Shape Indexing Using Relational Vectors and Neural Networks

P. N. Suganathan
Block S2, School of Electrical and Electronic Engineering
Nanyang Technological University, Republic of Singapore 639798
Email: epnsugan@ntu.edu.sg

ABSTRACT: In this paper, we propose a novel approach to generating topology preserving mapping of structural shapes using the self-organising maps (SOM). The structural information of the geometrical shapes is captured by the relational vectors. These relational attribute vectors are quantised using an SOM. Using this quantisation SOM, a histogram is generated for every shape. These histograms are treated as inputs to train another SOM which yields a topology preserving mapping of the geometric shapes. By appropriately choosing the relational vectors, it is possible to generate the mapping invariant to some chosen transformations such as rotation, translation, scale, affine or perspective. These SOMs may be organised into a tree-structure so that during the application phase the histogram of the query shape and the shapes most similar to the query shape can be retrieved efficiently.

1 Introduction

Due to the recent advances in storage, communications, image compression and internet technologies, multimedia information has become more popular. Consequently, more and more multimedia documents containing video clips, images and audio are being generated in diverse application areas including education, medicine, entertainment, sport, remote sensing and online information services. With this explosive growth in the volume of multimedia information archives, the effective management of these multimedia archives for efficient browsing and retrieval of desired information is of paramount importance. In recent years, several intelligent techniques have been developed to perform these tasks, as evidenced by the recent publications [4,14,16] on the subject of content-based image retrieval. The most commonly used properties of images for visual content-based retrieval are colour [15], texture [10], shape [2,5,13,16], spatial relationships between various properties [3] or a combination of these properties [4,6,11,14]. Although several methods have been proposed to retrieve images, the most popular approach for indexing into image databases has been the histogram indexing using the above listed properties [15].

In this paper, we employ the SOM to organise structural shapes in a topographical manner for efficient shape retrieval. In the past, the SOM has been applied to solve several complex problems such as vector quantisation, speech recognition, combinatorial optimisation, control, pattern recognition, organisation of documents and modeling the structure of the visual cortex [7] as well as image retrieval [12,17]. The concept of mapping shapes in a topology conserving manner is novel. The structural information contained in geometrical shape is extracted using the pairwise relational attribute vectors. These vectors are quantised using an SOM [7], as the SOMs offer a number of advantages such as the ability to quantize adaptively depending on the dynamic ranges of the attributes and the ability to deal with the curse of dimensionality in the histograms-based methods more efficiently. For example, Hu et al. [5] define a number of relational attributes but use the best two attributes in their histogram based line patterns retrieval system. If they were to use all five or so attributes, then the line patterns retrieval will become a five dimensional histogram matching problem. Further, if the problem [5] does not
require scale invariance, then quantisation of absolute lengths and distances may become a difficult problem. All these difficulties can be resolved satisfactorily by employing the SOM to perform the quantisation. As the SOM preserves the topological relationships in the data (i.e. similar relational attribute vectors are mapped to neighboring SOM units), we can combine neighboring elements in the histogram if we desire to perform a coarse-to-fine search.

Using this trained quantisation SOM referred to as $\text{SOM}^1$, a global histogram of relational attribute vectors is generated for every structural shape. These histograms are treated as input vectors to another SOM referred to as $\text{SOM}^2$. As these global histograms capture the shape properties of the objects, the $\text{SOM}^2$ trained using these histograms naturally generates a topology conserving mapping for the structural shapes. This structural topology preserving maps can be made invariant to chosen transformations such as the similarity or affine transformations by choosing relational attribute vector appropriately. Although we employ a single layer SOM, it is possible to employ a tree-structured SOM [8] to perform the search efficiently to identify the winner neuron or to have structure adaptive models to facilitate flexible inclusion or removal of shapes. The tree-structured SOM [9] would be more appropriate for fast shape retrieval.

The paper is organized as follows. In Section 2, we present the relational attributes used to characterize a pair of line segments. In Section 3, the SOM algorithm as well as the applications of this algorithm for generating one-dimensional histograms and organization and indexation of databases are described. Experimental results are presented in Section 4 and the paper is concluded in Section 5.

2 Relational Vectors

We have chosen to represent the shapes by line segments. Depending on the requirements, the relational attributes between a pair of line segments may be chosen to possess various invariance properties such as invariance to translation, rotation, scale and more general affine transformations [5].

In this implementation, we have decided to include invariance to translation and rotation. Prior to computing the attributes, the intersection point between the two lines are computed as shown by “i” in Fig. 1. The end point of the first line also known as the reference line closer to the intersection point is labeled as “a”. The other end point of the first line is labeled as “b”. Likewise the end points of the second line are also labeled as “c” and “d” as shown in Fig. 1. Therefore, given a pair of line segments, all four end points can be unambiguously labeled. The following relational attributes are used: 

\[ \theta_{ab, cd} = \arccos \left( \frac{\overrightarrow{ab} \cdot \overrightarrow{cd}}{|\overrightarrow{ab}| \cdot |\overrightarrow{cd}|} \right). \]

The angle returned is between 0 and $\pi$. However, if we identify the rotation from $\overrightarrow{ab}$ to $\overrightarrow{cd}$ as clockwise or counter-clockwise by evaluating the vector product between vectors $\overrightarrow{ab}$ and $\overrightarrow{cd}$, then we can compute the angle attribute between $-\pi$ to $\pi$ in order to improve the discrimination quality of this attribute.

(2) Length of the reference line $ab$

(3) Length of the second line $cd$

(4) Distance $ac$

(5) Distance $bd$

(6) Distance $ad$

(7) Distance $bc$

Figure 1: A figure illustrating the computation of directed relational attributes.

We make use of all pairwise relational attribute vectors. If there are 40 line segments, then there will be $40 \times 39$ pairwise relational vectors. Every line is treated as the reference line in turn and every pair of line segments generates a seven dimensional vector. The vectors obtained from all shapes and all pairs are used to train an SOM as explained in the next section to perform adaptive quantisation of relational vectors.
3 Quantisation of Relational Vectors

In this section, first the SOM algorithm [1,7] is briefly explained. Then the procedure for generating a one-dimensional relational histogram for every shape using the trained quantisation map SOM$^1$ and developing the SOM$^2$ for topologically mapping the shapes are described.

3.1 Self-organizing Maps

In this application, it is desirable to have an equi-probable map. In other words, it is desirable to have each neuron to be the winner with the probability $\frac{1}{M}$ where $M$ is the total number of nodes in the SOM. Although the usage of topological neighbourhoods attempts to provide a uniform utilisation of all nodes, it does not completely resolve the problem. There were several approaches proposed in the literature [7]. Three of these approaches, namely convex combination, competitive learning with conscience and competitive learning with attention, are reviewed and evaluated recently by Bebis et al. [1]. According to their findings, the competitive learning with conscience appears to yield the best performance. In our experiments, we made use of the same approach to train SOM$^1$ and SOM$^2$.

The SOM algorithm with conscience is summarised in Table 1 [1,7]. The dimension of input vector is seven for SOM$^1$. The number of output neurons will be identical to the number of bins that we wish to have in the one-dimensional histogram. As the relational attribute vectors have varying dynamic ranges, it is necessary to make use of all relational attribute vectors to train the SOM$^1$.

3.2 Shape Histograms and Indexing

After completing the training of the SOM$^1$, we can extract the one-dimensional histogram by performing the following steps: (1) For every shape, initialize to zero a one-dimensional array of length identical to the number of output neurons in the SOM$^1$. (2) Present every seven dimensional pairwise relational vector to the SOM$^1$ and identify the winning neuron. (3) Increment the element in the one-dimensional histogram array corresponding to the winning neuron. (4) Repeat steps 2 and 3 for every relational attribute vector of the shape. (5) Repeat steps 1-4 for every sample shape model.

At the completion of executing the above steps, we have a one-dimensional histogram for every shape in the database. These histograms are treated as the input vectors to construct the SOM$^2$ using the same self-organising map algorithm in Table 1. Hence, the input feature vectors’ dimension of SOM$^2$ is identical to the number of nodes in SOM$^1$. Having trained the SOM$^2$, we associate the shapes with the neurons. Every shape is associated with three best matching neurons. With this, the organization phase is complete. During the application phase, we perform the steps outlined below to extract potentially similar model candidates to the query shape: (1) Extract all pairwise relational attribute vectors from the query shape. (2) Present every vector to the trained SOM$^1$ and obtain the one-dimensional histogram. (3) Present the one-dimensional normalized histogram to the SOM$^2$ and identify the three best matching units in the SOM$^2$ using the histogram intersection similarity measure. (4) Extract all the shapes associated with the best matching units and eliminate any duplication in the shape numbers to obtain the potentially similar model candidates to the query shape.

In order to identify the best matching unit in step 3 in the last paragraph, the histogram intersection similarity measure is used. It was shown that the histogram similarity measure [15] is superior when there are multiple shapes in the query or only a partial shape is presented in the query.

$$HI = \frac{\sum_{j=1}^{m} \min(Q_j, M_j)}{\sum_{j=1}^{m} M_j}$$

where $HI$, $n$, $Q$ and $M$ are the histogram intersection similarity, number of bins in the histogram (or the number of nodes in SOM$^1$), histogram of the query shape, and an arbitrary feature vector of SOM$^2$ respectively.
4 Experimental Results

Experiments were conducted using 363 shapes. There are 33 different shapes. These shapes are scaled using different scaling factors in x and y directions to obtain the total 363 shapes. As the relational vectors are scale sensitive, there are altogether 363 different shapes as shown in Fig. 2. The number of line segments in these shapes is about 50. In our experiments, the number of neurons in the SOM$^1$ is 1600. The SOM$^2$ has 225 neurons. Our results showed that the topologically similar shapes are mapped to the same neuron or neighbouring neurons. As naturally expected, the system was able to always retrieve the original shape when noiseless query objects were presented to the system. On average the SOM$^2$ extracted about 15 shapes for further rigorous matching. This is about 5% of the total shapes in the database. The final retrieval can be performed using a more accurate matching method on the 15 or so potential candidates. Further experiments were conducted using some modified shapes and partial shapes. The SOM$^2$ was able to retrieve the similar shapes using the histogram intersection similarity measure. The results clearly show the ability of the proposed approach to map the shapes in a topologically conserving manner and to retrieve structurally similar shapes for a given (partial) query shape.

5 Conclusion

In this paper, we proposed a novel topology preserving mapping scheme for geometric structural objects for the purposes of indexing into large structural databases by the SOM. We made use of seven dimensional relational vectors. These relational vectors are quantized using SOM$^1$ and subsequently converted into a one dimensional histogram for every structural shape. These one dimensional histograms are then used to train the SOM$^2$. The SOM$^2$ can be used to retrieve structural objects from a database given a query shape. The SOM$^2$ can also be used to organise structural objects in a model-based object recognition system and when test scene is given, models which are probably present in the test scene can be retrieved efficiently. If there are multiple shapes in the query scene, then it would be beneficial to use only the neighbouring line pairs [5] to generate the relational vectors, as distant line pairs may belong to different shapes. This scenario is not considered in this study.

The proposed approach offers a number of advantages such as the ability to make use of several relational attributes (as opposed to the limited number of attributes that can be used in the histogram-based relational indexing methods [5]), the ability to perform a dynamic quantization, the flexibility in including and removing model objects and database images, possibility for efficient implementation such as the tree-structured SOM [7,8,9] and the ability to handle other attributes like color and texture in a homogeneous manner by the SOM [12,17]. The proposed approach is also capable of incorporating relational attributes invariant of chosen transformations.

References


---

**Table 1: The SOM algorithm**

**SOM1**: Randomly initialize the weights for the given size map. Initialize the learning rate parameter $\alpha_0$, neighborhood size $\sigma_0$, and set the number of unsupervised learning iterations $K$. Set the biases $p_i(0)$ to $\frac{1}{M}$ where $M$ is the number of nodes in the map. Initialise constants $C$ and $D$ [1].

**SOM2**: Present the input feature vector $x = [x_1, x_2, \ldots, x_n]$ in the training data set to the network.

**SOM3**: Determine the winner node $c$ such that

$$||x-w_c|| = \min_i(||x-w_i|| - C(\frac{1}{M} - p_i(t)))$$

**SOM4**: Update the weights within the neighborhood of node $c$, $N_r(t)$ using the standard update rule [7]:

$$w_i(t+1) = w_i(t) + \alpha(t)[x-w_i(t)]$$

where $i \in N_r(t)$. The neighborhood wraps around at edges, i.e. column and row indices are in modulo representation. Update the biases:

$$p_i(t) = p_i(t) + D(1-p_i(t))$$

**SOM5**: Update learning rate and neighborhood size.

$$\alpha(t+1) = \alpha(0)\{1 - \frac{t}{K}\}$$

$$|N_r(t+1)| = |N_r(0)|\{1 - \frac{t}{K}\}$$

where $K$ is a constant and is usually set to be equal to the total number of iterations in the self-organising phase.

**SOM6**: Repeat SOM2-5 for the specified $K$ number of unsupervised learning iterations.
Figure 2: The 33 synthetic base shapes used in the experiments.