AN INTERACTIVE REGION-BASED IMAGE CLUSTERING AND RETRIEVAL PLATFORM

Ying Liu, Xin Chen, Chengcui Zhang, *Alan Sprague*

Department of Computer and Information Sciences,
University of Alabama at Birmingham, Birmingham, Alabama 35294, U.S.A.

ABSTRACT

Content-based Image retrieval has become an important part of information retrieval technology. Images can be viewed as high dimensional data and are usually represented by their low-level features. How to effectively find the semantic meanings of images is a central challenge in the area. In this paper, we propose an interactive platform for region-based image clustering and retrieval. A Genetic Algorithm is used to perform the initial clustering. In order to further refine the clustering results, we adopt the maximum flow/minimum cut theorem from graph theory to do outlier/outlier group detection. Outlier detection can help identify misclustered image segments and is used to improve the quality of clusters in this paper. In the interactive retrieval phase, user feedback is used to dynamically locate candidate images from clusters and outliers/outlier groups. Through Relevance Feedback, more information is gathered and fed to the learning algorithm – One-class SVM. Experiments show the effectiveness of the proposed platform.

1. INTRODUCTION

Most of the existing content-based image retrieval (CBIR) approaches consider each image as a whole, which is represented by a vector of *N* dimensional image features. However, a single image can include regions/objects with completely different semantic meanings. A user's interest is often just one part of the image i.e. a region in the image that has an obvious semantic meaning. Therefore, rather than viewing each image as a whole, it is more reasonable to view it as a set of semantic regions. In this context, the goal of image retrieval is to find the semantic region(s) of user's interest. In our preliminary work [3], we developed a region-based CBIR system with a clustering component. Since an image is typically split into 7-8 segments, the search space of a region-based system will be 7-8 times as large as otherwise it could be. The enlarged search space makes retrieval efficiency an issue. To improve the efficiency, we impose clustering of all the segments as a preprocessing step. The new problem that arises is the quality of the clustering. We found that in some clusters, many segments actually do not belong to this cluster. Such segments are outliers with respect to the other segments in the cluster. To find such misclustered segments and to repair the clustering results, we propose an outlier detection and cluster repairing algorithm in this paper. Our experiments show that the proposed outlier detection algorithm can improve the quality of the clustering by identifying noisy data and thus can improve the accuracy of retrieval.

With clustering and outlier detection, image segments are grouped into clusters and outliers/outlier groups. In the retrieval phase, in order to correctly locate the candidate image segments, user-provided information needs to be incorporated. Later, in the learning and retrieval phase, user interaction provides more information to facilitate the retrieval process through Relevance Feedback.

Since each image is composed of several regions and each region can be taken as an instance, region-based CBIR is then transformed into a Multiple Instance Learning (MIL) problem [4]. In MIL, each image is viewed as a bag of semantic regions (instances). The labels of individual instances in the training data are not available, instead the bags are labeled. When applied to region-based CBIR, this corresponds to the scenario that the user gives feedback on the whole image (bag) although he/she may be only interested in a specific region (instance) of that image. The goal of MIL is to obtain a hypothesis from the training examples that generates labels for unseen bags (images) based on the user's interest on a specific region. In our previous work [6], we proposed a framework combining Relevance Feedback and One-class Support Vector Machine (SVM) to solve this MIL mapping problem. In this paper, with an intention to solve the efficiency problem, we further extend our previous framework by adding outlier detection and cluster repair.

In Section 2, we introduce the first phase i.e. clustering and outlier detection. The retrieval phase is presented in Section 3. Section 4 is our system overview. Section 5 shows our experimental results.

2. IMAGE SEGMENT CLUSTERING

The purpose of clustering is to reduce the search space for retrieval. In the proposed platform, the clustering phase can be divided into two parts: initial clustering and cluster refinement.

2.1. Initial Clustering by Genetic Algorithm

Genetic Algorithm (GA) [5] is a well-known global optimum finder. We adapt it to suit our needs of clustering image regions/segments. Figure 1 shows an overview of the GA based clustering algorithm.

The first step of GA-based clustering is to generate the possible solutions to a clustering problem by encoding cluster centroids into chromosomes. Each segment is given an integer ID. Genes in each chromosome are the cluster centroids represented by their IDs. Note that the term "centroid" here represents an actual image region but not the "center" i.e. a "virtual point" of a cluster. The number of genes in each chromosome corresponds to the number of clusters to be generated.

1-4244-0367-7/06/$20.00 ©2006 IEEE

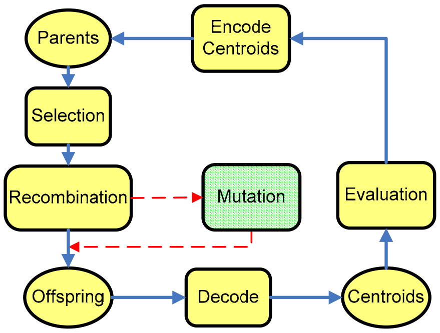


Figure 1. GA-based Initial Clustering

In the initialization step, a pool of chromosomes is randomly generated. Each chromosome is evaluated by an Objective Function [3] and therefore is associated with a fitness value. After evaluation, parent chromosomes are selected, recombined and mutated. Chromosomes with higher fitness values are more likely to be selected. In other words, such chromosomes will have a better chance to produce their offspring generations. Mutation is performed at a very low frequency and is controlled by a small randomly generated number. It is denoted by dashed arrows in Figure 1. A new offspring generation of chromosomes is generated after all these operations. This whole process goes through several iterations until a certain criterion is met. The chromosome with the highest fitness value in the last generation is picked as the final clustering result. For more details, please refer to [3].

2.2. Refine Clustering Results by Outlier Detection

We further refine the clustering results generated by the above mentioned clustering method by finding outliers and outlier groups inside the clusters. A novel outlier detection and evaluation algorithm is proposed, with its origin from the Network Flow of Graph theory [8]. The basic idea of the algorithm is as follows.

Each cluster is associated with a set *C* of labeled data points ("vertices"). We wish to determine if *C* is a good cluster, or instead contains points that are only weakly related to the rest. Those points are so-called outliers. We propose to use the network flow method to analyze clusters. Consider the set *C* of points as a network/graph: for each pair of points (*s, s'*) in *C*, the edge *ss'* has a capacity equal to $(100/(1+l(ss')))^n$ where *l*(*ss'*) is the length of the edge *ss'*. We scale the capacity to [0, 100] by multiplying by 100. The n^{th} power is used to make low capacities lower and high capacities higher. *n* is an integer as long as the maximum flow is much lower than the high capacities. Specifically, we use n=4 in our experiments. Thus if *s* is close to *s'*, *ss'* has a high capacity, and vice versa. If *s* and *s'* is an outlier group, maximum flow would probably saturate low-capacity edges around *s* and *s'*. This fact can be used to identify the *s* and *s'* outlier group. In our case, upscaling capacities helps us to find the weakest edges faster.

Our intuition is based on the following ideas. Let *s* be an outlier and *t* be the point in *C* that is the farthest from *s*. Suppose that *s*, as an outlier, is also far from all other points in *C*. Then, each edge *ss'* has a small capacity, and ({*s*}, *C*-{*s*}) represents a cut of small capacity. Then the network flow algorithm will tell that the maximum flow from *s* to *t* is also small, and quite likely the minimum cut will be ({*s*}, *C*-{*s*}). There is another possibility that *t* is also an outlier and will be singled out by minimum cut.

In the case of outlier groups, let *s* and another couple of points *s'* and *s''* form a group of outlier points, and *t* be the point in *C* that is the farthest from *s*. Since *s*, *s'* and *s''* are very similar to each other, the capacities are very high or even full when they are identical. When the total flow from *s* to *t* is less than the capacities between *ss'* and *ss''*, the minimum cut is ({*s,s',s''*}, *C*-{s,*s',s''*}). The algorithm is applied to each cluster separately. The basic steps of this algorithm are as follows:

1. Find the *k* nearest neighbors of each data point (image segment).
2. Set up a graph for the data points in the cluster.
3. Select a random vertex as the starting source *s*.
4. Find the sink *t* that is the farthest from *s*.
5. Use the Maximum-Flow/Minimum-Cut algorithm [8] to find the flow from the source to the sink, get the cut separating *s* and *t*, and use the smaller group as the candidate outlier or outlier group.
6. Remove the candidate outlier or the candidate outlier group from the graph.
7. Choose the next source and go back to Step 4 until the stop criterion is satisfied.
8. Coarsen the graph and run the maximum-flow/minimum-cut algorithm again on the coarsened graph.
9. Select outliers from the pool of candidate outliers.

As for the stop criterion in Step 6, instead of using a maximum number for outlier groups [7], we use an auto-stop criterion in this paper. The maximum flow is translated into a pseudo distance *d'* as follows:

$$d' = \frac{100}{\sqrt[n]{\max flow/\#cross\_edge}} - 1$$

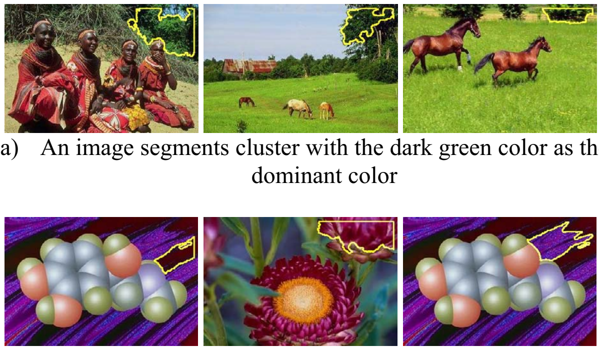
where *#cross_edge* is the number of edges crossing the source side (main part of the data) and the sink side (outliers). Thus, *max_flow/#cross_edge* indicates the average capacity per edge, and the n^{th} root of the average capacity converts the average capacity back to its original scale (*n* is a scaling factor and equals 4 in this study.) If *d'* is smaller than the average distance within the data on the source side, the process stops.

For image data, each image segment is represented by an *N* dimensional feature vector. In this study, we adopted the automatic image segmentation method Blobworld [1] to segment each image into a set of regions. Eight features, (three texture features, three color features, and two shape features [1]) are extracted for each blob, i.e. semantic region. Hence, each image region/segment is represented by an 8-dimension feature vector. We then use the software ANN 0.1 [1] to find the *k* nearest neighbors of each data point. Specifically, Euclidean norm is applied to measure the distance. As mentioned earlier, Maximum-Flow/Minimum-Cut outlier detection algorithm is a graph-based method. When we set up the graph for image segments, vertices in the graph represent the actual image segments, while edges indicate the distances between segments based on Euclidean norm. It should be noted here that although different distance measures can be used in building the graph, we use the Euclidean distance for simplicity and our experimental results demonstrate the effectiveness of the proposed framework.

From the initial clustering results by using the adapted Genetic Algorithm (see Section 2.1), we observed that some clusters are

[1] ANN 0.1 developed by David M. Mount, Dept. of Computer Science and Institute for Advanced Computer Studies, University of Maryland.

dominated by one single feature such as the shapes of the image segments or the color characteristics of the segments. The outliers in each cluster are often defined as those segments with the opposite properties to the dominating features. Figure 2(a) shows an example of a cluster whose dominant color is dark green and Figure 2(b) shows some "purple" outliers detected by our algorithm.



(a) An image segments cluster with the dark green color as the dominant color

(b) Sample outliers

Figure 2. Outlier detection example.

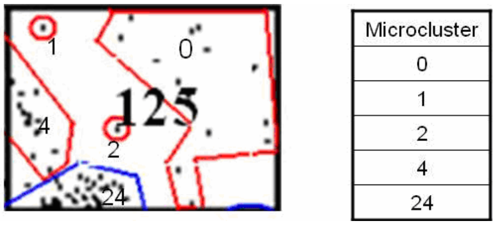
2.3. Locate the Candidate Image Segments for a Given Query Segment

Clustering of image segments, which is just the first step towards interactive retrieval of image regions, will be used to reduce the search space in the later phase of retrieval. In the proposed platform, the user submits a query image and specifies explicitly which region of the query image he/she is interested in. Given this information, the search engine will locate the candidate image segments from the whole data set.

By using the network flow, the outlier segments, as defined in Section 2.2, will be separated from the cluster to which they belong. The cleaned clusters, together with the outliers/outlier groups, are called microclusters. We could simply reduce the search space to the one microcluster that the query region falls into. However, this might not be an effective solution for two reasons. First, the microcluster might just have a small number of segments in it if it is an outlier or an outlier group. Second, even if the microcluster is one of the regular clusters, it may happen that the query region, though it belongs to that cluster, is more similar to some regions in another cluster than some other regions in the same cluster. Therefore, we need to consider not only the microcluster to which the query region belongs, but also several other microclusters that are close to it.

For high dimensional data, the relationship among points can be very complex. For this high dimensional image segment data set, we use the concept of 'bucket' to check the relationship of a microcluster to its neighbors. 'Bucket' is a concept from kd-tree. In Figure 3, a set of two-dimensional data points is used to illustrate the idea of kd-tree bucket. In our case, the whole data set has 82,552 points/segments. By indexing the original data set into the kd-tree with bucket size 500, there are 450 buckets, and 364 of them include points from 2 or more regular clusters in them.

Instead of just checking the microcluster of the query segment, we also check the microclusters within the bucket where the query region is located. A bucket in kd-tree is like a microscope - the bigger the bucket size, the more microclusters the bucket will include, and the more segments need to be checked in retrieval. We do not limit the number of clusters to check. The bucket helps us find the nearest microclusters related to the query segment.



Microcluster
0
1
2
4
24

Figure 3. No.125 bucket with 5 microclusters

According to user-specified query region, we pull out all the microclusters that fall into the same bucket as the query image segment. All the images that have at least one segment falling into these chosen microclusters are pulled out and grouped together as the reduced image database for the next phase -- learning and actual retrieval.

3. INTERACTIVE LEARNING AND RETRIEVAL

Given a query image, in the initial query, the user needs to identify a semantic region of his/her interest. Since no training data is available at this point, we simply compute the Euclidean distances between the query semantic region and all the other semantic regions in the reduced image database. The smaller the distance, the more likely this semantic region is similar to the query region. The distance between an image and the query segment is the smallest distance from the query segment to the segments contained in the image. We compute such distances for all images in the reduced database and return the top 30 image to the user. The training sample set is then constructed according to the user's feedback. If an image is labeled positive, its semantic region that is the least distant from the query region is labeled positive. All the other regions of this image are then labeled negative. If an image is identified as negative, then all regions in this image are labeled negative. With the training sample set, One-class Support Vector Machine is used to learn from the user's feedback and retrieve images from the reduce image database. The idea of One-class SVM is to model the positive image regions as a hyper-sphere. Positive image regions are inside and negative ones are outside. The goal is to make this hyper-sphere as small as possible while keeping it as "pure" as possible.

One-class SVM learns from training set and returns the refined results to the user who will provide feedback. This whole process goes through several iterations. Our previous work shows its effectiveness [3].

4. SYSTEM OVERVIEW

Figure 4 shows the architecture of our system. Images are segmented into semantic regions, with each represented by an 8-feature vector. After the initial clustering by Genetic Algorithm, network flow is used to refine the clustering result by separating outliers/outlier groups from clusters. The user interacts at two places. The user-specified query region is first used to locate candidate image segments i.e. microclusters in one bucket. After the initial query, the user gives feedback to the retrieved images and this feedback is returned to the system. Our One-Class SVM based algorithm learns from this feedback and starts another round of retrieval.

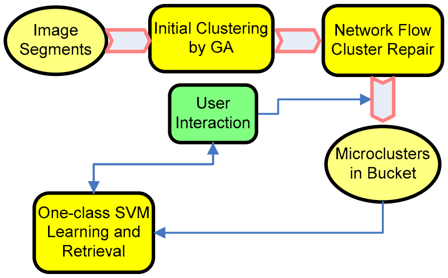


Figure 4. System architecture

5. EXPERIMENTS

The experiment is conducted on a Corel image database consisting of 9,800 images from 98 categories. After segmentation by Blobworld [1], there are in total 82,552 image segments. In the initial clustering, the number of clusters is k=100. By indexing the data using kd-tree, there are 450 buckets. In our experiments, twenty images are randomly chosen from 15 categories as the query images.

In order to examine the quality of clustering, we compare our system with that of the full sequential search as well as the Genetic clustering. We also compare the performance of our system with another algorithm -- a general feature re-weighting relevance feedback algorithm [2]. Five rounds of relevance feedback are performed for each query image - Initial (no feedback), First, Second, Third, and Fourth. The accuracy rates with different scopes, i.e. the percentage of positive images within the top 6, 12, 18, 24 and 30 retrieved images, are calculated. Figure 5 shows the results after the Fourth Query. “RF” is the general re-weighting Relevance Feedback algorithm without a region-based learning component. “GACluster” is the system that incorporates Genetic Algorithm based clustering and SVM-based retrieval. “Network Flow” is “GACluster” plus the proposed clustering refinement module. “SVM” refers to the same learning and retrieval mechanism without clustering. It can be seen from Figure 5 that the performance of the proposed system is better than that of “SVM” and “GACluster”. By repairing the clusters, we improve the retrieval accuracy by about 4%. To further improve the accuracy, we can let the outlier detection run more iterations. Another thing worth pointing out is that when incorporated with the One-class SVM learning algorithm, Relevance Feedback performs much better than it would otherwise.

6. CONCLUSION

The paper proposes an interactive platform for region-based image clustering and retrieval. With the ability to find global optimum, Genetic Algorithm is chosen to perform the initial clustering. Seeing some space for improvement in clustering, we explore Network Flow for cluster refinement. In doing so, user interaction needs to be incorporated so as to locate the reduced search space for retrieval phase. One-class SVM is integrated with Relevance Feedback technique for region-based learning and retrieval. The design of the platform targets at two main problems in the region-based CBIR systems – the efficiency of retrieval and the mapping from low level features to high level semantic concepts (object-of-interest). Our experimental results show the effectiveness of the platform.

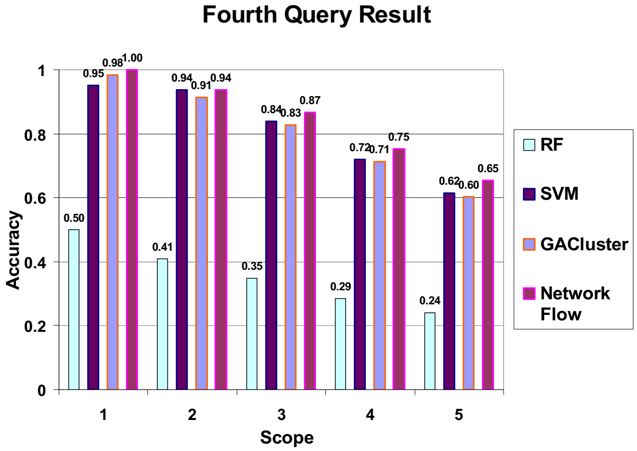


Figure 5. The comparison of retrieval accuracy after the 4th query.

7. ACKNOWLEDGMENT

The work of Chengcui Zhang was supported in part by SBE-0245090, the UAB ADVANCE program of the Office for the Advancement of Women in Science and Engineering, and the UAB Faculty Development Award.

8. REFERENCES

[1] C. Carson, et al., “Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying,” *IEEE Trans. on Pattern Analysis and Machine Intelligence,* v(24):8, 2002.

[2] Y. Rui, T. S. Huang, and S. Mehrotra, “Content-based Image Retrieval with Relevance Feedback in MARS,” *Proc. of Intl. Conf. on Image* Processing, pp. 815-818, 1997.

[3] C. Zhang and X. Chen, “Region-based Image Clustering and Retrieval Using Multiple Instance Learning,” *Proc. of ACM SIGIR International Conference on Image and Video Retrieval,* pp. 194-204, July 20-22, 2005, Singapore.

[4] O. Maron and T. Lozano-Perez, “A Framework for Multiple Instance Learning,” *Advances in Natural Information Processing System 10,* Cambridge, MA, MIT Press, 1998.

[5] J. H. Holland, “Adaptation in Natural and Artificial Systems,” University of Michigan Press, 1975.

[6] C. Zhang, X. Chen, M. Chen, S.-C. Chen, and M.-L. Shyu, “A Multiple Instance Learning Approach for Content-based Image Retrieval Using One-class Support Vector Machine,” *Proc. Of IEEE International Conference on Multimedia & Expo (ICME),* pp. 1142-1145, July 6-8, 2005, Amsterdam, the Netherlands.

[7] Y. Liu and A. P. Sprague, “Outlier Detection and Evaluation by Network Flow,” *Proc. of International Conference on Machine Learning and Applications (ICMLA)*, pp.436-442, Louisville, KY, USA, 2004.

[8] S. Even. Graph Algorithms. 1979. Computer Science Press.