Cluster Tree Self Organizing Map for Developing Image Retrieval System

Hamed Shahbazi[1], Mohsen Soryani [2], Nasser Mozayani [2], and Mahmood Fathy [2]

Computer Engineering Department, Iran University of Science and Technology
Narmak, Tehran 16844, Iran
[1] hshahbazi@comp.iust.ac.ir
[2] {soryani, mozayani, mahfathy}@iust.ac.ir

(Paper received on June 18, 2007, accepted on September 1, 2007)

Abstract. In this paper, a novel structure of hierarchical self organizing map called Cluster Tree Self Organizing Map (CT-SOM) is proposed. This structure is a hierarchical representation of the cluster of a data set that will be used for image retrieval. Neural units of CT-SOM are given multi labels and each label hints at one image in database. For each different image's feature, a CT-SOM is formed. These representations include color, texture and shape. Using evidence accumulation we have facilitated automatic combination of responses from multiple CT-SOMs and their hierarchical levels. A new relevance feedback technique is also used based on user's preferences for finding image resemblance in each category. We have performed experiments and tested the proposed approach on an image database constructed from Corel photo gallery.

1 Introduction

In recent years Content-Based Image Retrieval (CBIR) has been a subject of very extensive research field and many projects have been started to research and develop efficient CBIR systems. Despite some breakthroughs made in the field, it is generally understood that the problem is still far from being solved. Some of the popular CBIR systems include QBIC project [1], MIT's Photobook [2], VisualSeek [3], PicSOM [4] and lot more.

Heretofore, variant structure of hierarchical self-organizing map is used to develop indexing structure. For instance tree structured SOM which are introduced in PicSOM, and four-level R-tree SOM [5] which are used for image retrieval. The mentioned indexing structures have a problem of having a large overlapping area among nodes, causing the retrieval process to inspect a large number of image items. Also until relevance feedback modifies the structures remarkably, the approach for combination of the structures' results is weak.

A cluster tree [6] is a hierarchical representation of the cluster of a data set. This index structure organizes the data based on their different level of clustering information from coarse to fine. Here, we develop structures of SOM which represent a cluster tree of data and decrease overlapping area among nodes. These structures are called Cluster Tree Self Organizing Map (CT-SOM). For each visual feature, one CT-

Special Issue in Neural Networks and Associative Memories
Research in Computing Science 28, 2007, pp. 81-90

SOM is developed. Partition of clusters is organized from precise bottom level to coarse top level of CT-SOMs. Following production of partitions in CT-SOMs' levels, evidence accumulation [7] is used to facilitate automatic combination of responses from multiple hierarchical structures. A new relevance feedback technique is also used based on user's preferences for finding image resemblance in each category.

The remainder of this paper is organized as follows. Section 2 describes cluster tree self organizing map. Relevance feedback to refine query is proposed in section 3. Visual content features, is given in section 4 and section 5 gives the performance and experimental results.

2 Cluster Tree Self Organizing Map

The SOM [8] carries out vector quantization and multi-dimensional scaling at the same time. In step index $t = 0,\ 1,\ldots\ t_{\max} - 1$, an input vector $X(t)=[X_1(t),..X_m(t)]^T$ is presented to the network and unit $i(X)$ with synaptic weight vector $W_j(t)=[W_{j1}(t),..W_{jm}(t)]^T$ is selected as the Best Match Unit (BMU), based on the best matching criterion (1).

$$i(X) = \arg\min_j \| X(t) - W_j(t) \|,\quad j = 1, 2, \ldots, l \tag{1}$$

Where l is the number of units and $\| \|$ represents Euclidian distance. Consequently, the weight vectors are updated according to (2).

$$W_j(t+1) = W_j(t) + \eta(t) h_{j,i(x)}(t)(X(t) - W_j(t)) \tag{2}$$

Where $h_{j,i(x)}(t)$ is topological neighborhood function which the typical choice of it is Gaussian function (3).

$$h_{j,i(x)}(t) = \exp(-\frac{d_{j,i}^2}{2\sigma^2(t)}) \tag{3}$$

Where $d_{j,i}^2 = \| r_j - r_i \|^2$ and r_j defines the position of excited neuron j, and r_i defines the discrete position of winning neuron i. In order to construct the CT-SOM, below steps are done:

Step1) The size of first level is $r_1 * c_1$ and initial standard deviation of neighborhood function is σ_0. This level, after training is fixed and each neural unit on it is given labels from the database image nearest to it. In other words CT-SOM's neural units are given multi labels and each label hints at one image in database. We called the neural unit and its labels as Unit Cluster (UC).

Step2) The map is divided into squares with size $\Lambda \times \Lambda$ (Λ is odd) and the k-mean algorithm is performed on map as follow:

- Begin with $K = (r_1 \times c_1)/(\Lambda \times \Lambda)$ clusters which their centroids are initialized with the centers of mentioned squares. The cluster prototype matrix is shown with $W = [w_1, ..., w_K]$.
- Assign each unit of map to the nearest cluster C_l .i.e.

$$U_j \in C_l, if \left\| U_j - w_l \right\| < \left\| U_j - w_i \right\| \quad (4)$$
$$for \ \ j = 1, ..., (r_1 \times c_1), i = 1, ..., K \ \ and \ \ i \neq l.$$

- Recalculate the cluster prototype matrix based on the current partition.
- Repeat steps 2 and 3 until there is no change for each cluster.

We called these clusters as Map Cluster (MC) which consists of some UCs. Fig.1 shows two MCs for Λ=3 .

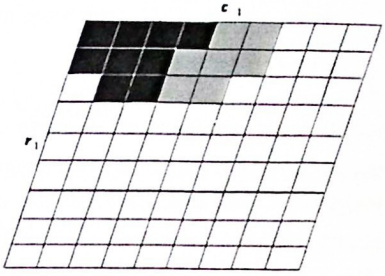

Fig. 1. Two Map Clusters (MCs) are shown. Each MC contains some Unit Clusters (UCs)

Step3) Every later level is developed on previous level which its map's size is number of MCs on previous level. This means that every unit on later level represents one MC on previous level.

Each later level, after construction is trained and its UCs and MCs are determined. In training of later *jth* level, the BMU is determined in this way that, the input object is presented to the first level and the BMU and therefore the MC that this unit belongs to it is determined. Then the BMU of the next level is unit that represents the determined MC on previous level. This iteration goes on until reach *jth* level.

Step3 is iterated until efficient partitions with desirable number of clusters (size of last level) are gained. In the end of this step, cluster tree is produced which is a hierarchical representations of the cluster of a data set. This hierarchical structure organizes the data based on their different level of clustering information from coarse to fine, providing an index structure of data. For each visual feature we construct a CT-SOM separately. Since various feature classes are not necessarily linearly-related so we consider each map as a partition of UCs and use evidence accumulation to combine them. Evidence accumulation gives us desired co-association matrix that indicates the degree of resemblance between images.

Assume that N is the number of database images, B is the number of partitions and the final partitions of UCs which acquired from CT-SOMs' hierarchical levels are $\mathrm{P} = \{P^1, P^2, ...P^B\}$. Since processing of N^*N proximity matrix has computational complexity for large value of N, so we pre-compute a N^*q matrix which stores the indices of the q nearest neighbors for each of the N images. In our experiment, we set q equal to 15. The nearest-neighbor matrix can be computed as a preprocessing step by using branch and bound algorithm [9].
Co-association matrix is calculated as follow:

1. C = co-association matrix with dimension N^*q is initialized to a null matrix.
2. For each data partition $p^l \in \mathrm{P}$, do:
   2.1. Update the co-association matrix as:
   For each image pair (i, j) in the qth neighbor list which belongs to the kth UC in p^l, set:

$$C(i,j) = C(i,j) + \frac{\gamma_{l,k}}{B} \tag{5}$$

Where $\gamma_{l,k} \in [0,1]$ and its value is minimum for coarse clusters (top levels) and maximum for precise one (low levels). It means that two images in precise clusters are more similar than two images in coarse one. Each UC has its $\gamma_{l,k}$ value and gives special amount of similarity between its members. We use relevance feedback mechanism to modify the value of $\gamma_{l,k}$ and improve queries result. $\gamma_{l,k}$ may be dependent to type of visual features.

In actual implementation to retrieve image, CT-SOMs are browsed from top to bottom and in each level, the BMU and consequently the UC which is bound to it, are determined. The labels of this UC together with 10 labels from co-association matrix with highest resemblance value to prior selected labels are selected. In browsing to lower levels, the search space for BMU is restricted to the MC which BMU on top previous level represents it.

Finally for each CT-SOM, one set of labels is acquired which is shown with $S = \{S_1, S_2, ..., S_\Delta\}$. Where Δ is the number of CT-SOMs. Images that are presented to user, to get his/her preferences, are intersection between these sets:

$$R = \bigcap_{i=1}^{\Delta} C_i \tag{6}$$

3 Refining Queries Using Relevance Feedback

The System tries to learn the user's preferences from the interaction with him/her and then satisfies own responses accordingly. We use a relevance feedback approach in which the results of multiple CT-SOMs are combined automatically by using the implicit information from the user's responses during the query session. This can be implemented simply by marking each kth UC on lth map with $\gamma_{l,k} \in [0,1]$ as similarity weight. Initial values of $\gamma_{l,k}$ is minimum for coarse clusters (top levels) and maximum for precise (low levels).

The user's preferences for each images is either relevant or irrelevant. For each image pair (i, j) in the images which are shown to user, if they are relevant then the $\gamma_{l,k}$ of UCs which this pair are belong to it, is increased and vice versa. As seen in (5), the modification of $\gamma_{l,k}$ affects the value of co-association matrix which will be recomputed after some interactions.

4 Visual Content Features

Feature selection is not restricted and every feature or description of it can be added. In our experiment fuzzy color histogram, entropy and shape histogram are selected which are described in later subsection.

4.1 Fuzzy Color Histogram

The Fuzzy Color Histogram (FCH) [10] of image I can be expressed as $F(I) = [f_1, f_2, \ldots f_n]$, where

$$f_i = \sum_{j=1}^{N} \mu_{ij} P_j = \frac{1}{N} \sum_{j=1}^{N} \mu_{ij} \tag{7}$$

μ_{ij} is the membership value of the jth pixel in the ith color bin and p_j is the probability of jth pixel selected from image I. Let M (8) is the membership matrix and m_{ij} is the membership value of the jth fine color bin distributing to the ith coarse color bin:

$$M = [m_{ij}]_{n^* n}. \tag{8}$$

The FCH of an image can be directly computed as follows:

$$F_{n^{*}1} = M_{n^{*}n'} H_{n'^{*}1} \tag{9}$$

Where membership matrix M is pre-computed only once and can be used to generate FCH for each database image. M is computed as follow:

- Fine uniform quantization in RGB color space is performed by mapping all pixel colors to n' histogram bins. Then, the n' colors are transformed from RGB to CIELAB color space.
- Using FCM clustering technique [11], these colors in CIELAB color space is classified to clusters, which each cluster representing an FCH bin.

The FCM minimizes an objective function J_m, which is the weighted sum of squared errors within each group, and is defined as follows:

$$J_m(U,V;X) = \sum_{k=1}^{n}\sum_{j=1}^{c} u_{ik}^{m} \left\| x_k - v_i \right\|_A^2 \tag{10}$$

$$1 < m < \infty$$

Where, $V = [v_1, v_2, ...v_c]^T$ is a vector of unknown cluster prototypes. The value of u_{ik} represents the membership of the data Point x_k from the set $X = \{x_1, x_2, ...x_n\}$ with respect to the ith cluster. The inner product defined by a norm matrix A defines a measurement of similarity between a data point and the cluster prototypes, respectively. The fuzzy clustering result of FCM algorithm is represented by matrix $U = [u_{ik}]_{n^{*}n'}$. u_{ik} is referred to as the grade of membership of color x_k with respect to cluster center v_i. Thus, the obtained matrix $U_{n^{*}n'}$ can be viewed as the desired membership matrix $M_{n^{*}n'}$ for computing FCH, i.e. $M_{n^{*}n'} = U_{n^{*}n'}$. Moreover, the weighting exponent m in FCM algorithm controls the extent of membership shared among the fuzzy clusters.

4.2 Entropy Histogram

The co-occurrence matrix [12] is a two-dimensional histogram which estimates the pair-wise statistics of gray level. The $(i, j)th$ element of the co-occurrence matrix represents the estimated probability that gray level i co-occurs with gray level j at a specified displacement d and angle θ. Entropy histogram is acquired as follow:

- Conversion of color image to gray image.
- Dividing image into 2*2, 4*4, 8*8, and 16*16 rectangular regions as in color case.

- Obtaining co-occurrence matrix of four (horizontal 0^{o}, vertical 90^{o} and two diagonal $45^{o}, 135^{o}$) orientation in region and normalize entries of four matrixes to [0, 1], by dividing each entry by total number of pixels.
- Extracting average entropy value from four matrixes.

$$e = \frac{-\sum_k \sum_i \sum_j p(i,j)\log(p(i,j))}{4}, k = 1,2,3,4 \tag{11}$$

- Constructing entropy histogram of regions' entropy.

4.3 Shape Histogram

This feature describes the distribution of edge directions in various parts of the image and thus reveals the shape in a low-level statistical manner [13]. It is calculated in five separate zones of the image. The first zone is formed by extracting from the center of the image, a circular zone whose size is approximately one-fifth of the area of the image. Then the remaining area is divided into four zones with two diagonal lines. Shape histogram feature is based on the histogram of the eight quantized directions of edges in the image. When the histograms are separately formed in the same five zones, as before, an 8* 5 = 40 dimensional feature vector is obtained.

5 Performance and Experimental Result

Corel gallery product [14] contains 59995 photographs and artificial images with a very wide variety of subjects. Image collection which we used in our experiment is a set from the Corel Gallery. First, we consider 48 semantic concepts (Classes) and then select 12000 images from Corel gallery and give 48 membership values to each of them. Each membership value determines the belonging degree of image to one concept. The evaluation of performance for retrieval system can be mathematically formulated as follow: Suppose that the size of database is N and the number of semantic concept is C. Membership value of ith images to semantic concept can be expressed by $H_i = \{h_{i1}, h_{i2}, ..., h_{ic}\}$, and for all images in database it is expressed by $M = \{H_1, H_2, ..., H_N\}$. Let in time t, Queries on image with membership $Q_t = \{q_{t1}, q_{t2}, ..., q_{tc}\}$ is given to system and $K(t)$ images are retrieved as a query result. Recall is defined as:

$$\text{Re}call = \frac{Number\ of\ images\ retrieved\ and\ relevant}{Total\ number\ of\ relevant\ images\ in\ the\ database} \tag{12}$$

Recall is expressed with $R(t)$ for time t, and Precision is defined as:

$$\Pr ecision = \frac{Number\ of\ images\ retrieved\ and\ relevant}{Total\ number\ of\ retrieved\ images} \tag{13}$$

$P(t)$ is used to determine Precision for time t. We have chosen to show the evolution of precision as a function of Recall. Using above assumption, $R(t)$ and $P(t)$ is defined as follow:

$$R(t) = \frac{\sum_{i=1}^{K(t)} \frac{1}{1 + \sum_{j=1}^{c} (\|q_{ij} - h_{ij}\|)^2}}{\sum_{i=1}^{N} \frac{1}{1 + \sum_{j=1}^{c} (\|q_{ij} - h_{ij}\|)^2}} \tag{14}$$

$$P(t) = \frac{\sum_{i=1}^{K(t)} \frac{1}{1 + \sum_{j=1}^{c} (\|q_{ij} - h_{ij}\|)^2}}{K(t)} \tag{15}$$

The intermediate values of $P(t)$, first, display the initial accuracy of the system and then, show how RF mechanism is able to adapt the class. For our experiment, average precision, as a function of average recall is changed as indicate in Fig 2.

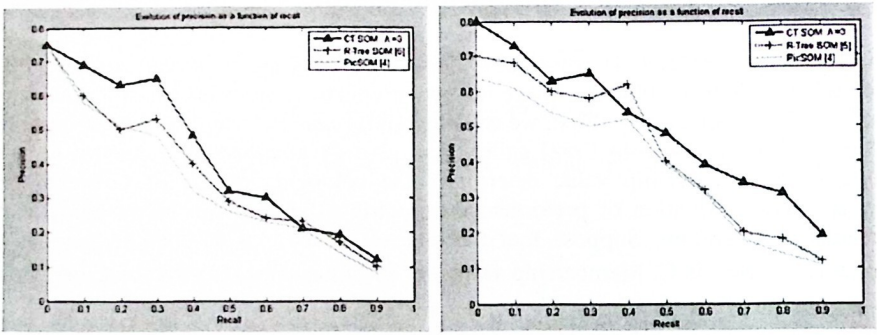

Fig. 2. The evolution of precision as a function of recall. (a) Building query. (b) Tiger query.

Some of Exemplar Queries that is given to the system are shown in Fig. 3 and 4.

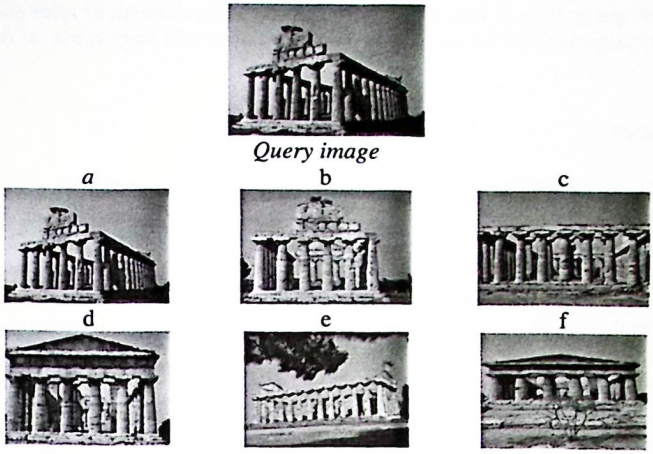

Fig. 3. Retrieval result for query of image with Building semantic.

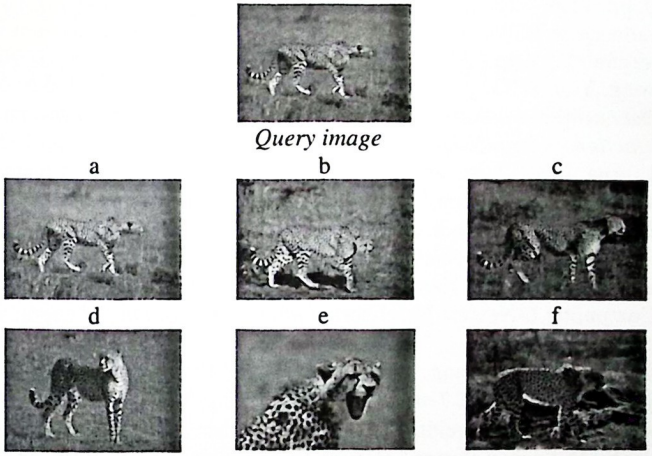

Fig. 4. Retrieval result for query of image with Tigers semantic.

6 Conclusion

The CT-SOM that is presented in this paper is very useful for large image data sets. This system has three advantages: First, it used the SOM with multi labeled neural units and thus organizes images into a hierarchical structure without overlapping area between nodes. Second, hierarchical maps are from coarse top level to precise bottom

level so the query time is largely reduced. Third the mechanism of relevance feedback used in the proposed system improves the performance of image retrieval drastically.

References

1. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Pettovic, D., Steele, D., and Anker, P. "Querying by image and video content: The QBIC system," *IEEE Trans. Computers 25*, 1995, pp.23–32.
2. Pentland, A., Picard, R.W., and Sclaroff, S. "Photobook: tools for content-based manipulation of image databases. In: Storage and Retrieval for Image and Video Databases," II. In: *SPIE Proceedings Series*, Vol. 2185. San Jose, CA, USA, 1994.
3. Smith, J. R., and Chang, S. F. "VisualSeek: a fully automated content-based image query system," *Proc. ACM Multimedia*, 1996, pp. 87–98.
4. Laaksonen, J., Koskela, M., Laakso, S., and Oja, E. "PicSOM: content-based image retrieval with self-organizing maps," *Elsevier, Pattern Recog.* Lett.21, 2000, pp. 1199–1207.
5. Subramanyam Rallabandi, V. P., Sett, S.K. "Image retrieval system using R-tree self-organizing map," *Elsevier, Data & Knowledge Engineering*, 2006.
6. Dantong, Y., and Zang, A. "Cluster Tree: Integration of cluster representation and nearest neighbor search for large data base in high dimensions," *IEEE Transaction on knowledge and Data Eng*, Vol.15, No.5, 2003, pp.1316-1337.
7. Fred, Ana L.N., Jain, and Anil K. "Combining Multiple Clusterings Using Evidence Accumulation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, No.6, 2005.
8. Kohonen, T. *"Self-Organizing Maps,"* third ed., Springer, New York, 2001.
9. Kamgar-Parsi, B., and Kanal, L.N. "An Improved Branch and Bound Algorithm for Computing k-Nearest Neighbors. *Pattern Recognition ," Letters*, vol. 1, 1985, pp.195-205.
10. Ju, Han., and Kai-Kuang, Ma. "Fuzzy Color Histogram and Its Use in Color Image Retrieval," *IEEE Trans. Image Processing*, Vol. 11, No. 8, 2002.
11. Rezaee, M. R., LeLieveldt, B. P. F., and Reiber, J. H. C. "new cluster validity index for the fuzzy c-means," *Pattern Recognition,* vol. 19, 1998, pp.237–246.
12. Haralick, R. M., Shanmugam, K., and Dinstein, I. "Texture features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(6)*, 1973, pp.610–621.
13. Brandt, S., and Laaksonen, J. E. Oja. "Statistical shape features in content-based image retrieval," *In: Proceedings of 15th International Conference on Pattern Recognition*, Barcelona, Spain, Vol. 2, 2000, pp.1066±1069.
14. Gunther, N.J., Beretta, and G. "A benchmark for image retrieval using distributed system over the internet," BIRDS-I HP Labs, 2000 Available from: <www.hpl.hp.com/techreports/2000/HPL-2000-162.html>.

Feature Extraction and Dimensionality Reduction