Semantic Clustering for Region-based Image Retrieval

Ying Liu, Xin Chen, Chengcui Zhang, and Alan Sprague
University of Alabama at Birmingham, Birmingham, AL 35294, USA
{liuyi, chenxin, zhang, sprague}@cis.uab.edu

Abstract

This paper proposes a semantic clustering scheme to reduce search space and semantic gap, two most challenging tasks in content-based image retrieval. By performing clustering before image retrieval, the search space can be reduced to those clusters that are close to the query target. In the proposed method, image sub-regions/segments are grouped into clusters in terms of their semantic meanings in addition to their low level features. Ideally, one cluster approximates one semantic concept or a small set of closely related concepts; hence the "semantic gap" in the retrieval phase is reduced. The experimental results show the effectiveness of the proposed method.

1. Introduction

Content-based Image Retrieval has become an important part of information retrieval technology. One challenge in this area is that the ever-increasing number of images acquired through digital world makes the brute force searching almost impossible. To improve the efficiency, we propose to impose a clustering component in the region-based image retrieval system, which makes it possible to only search the clusters that are close to the query target, instead of the whole search space.

Image segments (regions) can be viewed as high dimensional data and are usually represented by their low-level features. There exists a gap between the high level semantic meanings of images and their low level features (i.e., "semantic gap"). How to effectively find the semantic meanings of images is another key challenge in the area. "Relevance Feedback" [12] is a technique that has been well studied and proved effective in reducing the gap. In our previous work [16], we developed a region-based CBIR system. The query log of this system collects the user feedback, which provides hints to the semantic meaning of image regions. The query log is an affinity matrix in which rows are composed of query regions/segments (query targets) and columns are composed of all the images in the database. Entries in the matrix are accumulated scores acquired through Relevance Feedback. All entries are set to zero at the beginning. In the retrieval process, if the user labels a certain image as "positive" to the query target, the score of the corresponding entry in the matrix will be increased by 1. Otherwise, if the image is labeled "negative", the value will be reduced by 1. Thus, after querying the image database for certain period of time, this matrix is filled up by integers that represent the semantic closeness among images and image regions. In this paper, we utilize this log file and design a semantic clustering scheme, whose purpose is two-fold: to reduce the "semantic gap" and to reduce the search space.

Gong et al. [7] proposed to integrate the captions of images for semantic clustering. The work proposed in [13] requires that the semantics of the image database be pre-defined by domain experts. However, in many cases, neither the captions (texts) nor the semantic categories are readily available. In [14] and [3], it is proposed that semantic clustering is performed using Relevance Feedback. These works are based on the whole image instead of the image regions/segments. The clustering method in [14] is based on a method called CAST [1] while the one in [3] is based on the Association Rule Hyper-graph Partitioning algorithm [8]. While both these clustering schemes are based on existing clustering methods, we propose a new method that constructs clusters based on the semantics of image regions. The number of possible semantic meanings/categories (i.e., number of clusters) in the image database can thus be estimated.

In our proposed method, the initial cluster centers are the query targets (query regions/segments) in the log file. Initially, each cluster is assumed an independent semantic unit. Each image segment is assigned to one or more clusters according to the users' feedbacks on its closeness to the semantic meaning of the cluster represented by the query target. The feedback information is extracted from the affinity matrix in the log file. Since one image segment can have more than one semantic meaning, it is natural to allow it to belong to multiple clusters. For example, a "red flower" can be assigned to both "red" cluster and "flower" cluster.

0-7695-3084-2/07 $25.00 © 2007 IEEE
DOI 10.1109/ISM.Workshops.2007.37

The new problem that arises is the quality of the clustering. There is no prior knowledge to the number of semantic meanings among image segments in the database. Although the query targets in the log file may represent a certain number of semantic meanings, they are definitely not all. It is also hard for an expert to enumerate all the semantic meanings of a database containing a large number of natural images. Another problem is that for some image segments, since they have never been queried and/or retrieved before, no semantic information can be extracted from the log file. However, these segments may either belong to the existing semantic clusters or represent a new semantic meaning. We solve the above two problems by first assigning the "unknown" segments to the closest existing semantic clusters according to its distance to the cluster center. Then, we further divide each cluster into microclusters by an outlier detection method based on our previous study [10]. Segments that are misclustered can be considered as outliers and/or outlier groups of the semantic cluster. From another point of view, these outliers and/or outlier groups are new semantic clusters that emerged from the existing ones. Through outlier detection and cluster repairing, new semantic clusters are generated. We call them new and old microclusters and do not differentiate them as clusters and outliers in the subsequent region-based image retrieval. Our experimental results show that the proposed outlier detection algorithm can improve the quality of the clustering and thus improve the accuracy of the retrieval.

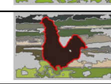
Details of the semantic clustering are illustrated in Section 2. Section 3 briefly introduces the retrieval system. Section 4 shows our experimental results. Section 5 concludes the paper.

2. Semantic clustering

2.1 Affinity matrix extracted from the log file

The users' feedbacks collected over time are recorded in a log file. According to the query log file, we get the user's feedback history for each query segment. The log file is thus represented by an affinity matrix. Its structure is shown in Table 1. I_1, I_2, … represent images and S_1, S_2,… are image segments. In total there are 9800 images in the database. Therefore, there are 9800 columns in the matrix. Each row of the matrix is composed of the users' feedbacks on images given a query segment. There are 1188 queries recorded. Entries of the matrix are positive/negative integers or 0s (no feedback for that image). Positive integers signify that the corresponding image matches the query segment and therefore the positive user feedback. In other words, the user thinks there is at least one segment in that image that is related to the query segment. The negative integers imply that no segment of that image matches the query segment and is therefore given a negative feedback. '0' means we do not have any information on the relevance of this image to the query segment. This happens when that image is not among the top ranked images retrieved for that query region. As we do not want to burden the user by asking him to provide feedback to too many images (i.e., >20 images), those images with lower similarity scores will be ignored in the feedback process. Among the 1188 queries, some of them have the same query target/region. We merge the users' feedback on those duplicate queries and finally obtain 833 unique query targets (Table 1).

Table 1 Affinity Matrix

		I_1	I_2	I_3	…	I_{9800}
S_1		0	0	2	…	-2
S_2		1	1	-1	…	2
S_3		0	0	0	…	0
…	…				…	
S_{833}		0	1	2	…	0

2.2 Initial semantic clustering

Our clustering method is based on the user's query log file. For the 9800 images in the database, there are 82552 segments. Each image segment is represented by an N dimensional feature vector. In this study, we adopted the automatic image segmentation method Blobworld [2] to segment each image into a set of regions. 32 low-level features (27 color, 3 texture, and 2 shape) are extracted for each blob, i.e., each region. Hence, each image region/segment is represented by a 32-dimensional feature vector. For the affinity matrix mentioned in the previous section, we use each query segment as a semantic cluster center. Although these query segments may not accurately represent the "centroid" of a cluster in terms of its low level features, the semantic meaning it conveys can be reasonably regarded as an estimate of the center of that semantic cluster. For those images which are labeled positive, we find out which segment of this image has the shortest Euclidean distance with the query segment, and put this segment (positive segment) in the same cluster of the query segment. The total number of positive segments identified from the log file by this method is 3586, and 9535 for negative segments.

As an initial step of semantic clustering, we first examine the segments with positive labels. For the 833 unique query segments, their positive feedbacks could be overlapped, i.e., the same image segment could be labeled positive in different queries. In other words, an image segment may belong to multiple clusters represented by different query segments. For example, when we search for a white object, the images containing one or more white horse are labeled positive. When we search for white horses, those segments containing white horses will also be labeled positive and therefore shall be assigned to both the "white object" cluster and the "white horse" cluster. If we simply combine the overlapped query results, 506 positive query sets will be obtained. However, from the semantic clustering point of view, semantic meanings are often ambiguous, especially because different user's interpretation can be different in many ways. Therefore, in the following experiment, we allow a segment to belong to different clusters.

In the next step, we try to cluster the negative segments and those segments without labels. For negative segments, we simply do not assign them to the clusters represented by their corresponding query segments. Instead, we assign them to the next nearest semantic centers by computing their Euclidean distances to each cluster center (query segment). For those segments without any labels, we just assign them to the same cluster of their nearest query segments. In this way, we obtain 833 updated clusters represented by the 833 query segments whose semantic meanings serve as cluster centers.

2.3 Refine clustering results by outlier detection

The semantic meanings expressed in the query segments in the affinity matrix are definitely not inclusive of all possible semantics in the image database. Therefore, the number of semantic clusters cannot be simply decided by the number of distinct queries in the affinity matrix. In order to explore for more potential semantic clusters, we try to partition the existing clusters by singling out those loosely connected segments. These segments are outliers of the original clusters, and will be formed into new clusters. Therefore we refine the semantic clustering generated by the above mentioned clustering method by finding outliers and outlier groups inside the clusters. However, the information of disconnected outliers is very difficult to analyze. A general outlier detection and evaluation algorithm is proposed in our previous work [10], with its origin from the Network Flow of Graph theory [4]. In this study, we adapt it to suit the needs of both semantic clustering and region-based image retrieval. The basic idea of the algorithm is as follows.

Each data point of a cluster is a "vertex" of the network. Vertices are connected by edges. If a point is far away from the majority of points, this point is a so-called outlier. We want to set the network and use the edge capacity to represent the relationship among points. The goal is to determine if a cluster C contains points that are only weakly related to the rest [10].

This algorithm consists of three main phases. First, a network is set up by k nearest neighbor graph. The capacity of each edge is reflected by the distance between the two connecting points. If the two connecting points are far away from each other, the capacity between them is low; if two points are close, the capacity between them is high. When a point is far away from the majority of the data, its total edge capacities are low. We hope to find this outlier by separating those edges with low capacities. We start with a vertex with the longest average edge length (minimum average capacity) as the source in the network, and then search for its farthest vertex as the sink and run the network flow algorithm. The farthest vertex is the most different vertex from the source vertex in the network [10]. After the candidate outliers are identified and removed, the network is updated, and the next iteration starts. The maximum flow is equal to the total capacities of the edges on the minimum cut which separates the source and sink. This iterative process stops when the average capacity of those edges on the minimum cut is less than the average edge capacity of the original network

The second phase is to adjust the total maximum flow and the outlier degree produced at each iteration. In Phase 1, due to the order of removing candidate outliers or outlier groups, outliers removed later may have artificially low network flow because of the removal of previous outlier groups that may have been near them which would be interpreted as high outlier degrees. To solve this problem, we coarsen each minimum cut into a new vertex.

The third phase is to select outliers from the candidate outlier/outlier groups. Different applications can have different requirements on outliers. Users can specify a threshold, such as the outlier percentage to single out outliers.

The basic steps of this algorithm are as follows:

/* Phase 1 */

1. Set up k nearest neighbor network.
2. Select a source s and its farthest vertex as the sink t. Find a maximum flow from s to t. Find a minimum cut separating s and t and use the smaller side as the candidate outlier or outlier group.
3. Remove the candidate outlier or outlier group from the graph. Repeat Steps 1 to 3 until the stop criterion is met.

/* Phase 2 */

4. Coarsen the original graph and construct the Gomory-Hu Tree [6] on the coarsened graph.

/* Phase 3 */

5. Select outliers from candidate outliers.

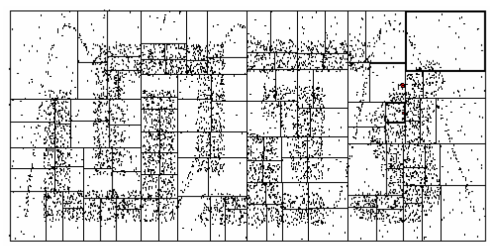
From the initial semantic clustering results, we found some obviously misclustered segments that do not actually belong to the clusters they are assigned to. This often happens to those segments with negative feedbacks or those with no feedback records in the query log file. Although these segments are assigned to their nearest query centers, the Euclidean distance between these segments and the query centers could still be large. The average distance between those segments and their nearest query centers is 1.85, while the maximum and minimum distances are 292.32 and 0.02, respectively. In total, there are 5340 out of 82552 segments whose distance to the query center is greater than the average. We do not deal with those segments separately. In the outlier detection step, those segments can be detected automatically. Therefore, after the outlier detection, new semantic clusters (outliers) are generated and the total number of semantic clusters in the image database can be approximated by this number based on the best knowledge extracted from the log file. This is desirable since we do not have a priori knowledge on the appropriate number of clusters, which is often a requirement for traditional clustering methods such as *K*-means.

2.4 Locate candidate image segments for region-based image retrieval

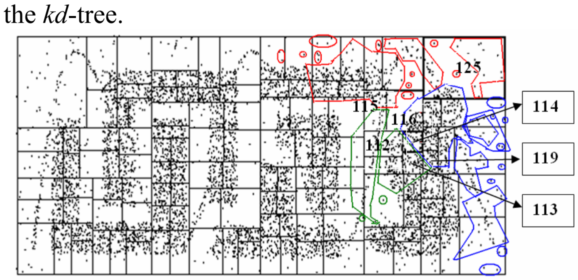
In our experiments, after the semantic clustering and the outlier detection which refines the clustering results, the whole image segment data set is clustered into 1407 semantic clusters (833 clusters and 574 outlier/outlier groups). The cleaned clusters, together with the outliers/outlier groups, are called microclusters. We then locate our search space from these microclusters.

A query segment specified by the user could be located in any microcluster. The size of this microcluster could be 1, such as the case of one single outlier, or more than one. It is obviously not a good idea to simply reduce the search space to the microcluster that the query region falls into because this might cause low retrieval accuracy for two reasons. First, if the microcluster is an outlier/outlier group, it might just have a small few segments in it. Second, even if the microcluster is one of the regular clusters, the query region could be more similar to some regions in another micorcluster than its own. 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. In [9], we successfully use buckets to locate the search space for an 8-dimmensional data set. "Bucket" is a concept from *kd*-trees [5] which is used to find the *k* nearest neighbors in logarithmic expected time. Buckets are the leaf nodes of a *kd*-tree where data are stored. Usually, the bucket size is determined by the desired number of nearest neighbors. For example, if we want to get the 7 nearest neighbors of the query point, we can simply set the bucket size to 7. Therefore, the search space will be first reduced to the bucket where the query segment is located. If its distance to the 7th nearest neighbor is larger than that to one of the neighboring buckets, *kd*-tree will search the neighboring buckets until all the nearer buckets are checked. However, in our case, buckets are not used for locating the *k* nearest neighbors. We use buckets to locate the search space for region-based image retrieval. In our experiments, the whole data set contains 82,552 points/segments. By indexing the original data set into a *kd*-tree with a bucket size of 500, there are in total 1197 buckets.



(a) 8000 data points are held by a *kd*-tree with a bucket size of 100. Rectangles are called buckets of the *kd*-tree.



(b) The No. 125 bucket on the upper righter corner overlaps with 5 microclusters. The closed curves show the boundaries of microclusters.

Figure 1. A 2-dimensional data set example showing the relationship between buckets and microclusters.

In addition to checking the microclusters that the query segment belongs to, we also check the microclusters which overlap with the bucket where the query region is located. In Figure 1, we use a 2-dimensional data set to illustrate the relationship between the buckets and microclusters. We use buckets as a microscope - the bigger the bucket size, the more microclusters the bucket will overlap with, and the more segments need to be checked in the retrieval. We do not limit the maximum number of microclusters to check. In stead, it is automatically determined by the number of microclusters that overlap with that bucket. The search space is reduced to those microclusters.

3. The retrieval system

We test the proposed semantic clustering method with a region-based image retrieval system [16]. In this section, we will briefly introduce this system.

This region-based image retrieval system is based on Multiple Instance Learning (MIL) [11]. Since each image is composed of several regions and each region can be taken as an instance, a region-based CBIR is transformed into a Multiple Instance Learning (MIL) problem, in which 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 interested in only 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 [16], the system successfully maps the region-based image retrieval problem to a MIL problem.

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 region and all the other semantic regions in the reduced search space. This is obtained by first locating the bucket that the query region falls into. Then, all the microclusters that overlap with the bucket constitute the reduced space where search and retrieval is performed.

The smaller the distance, the more likely a region is similar to the query region. The distance between an image and the query segment is thus equal to the smallest distance between the query segment and the segments contained in the image. We compute such distances for all images in the reduced search space and return the top 30 images to the user for feedbacks. 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 the 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 reduced search space. 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 [16].

One-class SVM learns from the training set and returns the refined results to the user who will provide further feedback. This whole process goes through several iterations until a satisfactory retrieval result is obtained. Our previous work shows its effectiveness.

The database log keeps track of the users' feedbacks. After a period of time, the log file will be used to update the semantic clustering.

4. Experimental results

The experiment is conducted on a Corel image database consisting of 9,800 images from 98 categories. After segmentation by Blob-world [2], there are in total 82,552 image segments. 833 clusters are initially constructed directly from the log file. After outlier detection, there are altogether 1407 microclusters. By indexing the data using *kd*-tree, there are 1197 buckets. In our experiments, twenty images are randomly chosen from 15 categories as the query images. After clustering, the average number of images that need to be search in each query is reduced to 25.7% of the whole image database.

In order to examine the quality of semantic clustering, we test the clustering result with the region-based retrieval system [16]. The proposed algorithm is compared with the one that uses distance-based clustering [15], which uses the Genetic Algorithm for initial clustering and improves the clustering result through outlier detection [10].

Five iterations 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 2 shows the accuracy rates after the first fourth iteration of relevance feedback. "Genetic Clustering" is the distance-based clustering without considering semantic relationships among image segments. "Semantic Clustering" is the proposed clustering scheme. The proposed algorithm outperforms the distance based clustering. Since both algorithms use the same retrieval system and only the search space is different, it can be concluded that by incorporating semantic meaning in the clustering scheme, the “semantic gap” is reduced.

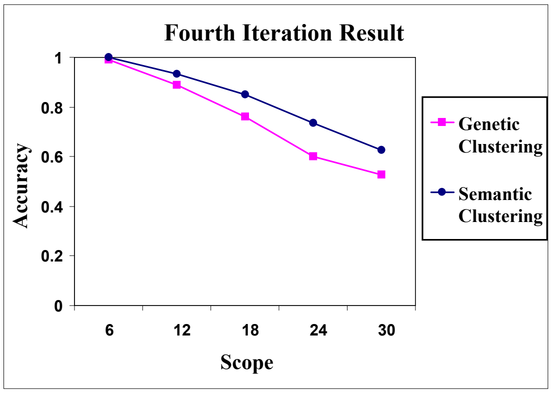


Figure 2. Fourth iteration comparison between semantic clustering and distance-based clustering

5. Conclusion

This paper proposes a semantic clustering method for region-based image retrieval. The method considers each cluster as a semantically independent unit. The initial clusters are constructed from the database log file containing the users’ relevance feedbacks. Semantic meanings (i.e., semantic clusters) that are not represented in log files are further constructed through an outlier detection method. The proposed method is a novel way to use database logs and hence the users’ feedbacks. Another merit of the algorithm is that it does not require a prior knowledge as to the number of clusters. This is a desirable feature since this prior knowledge is hard to acquire. The experimental results demonstrated the effectiveness of the semantic clustering in reducing the “semantic gap” while reducing the search space.

6. Acknowledgement

The work of Chengcui Zhang was supported in part by SBE-0245090 and the UAB ADVANCE program through the sponsorship of the National Science Foundation.

7. References

[1] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering Gene Expression Patterns," *Journal of Computational Biology*, 1999.

[2] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026-1038, 2002.

[3] L. Duan, Y. Chen, and W. Gao, "Learning Semantic Cluster for Image Retrieval Using Association Rule Hypergraph Partitioning," in *Proc. of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and the Fourth Pacific Rim Conference on Multimedia*, 2003.

[4] S. Even, *Graph Algorithms*: Computer Science Press, 1979.

[5] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Trans. on Mathematical Software*, vol. 3, no. 3, pp. 209-226, 1977.

[6] R. E. Gomory and T. C. Hu, "Multi-terminal Network Flows," *J. SIAM*, vol. 9, pp. 551-570, 1961.

[7] Z. Gong, L. H. U, and C. W. Cheang, "Web Image Semantic Clustering," in *Proc. of the International Conference on Ontologies, Databases and Applications of Semantics*, 2005.

[8] E. H. Han, G. Karypis, V. Kumar, and B. Mobasher, "Clustering based on Association Rule Hypergraphs," in *Proc. of the Workshop on Research Issues on Data Mining and Knowledge Discovery*, Tucson, Arizona, USA, 1997.

[9] Y. Liu, X. Chen, C. Zhang, and A. Sprague, "An Interactive Region-based Image Clustering and Retrieval Platform," in *Proc. of the IEEE International Conference on Multimedia & Expo (ICME 2006)*, Toronto, Ontario, Canada, 2006.

[10] Y. Liu and A. P. Sprague, "Outlier Detection and Evaluation by Network Flow," in *Proc. of the International Conference on Machine Learning and Applications*, 2004.

[11] O. Maron and T. Lozano-Perez, "A Framework for Multiple Instance Learning," *Advances in Natural Information Processing System 10*, 1998.

[12] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based Image Retrieval with Relevance Feedback in MARS," in *Proc. of the International Conference on Image Processing*, 1997.

[13] G. Sheikholeslami, W. Chang, and A. Zhang, "SemQuery: Semantic Clustering and Querying on Heterogeneous Features for Visual Data," *the IEEE Trans. on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 988-1002, 2002.

[14] X. Yin, M. Li, L. Zhang, and H. J. Zhang, "Semantic Image Clustering Using Relevance Feedback," in *Proc. of the International Symposium on Circuits and Systems (ISCAS)*, 2003.

[15] C. Zhang and X. Chen, "OCRS: An Interactive Object-based Image Clustering and Retrieval System," in *Proc. of the Fourth International Workshop on Multimedia Data Mining (MDM/KDD2005), in conjunction with the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Chicago, USA, 2005.

[16] 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," in *Proc. of the IEEE International Conference on Multimedia & Expo (ICME)*, Amsterdam, the Netherlands, 2005.