CAPTURING SEMANTIC RELATIONSHIP AMONG IMAGES IN CLUSTERS FOR EFFICIENT CONTENT-BASED IMAGE RETRIEVAL

Robert A. Davis, Zhongmiao Xiao, and Xiaojun Qi

davisbo@carleton.edu
Computer Science Department, Carleton College, Northfield, MN 55057-4000
zhongmiao.xiao@aggiemail.usu.edu and Xiaojun.Qi@usu.edu
Computer Science Department, Utah State University, Logan, UT 84322-4205

ABSTRACT

This paper presents an efficient content-based image retrieval system that captures users' semantic concepts in clusters. These semantically homogeneous clusters aid in the retrieval system to accurately measure the semantic similarity among images and therefore reduce the semantic gap. They also aid in the retrieval system to find matched images in a few candidate clusters and therefore reduce the search space. The extensive experiments demonstrate that the proposed retrieval system outperforms the peer systems to quickly retrieve the desired images in a few iterations.

Index Terms— Affinity relations, content-based image retrieval, semantic clustering, semantic similarity

1. INTRODUCTION

With the increasing amount of digital storage space and the increasing popularity of image hosting websites, content-based image retrieval (CBIR) has become an increasingly important information retrieval technique. Recent research effort mainly aims to address two major challenges: 1) reduce the semantic gap between the high-level concepts and the low-level features to make retrieval effective; and 2) reduce the search space to make retrieval efficient.

Relevance feedback (RF) techniques are effective in reducing the semantic gap and improving the retrieval performance. RF is an interactive process in which the user labels correctly retrieved images as relevant to the query. This feedback reveals the semantic relations among images since relevant images share similar semantics. It is therefore used to refine the query in a retrieval session. Recently, RF techniques are further expanded to derive the semantic relationships among images by studying the accumulated feedback collected from multiple users in different retrieval sessions. Researchers propose to use the semantic space [1], knowledge memory model [2], retrieval log [3], virtual feature matrix [4], and affinity matrix [5] to store the search history. Correspondingly, they propose to employ the dot product correlation [1], memory learning [2], statistical correlation [3], cross session learning [4], and manifold techniques [5] on the stored historical feedback to estimate the semantic relevance between images. These learning techniques reduce the semantic gap and achieve impressive retrieval results. However, they do not reduce the search space. Clustering techniques are therefore proposed to estimate the semantic categories of an image database. Researchers apply the semantic grouping [6], cluster affinity search [7], and dynamic semantic clustering techniques [8] on accumulated RF to extract semantically homogeneous clusters of images. These clustering techniques effectively reduce the search space. However, the complexity of constructing semantic clusters is high.

In this paper, we propose a novel image clustering approach that constructs clusters by capturing users' semantic concepts and deriving the semantic relationships among images. We use these semantically homogenous clusters to reduce the semantic gap and the search space to facilitate the image retrieval task. Our system offers the following advantages: 1) The learning mechanism captures users' query concepts in clusters; 2) The clusters are created and updated based on the users' labeled retrieved images; 3) The user's RF dynamically updates the semantic similarity measure; 4) Search space reduction significantly reduces the size of the candidate image pool for a query. The remainder of the paper is organized as follows: Section 2 describes the general methodology of the proposed system. Section 3 presents the experimental results. Section 4 concludes the paper with a discussion of future work.

2. THE PROPOSED SYSTEM

The proposed system consists of four components: feature extraction, semantic concepts capturing, semantic clustering, and image retrieval. We explain each component in the following subsections.

2.1. Feature Extraction

Each image is represented by 100 low-level features. These global features include the 64-bin HSV color

histogram, 9 color statistics (e.g., the first three moments for each of the HSV color space), 18-bin edge direction histogram, and 9 texture statistics calculated from the entropy of each of nine subbands of a 3-level wavelet transformed grayscale image. These global features are easy to compute and effective in image retrieval. However, they are limited in describing objects in an image. As a result, we also divide each image into five regions as shown in Fig. 1. This block-based region division scheme has been proven to achieve comparable image annotation results as the complicated image segmentation scheme. The same 36 features used in the global feature extraction are calculated for each of the five regions. We finally normalize the regional features so each value indicates the likelihood of a feature observed from an image and the sum of the likelihood of all regional features is 1.

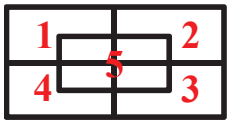

Fig. 1: Region division scheme

2.2. Semantic Concepts Capturing

In our system, the offline training is implemented to record the history of user retrieval patterns and retrieval frequencies on the image database. User retrieval patterns denote the affinity among images retrieved by user queries, while retrieval frequencies denote how often a query was submitted by users. The affinity indicates the co-occurrence relationship among images. That is, the jointly, positively labeled images in a search session likely contain similar semantic content. The higher the number of sessions in which the affinity exists, the higher the semantic similarity among the affinity-related images may be expected. The randomly chosen training images are then used to record user concepts and construct the semantic clusters offline.

For each training image, the initial retrieval starts with returning the 40 closest images by computing Euclidean distance on the normalized low-level features. The user marks the retrieved images that are similar to the query image as relevant images. That is, all positively labeled images are believed by the user to be semantically related to query. We then build a matrix (*pMatrix*) to store these user retrieval patterns (semantic concepts). The width of this matrix is the number of database images, and the length is the number of unique queries submitted so far. Initially each row is all 0's. Based on the user's RF, all relevant (positive) images have their respective slots incremented by 1 and all irrelevant (negative) images have their respective slots decremented by 1. We also use a vector (*fVec*) to store retrieval frequencies, which keep track of the times each image is returned as positive or is submitted as a query. The length of this vector is the number of database images.

For all future iterations of training on the query, we train an RBF-based SVM on the accumulated user responses and return the 40 closest images that have not been returned. We then update *pMatrix* and *fVec* using the same strategy mentioned in the initial retrieval process.

For a query image submitted multiple times, its current retrieval patterns merge with its prior retrieval patterns by adding their respective rows. Therefore, each row in *pMatrix* represents retrieval patterns for a distinct query.

2.3. Semantic Clustering

The semantic clustering process starts after capturing users' semantic concepts. It is different from most clustering techniques in the sense that the number of possible semantic categories (clusters) in the image database is automatically estimated. It consists of three phases: initial semantic clustering, semantic cluster merging, and addition of non-clustered images. The first phase directly translates *pMatrix* to initial clusters. The second phase merges the initial clusters. The third phase places all images not yet clustered in an already created cluster or in their own new cluster. In the following, we explain these three phases in detail.

2.3.1. Initial Semantic Clustering

We create one cluster for each of the rows in *pMatrix*. Each cluster comprises of all images with a positive value in the corresponding row of *pMatrix*. That is, all images in a cluster are labeled as relevant to the query and are likely to contain similar semantics. It should be noted that the same images could be labeled positive in different queries. That is, an image may belong to multiple clusters represented by different query images. This allows the clusters to be merged in the next phase.

2.3.2. Semantic Cluster Merging

We iteratively merge initial clusters using an adaptive threshold. If the number of images coexisting in clusters A and B is larger than a quarter of the number of images in the smaller cluster of A and B, we will merge these two clusters by combining the images in both clusters. The algorithmic view of this merging process is summarized in Fig. 2.

1. Let $\boldsymbol{C}=\{C_1, C_2, \ldots, C_n\}$ denote n clusters created in the initial semantic clustering phase.
2. Let ***NewC*** denote clusters after the merging process. Initially, set ***NewC*** = $\{NewC_1\}$ where $NewC_1 = C_1$.
3. For each C_i (i=2, 3, …, n)
   - 3.1 Set num = |***NewC***|, where | | represents the number of clusters in ***NewC***.
   - 3.2 Set *Flag*, a vector of num elements, to all 1's.
   - 3.3 For each j (j=1, …, num)
     If $NewC_j$ and C_i can be merged, enlarge C_i by adding all images in $NewC_j$ and set the jth element of *Flag* as 0's.
   - 3.4 If $Flag_j$ is 0, remove $NewC_j$ from ***NewC***.
   - 3.5 Add C_i to ***NewC***.

Fig. 2: Algorithmic view of the cluster merging

After merging overlapped clusters, we also compress *pMatrix* by merging rows corresponding to the merged clusters. It should be noted that the images that are not

returned or not positively labeled in any query session are not placed to any merged cluster. Therefore, these non-clustered images are assigned to the appropriate clusters, as summarized in the next phase.

2.3.3. Addition of Non-clustered Images

To reduce the computational cost, we use the 36 features of region 5 of each image to calculate cluster related distances since region 5 usually has a higher probability to include an important object in an image. Specifically, the cluster center is computed as the average of 36 features of all images in the cluster. The threshold (*addThresh*) for adding a non-clustered image to a cluster is computed as one half of the average of all paired inter-class distances. For each non-clustered image, we compute its distance to each of the cluster centers. We assign each non-clustered image to its nearest non-negatively marked cluster. That is, we don't add the non-clustered image to a cluster if it was marked as being negatively related to any of the queries related to that cluster. We also include a maximum distance threshold (*addThresh*) that an image can be from a cluster before allowing it to join a cluster. If a non-clustered image reaches this point, it creates a new cluster comprised of only the image itself.

2.4. Image Retrieval

Image retrieval consists of three steps: search space reduction, search space ranking, and refinement. The first step finds the candidate clusters to reduce the search space. The second step ranks the images in candidate clusters. The third step updates *pMatrix*, the semantic distance measure, and the search space.

2.4.1. Search Space Reduction

We first apply the "bucket" concept from the *kd*-tree to quickly find the *k* (initially set to be 15) nearest neighbors in logarithmic time. We then find all candidate clusters that contain at least one of the 15 nearest neighbors. If applicable, we also eliminate the candidate clusters that have more than 2 images being negatively related to the query. If the number of images in the remaining candidate clusters is less than the number of images we wish to return per testing iteration, we increase the number of nearest neighbors we search by 2 and repeat the above steps until a sufficient number of candidate images are collected.

2.4.2. Search Space Ranking

Search space reduction significantly reduces the size of the candidate image pool for a query. Search space ranking aims to find the most matched images from the candidate pool based on image features and affinity learned from the training process. To this end, we use *pMatrix* and *fVec* to compute the affinity relation $A_{q,i}$ between query image q and each database image i by:

$$A_{q,i} = \sum_{r \in rows} pMatrix(r,q) \times pMatrix(r,i) \times fVec(r) \quad (1)$$

where *rows* is a set containing all rows in *pMatrix* whose q^{th} and i^{th} columns having positive values. We then normalize $A_{q,i}$ as $NA_{q,i}$ by dividing $A_{q,i}$ by the sum of affinity relations between q and all database images. Finally, we compute the semantic similarity $SS_{q,c}$ between query image q and each candidate image c by equations (2) through (4):

$$W_1(q,c) = NA_{q,c}(1 - |f_c(o_1) - f_q(o_1)| / f_q(o_1)) \quad (2)$$

$$W_{t+1}(q,c) = W_t(q,c)(1 - |f_c(o_{t+1}) - f_q(o_{t+1})| / f_q(o_{t+1})) \quad (3)$$

$$SS_{q,c} = \sum_{t=1}^{T} W_t(q,c) \quad (4)$$

where $f_c(o_t)$ is the t^{th} feature of image c, $f_q(o_t)$ is the t^{th} feature of query q, and T is the length of the image features. It should be noted that some values of $SS_{q,c}$ may be 0's since their affinity relations with the query may not have been explored during the training process. In this case, the Euclidean distance is applied on the image features to compute the similarity scores to break the tie. The images with larger semantic scores indicate a higher similarity with the query and the images with lower Euclidean distances indicate a higher similarity with the query. Top images are returned based on the positive semantic scores and the Euclidean distance.

2.4.3. Refinement

The refinement step employs the user's RF on the retrieved images to update *pMatrix* and *fVec* as described in section 2.2. It also updates the affinity relations between query and all database images using a maximum of 5 positive images not returned in the previous iterations. Specifically, the normalized affinity relations between each of these positive images and all database images are computed. These relations are then used to update the normalized affinity relation between query and all database images as follows:

$$NA_{q,i} = \begin{cases} NA_{p_j,i} & if \ NA_{q,i} = 0 \\ NA_{q,i} + 0.2 \times NA_{p_j,i} & otherwise \end{cases} \quad (5)$$

where $NA_{p_j,i}$ denotes the normalized affinity relation between the positively labeled image P_j (j=1 to 5) and image i, and $NA_{q,i}$ denotes the affinity relation between query q and image i. A maximum of 5 positive images is chosen to keep this update process fast. The candidate image pool is finally re-calculated by considering the new negative images returned in the current iteration.

3. EXPERIMENTAL RESULTS

We test our CBIR system on three data sets: 2000-Flickr DB, 6000-COREL DB, and the combined 2000-Flickr and 6000-COREL DB. Flickr and COREL DBs contain 20 and 60 categories with 100 images per category, respectively. To test the online retrieval performance, we perform 5

iterations per query and return top 25 images per iteration. The retrieval precision is computed as the ratio of the relevant images to the total returned images.

In our system, we assign a weight of 44% to region 5 and a weight of 14% to the other 4 regions. We also assign a weight of 40% to the color features and a weight of 30% to the edge and texture features for each region. These weights are empirically selected to be optimal for the task.

We design the first experiment to show the retrieval performance using 5% randomly chosen images to capture users' semantic concepts and build initial clusters. Specifically, we test with 6 and 8 training iterations, with both using 40 returned images. Fig. 3 shows our results on each DB with the two varying number of training iterations. Since no significant gains in precision are made after the 3rd iteration, we only show the average retrieval precision for the first 3 iterations. It clearly demonstrates the retrieval performance is better when the dataset is smaller since it is easier to learn semantic relations in a small database. The retrieval performance also significantly improves when the number of training iterations increases. Therefore, we use 8 iterations for each training image to build initial clusters.

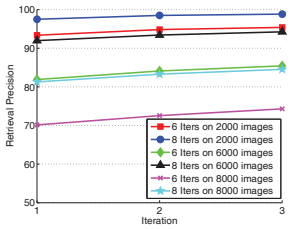

Fig. 3: Results using 6 and 8 training iterations on 3 DBs

We compare our system with memory-based [2], log-based [3], virtual-feature-based [4], manifold-based [5], and collaborative learning-based [8] systems on two larger DBs. Fig. 4 shows the average retrieval precision of 6 systems on the 6000-COREL DB and the combined DB. All these systems use 8 iterations and 40 returned images per iteration for each training image to build their perspective learning base. However, our system uses 5% of database images in training and the other systems use 10% of database images in training. Our system outperforms the peers on both DBs. Comparing to the second best, collaborative learning-based system, our system makes 2.26% and 4.75% improvement on 6000 and 8000 images, respectively. It should be mentioned that this improvement is achieved by using a half of the training images to build the initial learning base. This minimum learning mechanism is preferred since little user's involvement is required in the training process. Our retrieval time is also faster than its peers due to the reduction of the search space and the simple clustering and update process. Using 10% of database images as the training images, our system further makes a 2.45% and 4.38% improvement on 6000 and 8000 images, respectively.

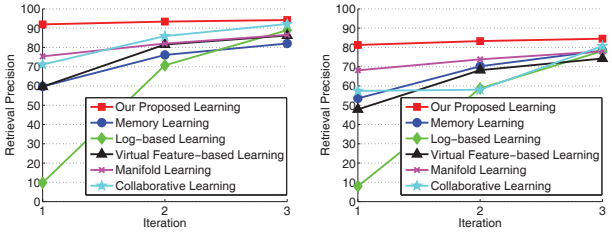

Fig. 4: Comparison of 6 systems on 6000 images (left) and 8000 images (right)

4. CONCLUSIONS AND FUTURE WORK

We propose a novel image clustering approach that constructs clusters by capturing users' semantic concepts. Major contributions are: 1) Applying the learning mechanism to capture users' semantic concepts in clusters. 2) Applying local and dynamic clustering to create and update clusters based on the users' labeled retrieved images. 3) Updating the semantic similarity measure using the user's RF. 4) Reducing the search space to find the candidate image pool for a query. Experimental results show that our system achieves better performance with a larger training iteration number and a larger training set. It outperforms peer systems even using a half of the training images.

In the future, we will improve the system to make it work well when a small amount of erroneous feedback is involved. We will also improve our clustering technique to group semantic related images faster and more reliable.

.

5. REFERENCES

[1] X. He, O. King, W. Y. Ma, M. Li, and H. Zhang, "Learning a Semantic Space from User's Relevance Feedback for Image Retrieval," *IEEE Trans. CSVT*, Vol. 13, pp. 39-48, 2003.

[2] J. Han, K. N. Ngan, M. Li, and H. J. Zhang, "A Memory Learning Framework for Effective Image Retrieval," *IEEE Trans. Image Processing*, Vol. 14, No. 4, pp. 511-524, 2005.

[3] S. Hoi, M. Lyu, and R. Jin, "A Unified Log-Based Relevance Feedback Scheme for Image Retrieval," *IEEE Trans. KDE*, Vol. 18, No. 4, pp. 509-524, 2006.

[4] P. Y. Yin, B. Bhanu, K. Chang, and A. Dong, "Long-term Cross-Session Relevance Feedback Using Virtual Features," *IEEE Trans. KDE*, Vol. 20, No. 3, pp. 352-368, 2008.

[5] R. Chang and X. Qi, "Semantic Clusters Based Manifold Ranking for Image Retrieval," *Proc. of IEEE Int. Conf. on Image Processing*, pp. 2473-2476, 2011.

[6] T. Yoshizawa and H. Schweitzer, "Long-Term Learning of Semantic Grouping from Relevance Feedback," *Proc. of the 6th ACM SIGMM Int. Workshop on Multimedia Information Retrieval*, pp. 165-172, 2004.

[7] Y. Liu, X. Chen, C. Zhang, and A. Sprague, "Semantic Clustering for Region-Based Image Retrieval," *J. of Visual Communication and Image Representation*, Vol. 20, No. 2, pp. 157-166, 2009.

[8] X. Qi, S. Barrett, and R. Chang, "A Noise Resilient Collaborative Learning Approach to Content-Based Image Retrieval," *Int. J. of Intelligent Systems*, pp. 1-23, 2011.