Clustering Approach to Content Based Image Retrieval

P J Dutta[1], D K Bhattacharyya[1], J K Kalita[2], M Dutta[1]
[1]*Department of Comp. Sc & IT, Tezpur University, Napaam, India*
[1]*{pjd,dkb,malay}@tezu.ernet.in*
[2]*Dept of Comp. Sc, University of Colorado, Colorado Spring, USA*
[2]*Kalita@pikespeak.uccs.edu*

Abstract

This paper presents an efficient spatial indexing technique based on Silhouette moments that makes the index robust subject to the three basic transformations for CBIR. Spatial index is generated based upon a fast and robust clustering technique, which can recognize color clusters of any shape. The new clustering technique has been found to be efficient in terms of time complexity and cluster quality than many of its counterparts. A matching engine has been devised to retrieve images from the image database, which has the capacity for global and regional similarity search.

1. Introduction

Any colored image can be viewed as a distribution of colored pixels in a 2-D plane. This distribution of colored pixels forms *color clusters* of arbitrary shape within the image. A color cluster can be viewed as a data set having a color and a position. Moreover, an image is usually formed by a number of constituent objects. Each such object can be characterized by a single color cluster or a combination of several color clusters. Figure 1 depicts an image having four color clusters marked by cluster numbers 1, 2, 3 and 4. Also there are five objects viz. a tree constituted of cluster numbers 1 and 2, the background constituted by cluster number 3, the earth constituted by cluster number 4, the leaf constituted by cluster number 1 and the trunk of the tree constituted by cluster number 2. The similarity search may be *global* or *regional*. In global search, all color clusters present in the query image are useful. In regional or object level search, the selected object formed by the color clusters of the query image is of particular help.

In our approach, we have utilized the power of BOO-Clustering algorithm [8] and GDBSCAN [11] to extract the color clusters of the image. Once the color clusters are determined, the *objects* are formed by selecting one or a few color clusters of the image in an interactive manner. Second order *rotation, translation, uniform* and *non-uniform scaling silhouette moments* are used for generation of indices separately for color clusters produced by both the clustering algorithms and the objects generated in an interactive manner and are finally stored in the cluster object index database respectively.

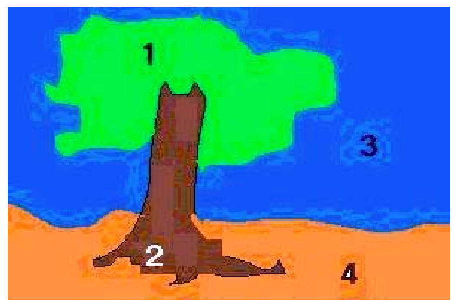

Figure 1. An example image having 4 clusters

2. Related Works

There are several techniques proposed to integrate spatial information in an image. Stricker and Dimai [1] have used the first three moments, Pass and Zabih [3] have used color coherent vector, Rao *et al.* [2] have used annular, angular and hybrid histograms for indexing spatial distribution density of colors. Stricker *et at.* [9] and Mandal *et al.*[10] have employed cumulative histograms and Legendre histogram moments respectively for CBIR. George *et al.* [6] have used chromaticity moments for indexing spatial distribution density of colors. This method is based on the concept of *chromaticity diagram* as defined within the CIE xyY color space. In particular, each pixel in a given image yields a pair of (x,y) chromaticity values, thus forming a set of chromaticities for the entire image. This set of (x,y) pairs is called the *chromaticity set* of the image, while the corresponding *xy* Euclidean space is called the *chromaticity space*. In addition, more than one pixel in a given image may yield the same chromaticity pair of values (x,y), thus leading to the formation of a 2D distribution (histogram) over the chromaticity space.

Proceedings of the Geometric Modeling and Imaging— New Trends (GMAI'06)
0-7695-2604-7/06 $20.00 © 2006 IEEE

This paper presents a new *silhouette moment* based *spatial color indexing* scheme.

3. The Spatial Indexing Scheme

Our indexing scheme works in *four* steps. In *step 1*, it accepts the input query image and generates all the possible color clusters of the image. All of these color clusters of the input image may not be of interest for index generation. Hence color clusters and objects formed by the color clusters which are of importance for indexing purpose are identified in *step 2* in an interactive manner. In *step 3,* indices are generated for the identified color clusters and objects of interest. In *step 4*, the query results are given based on inferences made by a matching engine. Next, we describe each of the modules present in the spatial indexing scheme, in brief.

3.1. The BOO-Clustering technique

Figure 2 depicts an image where each block represents a pixel. Here we have not shown a color in each block but some blocks are marked as classified (C), some are marked as unclassified (U), some are marked as pixels from a template (TC). One block is marked as target pixel (t). The process of clustering starts from the position (0,0) of the image .and successively proceeds till the last pixel is classified. After completion of the clustering process, each pixel is assigned a *Cluster Id.* The concept behind BOO-Clustering is to assign a Cluster Id to a target pixel (t) is that it searches in the template (T) of the target pixel for similar colored pixels. The template (T) of the target pixel (t), is constituted of a set of pixels (TC), which are already classified i.e., pixels which have already been assigned a Cluster Id. Three cases may arise after the search operation. First, not a single pixel is found which is of the same color as that of the target pixel. In that case, a new Cluster Id is assigned to the target pixel (t). It implies that there is not a single cluster having the same color as that of the target pixel nearby. Second, if some or all pixels have the same color as that of the target pixel and those pixels have a common Cluster Id, then it assigns that common Cluster Id to the target pixel (t). This implies that there is a single cluster having the same color as that of the target pixel nearby. Third, if some or all pixels have the same color as that of the target pixel and bear different Cluster Ids, it implies that there is more than one cluster having the same color as that of the target pixel. But, as they are appearing in the same template of the target pixel, they should belong to the same cluster. Hence, in such a situation the algorithm assigns any one of the found Cluster Ids to the target pixel and merges those found color clusters of the template having different Cluster Ids to one cluster. While merging the algorithm traverse back re-clustering successively pixel by pixel till it reaches the first pixel or does not find any one of those clusters in the template of the backtracking pixels. After assigning a Cluster Id (new or already assigned) to the target pixel, the algorithm takes up the next pixel as the target pixel and starts the process of clustering. This algorithm is of order $\theta(N^2)$. To make it linear, instead of backtracking any one of the Cluster Id's is assigned to the target pixel and a link of Cluster Ids is maintained as a list for those Cluster Ids whose pixel values are same as that of the target pixel but having different Cluster Ids. Once the clustering is over, the algorithm scans the list to reassign a common Cluster Id to those clusters which are connected. Hence the time complexity of the algorithm is $\theta(N \times 2(d^2 + d))$. Here d is very small as compared to N and hence can be neglected. Also the time taken for scanning the list is smaller compared to N. Thus, the complexity of BOO-Clustering is $\theta(N)$ where d is a number that defines the template.

If the d value is large then $2(d^2 + d)$ part of the complexity $\theta(N \times 2(d^2 + d))$ can not be neglected and then the complexity of the algorithm does no longer remains linear. To avoid this problem, the algorithm has been modified if the third case arises.

3.2. Modified BOO-Clustering algorithm

While clustering of the pixel values of the image if the third case arises the algorithm assigns the smallest Cluster Id of the found Cluster Ids to the target pixel and assigns that smallest Cluster Id to those pixels having the same color as that of the target pixel and bearing different Cluster Ids. Thus this modified algorithm eliminates the links of Cluster Ids that has to be maintained if the third case arises. Thus this modification makes the BOO-Clustering algorithm more linear.

3.3. Identification of color clusters and object of interest in an image

An interactive method is the only effective way for identification of proper color clusters and objects of interest of an image because the color clusters and objects depend on the distribution of color within the image (controlled by the *d* parameter) and quantization of the color of the pixels (controlled by the *quantization* parameter). Both these parameters vary from image to image and a global method can not be applied for all kinds of images.

Hence, during color cluster generation, two parameters are given as input to the BOO-Clustering algorithm. They are *quantization* over the *hue* component of the color of the image and *d* the size of the neighborhood template. Where as GDBSCAN takes three parameters: *d*, *quantization* parameter and *min_points*.

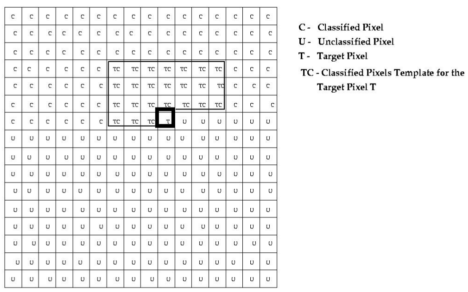

Figure 2. A sample template shown in an example image

3.4. The spatial cluster index and object index

A color cluster or an object of interest generated consists of a set of pixels denoted by S_q having an average color C_p that defines an area in the image having sharp or diffused boundary. Hence these image segments can be thought of as silhouette images. Silhouette images are binary images whose intensity value takes only two values viz., 0 and 1 [7]. In our case, a pixel that forms a cluster or object is assigned a value 1 i.e. $f(x,y) = 1$ and other points are assigned a value 0 i.e. $f(x,y) = 0$.

For a silhouette image, the point (x_0, y_0) gives the geometric center of the image region. When moments are calculated by shifting the origin of the reference system to the intensity centroid of the image, they are called central moments. This transformation makes the moment computation independent of the position of the image reference system; hence translation invariant. The central moment [7] can be defined as

$$\mu_{pq} = \frac{1}{m \times n} \sum_{x}^{m} \sum_{y}^{n} (x - x_0)^p (y - y_0)^q f(x,y) \qquad (1)$$

where $p,q = 0,1,2,3\ldots$ is the order of the moment. To make the index scale invariant, let us scale the image by a uniform factor *k*, then we have [7]

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{(p+q+2)/2}} \qquad (2)$$

where η_{pq} is translation and scale invariant central moment. If an image is transformed with unequal scale factors k_1, k_2 along *x* and *y* axes respectively, then we have [7]

$$\eta_{pq} = \frac{(\mu_{00})^{(p+q+2)/2}}{(\mu_{20})^{(p+1)/2} (\mu_{02})^{(q+1)/2}} \mu_{pq} \qquad (3)$$

where η_{pq} is translation and aspect ratio invariant central moment. Finally the translation, aspect ratio, scale and rotation invariant moments [7] of second order are given by

$$\varphi_1 = \eta_{20} + \eta_{02} \qquad (4)$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}{}^2 \qquad (5)$$

$$\varphi_3 = \eta_{20}\eta_{02} - \eta_{11}{}^2 \qquad (6)$$

In our experiment, equations (4), (5), and (6) are utilized for index generation of the color clusters and the objects generated. Hence the indices for the color cluster or objects of interest consists of four parameters viz., <*average color of the clusters,* φ_1, φ_2, φ_3> termed as *spatial cluster index* and <*average color of the objects,* φ_1, φ_2, φ_3> is termed as *spatial object index* respectively. Hence, a search on spatial cluster index of all the clusters present in the image represents a global similarity search of the image. A search on spatial object index, or one or a few spatial cluster indices of the image represent regional similarity search of the image. The spatial cluster indices and spatial object indices are stored in a spatial cluster object tree (Figure 3).

3.5. Database organization

The output of the BOO-Clustering algorithm is a set of color clusters from which the objects of interest are separated out from the image in an interactive manner. The spatial cluster indices and object indices are stored in a *spatial cluster object tree.* The spatial cluster object tree is a variant of B-tree (Figure 3). The root node termed as Cluster_Object Root Node, of the tree points to *k* independent tree structures, where *k* is the number of parameters in the index. If a new parameter is to be accommodated (i.e., for a *k*+1 dimensional index), the root node has to be updated by insertion of a new pointer and accordingly an associated tree structure will have to be generated. Each of the parameter trees will maintain the parameter key value (i.e., say, *Cluster/Object Color* value, φ_1 value, φ_2 value, φ_3 value), along with a pointer to a list of Image IDs (i.e., PIDs).

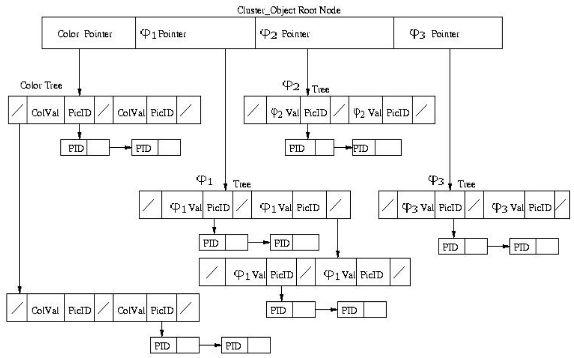

Figure 3. Spatial cluster_object tree

3.6. Matching engine

The matching engine has the facility for searching images in the image database based on the spatial cluster and object indices using the cluster_object tree (Figure 3). For global search, the matching engine fetches all those images from the image database whose cluster indices matches with all the cluster indices of the query image. For example in Figure 1 there are four clusters (cluster 1, 2, 3 and 4) are present in the query image. For each cluster, the query image will have one cluster index. Let these cluster indices be denoted by <$color^1$, φ_1^1, φ_2^1, φ_3^1>, <$color^2$, φ_1^2, φ_2^2, φ_3^2>, <$color^3$, φ_1^3, φ_2^3, φ_3^3> and <$color^4$, φ_1^4, φ_2^4, φ_3^4>. For global search, the matching engine searches for all those images that have all the four cluster indices, as that of the query image, present in image database.

For object level search, at first the objects present in the query image are determined from the set of clusters present in the query image. The objects may consist of either a single color cluster or a combination of color clusters. For example, in Figure 1 the query image has four color clusters (1, 2, 3 and 4). Cluster 1 and 2 forms the object "tree"; color cluster 3 is the background object; color cluster 4 is the soil object. Also color cluster 1 is the leaf object of the tree and color cluster 2 is the trunk object of the tree. Thus there are all total five objects in the query image as depicted in Figure 1 and hence there will be five object indices in the query image. Let these object indices be denoted by <Leaf_object^1, φ_1^1, φ_2^1, φ_3^1>, <Trunk_ object^2, φ_1^2, φ_2^2, φ_3^2>, <Background_ object^3, φ_1^3, φ_2^3, φ_3^3>, <Soil_ object^4, φ_1^4, φ_2^4, φ_3^4> and <Tree_ object^5, φ_1^5, φ_2^5, φ_3^5>. For object level search, one of the search may be "Find all the images from the image database that have the object *tree*". For this query the matching engine will fetch all the images from the image database that have the tree object. That is the index <Tree_ object5, φ_1^5, φ_2^5, φ_3^5> of the query image is matched with the object indices of the images in the image database and all those matching images are fetched that matches at least one of the object index of the stored images.

More over the matching engine has the facility for searching in the image database for color clusters and objects using both the conjunction 'AND' and disjunction 'OR'. For example, the query image as depicted in Figure 1 may have a query that "Find all images that have color cluster (1 AND 2) OR 3; i.e. find all those images which have either both the colors 1 and 2 or color 3. For this query, the indices <$color^1$, φ_1^1, φ_2^1, φ_3^1>, <$color^2$, φ_1^2, φ_2^2, φ_3^2> and <$color^3$, φ_1^3, φ_2^3, φ_3^3> of the query image will be utilized and the search will initiate from color cluster 1. At first it will search in the color tree that matches $color^1$ and a Picture Id list will be fetched. Next it will take up φ_1^1 value and search will be done in the φ_1 tree and will fetch another Picture Id list. Similarly, the matching engine will fetch another two Picture Id lists for φ_2^1 and φ_3^1 value by searching in the φ_2 and φ_3 tree respectively. After that all the four Picture Id lists will be concatenated to form a single Picture Id List (PID^1) which matches the first color index <$color^1$, φ_1^1, φ_2^1, φ_3^1> of the query. Thus another two Picture Id lists (PID^2 and PID^3) for the indices <$color^2$, φ_1^2, φ_2^2, φ_3^2> and <$color^3$, φ_1^3, φ_2^3, φ_3^3> will be generated. Now, according to the query the Picture Id lists PID^1, PID^2 and PID^3 will be combined together according to (PID^1 AND PID^2) OR PID^3 to produce the final set of Picture Id list that corresponds to the images in the image database which satisfy the given query. This type of query can also be applied to object level search and combination of object and cluster level search.

3.7. Efficiency of retrieval

There are two phases of computations involved in querying an image database. First, calculation of index for the query image and second, comparison of the generated index with the stored indices of the images in the image database and subsequently retrieval of the similar images from the image database.

The total retrieval time by chromatic moment [6] method is θ (Q^2 + 2S + N) and that of by BOO-Clustering and GDBSCAN methods are θ (5N + S) and θ (N×logN + 4N + S). Where Q is the total number of histogram bins, N is the total number of pixels present in the image and S is the total number of images in the image database.

Here $Q^2 < 5N$ and $Q^2 < N \times \log N + 4N$ but $2S >> S$. If the number of images in the image database is smaller in size, the Chromatic Moments index has time advantage over the proposed method at the cost of precision and recall. As the number of images in the image database increases in size, the proposed method has a time advantage over chromatic moments and GDBSCAN with a good precision and recall.

4. Experimental Results

To test the technique, we used a downloaded database ((a) Cohn-Kanade Facial Expression Database; http://www.pitt.edu/~jeffcohon, (b) ftp://ftp.eecs. umich.edu. groups /ai/ dberwick/ essbthm.zip and other images) consisting of 5000 real world and synthetic images divided into 100 similar groups such as *facial expressions*, *scenery, animals, cars, flowers* and *space.* Implementation was carried out for chromatic moment based technique along with the proposed technique using the BOO-Clustering and GDBSCAN method in the HSV color space. The proposed method has been implemented in Java. For any input image, indices are generated for the color clusters and objects of interest in an interactive manner. Similarity search is done on clusters of interest or objects of interest using a matching engine. *Cosine* measure is used as the similarity measure for the parameters.

4.1. Performance comparison

In our experiment, the *average precision recall (APR)* [4] was plotted for the best 200 queries out of the calculated 500 queries taken at random over the downloaded database. The proposed method was compared with chromatic moment based technique [6] along with the GDBSCAN method for global and regional search. Based on experimental results (Figure 4), it is found that the performance area for the curves are: 387 (BOO-Clustering method), 321 (GDBSCAN method) and 271 (Chromatic Moment method). Hence, the improvement produced by the proposed method over the GDBSCAN and Chromatic Moment method are: 20.56% and 42.8% respectively for global search. The Proposed method was also tested based on some specific clusters and objects of the query image along with its counterparts. Figure 5 reflects the query results of regional search of the three methods. Based on experimental results, it is found that the performance areas for the curves are: 583 (BOO-Clustering based retrieval), 411 (GDBSCAN based retrieval). Hence the improvement produced by the proposed BOO-Clustering based method over the GDBSCAN is: 41.85% for regional search.

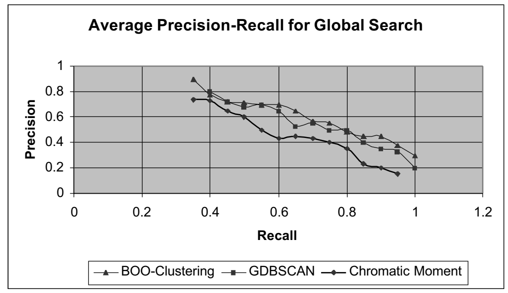

Figure 4. Average precession recall for BOO-Clustering vs GDBSCAN and Chromatic Moment based techniques for global search

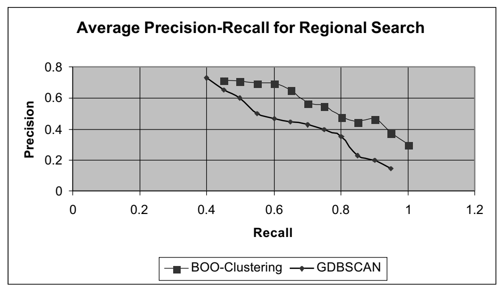

Figure 5. Average precession recall for BOO-Clustering vs GDBSCAN clustering technique for regional search

4.2. Retrieval time comparison of BOO-Clustering vs GDBSCAN and Chromatic moment based methods

In this experiment we have tried to plot the average time taken by the three methods vs the number of images in the image database for global search. In global search, all the clusters generated by the GDBSCAN and BOO-Clustering method are taken into account for the search operation. To test the retrieval time comparison, we have taken 100 images from the image database and the average retrieval time for the 100 images for global search has been plotted for BOO-Clustering, GDBSCAN and Chromatic Moment method for different database size (Figure 6). It can be clearly seen that as the size of the image database increases the BOO-Clustering method has time advantage over the GDBSCAN and Chromatic Moment based methods.

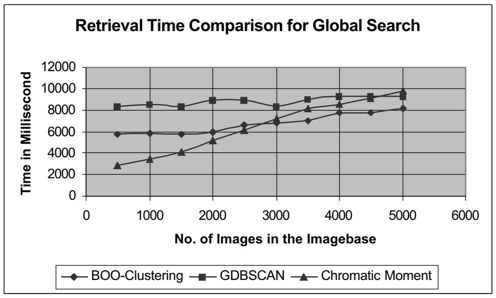

Figure 6. Retrieval time comparison for global search

4.3. Conclusion

A improved content-based indexing scheme has been presented in this paper. The scheme generates a compact, 4-dimensional transformation invariant index for the color clusters and objects of interest produced by a robust data clustering technique of any color image. The indices produced help in global and regional similarity search of images by a robust matching engine. The proposed scheme can be found to be superior in comparison to its counterparts [6].

5. References

[1] M. Stricker and A. Dimai, "Color Indexing with Weak Spatial Constraints" *SPIE Proceedings,* February, 1996 2670: pp. 29-40.

[2] A. Rao, R K. Srihari and Z. Zhang, "Spatial Color Histograms for Content-Based Image Retrieval", *11th IEEE International Conference on Tools with Artificial Intelligence,* Chicago, Illinois, November, 1999.

[3] G. Pass and R. Zabih, "Histogram Refinement for Content-Based Image Retrieval", *IEEE Workshop on Applications of Computer Vision*, December, 1996, pp. 96-102.

[4] R. K. Srihari, Z. F. Zhang, and A. Rao, "Image Background Search: Combining Objects Detection Techniques with Content-Based Image Retrieval (CBIR) Systems", *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '99), in conjunction with CVPR '99*June 1999.

[5] J. Sander, M. Ester, H. P. Kriegel, X. Xu, "Density-Based Clustering Spatial Databases: The Algorithm GDBSCAN and its Applications," http://Citeseer.nj.nec.com.

[6] G. Paschos, I. Radev, N, Prabakar, Image "Content-Based Retrieval Using Chromaticity Moments", *IEEE Transactions on Knowledge and Data Engineering*, September/October 2003, Vol 15, No 5, pp. 1069-1072.

[7] R. Mukundan and K. R. Ramakrishan, *Moment Functions in Image Analysis*. World Scientific 1998. pp. 13-19.

[8] P. J. Dutta, D. Bhattacharryya, J. Kalita, M. Dutta, "Spatial Color Indexing Using Clustering Technique", *The 8th World Multi-Conference on Systemics, Cybernatics and Informatics (SCI' 2004). Vol VI: Image, Acoustics, Signal Processing & Optical Systems, Technologies and Applications*, Orlando, Florida, July 2004, pp. 216-221.

[9] M. Stricker and M. Orengo, "Similarity of Color Images. SPIE Proceedings: Storage and Retrieval for Image and Video Databases III", 1995, Vol 2420, pp. 381-392.

[10] M. K. Mandal, T. Aboulnasr and S. Panchanathan, "Image Indexing Using Moments and Wavelets", *IEEE Transaction on Consume Electronics,* 1996, Vol 42, No 3, pp. 557-565,

[11] J. Sander, M. Ester, H. P. Kriegel, X. Xu, "Density-Based Clustering Spatial Databases: The Algorithm GDBSCAN and its Applications", http://Citeseer.nj.nec.com.

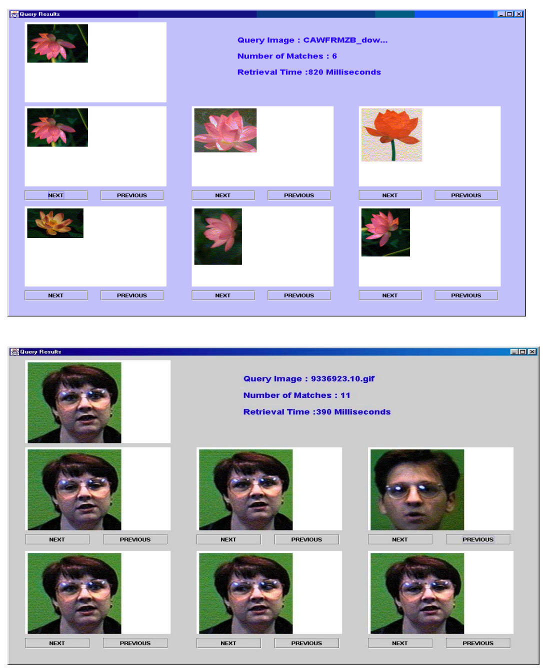

Figure 7. Some retrieval result of the proposed method