J. Vis. Commun. Image R. xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Semantic clustering for region-based image retrieval

Ying Liu, Xin Chen, Chengcui Zhang *

Department of Computer and Information Science, University of Alabama at Birmingham, CH 127, 1530 3rd Avenue S., Birmingham, AL 35294, USA

ARTICLE INFO

Article history:
Received 30 May 2008
Accepted 12 November 2008
Available online xxxx

Keywords:
Semantic clustering
Content-based image retrieval
Outlier detection
Network flow

ABSTRACT

With the proliferation of applications that demand content-based image retrieval, two merits are becoming more desirable. The first is the reduced search space, and the second is the reduced "semantic gap." This paper proposes a semantic clustering scheme to achieve these two goals. By performing clustering before image retrieval, the search space can be significantly reduced. The proposed method is different from existing image clustering methods as follows: (1) it is region based, meaning that image sub-regions, instead of the whole image, are grouped into. The semantic similarities among image regions are collected over the user query and feedback history; (2) the clustering scheme is dynamic in the sense that it can evolve to include more new semantic categories. Ideally, one cluster approximates one semantic concept or a small set of closely related semantic concepts, based on which the "semantic gap" in the retrieval is reduced.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

Content-based image retrieval (CBIR) has become an important part of information retrieval technology. One challenge in this area is that the ever-increasing number of images acquired through the digital world makes the brute force searching almost impossible. Most of the existing Content-based Image Retrieval systems consider each query image as a whole, which is represented by a vector of N dimensional image features. However, a single image can include multiple regions/objects with completely different semantic meanings. A user's query interest is often focused on one particular 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 the user's interest. However, this makes the exhaustive search even less feasible for region-based image retrieval systems since an image is typically split into 7–8 regions (regions), and the search space of a region-based system will be 7–8 times as large as otherwise it could be. The expanded search space makes retrieval efficiency a critical issue. 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 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 challenge in the area. "Relevance Feedback" [23] is a technique that has been well studied and proved effective in reducing the semantic gap. In our previous work [26], 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/regions (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 phase, if the user labels a certain image "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 a 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 affinity matrix in the log file and design a semantic clustering scheme, whose purpose is twofold: to reduce the "semantic gap" and to reduce the search space.

In the proposed method, the initial cluster centers are the query targets (query sub-regions/regions) in the log file. Each cluster is assumed to be an independent semantic unit. Each image region is compared with the initial cluster centers and assigned to one or more clusters according to the users' feedbacks (recorded in the log file) regarding its closeness to the semantic categories of the clusters represented by the query targets. Since one image region 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 the "red" cluster and "flower" cluster. This also differentiates our proposed method from other existing image clustering methods.

* Corresponding author.
E-mail addresses: liuyi@cis.uab.edu (Y. Liu), chenxin@cis.uab.edu (X. Chen), zhang@cis.uab.edu (C. Zhang).

1047-3203/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2008.11.006

UNCORRECTED PROOF

The new problem that arises is the quality of the clustering. There is no prior knowledge on the number of semantic meanings among image regions 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 even for an expert to enumerate all the unique semantic meanings in a database containing a huge number of natural-scene images. Another problem is that for some image regions/segments, since they have never been queried and/or retrieved before, no semantic information can be extracted from the log file. However, these regions 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" regions to the closest existing semantic clusters according to its distance to the cluster center. Then, each cluster is further divided into microclusters by an outlier detection method based on our previous study [21]. Regions 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 refer to them new and old microclusters, respectively, and do not differentiate them as clusters and outliers in the subsequent region-based image retrieval phase. A detailed introduction to 'outlier' and a brief review of the state of the art of outlier detection are presented in Section 3.4.

In summary, this paper proposes a novel clustering method based on the semantics of image regions in the database. An outlier detection method is applied in the clustering phase to refine the clustering results and discover new semantic clusters. The effectiveness of the proposed clustering method is tested on an image database and our experimental results show that the proposed algorithm can improve the quality of the clustering and thus improve the accuracy of the region-based image 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. Related work

"Texts" in natural languages are the main means to convey semantics among human being. Therefore, the status quo of semantic image clustering is to incorporate "texts", e.g., captions to facilitate the understanding of images. Gong et al. [12] proposed to integrate the captions of images for semantic clustering. The work proposed in [24] requires that the semantics of the image database are pre-defined by domain experts. However, in many cases, neither the captions (texts) nor the semantic categories are readily available. A web image search engine is proposed in [27], which filters images by performing clustering based on image captions.

There are also some researches on semantic image clustering that do not directly reply on image captions. For example, a semantic tolerance model is built by Dai and Cai [28], which first represents images based on semantic classification. For this purpose, it is necessary to gather the categorized training images. Another useful source of information for image semantics is the web. For example, Hai [29] proposes to understand the images through the analysis of semantic links existing among web pages.

In [7,25], it is proposed that semantic clustering is performed using relevance feedback. These works are based on the whole image instead of image sub-regions/regions. The clustering method in [25] is based on a method called CAST [2] while the one in [7] is based on the Association Rule Hyper-graph Partitioning algorithm [14]. While both of these clustering schemes apply existing general-purpose clustering methods, we propose a new method that constructs clusters based on the semantics of image regions. Another work that uses RF for semantic clustering is proposed in [30]. In this research, the users are asked about the similarity groups of images. Images are then clustered based on these answers. This is different from the traditional way of doing RF, when users are asked for feedback in the process of image retrieval. The proposed method in this paper uses the RF to the retrieval results both in the retrieval process and the clustering process. It is not feasible to solicit users' feedback for retrieval and clustering separately as this will introduce too much trouble to the users.

The proposed semantic clustering algorithm is also different from most of the clustering algorithms in general. Most existing clustering schemes depend on a heuristic on k – the number of clusters. In the proposed method, the number of possible semantic meanings/categories (i.e., number of clusters) in the image database is automatically estimated.

3. Semantic clustering

In this section, the detailed semantic clustering algorithm is presented. The semantic relations between and among image regions are obtained from the database query log. In particular, an affinity matrix is constructed from users' feedbacks based on which the initial clustering is performed. The initial clustering results are further refined using an outlier/outlier group detection algorithm. The region-based image search is then performed on "microclusters" rather than an exhaustive search.

3.1. The affinity matrix extracted from the log file

The semantic similarity information used in the semantic clustering is from the users' feedbacks stored in the database query log. Before the details of the proposed semantic clustering scheme are elaborated, a concrete view of the data format used to store those semantics is presented as follows.

The users' feedbacks collected over time are recorded in a query log file. In this log file, the user's feedback history for each query region is recorded. The log file is thus represented by an affinity matrix. Its structure is shown in Table 1. $I_1, I_2, , \ldots$ represent images and $S_1, S_2, , \ldots$ are image regions. 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 region. 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 region and is therefore has a positive user feedback. In another word, the user thinks there is at least one region/object in that image that is relevant to the query region. The

Table 1
Affinity matrix.

		I_1	I_2	I_3	…	I_{9800}
S_1		0	1	0	…	−1
S_2		1	2	−1	…	3
S_3		3	0	1	…	−2
…	…				…	
S_{833}		0	0	−1	…	0

UNCORRECTED PROOF

negative integers imply that no region of that image matches the query region and is therefore given a negative feedback. '0' means we do not have any information on the relevance of this image to the query region. 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/her to provide feedback on too many images (i.e., >20 images), those images with lower similarity ranks will not receive a specific (positive/negative) feedback score. Among the 1188 queries (image regions), some of them are the same, i.e., they are queried more than once. We merge the users' feedback on these duplicate queries and finally obtain 833 unique query targets/regions (see Table 1).

Since different users may provide different feedback even for the same query region, one more study is conducted to show the difference among user feedbacks. In particular, those queries in the log file that have the same query region are analyzed to reveal the difference in feedback by different users with respect to perceptual subjectivity. There are 253 such queries in total in the log file. We measure the 'similarity' (the level of agreement) in positive and negative feedback from different users in the following way. For each group of queries that have the same query region, we collect the set of positive (or negative) images marked by all users. For each query in that group, we compare its positive (or negative) image set with the positive (or negative) image set of that group. The similarity score between an individual query and the query concept represented by the positive (or negative) image set of that group, is calculated as the ratio of the number of common positive (or negative) images to the total number of positive (or negative) images in the group set. For example, if there are three positive images in the group set, and two of them belong to the positive set of some query in that group, the similarity score between that query and the group's positive image set is 66.67%. After calculating the similarity score between each query and the group it to which it belongs for all 253 queries, the average similarity score for positive feedback is 84.37%, while the similarity for negative feedback is 88.80%. The users who helped to collect the feedback information in the log file include one Ph.D. student, whose research area is content-based image retrieval, two other Ph.D. students in the fields of Bioinformatics and Software Engineering, respectively, and four MS students who are not involved in any kind of research.

3.2. Semantic gap

"Semantic gap" is the gap between low-level features and high-level human concepts, which is a well-known challenge in the content-based image retrieval. As mentioned above, the query log file contains positive and negative feedbacks from the users of the image retrieval system. In another word, the log file reflects human understanding of the image database. Although it is not the complete ground truth, we can still peek through it and have a sense of the existence of "Semantic Gap". For this purpose, we design the following experiment. We use the 20 nearest neighbors of each query region to represent the machine understanding of the closest regions to this query region. These nearest neighbors are obtained from a *kd*-tree [10] by Euclidean distance. We compare these 20 nearest neighbors with the user's feedbacks in log file, which represents human understanding of the image regions' semantic meaning. The data in Table 2 shows the difference between the semantic understanding of a human and the machine. As can be gleaned from the first row of Table 2: (1) 49.5% of the positive images in the log file are from the 20 nearest neighbors; (2) 9.5% of the 20 nearest neighbors are labeled positive by the users; (3) the average number of positive regions for each query region is 5.3. The second row is about negative feedbacks: (1) 25.7% of the negative images in the log file are from the 20 nearest neighbors;

Table 2
The experimental result on measuring the query semantic gap.

	Percentage of P/N images in the 20 nearest neighbors	Percentage of the 20 nearest neighbors labeled as P/N	Average number of P/N regions
Positive (P)	49.5	9.5	5.3
Negative (N)	25.7	69.7	62.4

(2) 69.7% of the 20 nearest neighbors are labeled negative by users; (3) the average number of negative regions for each query is 62.4.

From the log file, we can also see that some queries do not have any positive feedbacks. For example, if a user wants to search for a tiny white region on a flower image, the system will give images containing white regions that are similar to the query region in terms of low-level color features, such as cloud or snow. However, none of them contains what the user really wants. This could cause the values for positive feedbacks in Table 2 very low while the values for negative feedbacks very high. On the other hand, if the query region is well segmented, such as a perfect region of a tiger, a horse, or a red flower, the system may be able to return many positive images. In either case, there still exists a semantic gap. The purpose of providing these figures in this paper is not to measure exactly how big the semantic gap is, and these figures may change with the update of the log file. However, we can still get a sense of the semantic gap between the machine and human understandings. To reduce this semantic gap is one of the main objectives of this paper.

3.3. Initial semantic clustering

Our clustering method is based on the affinity matrix in the query log file. In this study, there are altogether 9800 images in the database with 82,552 image regions segmented by an automatic image segmentation method Blobworld [5]. Each image region is represented by an N dimensional feature vector. Specifically, 32 low-level features (e.g., color, texture, and shape) are extracted for each image region, i.e., each image region. Hence, each image region/region is represented by a 32-dimensional feature vector. Based on the affinity matrix mentioned in the previous section, we use each query region as a semantic cluster center. Although these query regions may not accurately represent the "centroids" of clusters in terms of its low-level features, the semantic meaning it conveys can be reasonably regarded as an initial estimate of the center of that semantic cluster. For every image that is labeled positive given a query region, we find out which region of that image has the shortest Euclidean distance to the query region, and put this region (positive region) in the same cluster of the query region. All the other regions in that image have a label of 'unknown.' And all the regions in negative images are marked 'negative' with regard to that query region. The total number of positive regions identified from the log file by this method is 3586, and 9535 for negative/unknown regions.

As an initial step of semantic clustering, we first examine the regions with positive labels. For the 833 unique query regions, their positive feedback sets could overlap, i.e., the same image region could be labeled positive in different queries. In other words, an image region may belong to multiple clusters represented by different query regions. For example, when we search for a white object, the images containing one or more white horses are labeled positive. When we search for white horses, those regions 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,

UNCORRECTED PROOF

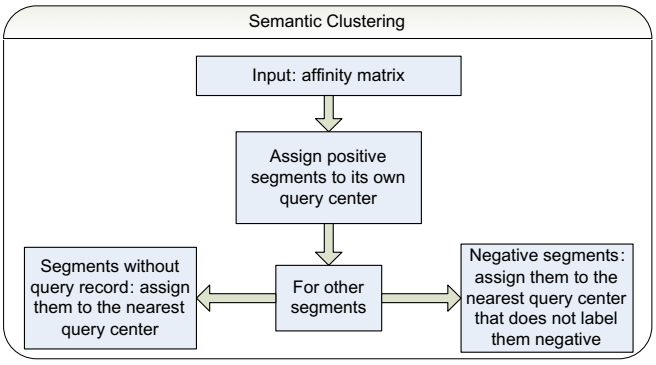


Fig. 1. An overview of the semantic clustering process.

especially because different users' subjective perceptions can be different in many ways. Therefore, in the following experiment, we allow a region to belong to different clusters.

In the next step, we try to cluster the negative regions and those regions without labels (no feedbacks). For negative regions, we first exclude them from the clusters represented by their corresponding query regions. Then we assign them to the next nearest semantic centers by computing their Euclidean distances to each cluster center (query region). For those regions without any labels, we just assign them to the same cluster of their nearest query regions. In this way, we obtain 833 updated clusters represented by the 833 query regions whose semantic meanings serve as cluster centers. An overview of the method is illustrated in Fig. 1. With more data being collected, the affinity matrix will grow larger, making scalability an issue. Therefore, in an image retrieval system, this step and the rest of the clustering steps are performed offline and shall not be frequently updated. The update is performed only when the growth of data in the database reaches a certain amount.

3.4. Refine clustering results by outlier detection

In this section, the general meaning of 'outliers' is presented first, followed by the detailed explanation of how outlier detection can be used to refine clustering results and discover new semantic clusters, and how it is adapted to solve our particular problem in semantic image region clustering. A comparison of the outlier detection method used in this study with other existing outlier detection methods is presented in Section 3.4.6.

3.4.1. Outliers

Outliers are those points which are different from or inconsistent with the rest of the data. Novel, new, abnormal, unusual or noisy information can all be called outliers. Sometimes the outliers are more interesting than the majority of the data, such as the applications of intrusion detection and unusual usage of credit cards. With the increase of the complexity and variety of datasets, the challenges of outlier detection are how to catch similar outliers as a group, and how to evaluate the outliers.

Traditional outlier detection methods are statistical, especially for discarding of noise. Those unwelcome errors will affect the observations and contaminate the computation results. Therefore, the early attitudes toward outliers considered outliers to be bad, and could cause mistaken analysis of data [1].

Like the definition of outliers, outliers can also represent novel properties. As mentioned above, sometimes the outliers are more interesting than the rest of the data. For example, outlier detection is useful in medical analysis for finding unusual responses to various medical treatments. With the development of computer sciences and the internet, many unexpected situations need outlier detection.

3.4.2. Outlier detection for semantic clustering

The semantic meanings expressed in the query regions 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. Another problem is that although some regions are assigned to their nearest query centers, they might not really belong to that cluster in terms of semantic meanings or from the density connection's point of view. In order to explore for more potential semantic clusters, we try to partition the existing clusters by singling out those loosely connected regions or region groups. These regions or region groups are outliers of the original clusters, and will be grouped into new clusters. Therefore, we refine the semantic clustering generated by the initial clustering method by finding outliers and outlier groups inside the clusters. After we find these outliers/outlier groups, we regard them as new clusters. Together with the original clusters, a refined clustering result is formed with each cluster representing a distinctive semantic meaning. However, the information of disconnected outliers is very difficult to analyze. A general outlier detection and evaluation algorithm is proposed in our previous work [21], with its origin from the Network Flow of Graph theory [9]. In this study, we adapt it to suit the needs of both semantic clustering and region-based image retrieval.

In our outlier detection method, in order to analyze the quality of each cluster, the first step is to construct a network for each cluster. 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 setup the network in a way that the edge capacity can be used to represent the relationship among points (vertices). The goal is to determine if the cluster contains points that are only weakly related to the rest [21].

In the following subsections, a brief review of the Network Flow theory is presented followed by the outlier detection algorithm used in this study. This algorithm will be used to detect the outliers in image region clusters and refine the clustering result.

3.4.3. Network flow

Let $G = (V, E)$ be a directed graph with no self-loops and no parallel edges, and let each edge have a capacity which is a nonnegative real number. Let vertices s and t be specified; s is called the source and t the sink. An edge capacity is represented by $c(e)$. A flow is a function f from the edges to the real numbers satisfying:

(1) For every edge $e \in E$, $0 \leqslant f(e) \leqslant c(e)$.
(2) For every vertex v except the source and the sink, the flow incoming to v is equal to the flow outgoing from v.

The maximum flow problem is to find a flow function f which maximizes the total flow, where the *total flow* is defined as the amount of flow leaving the source, minus the amount entering the source. Let X be a subset of the vertex set V and $\overline{X}$ be the complement of X. If $s \in X$ and $t \in \overline{X}$ then $(X, \overline{X})$ is called a *cut* separating s and t. The *capacity* of cut $(X, \overline{X})$ is the sum of capacities of edges from X to $\overline{X}$ The Maximum Flow Minimum Cut theorem states that the maximum amount of flow from s to t equals the minimum of the capacities of cuts separating s and t.

3.4.4. Outlier detection algorithm

The basic idea of our outlier detection algorithm is as follows. Suppose that s is an outlier. Let t be the point in cluster C that is farthest from s. Suppose further that s is far from all the other points in C, then each edge connected to s has small capacity. Then the network flow algorithm will tell us that the maximum flow from s to t is small, and quite likely the minimum cut will be $(\{s\}, C - \{s\})$. Alternately, if t is also an outlier, the minimum cut may single out t as well.

UNCORRECTED PROOF

This whole algorithm consists of three main phases. First, a network for each cluster is setup by k nearest neighbor graph. The capacity of each edge is reflected by the distance between the two connecting vertices. If the two connecting vertices are far away from each other, the capacity between them is low; if two vertices are close, the capacity between them is high. When a vertex is far away from the majority of the data, its total edge capacities are low. We hope to separate this outlier by cutting those edges with low capacities. We start with a vertex with the longest average edge length (minimum average capacity) as the source s in the network, and then search for the farthest vertex from the source as the sink t and run the network flow algorithm. The farthest vertex is the most different vertex from the source vertex in the network [21]. Then we find a maximum flow from s to t and a minimum cut separating s and t. The maximum flow is equal to the total capacities of the edges on the minimum cut which separates the source and sink. After minimum cut, the side that has smaller edge capacities is identified as the candidate outlier or outlier group. These candidate outliers are then removed, the network is updated, and the next iteration starts. 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.

Phase 2 of the algorithm is to adjust the maximum flow. In Phase 1, due to the order of removing candidate outliers and outlier groups, outliers removed later may have artificially low network flow which would be interpreted as being strong outliers. To solve this problem, each candidate outlier group is coarsened into a new vertex. When the stop condition in Phase 1 is satisfied, the remaining data is also coarsened into a new vertex, which is called the body vertex. The body vertex is used as the source and each other vertex as a sink to run the network flow and a Gomory–Hu Tree [11] is constructed on the coarsened network. Suppose the original network is called N_1, the coarsened network is called N_2 and the corresponding Gomory–Hu tree structure is called T. T satisfies the following properties. All the nodes of the T are vertices of N_2 and the root is the body vertex. A descendant's edge capacity is always less than or equal to its predecessor's edge capacity. The tree structure T represents the maximum flow between all pairs of vertices in the network. The minimal edge capacity between a pair of vertices in the original network N_1 is the minimum edge capacity along the connecting path of this pair in the tree T.

A subtle issue in Phase 3 is that different users may disagree on how many outliers are there, depending on different intended uses of the data, application-specific requirements, and other affecting factors. This study focuses on detecting those data points loosely connected to the main data. The user can either specify what percentage of the data should be considered outliers, or a threshold on outlier degrees can be specified to separate outliers from normal data points.

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 1–3 until the stop criterion is met.

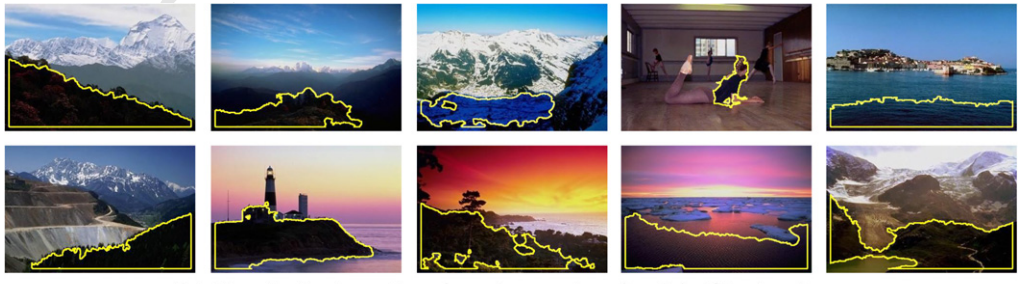
/* Phase 2 */
4. Coarsen the original network and construct the Gomory–Hu Tree [11] on the coarsened network.

/* Phase 3 */
5. Select outliers from candidate outliers.

3.4.5. Outlier detection in image region clusters

From the initial semantic clustering results, we found some obviously misclustered regions. This often happens to those regions with negative feedbacks or those with no feedback records in the query log file. Although these regions are assigned to their nearest query centers, the Euclidean distance between these regions and the query centers could still be large. The average distance between those regions 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 82,552 regions whose distances to the corresponding query center are greater than average. We do not deal with those regions separately. In the outlier detection step, those regions 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.

Fig. 2 shows some sample outlier regions and normal regions in a cluster. For Cluster A in Fig. 2, there are 350 regions (data points)



(a) Regularly shaped regions (normal regions) in Cluster A

(b) Irregularly shaped regions (outliers) in Cluster A

Fig. 2. Outlier detection for regions with irregular shapes and dark shade texture.

Please cite this article in press as: Y. Liu et al., Semantic clustering for region-based image retrieval, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.11.006

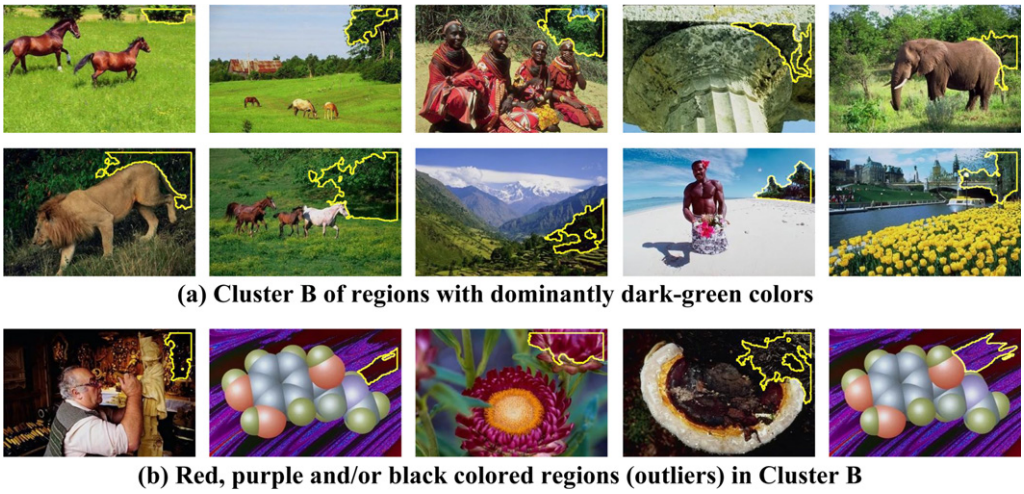


Fig. 3. Outlier detection for regions with non-dominant color features.

connected on the same network by four nearest neighbors. Two iterations of maximum flow/minimum cut find 117 outliers. We examine the low-level features of those outlier regions and find that they usually have irregular shape features, while normal regions in that cluster have more regular shape features. Another example in Fig. 3 shows a cluster (Cluster B) where the outlier regions have quite different colors compared with that of the normal regions in that cluster. In Cluster B, there are 1108 segments. By five iterations of network flow, we find five outlier groups, one per network flow iteration. These five outlier groups contain 52 outlier regions. In addition, since there are 95 points that cannot be connected onto the network, Cluster B has 147 outliers in total.

In brief, the use of outlier detection for cluster repairing actually alleviates the problem of misclustering by removing outliers or outlier groups which could potentially correspond to new semantic meanings that are not previously known by the log file. Fig. 4 is a two-dimensional clustering result we obtain by using another clustering algorithm hMETIS [1]. Our outlier detection algorithm can fix a bad clustering result in general. Actually, the worse the clustering results, the more evident the effectiveness of our outlier detection method.

3.4.6. The comparison of outlier detection algorithms

In this section, our outlier detection method is compared with another outlier detection method which is based on the density analysis [4]. Breunig et al. [4] created an outlier detection algorithm based on an object's neighborhood density, i.e., estimating the density at the point p by analyzing its k nearest neighbors. By measuring the difference in density between an object and its neighboring objects, this algorithm assigns each object a degree of being an outlier called local outlier factor (LOF). If the object is isolated with respect to the surrounding neighborhood, the LOF value would be high, and vice versa. First, the algorithm finds every object's k nearest neighbors. Then, the reachability distance of an object p with respect to object o filters out small changes of reachability distance in a uniform density area. For points far away, the reachability distance is the original distance from o to p, written as $d(p, o)$; for points within the kth neighborhood, the reachability distance is taken as the distance to the kth nearest neighbor of o, written k-distance(o). Thus, it smoothes small differences in uniform areas. The algorithm has a single parameter MinPts – the number of an object's nearest neighbors. When MinPts changes from low to high, an outlier's LOF value may change substantially. The objects with high LOF values are considered outliers.

Fig. 4(a) and (b) shows a two-dimensional cluster dataset, and list the top-20 LOF values based on MinPts = 10 and MinPts = 20, respectively. Fig. 4(c) shows the candidate outliers/outlier groups found from the same dataset by our outlier detection method (with $k = 15$). From the figures, we can see the top-20 outliers' positions in (b) are more accurate than those in (a). If we continue to in-

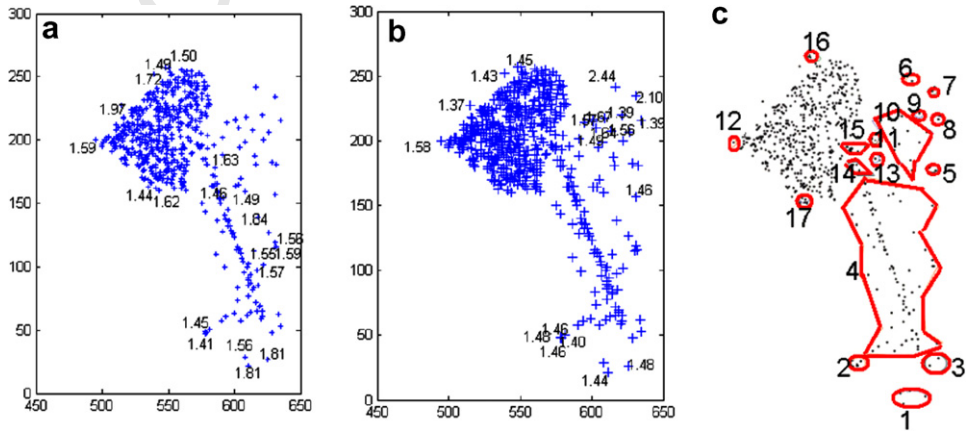


Fig. 4. (a) and (b) Local outlier factors for the sample dataset. Only the top 20 LOFs are shown here. (a) LOFs based on 10 nearest neighbors. (b) LOFs based on 20 nearest neighbors. (c) The candidate outliers found in the same dataset with $k = 15$. The proposed algorithm automatically stops after 13 iterations.

Please cite this article in press as: Y. Liu et al., Semantic clustering for region-based image retrieval, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.11.006

crease the MinPts, the top outliers' positions will move to the bottom-left corner of the dataset. This is because MinPts is not the only factor that decides the LOF values. The other crucial factor is the surroundings of each object, i.e., the surrounding points within each object's k nearest neighbors. We can also see that the LOF is not good at finding outlier groups. Some members of an outlier group have low LOF values. The LOF algorithm cannot find outlier groups, which is not a limitation of our algorithm. As shown in Fig. 4(c), not only isolated outliers, but several outlier groups are found by our algorithm (with $k = 15$). Due to the difficulty in deciding the proper size of small clusters or outlier groups, some members of an outlier group could be missed. In addition, because of the unknown data distribution, nested outliers can be hidden. This example uses two-dimensional data; in higher dimensional data, the situation could be much more complicated.

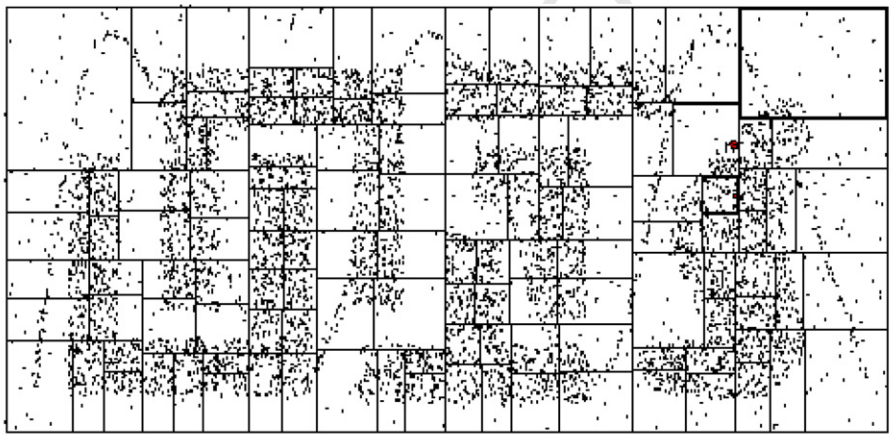
3.5. Locate candidate image regions for region-based image retrieval

In our experiments, we examine the semantic clustering algorithm proposed in this paper with a region-based image clustering system. Semantic clustering of image regions is just the first step towards interactive retrieval of image regions. It will be used to reduce the search space in the later phase of the retrieval.

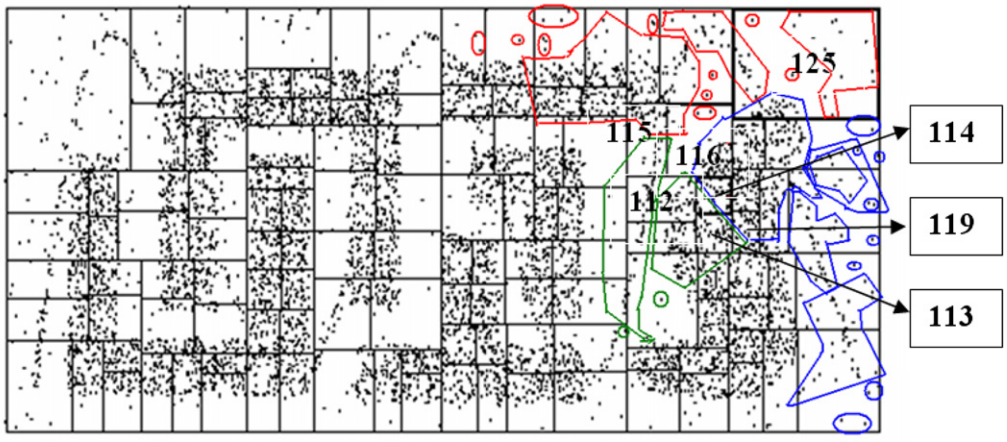
In our experiments, after the semantic clustering and the outlier detection which refines the clustering results, the whole image region data set is clustered into 1407 semantic clusters (833 clusters and 574 outlier/outlier groups). The clusters, together with the outliers/outlier groups, are called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

A query region 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 regions 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 microcluster 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 data points can be very complex. In [20], we successfully use buckets to locate the search space for an eight-dimmensional data set. "Bucket" is a concept from *kd*-trees [10] 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, to get the seven nearest neighbors of the query point, we can simply set the bucket size to 7. Therefore, the search space will be first



(a) 8000 data points are organized 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.

Fig. 5. A two-dimensional data set example showing the relationship between buckets and microclusters.

Please cite this article in press as: Y. Liu et al., Semantic clustering for region-based image retrieval, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.11.006

reduced to the bucket where the query region is located. If its distance to the seventh 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/regions. By indexing the original data set into a *kd*-tree with a bucket size of 500, there are in total 1197 buckets.

In addition to checking the microclusters that the query region belongs to, we also check the microclusters which overlap with the bucket where the query region is located. In Fig. 5, we use a two-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 regions 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. Buckets help us find the nearest microclusters potentially related to the query region. The search space is reduced to those microclusters. The regions in those microclusters are used as input for the next phase of actual learning and retrieval.

Our motivation in using the proposed semantic clustering algorithm is to reduce the search space for the retrieval. LOF cannot be used in this step because it only gives the outlier degree of every data point. This information is not enough to locate the close image regions of a given query target. However, the network flow based outlier detection algorithm is able to refine the clustering result; furthermore, it can locate candidate image regions through buckets. This allows the retrieval algorithm to search image regions within a subset of the whole dataset. Due to this reason, the LOF is not compared with the proposed semantic clustering algorithm on the image retrieval system since it cannot be directly applied in this application.

4. The retrieval system

We test the proposed semantic clustering method with a region-based image retrieval system [26]. Fig. 6 is an example query performed by the system. The image on the upper left corner is the query image provided by the user. Its regions are listed next to it, which are outlined by red lines. In each round, there are 30 images returned as the query results.

This region-based image retrieval system is based on multiple instance learning (MIL) [22]. 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 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 re-

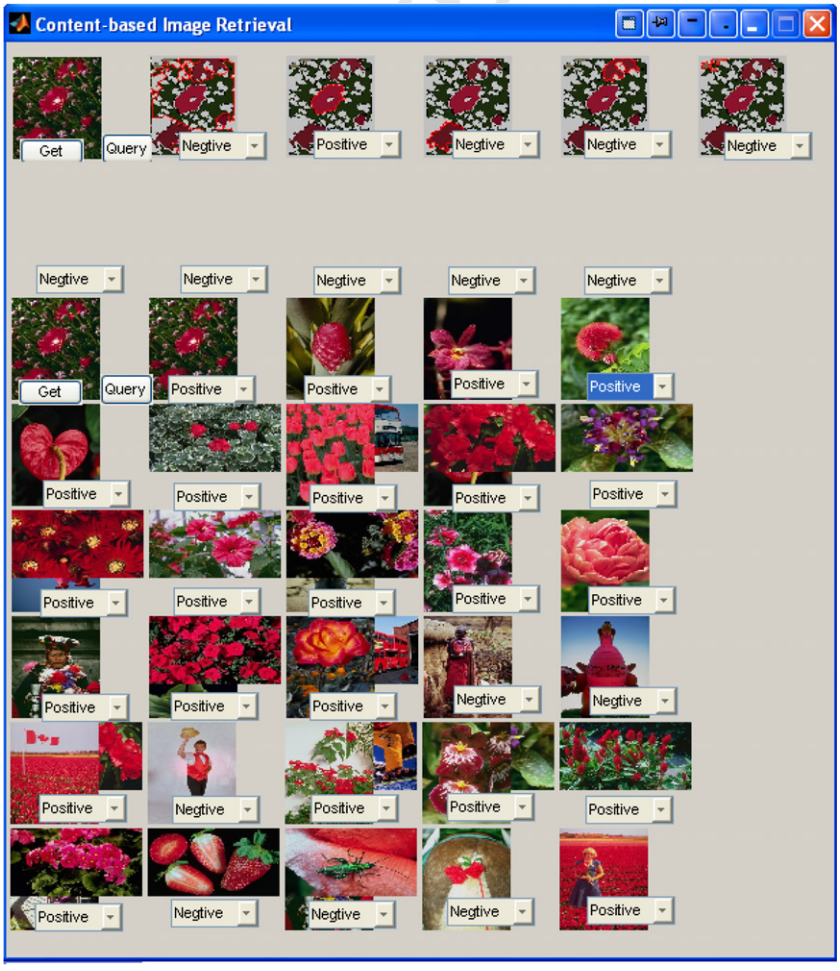


Fig. 6. An example query performed by the system.

Please cite this article in press as: Y. Liu et al., Semantic clustering for region-based image retrieval, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.11.006

gion. In [26], 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 region is thus equal to the smallest distance between the query region and the regions 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. Note that in case there is no relevant image returned in the first round, the mechanism used in the initial retrieval will return the next 30 images that are close to the query image.

With the training sample set, One-class Support Vector Machine (SVM) 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 [26].

One-class SVM learns from the training set and returns the refined results in-time 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.

5. Experimental results

The experiment is conducted on a Corel image database consisting of 9800 images from 98 categories. After region segmentation by Blobworld [5], there are in total 82,552 image regions. Each region is represented by a 32-feature vector – three texture features, two shape features and 27 color features. 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 searched in each query is reduced to 25.7% of the whole image database. In order to examine the quality of semantic clustering, we integrate the clustering component with a region-based image retrieval system [26] and evaluate the performance of clustering in terms of the image retrieval accuracy. For the comparison purpose, the proposed algorithm is compared with another clustering framework proposed in [20], which is distance-based and uses the Genetic Algorithm for initial clustering and improves the clustering result through outlier detection [21].

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.

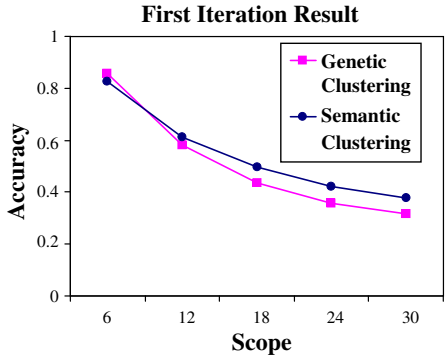


Fig. 7. First iteration result comparison between the proposed semantic clustering method and the distance-based clustering method [20].

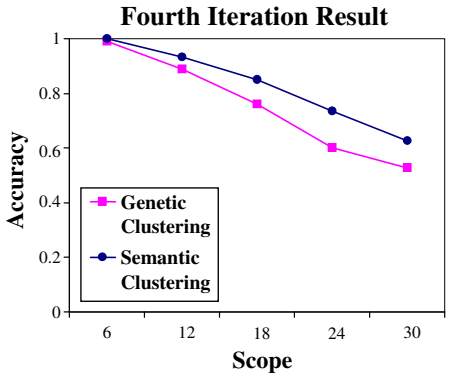


Fig. 8. Fourth iteration result comparison between the proposed semantic clustering method and the distance-based clustering method [20].

Figs. 7 and 8 show the accuracy rates after the first iteration and the fourth iteration of relevance feedback, respectively. "Genetic Clustering" is the distance-based clustering [20] without considering semantic relationships among image regions. "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.

In order to test the impact of the reduced search space on the retrieval performance of the proposed algorithm, we compare it with two other methods that perform exhaustive search [23,26]. Ref. [26] is a region-based learning and retrieval system that uses one-class SVM to solve a Multiple Instance Learning problem. Ref. [23] is a general feature re-weighting Relevance Feedback algorithm.

In Fig. 9, "RF" is the general query re-weighting Relevance Feedback algorithm without a region-based learning component. "SVM" refers to the learning and retrieval mechanism used in testing the proposed semantic clustering method [26]. Neither of the two algorithms have a clustering module prior to retrieval. Therefore, the search space is the whole database. It can be seen from Fig. 9 that the performance of the proposed system is better than that of "SVM" and "RF" although the search space is reduced by 74.3%.

6. Conclusions

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

Please cite this article in press as: Y. Liu et al., Semantic clustering for region-based image retrieval, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.11.006

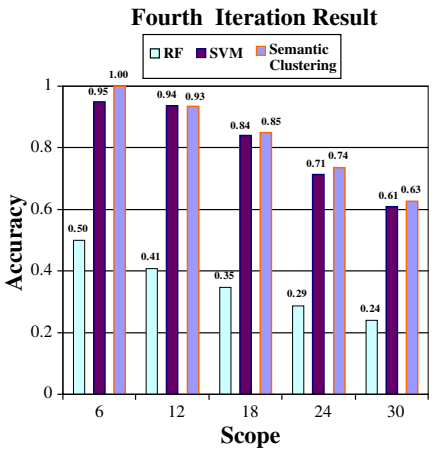


Fig. 9. Fourth iteration result comparison between the proposed algorithm and two other full-search algorithms.

from the database log file containing the users' relevance feedbacks. The affinity matrix obtained from the log file is sparse in nature, given the large scale of the database, which will directly affect the clustering results due to the incomplete information about the semantic categories in the database. This motivates us to further refine the clustering results by using an outlier detection approach, which alleviates the problem of misclustering caused by the incomplete semantic information contained in the log file. By this way, those semantic meanings (i.e., semantic clusters) that are not well 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, which, in our case, is also the number of semantic meanings in the database. This is a desirable feature since this prior knowledge is either hard or impossible to acquire. In our experiments, we test the proposed clustering method with a region-based image retrieval system. The results demonstrated the effectiveness of the semantic clustering in reducing the "semantic gap" while reducing the search space.

7. Uncited references

Q1 [3,6,8,13,15–19].

Acknowledgments

The work of Chengcui Zhang was supported in part by the UAB ADVANCE program and NSF DBI-0649894.

References

[1] V. Barnett, T. Lewis, Outliers in Statistical Data, Wiley, New York, 1984.
[2] A. Ben-Dor, R. Shamir, Z. Yakhini, Clustering gene expression patterns, Journal of Computational Biology, 1999.
[3] C. Bilen, S. Huzurbazar, Wavelet-based detection of outliers in time series, Computational & Graphical Statistics 11 (2002) 311–327.
[4] M.M. Breunig, H. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of SIGMOD International Conference on Management of Data, 2000, pp. 93–104.
[5] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: image segmentation using expectation–maximization and its application to image querying, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1026–1038.
[6] S.C. Chapra, R.P. Canale, Numerical methods for engineers, fifth ed., McGraw-Hill, New York, 2006.
[7] L. Duan, Y. Chen, W. Gao, Learning semantic cluster for image retrieval using association rule hypergraph partitioning, in: Proceedings 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, pp. 1581–1585.
[8] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.
[9] S. Even, Graph Algorithms, Computer Science Press, 1979.
[10] J.H. Friedman, J.L. Bentley, R.A. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Transactions on Mathematical Software 3 (3) (1977) 209–226.
[11] R.E. Gomory, T.C. Hu, Multi-terminal network flows, SIAM 9 (1961) 551–570.
[12] Z. Gong, L. Hou U, C.W. Cheang, Web image semantic clustering, in: Proceedings of ODBASE, 2005, pp. 1416–1431.
[13] S. Guha, R. Rostogi, K. Shim, A robust clustering algorithm for categorical attributes, Information Systems 25 (2000) 345–366.
[14] E. H. Han, G. Karypis, V. Kumar, B. Mobasher, Clustering based on association rule hypergraphs, in: Proceedings of the Workshop on Research Issues on Data Mining and Knowledge Discovery, Tucson, Arizona, USA, 1997.
[15] V.J. Hodge, J. Austin, A Survey of Outlier Detection Methodologies, Kluwer Academic Publishers, Dordrecht, 2004.
[16] T. Johnson, I. Kwok, R. Ng, Fast computation of 2-dimensional depth contours, in: Proceedings of New York, NY, USA, 1998, pp. 224–228. Q2
[17] E. Knorr, R. Ng, A unified notion of outliers: properties and computation, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, 1997, pp. 219–222.
[18] E.M. Knorr, R.T. Ng, Algorithms for mining distance-based outliers in large datasets, in: Proceedings of the 24th VLDB, 1998, pp. 392–403.
[19] E.M. Knorr, R.T. Ng, Finding intentional knowledge of distance-based outliers, in: Proceedings of the 25th VLDB, 1999, pp. 211–222.
[20] Y. Liu, X. Chen, C. Zhang, A. Sprague, An interactive region-based image clustering and retrieval platform, in: Proceedings of the IEEE International Conference on Multimedia & Expo (ICME 2006), Toronto, Ontario, Canada, 2006.
[21] Y. Liu, A.P. Sprague, Outlier detection and evaluation by network flow, in: Proceedings of the International Conference on Machine Learning and Applications, 2004.
[22] O. Maron, T. Lozano-Perez, A framework for multiple instance learning, Advances in Natural Information Processing System 10 (1998).
[23] Y. Rui, T.S. Huang, S. Mehrotra, Content-based image retrieval with relevance feedback in mars, in: Proceedings of the International Conference on Image Processing, 1997, pp. 815–818.
[24] G. Sheikholeslami, W. Chang, A. Zhang, Semquery: semantic clustering and querying on heterogeneous features for visual data, IEEE Transactions on Knowledge and Data Engineering 14 (5) (2002) 988–1002.
[25] X. Yin, M. Li, L. Zhang, H.J. Zhang, Semantic image clustering using relevance feedback, in: Proceedings of the International Symposium on Circuits and Systems (ISCAS), 2003.
[26] C. Zhang, X. Chen, M. Chen, S.-C. Chen, M.-L. Shyu, A multiple instance learning approach for content based image retrieval using one-class support vector machine, in: Proceedings of the IEEE International Conference on Multimedia & Expo (ICME), Amsterdam, The Netherlands, 2005, pp. 1142–1145.
[27] F. Jing, C. Wang, Y. Yao, K. Deng, L. Zhang, W.-Y. Ma, IGroup: a web image search engine with semantic clustering of search results, in: Proceedings of ACM MM Demo, 2006.
[28] Y. Dai, D. Cai, Image clustering using semantic tolerance relation model, in: Proceedings on European Internet and Multimedia Systems and Applications, 2007.
[29] Z. Hai, Retrieve images by understanding semantic links and clustering image fragments, Journal of Systems and Software 73 (2004) 455–466.
[30] R.E. Patino-Escarcina, J.A.F. Costa, The semantic clustering of images and its relation with low level color features, in: Proceedings of International Conference on Semantic Computing, 2008, pp. 74–79.

UNCORRECTED PROOF