$HighSim(I_i, I_j) = \langle HSF(I_i), HSF(I_j) \rangle = \sum_{k=1}^{n} HSF(I_i(k)) \times HSF(I_j(k)) \quad (3)$$

where $HSF(I_i)$ represents high-level semantic features of an image I_i,$HSF(I_i(k))$ represents the kth value of high-level semantic features of an image I_i, and n is the maximum length of the high-level semantic features. Since the hidden meaning of the semantic features may contain non-relevant information, a special dot product operation is incorporated into our system. That is, if the kth values of two feature vectors are negative, we assign 0's instead of a positive value to $HSF(I_i(k)) \times HSF(I_j(k))$.

The overall dissimilarity between two images, I_i and I_j, is computed as follows:

$$OverallDisSim(I_i, I_j) = W_l \times LowDisSim(I_i, I_j) - W_h \times HighSim(I_i, I_j) \quad (4)$$

where W_l and W_h represent the weight contribution of the low-level dissimilarity measure and the high-level similarity measure, respectively. In our system, we set W_l to be 0.1 and W_h to be 0.9. The smaller the overall dissimilarity measure, the more similar the two images are.

2.3 RF model Θ: update the semantic repository R

RF model Θ aims to update the semantic repository $\boldsymbol{R}$ based on user's RF. This repository stores weighted semantic relationships between each database image and each possible semantic concept. Here, each database image can be described by several semantic concepts. We empirically set the number of semantic concepts as 2 % of the total number images in the database, which is a reasonable estimate for the maximal number of concepts contained in all images in a database. $\boldsymbol{R}$ can therefore be treated as a semantic feature matrix that stores the weights of semantic concepts contained in a database image. The weights in $\boldsymbol{R}$ are dynamically modified so that the existing hidden semantic concepts in the database become apparent over time. The dimensionality of $\boldsymbol{R}$ is the number of database images by the number of semantic concepts. The algorithmic view of updating $\boldsymbol{R}$ is summarized in Fig. 2.

In step 2, we initially update high-level semantic features for $\forall I_i \in AllPos$ by

$$HSF(I_i) = \frac{1}{pn} \sum_{i=1}^{pn} HSF(I_i) \quad (5)$$

where pn represents the total number of accumulated positively labeled images in $AllPos$. In step 3, we adopt two strategies to update the $NumD^{th}$ semantic feature in $\boldsymbol{R}$ for images in $AllPos$ and Neg, respectively. Specifically, we update the $NumD^{th}$ semantic feature for $\forall I_i \in AllPos$ by:

$$HSF(I_i(NumD)) = \begin{cases} HSF(I_i(NumD)) + 1 & if \;\; HSF(I_i(NumD)) = 0 \\ HSF(I_i(NumD)) \times 2 & otherwise \end{cases} \quad (6)$$

We update the $NumD^{th}$ semantic features for $\forall I_i \in Neg$ by:

$$HSF(I_i(NumD)) = HSF(I_i(NumD)) - 1 \quad (7)$$

It should be noted that $\boldsymbol{R}$ contains all 0's at the beginning of the retrieval process since no knowledge has been learned. After the first iteration of the first query session, the semantic

($\boldsymbol{R}$, *NumD*) = Function RFModelΘ (*AllPos*, *Neg*, $\boldsymbol{R}$, *NumD*), where each input parameter denotes the following:

- *AllPos*: The set of accumulated positively labeled images in a query session;
- *Neg*: The set of negatively labeled images in the current iteration of a query session;
- $\boldsymbol{R}$: The current semantic repository;
- *NumD*: T he number of learned semantic concepts, which equals to the valid dimensionality of the semantic features to date. Its maximum value is 2% of the total number of images in the database.

The update procedure within a query session:

1. Load in the semantic features of images in *AllPos* and *Neg* from $\boldsymbol{R}$.
2. Update high-level semantic features for every positively labeled image.
3. If *NumD* $\leqslant$ the maximum number of concepts
   3.1. Update the $NumD^{th}$ semantic feature in $\boldsymbol{R}$ for images in *AllPos* and *Neg*.
   3.2. Add *NumD* by 1.
4. Elseif *NumD* > the maximum number of concepts
   4.1. Find the column(s) with the most number of 0's in $\boldsymbol{R}$.
   4.2. If one such column exists, delete this column by moving all the afterwards columns one position ahead, which leaves the last column all 0's.
   4.3. Otherwise, find the first column whose sum of the absolute values is the smallest. Delete this column by moving all the afterwards columns one position ahead.
   4.4. Set *NumD* = the maximum number of concepts and update the $NumD^{th}$ semantic feature in $\boldsymbol{R}$ for images in *AllPos* and *Neg*.

Fig. 2 Algorithmic view of the RF model Θ

features of all the positively labeled images are all 0's (i.e., the semantic features are empty) and their semantic features still stay the same after applying Eq. (5). We then employ Eqs. (6) and (7) to assign 1's and −1's to the 1st column in $\boldsymbol{R}$ for images in *AllPos* and *Neg*, respectively. As a result, for the following iterations of the first query session, at least one positively labeled image contains the semantic information for the learned semantic concepts. We can then employ Eq. (5) to propagate the learned semantic relations to other positively labeled images that do not contain any semantic information (i.e., whose semantic features are empty). This propagation strategy is guided by the observation that all positively labeled images should have similar semantic concepts since they are all semantically similar to the query. In our system, we simply perform the average operation to make the semantic features of all positively labeled images in a query session as the same. We finally employ two strategies as summarized in Eqs. (6) and (7) to update the semantic relation between the current query and retrieved database images. These update strategies are guided by the following observations: 1) If one of the positively labeled images is labeled by the same or multiple users as relevant to the query image multiple times, the semantic relation between the positively labeled image and the query is reliably strong. 2) If one of the negatively labeled images is labeled by the same or multiple users as irrelevant to the query image multiple times, the semantic relation between the negatively labeled image and the query is reliably weak. 3) The reliable information can be propagated to other positively labeled images and negatively labeled images based on the semantic transitivity. To this end, we double the current semantic feature value for positively labeled images. Since the initial value of the current semantic feature may be 0's, this doubling operation is replaced by the addition by 1 operation as summarized in the first condition of Eq. (6) to ensure that the semantic feature value is increased. For negatively labeled images, we decrease their current semantic feature value by 1.

Figure 3 illustrates the basic idea of the propagation and update strategies. To facilitate discussion, we assume that $\boldsymbol{R}$ has been properly updated after performing three query sessions. Figure 3a shows the snapshot of $\boldsymbol{R}$ that contains only pertinent information related to three positively labeled images (e.g., Im3, Im5, and Im42) and two negatively labeled images (e.g., Im210 and Im420) for the 4th query, which is Im3 submitted by the user. Here, the semantic features of Im3, Im5, and Im42 are (2, 1, 0), (0, −1, 1), and (0, 0, 0), respectively. We perform

	Q1	Q2	Q3
Im3	2	1	0
Im5	0	-1	1
Im42	0	0	0
Im210	0.5	-1	0
Im420	0.3	0	0

(a)

	Q1	Q2	Q3
Im3	2/3	0	1/3
Im5	2/3	0	1/3
Im42	2/3	0	1/3
Im210	0.5	-1	0
Im420	0.3	0	0

(b)

	Q1	Q2	Q3	Q4
Im3	2/3	0	1/3	1
Im5	2/3	0	1/3	1
Im42	2/3	0	1/3	1
Im210	0.5	-1	0	-1
Im420	0.3	0	0	-1

(c)

Fig. 3 Illustrate of propagation and update strategies in the RF model Θ. **a** Snapshot of $\boldsymbol{R}$ for the first three queries. **b** Snapshot of $\boldsymbol{R}$ after the propagation strategy (Eq. (5)). **c** Snapshot of $\boldsymbol{R}$ after the update strategy (Eqs. (6) and (7))

the average operation on these three semantic features and replace each of these three semantic features by its averaged semantic feature (2/3, 0, 1/3), as shown in Fig. 3b. It clearly shows that Im42's semantic feature are no longer empty after this propagation operation and all three positively labeled images share the same semantic features. We then apply Eqs. (6) and (7) to assign 1's and −1's to the 4th value of the semantic features of the three positively labeled images and the two negatively labeled images, respectively. The snapshot of $\boldsymbol{R}$ is shown in Fig. 3c. It should be noted that the values in the other rows and columns are not affected after this propagation and update process.

At the end of each query session, we decide whether the column representing the current query session can be merged with the other columns in $\boldsymbol{R}$ by checking the record of past queries corresponding to each column. Specifically, we first record the names of query image(s) associated with each column in $\boldsymbol{R}$, which represent query's semantic concepts. We then find out whether any positively labeled image has been submitted as the query image in the past. If one past query image is located, we know that the current query image shares similar semantic meanings with the past query image. We can then merge these two columns, one representing semantic concepts of the past query image and one representing semantic concepts of the current query image, into one new column. If several past query images are located, we know that the current query image shares similar semantic meanings with these past query images. Based on the semantic transitivity, these past query images share similar semantic meanings as well. We can then merge the columns corresponding to these past query images with the column corresponding to the current query image to form one new column. To this end, we set the values of the column corresponding to the current query as the average values of these merged columns and set 0's to columns corresponding to the past query images. Correspondingly, we update the record of the names of query image(s) by updating the names of query images in the newly merged column as the union of the names of query images in all the merged columns and resetting the names of query images in the other columns as empty.

Figure 4 demonstrates this merging process. The first table shows the sample values in the first column of $\boldsymbol{R}$ after performing the first query session. The second table shows the sample values in the 1st and 20th column of $\boldsymbol{R}$ after performing 20 queries. Here, we assume that no merging operations occur during this query process for the ease of discussion. At the end of 20th query session (i.e., the query image is Im1460), we find that one positively labeled image Im1432 has been submitted as the query image in the past, which is the first query image. As a result, we merge the first and the 20th columns by performing the averaging operation. Its merged result is shown in the last table and the semantic concepts of this merged column are recorded as the union of the names of query images in the 1st and 20th columns. Correspondingly, we set the values in the 1st column as all 0's and record the semantic concept of this column as empty.

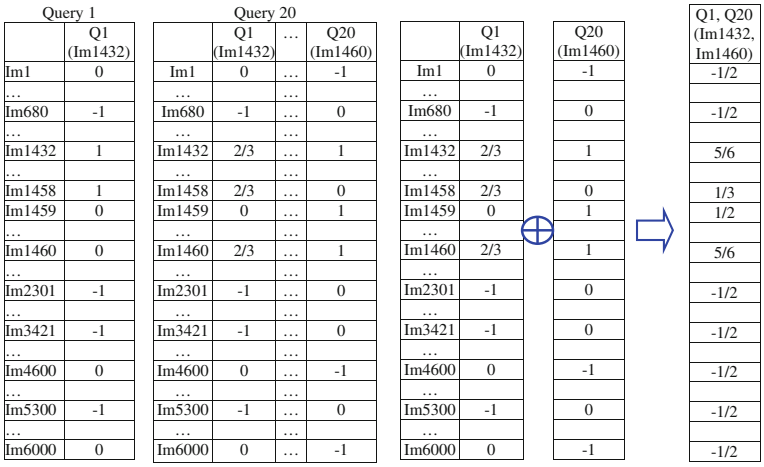

Fig. 4 Illustration of the merging strategy

2.4 RF model Ω: update query's visual and semantic features

RF model Ω aims to update query's visual and semantic features based on user's RF in the current query session. Our proposed update strategy moves the query vector towards the subspace that contains more relevant images by reinforcing semantically relevant features. The following observation supports our update strategy: Positively labeled images share similar semantic concepts as the query image. As a result, high-level semantic features of the query image and positively labeled images should be close to each other. Following this observation, we update the query's semantic features and calculate the semantic similarity between the query image and each database image as follows:

1. Calculate *HSFPosMean*, the average semantic features of all accumulated positively labeled images, as query's desired semantic features.
2. For each database image DIm_i
   2.1. Obtain its high-level semantic features $HSFIm_i$.
   2.2. Apply Eq. (3) to compute its semantic similarity *PosSim* between *HSFPosMean* and $HSFIm_i$.

Similarly, for the low-level visual features, we compute the average of visual features of all accumulated positively labeled images as the desired visual features of the query image. We then apply the above strategy to compute the visual similarity between the query image and each database image. Finally, we apply Eq. (4) to subtract the semantic similarity from the visual dissimilarity to obtain the overall dissimilarity between the query image and each database image.

3 Experimental results

We conduct a set of carefully designed experiments to evaluate the performance of our proposed complementary RF-based CBIR system on five image databases. In Section 3.1, we explain these five image databases containing 2,000 images, 6,000 images, 8,000 images, 12,000

images, and 22,000 images, respectively. In Section 3.2, we evaluate the performance of the proposed CBIR system together with seven state-of-the-art long-term-based peer CBIR systems by performing both in-depth and in-breadth analyses. In Section 3.3, we evaluate the performance of the proposed CBIR system together with its four variant systems by performing in-breadth analysis. In Section 3.4, we evaluate the complexity and the storage effectiveness of all eight state-of-the-art long-term-based CBIR systems.

3.1 Five image databases

To simplify the retrieval process, we manually organize the database images into several semantic classes. As a result, the image relevance is automatically determined by checking whether the returned images belong to the same manually defined class as the query. It should be noted that the ground truth is exclusively used to evaluate the retrieval performance and is not assumed to provide additional class-related information for our proposed system. Thus, our proposed technique can handle any new database. To this end, we collect the following images to evaluate our retrieval performance:

- 6000 COREL images: We carefully select 60 distinct categories from the COREL database. Each of these 60 categories contains exactly 100 images. These images cover a variety of real-world scenes such as cars, buildings, animals, etc.
- 2000 Flickr images: We download a large collection of images from the social photography site http://www.flickr.com through its API. Specifically, we use Flickr's API to download top 200 images (based on relevance) for each of the following 20 categories: American flag, boat, cat, Coca-Cola can, fire flame, fireworks, honey bee, Irish flag, keyboard, Mexico City taxi, Mountie, New York taxi, orchard, ostrich, Pepsi can, Persian rug, samurai helmet, snow boarding, sushi, and waterfall. We then manually pick 100 images that best represent the category.
- 4000 online images: We download another set of images from the following two websites: http://images.google.com and http://picasa.google.com through their APIs. Similarly to Flickr images, we download top 200 images for each of 40 distinct keywords and manually pick the most appropriate 100 images.
- 22000 NUS-WIDE images: We download a set of real-world web images from National University of Singapore [6]. This set contains 269,648 images and the associated tags from Flickr, with a total number of 5,018 unique tags. The authors organize semantic related images into their respective folder such as airport, albatross, cars, police, plants, etc. It should be noted that lots of folders have less than 100 images so we do not use them in our experiment. To this end, we randomly choose 100 images from each of 81 concepts, which are used for annotation evaluation and contain a sufficient number of images. We then choose 100 images from each of additional 139 concepts with a sufficient number of images.

Three graduate students are further asked to check the appropriateness for each image to be assigned to a particular semantic class based on the majority of the agreement. The inappropriate images are replaced by other appropriate images approved by the majority of the three graduate students. We then build five image databases as follows: 1) the 6000-image database containing 6000 COREL images;2) the 2000-image database containing 2,000 photos taken by different persons; 3) the 8000-image database containing 6000 COREL images and 2000 Flickr images; 4) the 12000-image database containing 6000 COREL images, 2000 Flickr images, and 4,000 online images; 5) the 22000-image database containing 22000 NUS-WIDE images

from 220 distinct categories, where each category has 100 images. Each image in the database is represented by a 100-dimensional low-level visual feature vector and a 2 % × N-dimensional high-level semantic feature vector, where N equals the number of images in the database.

3.2 Comparative performance evaluation

To simulate the practical retrieval process of online users, we randomly generate a sequence of query images to conduct various experiments. At each query session, the proposed CBIR system refines its retrievals by taking advantages of both short-term and long-term learning techniques and exploiting the synergism between them for several iterations. We use the retrieval precision (RP), which is defined as the ratio between the number of relevant images returned and the total number of images returned, as our performance measure. In each experiment, we perform four iterations of RF with the top 25 images returned in each iterative step.

To evaluate the performance of our CBIR system, we perform two sets of experiments. We calculate the average RP at each iteration of a query session that is achievable with any query image in the database. For example, at the 1200th query session, we calculate the average RP by using every database image as a query. The query session is used as a time stamp. The average RP is the performance measure that evaluates the capability achieved by our system at that time stamp. Thus, the RF learning Θ is not performed during this evaluation to make our system stay at a constant performance level within this period. The comparative performances are analyzed both in depth with respect to the number of queries (NQ) and in breadth with respect to the number of feedback iterations (NF).

3.2.1 In-depth performance analysis

For the in-depth performance analysis, we carry out various experiments using 2 %, 5 %, and 10 % of database images to build the long-term learning bases (repositories). In other words, we perform a different number of query sessions (e.g., 2 %, 5 %, and 10 % of database images) to evaluate the capability achieved by the proposed CBIR system at their corresponding time stamps, respectively. In addition, we compare the proposed learning system with the following seven state-of-the-art long-term-based CBIR systems at the same three time stamps:

- Our implemented Hoi's log-based system (i.e., Log-based + global soft label SVM) [12]: This system uses the soft label SVM to deal with the noisy log data collected in multiple RF sessions. We empirically chose the parameter γ in radial basis function (RBF) kernel to be 0.5 using a separate validation data set. The two weight parameters C_H and C_S for balancing the importance between hard-labeled data and soft-labeled data are set to be 1 and the absolute value of the high-level relevance score computed from the log data, respectively.
- Our implemented Han's memory learning system (i.e., Memory learning + global SVM) [10]: This system uses the RBF kernel SVM to classify the database images to return similar images. It computes the ratio of the co-positive feedback frequency and the co-feedback frequency to measure the semantic similarity among images. We empirically set the parameter γ in the RBF kernel as 0.5 using a separate validation data set.
- Our implemented Yin's virtual feature-based system (i.e., Virtual feature learning) [21]: This system uses the learned virtual features to compute the similarity among images. All the parameters can be automatically computed using the formulas specified in the paper. The dimensionality of the virtual features is set to be the number of training images.

- Qi's dynamic semantic clustering system (i.e., DSC + block based fuzzy SVM) [16]: This system applies block-based fuzzy SVM and DSC learning to adaptively learn and update the semantic categories for more accurate retrieval. The parameter γ in the RBF kernel is set to be 0.5 and the merging percentage is set to be 0.25 as specified in [16].
- Chang's long-term semantic clusters based manifold ranking system (i.e., SC-based manifold) [2]: This system uses SVM-based RF technique to create semantic clusters for computing the reliability score of each database image, which is incorporated into the affinity matrix to construct a weighted manifold structure for efficient image retrieval. The merging threshold for creating semantic clusters is set to be 0.5, the parameter σ (the overall variance of image features) for computing the weight in the affinity matrix is set to be 0.05, the convergence rate of the affinity matrix α is set to be 0.99, and the parameter γ in the RBF kernel is set to be 0.5.
- Chang's long-term weighted semantic manifold ranking system (i.e., Semantic manifold) [3]: This system builds upon SC-based manifold to include the fuzzy membership and the high-level semantic similarity score in the affinity matrix to construct a semantic manifold ranking system. All the fixed parameters are the same as the ones used in [2].
- Xiao's dynamic semantic feature-based long-term cross-session learning system (i.e., DSF-based cross-session learning) [20]: This system uses the learned dynamic semantic features to measure the similarity among images. All the parameters can be automatically computed using the formulas specified in [20].

Figure 5 shows a comparison among the average RP of the above eight CBIR systems using three kinds of NQs, which equal to 2 %, 5 %, and 10 % of the 6000 COREL benchmark images, to build the long-term repositories. Specifically, Fig. 5a shows the legend for each of the eight CBIR systems in comparison. Figure 5b, c, and d show the average RP for all database images obtained by eight CBIR systems at three time stamps, corresponding to the 120th query, the 300th query, and the 600th query, at three displays, namely, the one without any RF iteration (iteration 0), the one after the first iteration of RF (iteration 1), and the one after the second iteration of RF (iteration 2), respectively. We observe the following from Fig. 5:

- At each of the three iterations, the average RP of each of eight CBIR systems monotonically increases when the NQ increases.
- At iteration 0, the proposed CBIR system achieves the third best RP at the time stamps of the 120th query and the 300th query and the second best RP at the time stamp of the 600th query. The average RP at the time stamp of the 120th, 300th, and 600th query is 59.55 %, 72.19 %, and 79.95 %, respectively.
- At iteration 1, the proposed CBIR system achieves a worse RP of 80.16 % than the RP of 85.12 % for Chang's semantic manifold system at the time stamp of the 120th query. It achieves the best RP of 89.63 % at the time stamp of the 300th query and improves the second best CBIR system (Chang's semantic manifold system) by 0.74 %. It also achieves the best RP of 93.21 % at the time stamp of the 600th query and improves the second best CBIR system (Chang's semantic manifold system) by 2.60 %.
- At iteration 2, our proposed CBIR system achieves a comparable RP of 85.33 % as the RP of 85.76 % for Qi's dynamic semantic clustering system and the RP of 86.28 % for Chang's semantic manifold system at the time stamp of the 120th query. It achieves the best RP of 93.20 % at the time stamp of the 300th query and improves the second best CBIR system (Chang's semantic manifold system) by 3.35 %. It also achieves the best RP of 95.17 % at the time stamp of the 600th query and improves the second best CBIR system (Xiao's DSF-based cross-session learning system) by 2.56 %.

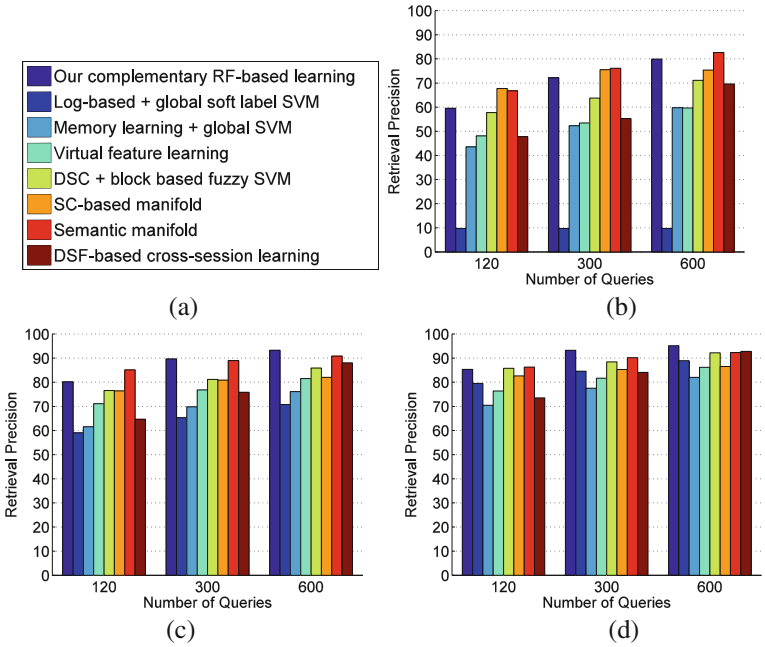

Fig. 5 Comparison of eight long-term-based CBIR systems at three time stamps (the 120th query, the 300th query, and the 600th query) at three displays. **a** Legend of the bar graph in (**b**), (**c**) and (**d**). **b** Comparison at the display of iteration 0. **c** Comparison at the display of iteration 1. **d** Comparison at the display of iteration 2

These observations lead to the following conclusion: Our proposed CBIR system efficiently exploits the synergism between RF model Θ and RF model Ω to achieve faster learning and therefore better retrieval performance after performing a sufficient number of queries. Our experiments on the other three databases using three kinds of NQs, which equal to 2 %, 5 %, and 10 % of the database images, further confirm these observations and the above conclusion. Their corresponding plots are similar to the plots demonstrated in Fig. 5b, c, and d. As a result, we omit these plots to save the space.

3.2.2 In-breath performance analysis

For the in-breath performance analysis, we carry out various experiments using 10 % of database images to build the long-term learning repository for five databases. This choice is supported by the RP results shown in Fig. 5. That is, larger NQ sessions lead to more learning. As a result, we evaluate the capability achieved by eight CBIR systems at the time stamp corresponding to 10 % of the total number of images in the current retrieved database. To this end, we compare our proposed complementary RF-based CBIR system with seven long-term-based CBIR systems in terms of the RP at each iterative step. Figure 6 shows the average RP for all database images obtained by eight CBIR systems at three displays (e.g., the 0th iteration, the 1st iteration, and the 2nd iteration). Specifically, Fig. 6a shows the legend of the bar graph in the following sub-figures. Figure 6b, c, d, e and f demonstrate the average RP obtained by eight CBIR systems at three displays for the 2000-image, 6000-image, 8000-image, 12000-image, and 22000-image databases respectively. It should be noted that Chang's SC-based manifold ranking system and Chang's semantic manifold

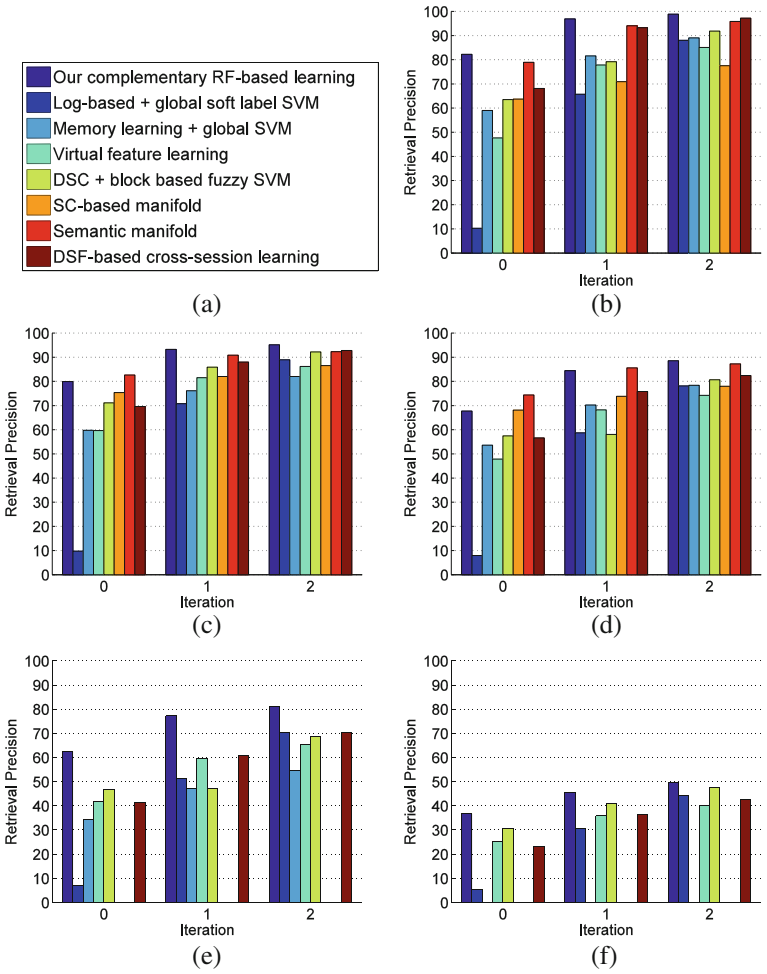

Fig. 6 Comparison of eight long-term-based CBIR systems in terms of RP at three displays for four databases. **a** Legend of the bar graph in (**b**), (**c**), (**d**), (**e**), and (**f**). **b** The 2000-image database. **c** The 6000-image database. **d** The 8000-image database. **e** The 12000-image database. **f** The 22000-image database

ranking system cannot run on our computer for the 12000-image database due to its requirement of several matrices of 12,000×12,000. As a result, the bars corresponding to these two systems in Fig. 6e are missing. Similarly, the bars corresponding to these two systems and Han's memory learning system are missing in Fig. 6f since they require several matrices of 24,000×24,000 and cannot run on our computer. We observe the following from Fig. 6:

- The average RP of all eight CBIR systems increases when the NF increases.
- The average RP of all eight CBIR systems decreases when the image database grows larger.
- Our CBIR system outperforms the other seven CBIR systems at all iterations for the two largest databases: the 12000-image and 22000-image databases. For the 12000-image database, our system achieves the average RP of 62.49 % at iteration 0, 77.21 % at iteration 1, and 81.26 % at iteration 2. At the last iteration, it improves the second best

system, Xiao's DSF-based cross-session learning system, by 15.23 % in terms of RP. For the 22000-image database, our system achieves the average RP of 36.71 % at iteration 0, 45.71 % at iteration 1, and 49.71 % at iteration 2. At the last iteration, it improves the second best system, Qi's dynamic semantic clustering system, by 4.35 % in terms of RP.

- Our CBIR system outperforms the other seven CBIR systems at the last iteration for the three smaller databases: the 2000-image, 6000-image, and 8000-image databases. For the 2000-image database, our system achieves the average RP of 82.20 % at iteration 0, 96.90 % at iteration 1, and 98.82 % at iteration 2. At the last iteration, it improves the second best system, Xiao's DSF-based cross-session learning system, by 1.69 % in terms of RP. For the 6000-image database, our system achieves the average RP of 79.95 % at iteration 0, 93.21 % at iteration 1, and 95.17 % at iteration 2. At the last iteration, it improves the second best system, Xiao's DSF-based cross-session learning system, by 2.56 % in terms of RP. For the 8000-image database, our system achieves the average RP of 67.73 % at iteration 0, 84.42 % at iteration 1, and 88.51 % at iteration 2. At the last iteration, it improves the second best system, Chang's semantic manifold ranking system, by 1.46 % in terms of RP.

The existing long-term CBIR systems use various different strategies to store the RF information in their respective long-term learning bases. They achieve inferior performance mainly because they exclusively store 1's and -1's in their long-term learning spaces to simply represent the relevancy and irrelevancy of the images to each query concept, respectively. In our system, we apply the average operation to propagate the learned semantic relations to other positively labeled images that do not contain any semantic information (i.e., whose semantic features are empty). We also apply the update strategies to ensure the relevance or irrelevance information is modified proportionally to the number of consistent labeling information provided by the same or multiple users. As a result, our semantic repository stores three kinds of values (positive values, negative values, and 0's) where a larger positive value indicates the database image likely to possess the corresponding semantic concept and a smaller negative value indicates the database image unlikely to possess the corresponding semantic concept. Compared to the other learning bases, ours provides more precise information and therefore our system achieves better retrieval performance especially when a larger database is involved.

3.3 Comparative performance evaluation of variants of the proposed CBIR system

To further evaluate the effectiveness of the proposed complementary RF-based CBIR system, we implement four variants of the proposed system:

- Variant 1: The CBIR system that uses 100-dimensional low-level visual features without integrating high-level semantic features learned from the long-term learning technique.
- Variant 2: The CBIR system that uses 100-dimensional low-level visual features and high-level semantic features with equal weight contribution (e.g., W_l=W_h=0.5 in Eq. (4)).
- Variant 3: The CBIR system that uses the 512-dimensional low-level GIST descriptor [15] and high-level semantic features with the proposed weight contribution (e.g., W_l=0.1 and W_h=0.9 in Eq. (4)).
- Variant 4: The CBIR system that uses the combined 100-dimensional low-level features and 512-dimensional GIST descriptor and high-level semantic features with the proposed weight contribution (e.g.,W_l=0.1 and W_h=0.9 in Eq. (4)).

We use the first two variant systems to evaluate the effectiveness of the combined low-level visual features and high-level semantic features with the proposed weight contribution

(e.g., W_l=0.1 and W_h=0.9). We use the last two variant systems to evaluate the effectiveness of the proposed 100-dimensional low-level visual features. Here, we implement the variant 3 system using the 512-dimensional GIST descriptor [15] to replace the 100-dimensional low-level visual features to represent an image in a large scale imagery database. The GIST descriptor is a set of perceptual features including naturalness, openness, roughness, expansion, and ruggedness to represent the dominant spatial structure of a scene. It is similar in spirit to the local SIFT descriptor and has been mainly used for same location/object recognition and copy detection [8]. We also implement the variant 4 system by incorporating the 100-dimensional low-level features with the 512-dimensional GIST descriptor to more accurately represent an image using color, edge, texture, and perceptual scene-related features without requiring any form of segmentation.

Figure 7 shows the average RP for two representative database images obtained by the proposed CBIR system and its four variant systems at three displays (e.g., the 0th iteration, the 1st iteration, and the 2nd iteration). Specifically, Fig. 7a and b demonstrate the average RP obtained by five CBIR systems at three displays for the 6000 COREL benchmark images and the 22000 NUS-WIDE images, respectively. We observe the following from Fig. 7:

- The four CBIR systems integrating low-level visual features and high-level semantic features, namely, the proposed system, variant 2, variant 3, and variant 4, achieve higher RP than the variant 1 system (using only low-level visual features) at each of three iterations for both databases.
- The proposed system achieves a slightly better RP than the variant 2 system at each of three iterations for both databases. This clearly shows that the effectiveness of the synergism between short-term and long-term learning techniques. In other words, high-level semantic features which are extracted from the long-term learning technique are paramount in helping the CBIR system to achieve impressive RP. It should be noted that the low-level visual features are also needed especially when the semantic features are not learned (i.e., empty).
- The proposed system achieves better RP than the variant 3 system at each of the three iterations for both databases. This clearly shows the proposed 100-dimensional low-level features can be better incorporated into the system to achieve impressive RP than the 512-dimensional GIST descriptor.
- The variant 4 system achieves better RP than the proposed system at each of the three iterations for both databases. This improvement is more obvious when retrieving images from a large image database. Specifically, for the 22000-image database, the variant 4

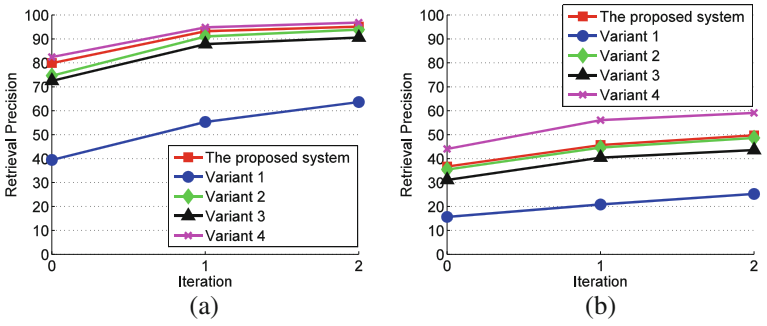

Fig. 7 Comparison of the proposed CBIR system and its four variant systems in terms of RP at three displays for two databases. **a** The 6000-image database. **b** The 22000-image database

system achieves the average RP of 44.06 % at iteration 0, 56.09 % at iteration 1, and 59.10 % at iteration 2. At the last iteration, it improves the proposed system by 18.89 % in terms of RP. This clearly shows that the effective low-level features significantly boost the learning system to achieve impressive RP at each iteration. To this end, we can incorporate the 512-dimensional GIST descriptor into the proposed system to further improve the RP with any additional computation cost.

Table 1 summarizes the average RP for five database images obtained by the proposed CBIR system and the variant 4 system at three displays. It clearly shows that incorporating the 512-dimensional GIST features significantly improves the average RP at each of three iterations for the larger databases. The larger the database, the more RP improvement is.

3.4 Comparative complexity and storage evaluation

Finally, we compare the eight CBIR systems from the perspectives of the storage and computational complexity.

To facilitate discussion, we denote N by the total number of images in the database. Our proposed CBIR system uses less storage space to save semantic knowledge learned to date. Specifically, it requires $O(N\times N\times 0.02)$ space where $N\times 0.02$ is a reasonable estimate for the maximal number of concepts contained in all database images. Qi's dynamic semantic clustering system requires $O(N\times NumC)$ space where $NumC$ is the number of learned clusters. Based on their experiments, $NumC$ was approximately 68, 139, and 326 for the 6000-image, 12000-image, and 22000-image databases, respectively. All the other long-term-based CBIR systems require $O(c\times N\times N)$ space. The constant c's in Yin's virtual feature learning system, and Xiao's DSF-based cross-session learning system are a fractional number (e.g., 0.1), the constant c's in Hoi's log-based system equals 1, the constant c's in Han's memory learning system equals 3, and the constant c's in Chang's SC-based manifold and Chang's semantic manifold systems equal 4. It clearly shows that the proposed CBIR system requires a little more storage space as Qi's dynamic semantic clustering system and a small fraction of storage space as required by the

Table 1 Comparison of the proposed and the variant 4 system in terms of RP at three displays for five databases

Databases	Methods	RP at iteration 0	RP at iteration 1	RP at iteration 2
2000 images	Proposed system	82.20 %	96.90 %	98.82 %
	Variant 4 system	82.77 %	96.82 %	98.18 %
6000 images	Proposed system	79.95 %	93.21 %	95.17 %
	Variant 4 system	82.49 %	94.84 %	96.82 %
8000 images	Proposed system	67.73 %	84.42 %	88.51 %
	Variant 4 system	71.56 %	87.89 %	90.88 %
12000 images	Proposed system	62.49 %	77.21 %	81.26 %
	Variant 4 system	68.73 %	84.25 %	87.45 %
22000 images	Proposed system	36.71 %	45.71 %	49.71 %
	Variant 4 system	44.06 %	56.09 %	59.10 %

other six long-term-based CBIR systems. This efficient storage is necessary for real-world situations with databases of millions of images.

The complexity of our proposed algorithm is $O(N \times N \times 0.02)$. The complexity of Qi's dynamic semantic clustering system is $O(N \times NumC + NumC \times NumC)$. The complexity of the other six long-term-based CBIR systems is $O(c \times N \times N)$. It clearly shows that our proposed is computationally efficient.

4 Conclusions and future work

In this paper, we propose a complementary RF-based CBIR system. This system exploits the synergism between short-term and long-term learning techniques to improve the retrieval performance. It combines these two complementary learning techniques in a hope to eradicate the weakness of individual learning. Our contributions are:

- Seamlessly integrating low-level visual features and high-level semantic features in short-term learning to effectively represent the query in a single retrieval session.
- Dynamically updating the high-level semantic features based on users' query concept to represent the image's semantic concept more accurately.
- Quickly constructing an adaptive semantic repository in long-term learning to store retrieval patterns of historical query sessions.
- Efficiently merging the similar semantic concepts to keep the adaptive semantic matrix compact.
- Effectively composing low-level visual and high-level semantic similarity measure to estimate the semantic relevance among images.

We plan to test the proposed technique for its effectiveness and scalability on a larger database by comparing with additional emerged state-of-the-art systems. We will incorporate erroneous feedback, which is resulted from the inherent subjectivity of judging relevance, user laziness, or maliciousness, into the current system to evaluate its resilience to the noisy feedback. Next, 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 explore the potential of applying the proposed technique in the image annotation task by propagating users' annotations to related images.

Acknowledgments This material is based upon work supported in part by the National Science Foundation under Grant No. 0850825.

References

1. Aksoy S, Haralick RM (2001) Feature normalization and likelihood-based similarity measures for image retrieval. Pattern Recogn Lett Spec Issue Image Video Indexing Retr 22(5):563–582
2. Chang R, Qi X (2011)Semantic clusters based manifold ranking for image retrieval. In: IEEE Int Conf on Image Processing, Brussels, Belgium, pp. 2473–2476

3. Chang R, Xiao Z, Wong KS, Qi X (2012). Learning a weighted semantic manifold for content-based image retrieval. In: IEEE Int Conf on Image Processing, Orlando, Florida, USA, pp.2402–2404
4. Chen Y, Rege M, Dong M, Fotouhi F (2007) Deriving semantics for image clustering from accumulated user feedbacks. In: 15th ACM Int. Conf. on Multimedia, Augsburg, Germany, pp. 313–316
5. Cheng H, Hua KA, Vu K (2008) Leveraging user query log: toward improving image data clustering. In: 7th ACM Int Conf on Image and Video Retrieval, Niagara Fall, ON, Canada, pp. 27–36
6. Chua T, Tang J, Hong R, Li H, Luo Z, Zheng Y (2009) NUS-WIDE: a real-world web image database from National University of Singapore. In: Proc of the ACM Int Conf on Image and Video Retrieval, Santorini, Fira, Greece, Article No. 48
7. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surv 40(2):1–60
8. Douze M, Jegou H, Sandhawalia H, Amsaleg L, Schmid C (2009) Evaluation of GIST descriptors for web-scale image search. In: Proc of the ACM Int Conf on Image and Video Retrieval, Santorini, Fira, Greece, Article No. 19
9. Fechser S, Chang R, Qi X (2010) Inter-query semantic learning approach to image retrieval. In: IEEE Int Conf on Acoustics, Speech, and Signal Processing, Dallas, Texas, USA, pp. 1246–1249
10. Han J, Ngan KN, Li M, Zhang HJ (2005) A memory learning framework for effective image retrieval. IEEE Trans Image Process 14(4):511–524
11. He X, King O, Ma WY, Li M, Zhang H (2003) Learning a semantic space from user's relevance feedback for image retrieval. IEEE Trans Circ Syst Video Tech 13(1):39–48
12. Hoi SCH, Lyu MR, Jin R (2006) A unified log-based relevance feedback scheme for image retrieval. IEEE Trans Knowl Data Eng 18(4):509–524
13. Lew MS, Sebe N, Djeraba C, Jain R (2006) Content-based multimedia information retrieval: state of the art and challenges. ACM Trans Multimed Comput Comm Appl 2(1):1–19
14. Liu Y, Chen X, Zhang C, Sprague A (2009) Semantic clustering for region-based image retrieval. J Vis Comm Image Represent 20(2):157–166
15. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175
16. Qi X, Barrett S, Chang R (2011) A noise-resilient collaborative learning approach to content-based image retrieval. Int J Intell Syst 26:1153–1175
17. Qi X, Chang R (2007) Image retrieval using transaction-based and SVM-based learning in relevance feedback sessions. In: 4th Int Conf on Image Analysis and Recognition, Montreal, Canada, pp. 637–649
18. Shah-Hosseini A, Knapp GM (2006) Semantic image retrieval based on probabilistic latent semantic analysis. In: 14th ACM Int Conf on Multimedia, Santa Barbara, CA, USA, pp. 703–706
19. Thomee B (2010) A picture is worth a thousand words: content-based image retrieval techniques. Ph.D. Dissertation, Leiden University, Netherlands
20. Xiao Z, Clark MJ, Wong KS, Qi X (2012) Dynamic semantic feature-based long-term cross-session learning approach to content-based image retrieval. In: IEEE Int Conf on Acoustics, Speech, and Signal Processing, Kyoto, Japan, pp. 1033–1036
21. Yin PY, Bhanu B, Chang KC, Dong A (2008) Long-term cross-session relevance feedback using virtual features. IEEE Trans Knowl Data Eng 20(3):352–368
22. Yoshizawa T, Schweitzer H (2004) Long-term learning of semantic grouping from relevance feedback. In: 6th ACM SIGMM Int Workshop on Multimedia Information Retrieval, New York, NY, USA, pp. 165–172
23. Zhou XS, Huang TS (2003) Relevance feedback in image retrieval: a comprehensive review. ACM Multimed Syst 8(6):536–544

Zhongmiao Xiao received the BS degree in Software Engineering from Beihang University, Beijing, China, in 2007, the MS degree in Computer Science from University of Montana, Missoula, USA in 2009. He is a PhD student in Computer Science at Utah State University since 2009. His research interests include content-based image retrieval, machine learning, and computer vision.

Dr. Xiaojun Qi received the BS degree in Computer Science from Donghua University, Shanghai, China, in 1993, the MS degree in Pattern Recognition and Intelligent System from Shenyang Institute of Automation, the Chinese Academy of Sciences, Shenyang, China, in 1996, and the PhD degree in Computer Science from Louisiana State University, Baton Rouge, USA in 2001. In Fall 2002, she joined the Department of Computer Science at Utah State University as a tenure-track assistant professor. In April 2008, she received tenure and was promoted to Associate Professor. She has authored and co-authored more than 60 peer-reviewed journal and conference publications. She has been served as technical program committees for 14 international conferences. She is a senior IEEE member since 2010. Her research interests include content-based image retrieval, digital image watermarking, pattern recognition, and computer vision.