LEARNING FROM RELEVANCE FEEDBACK SESSIONS USING A K-NEAREST-NEIGHBOR-BASED SEMANTIC REPOSITORY

Matthew Royal[1], Ran Chang[2], and Xiaojun Qi[2]

[1]matthew.royal@gmail.com
Pensacola Christian College, 250 Brent Lane, Pensacola, FL 32503
[2]ranchang@cc.usu.edu and [2]xqi@cc.usu.edu
Computer Science Department, Utah State University, Logan, UT 84322-4205

ABSTRACT

This paper introduces a flexible learning approach for image retrieval with relevance feedback. A semantic repository is constructed offline by applying the *k*-nearest-neighbor-based relevance learning on both positive and negative session-term feedback. This repository semantically relates each database image to a set of training images chosen from all semantic categories. The query semantic feature vector can then be computed using the current feedback and the semantic values in the repository. The dot product measures the semantic similarity between the query and each database image. Our extensive experimental results show that the semantic repository (6% size and 1/3 filling rate) based approach alone offers average retrieval precision as high as 94% on the first iteration. Comprehensive comparisons with peer systems reveal that our system yields the highest retrieval accuracy. Furthermore, the proposed approach can be easily incorporated into peer systems to achieve substantial improvement in retrieval accuracy for all feedback steps.

Index Terms — Content-based image retrieval, semantic repository, *k*-nearest-neighbor-based relevance learning

1. INTRODUCTION

Current research on content-based image retrieval aims to narrow down the semantic gap between low-level features and high-level semantics. As a viable solution, relevance feedback techniques are extensively studied to retrieve the user's desired images. These techniques can be generally classified into three categories: query reweighting, query shifting, and query expansion. The first two categories apply a nearest-neighbor sampling approach to refine query concept using the user's feedback upon the returned images. Specifically, query reweighting [1-3] assigns a new weight to each feature of the query, and query shifting [4-6] moves the query to a new point in the feature space. Query expansion [7, 8] uses a multiple-instance sampling approach to learn from samples around the neighborhood of the positive labeled instances. However, most approaches require seeding a query with appropriate positive examples and do not effectively use negative examples. In addition, these systems focus on the short-term learning by refining low-level features using current feedback. They do not utilize any previous feedback to narrow the semantic gap. To our knowledge, the semantic-space-based approach [9] is the only system that integrates short-term and long-term learning to improve retrieval performance. However, it is computationally intensive and intricate to incrementally construct the semantic space. Furthermore, that system does not integrate any negative examples, which correspond to the failure of the current classifier in learning.

To address the limitations of current retrieval systems, we propose a learning approach to retrieving the desired images using as few feedback steps as possible. To this end, we offline construct a semantic repository (SR) by applying the *k*-nearest-neighbor-based (*k*-nn) relevance learning on the session-term feedback, which contains the accumulated collection of all short-term positive and negative feedbacks for the current query search. This SR remembers the user's intent and, therefore, stores a semantic representation of each database image in terms of presence or absence of the semantics of each training image, which is randomly chosen from each semantic category in the image database. We further refine the query semantic features using the current feedback and the SR, and measure the semantic similarity between the query and each database image using the dot product. Information from positive and negative examples is also incorporated to improve learning.

The remainder of the paper is organized as follows: Section 2 presents our proposed approach in detail. Section 3 discusses several experimental results. Section 4 concludes the paper and shows the future work direction.

2. THE PROPOSED APPROACH

The block diagram of our proposed method is shown in Fig. 1. The system first returns top 20 images based on the low-level similarity between each database image and the query image. The user then indicates relevant and non-relevant images from the returned pool. This is designated as short-term feedback and is performed at each iteration step. The *k*-

nn-based relevance learning and SR-based semantic query modification techniques use session-term feedback to search the low-level feature database (LLFD) and the SR, respectively. The system then returns top 20 images ranked by SR-based semantic similarity scores. The user labels each returned image for the next iteration until he is satisfied with the results. The following subsections will explain each component of the proposed system in detail.

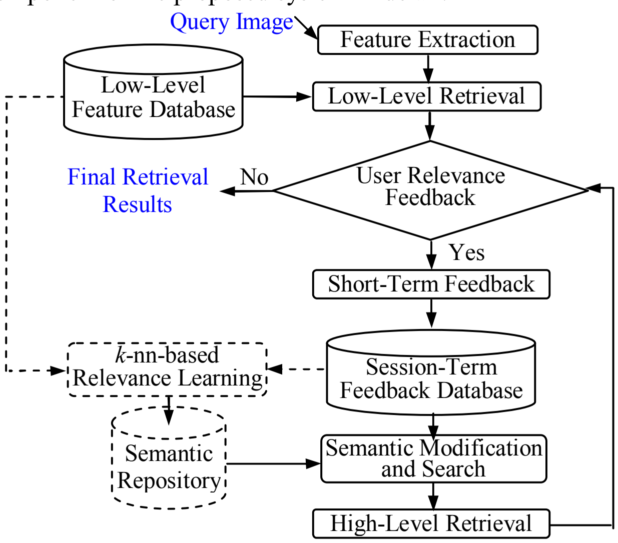

Fig. 1: The block diagram of our proposed system

2.1. Initial Retrieval Using Low-Level Features

The initial retrieval is essential to learning efficiency and iteration steps. We use the expanded MPEG-7 edge histogram descriptor (EHD) and the 64-bin (8×2×4) HSV-based scaled color descriptor (SCD) to extract low-level features for each database image. The expanded EHD is also constructed by appending the conventional 80-bin EHD with 5-bin global and 65-bin semi-global edges, which represent the sum of five edges in all blocks and 13 different block groupings, respectively.

To measure the similarity between the query and each image in the LLFD, the weighted and normalized Euclidean distances are computed for the EHD and SCD, respectively. Specifically, we assign a weight of 5 to the global edges. In our system, lower score corresponds to more similarity.

2.2. *K*-Nearest-Neighbor-Based Relevance Learning

We adopt the *k*-nn-based instance relevance technique [10] to assign each database image a query relevance score. Specifically, we characterize each image by a degree of relevance and non-relevance according to the dissimilarities from the nearest relevant and non-relevant images, respectively. This query relevance score is computed as:

$$relevance(I) = \left(1 + \frac{dR(I)}{dN(I)}\right)^{-1} \quad (1)$$

where I denotes a database image, and $dR(I)$ and $dN(I)$ respectively denote the Euclidean-distance-based dissimilarities from the nearest image in R and N, which correspondingly contain the subset of indices related to the relevant and non-relevant images retrieved so far. That is, $dR(I)$ and $dN(I)$ respectively measure the degree of relevance and non-relevance of image I, assuming small $dR(I)$'s are related to relevant images and large $dN(I)$'s are related to non-relevant images. The relevance score will be used to rank the images and return top matches. This learning and search process is as follows:

1. Perform initial retrieval (section 2.1) to return m images most similar to query image $q(t)$ and empty the session-term feedback database (STFD).
2. Let the user select relevant images, which are most similar to the user's query concept, while regarding the remaining returned images as non-relevant.
3. Add the new relevant and non-relevant images, which have not been retrieved in previous iterations, to STFD.
4. Compute the query relevance score for each database image using (1) and return m highest ranked images.
5. Repeat steps 2 through 4 until the user is satisfied with the retrieval results or the maximum iteration is reached.

2.3. The Semantic Repository Construction

The semantic repository (SR) stores semantic relationships between database images and semantic basis images (SBIs), which are composed of unique, randomly selected training images in each category. These SBIs correspond to columns of the SR and the database images correspond to rows of the SR. That is, the SR can be treated as an $M \times N$ matrix, where M is the number of database images and N is the number of SBIs. The SR is formally constructed offline as follows:

1. Initialize the $M \times N$ matrix as zeros.
2. For each row i in the SR
   2.1. Treat the database image at the ith row as a query.
   2.2. Perform 5-step k-nn-based relevance learning and search process to search the set of SBIs. Here, search stops at the maximum iteration of $\lceil N/m \rceil$, where m is the number of images returned at each iteration step.
   2.3. Update the columns corresponding to the relevant and non-relevant images retrieved so far at the ith row as 1's and -1's, respectively.

In our implementation, we ensure that all the images returned in each feedback step are exclusively new for the current query to increase the learning diversity and speed. Specifically, the ith row of the SR is the semantic feature vector (SFV) of database image x_i.

2.4. SR-Based Semantic Learning and Search

The SR-based semantic learning and search starts with finding the semantic rows corresponding to the relevant and non-relevant images labeled in the initial retrieval. The query's semantic feature vector (QSFV) is initialized as:

$$q_k(t) = (s_k^{R,1} \vee \ldots \vee s_k^{R,N_r}) \wedge \overline{(s_k^{N,1} \vee \ldots \vee s_k^{N,N_n})} \quad (2)$$

where $q_k(t)$ is the k^{th} element of the QSFV, $s_k^{R,i}$ and $s_k^{N,i}$ are the k^{th} element of the SFV of the i^{th} relevant and non-relevant images, respectively. The values of N_r and N_n correspond to the number of relevant and non-relevant images. Here, we treat all negative values as 0's.

The dot product computes the semantic similarity scores between query $q_k(t)$ and database image x_i.

$$S_{x_i}^{Sem} = x_i \cdot q(t) = \sum_k x_{i,k} q_k(t) \tag{3}$$

The higher the similarity score, the more semantically relevant the images are to the query. These scores ranging from -1 to 1 will be used to rank the database images and return the top matches for the user to judge.

For the following feedback iterations, relevant images reinforce the semantically relevant features of the QSFV and non-relevant images suppress the non-relevant features of the QSFV. This process is summarized by:

$$q_k(t+1) = \begin{cases} \alpha q_k(t) & (s_k^R = 1 \ or \ s_k^N = -1), q_k(t) \neq 0 \\ 1 & (s_k^R = 1 \ or \ s_k^N = -1), q_k(t) = 0 \\ q_k(t) & s_k^R = 0 \ or \ s_k^N = 0 \\ q_k(t)/\alpha & s_k^R = -1 \ or \ s_k^N = 1 \end{cases} \tag{4}$$

where $q_k(t+1)$ is the k^{th} element of the updated QSFV, s_k^R and s_k^N correspond to the k^{th} element of the SFVs of the relevant and non-relevant images, respectively. The parameter α is the adjustment rate and is set to 1.1.

3. EXPERIMENTAL RESULTS

To date, we have tested our retrieval system on 6000 images from COREL. These images have 60 distinct semantic categories with 100 images in each. A retrieved image is considered to be relevant if it belongs to the same category as the query image. To facilitate the evaluation process, we designed an automatic feedback scheme to construct the SR and model the SR-based query session. The retrieval accuracy is computed as the ratio of the relevant images to the total returned images. Three experiments have been specifically designed to evaluate the performance of different SRs, which are constructed by varying factors including the number of SBIs, the number of images returned per iteration, and the number of iterations. These three factors control the size and the fillings (i.e., the number of non-zeros) of the SR. In our experiments, we build a variety of SRs with different sizes using 5-step *k*-nn-based relevance learning and search. Specifically, the SRs with column numbers such as 1%, 2%, 4%, 6%, and 8% of the database images are constructed. The fillings using different returned images (i.e., 20, 30, 40, and 50) per iteration are further used to construct SRs using up to 6 iterations. For example, the largest SR has the size of 6000×480, where 8% of the database images in each category are chosen as the SBIs. The 50 returned images per iteration will fill in about 62.5% of each row with non-zeros. The initial Euclidean-distance-based retrieval accuracy for this database is 25.275% and is omitted from the figures to ensure readability of subsequent iterations. This initial accuracy can be improved by using more features. The use of the SR in the following feedback steps can also substantially improve the retrieval accuracy.

Experiment 1: The optimal number of SBIs. Fig. 2 shows the average retrieval accuracy for different sizes of the SRs with different fillings. It clearly shows the retrieval accuracy is above 90% for all tested sizes after the 1st iteration. Two large sizes (6% and 8%) achieve comparable retrieval results. Furthermore, the more fillings (the fewer 0's) in the SR, the better the retrieval accuracy. This interesting observation makes the smaller SR favorable. However, we decide to choose 6% of the database images as the SBIs (i.e., training images) to construct the SR in order to better represent the diversity of the images in each category. In addition, this SR can be built with rather low computational cost and storage requirement compared with larger SRs which achieve similar performance.

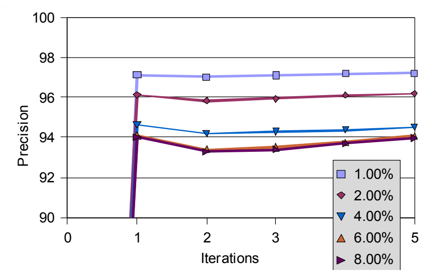

Fig. 2: Retrieval accuracy using different sizes of the SR

Experiment 2: The optimal number of images returned per iteration. Fig. 3 shows the average retrieval accuracy for different fillings of different sized SRs constructed by using different number of images returned per iteration. It clearly shows the retrieval accuracy is above 86% for all tested fillings after the 1st iteration. More fillings in the SR achieve higher retrieval accuracy. However, there is not much improvement in retrieval accuracy between using 40 and 50 returned images per iteration. As a result, we decide to choose 40 returned images per iteration to construct the SR. In addition, this SR can be reasonably built by a person, who is to label the returned images as positive or negative in the training.

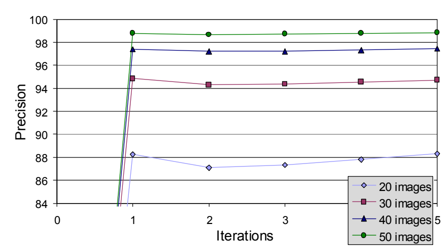

Fig. 3: Retrieval accuracy using different fillings of the SR

Experiment 3: The optimal number of iterations. Fig. 4 shows the average retrieval accuracy for different fillings of different sized SRs constructed by using different iterations. It clearly shows the retrieval accuracy is above 90% for all tested fillings after the 1st iteration. More fillings in the SR achieve higher retrieval accuracy. However, the improvement in retrieval accuracy, resulted from using multiple iterations to generate the SR, reduces sharply after the 4th iteration. As a result, we decide to choose 4 iterations to construct the SR. In addition, this choice minimizes the number of relevance feedback sessions a person must ensure in constructing the SR.

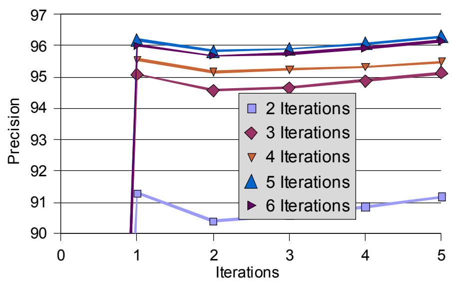

Fig. 4: Retrieval accuracy using different iterations

Experiment 4: Comparisons with peer CBIR systems. Based on the above experiments, we build our retrieval system by constructing the SR with 6% size and 1/3 filling rate (i.e., 40 returned images per iteration for a total of 4 iterations). Fig. 5 compares our proposed SR-based search, *k*-nn approach [10], a combined SR-based and *k*-nn approach, and MARS-1 [1] on 6000 COREL images. It shows our approach and the combined approach perform the best in all the feedback steps. Particularly, the average retrieval accuracy is dramatically improved to 94.9% after the 1st iteration for both approaches. Overall, the combined SR-based and *k*-nn approach almost doubles the retrieval accuracy for the *k*-nn approach. It also yields the retrieval accuracy of 1% higher than our proposed SR-based approach. However, our approach retrieves the desired images faster than the combined approach since *k*-nn-based relevance scores are not involved in the on-the-fly computation.

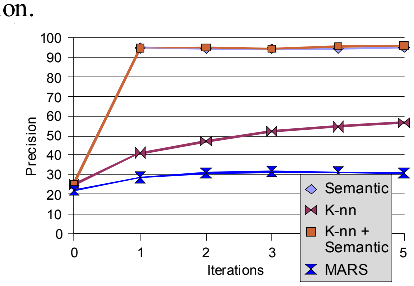

Fig. 5: Comparisons with other CBIR systems

4. CONCLUSIONS AND FUTURE WORK

This paper introduces a novel retrieval system with relevance feedback. The proposed SR-based system provides a flexible learning approach allowing users to benefit from past search results. It also returns a high percentage of relevant images on the first feedback iteration, yet has a negligible effect on speed. Major contributions consist of: 1) Construct the SR to learn the user's intention with appropriate number of SBIs and appropriate manner of fillings (the number of returned images per iteration and the total number of iterations); 2) Learn the semantic meaning of each database image using SBIs; 3) Apply the SR-based approach to achieve high retrieval accuracy with fewer than 3 feedback iterations; 4) Integrate SR into other relevance feedback based retrieval systems (e.g., *k*-nn-based system) to achieve higher retrieval accuracy. Experimental results show the proposed system outperforms peer systems and achieves remarkably high retrieval accuracy for a large database after the first iteration.

The principal component analysis will be considered to update the SR. Different learning approaches will be explored to construct the SR in a more systematic manner. The performance in a multi-class database, where an image may belong to multiple semantic classes, will be evaluated.

5. REFERENCES

[1] Y. Rui, T. S. Huang, M. Ortega, and S. Mahrota, "Relevance feedback: A Powerful Tool for Interactive Content-based Image Retrieval," *IEEE Trans. on Circuits and Video Technology*, Vol. 8, No. 5, pp. 644-655, 1998.

[2] G. Aggarwal, T. V. Ashwin, and G. Sugata, "An Image Retrieval System with Automatic Query Modification," *IEEE Trans. on Multimedia*, Vol. 4, No. 2, pp. 201-214, 2002.

[3] A. Kushki, P. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos, "Query Feedback for Interactive Image Retrieval," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 14, No. 5, pp. 644-655, 2004.

[4] Y. Rui, T. S. Huang, and S. Mahrota, "Content-based Image Retrieval with Relevance Feedback in MARS," *Proc. of Int. Conf. on Image Processing*, pp. 815-818, 1997.

[5] G. Giacinto, F. Roli, and G. Fumera, "Comparison and Combination of Adaptive Query Shifting and Feature Relevance Learning for Content-based Image Retrieval," *Proc. of 11th Int. Conf. on Image Analysis and Processing*, pp. 422-427, 2001.

[6] P. Muneesawang and L. Guan, "An Interactive Approach for CBIR Using a Network of Radial Basis Functions," *IEEE Trans. on Multimedia*, Vol. 6, No. 5, pp. 703-716, 2004.

[7] K. Porkaew, S. Mehrota, and M. Ortega, "Query Reformulation for Content-Based Multimedia Retrieval in MARS," *Proc. of Int. Conf. on Multimedia Computing and Systems*, Vol. 2, pp. 747-751, 1999.

[8] D. H. Widyantoro and J. Yen, "Relevant Data Expansion for Learning Concept Drift from Sparsely Labeled Data," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 17, No. 3, pp. 401-412, 2005.

[9] X. He, O. King, W. Y. Ma, M. J. Li, and H. J. Zhang, "Learning a Semantic Space From User's Relevance Feedback for Image Retrieval," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 13, No. 1, pp. 39-48, 2003.

[10] G. Giacinto and F. Roli, "Instance-Based Relevance Feedback for Image Retrieval," *Proc. of Advances in Neural Information Processing Systems*, Vol. 17, pp. 489-496, 2005.