An efficient symbolic image retrieval method based on TSR and CMM

Mojtaba Sabet [1], Saeed Ayat [2], Reza Askari Moghadam [3]

[1] Payame Noor University faculty of engineering, Tehran, Iran
Msabet63@gmail.com
[2] Payame Noor University faculty of engineering, Najafabad, Iran
Dr.ayat@pnu.ac.ir
[3] Payame Noor University faculty of engineering, Tehran, Iran
askari@pnu.ac.ir

Abstract

The growing availability of inexpensive computer memory means that large amount of on-line images are now stored routinely, raising problems in providing fast and flexible access to images. Recently researchers proposed different methods of storage and retrieval of symbolic images but these methods have various shortcomings, for example most of them have high time complexity and are not suitable for dynamic image databases.

In this paper a faster and more efficient method for storage and retrieval of symbolic images in databases is proposed. In this method TSR[1] is used to represent spatial relationships existing among the elements of an image and a kind of binary neural network called CMM[2] is used as a data structure. This proposed method has low time complexity and has the capability of different types of retrieval such as exact mach retrieval and partial match retrieval, it also has the ability to retrieve the transformed[3] images and can be used in dynamic databases. The study made in this work reveals that this method bears various advantages when compared to other existing methods.

Key words: Symbolic image database, Symbolic image retrieval, neural network, TSR, CMM

1 Introduction

Nowadays image database is a fundamental requirement of information systems, search engines and also many other types of softwares. Recently researchers have proposed different types of image storage and retrieval methods. The storage and retrieval methods present in alphanumeric databases such as indexing, hashing and B-tree are not suitable for image databases due to different natures of alphanumeric data and pictorial data. In early 90 content based image retrieval was proposed. In these kinds of systems, images are indexed automatically according to features like color, texture, shape and spatial relationships.

The symbolic image database is a kind of image database which consist of images and information about images. The symbolic image can be regarded as an abstract physical image, while the physical image is the real image itself (Guru & Nagabhushan, 2001). Generally there are two problems with making symbolic image database: first to find a

[1] Triangular Spatial Relationships
[2] Correlation Matrix Memories
[3] Such as rotated or flipped images

way to extract image visual features Second to store these features in a kind of data structure that can be both fast and flexible and also be able to do different types of searches and retrievals, in addition the database should be intelligent enough to retrieve transformed images.
Recently the researchers have proposed different methods to make image database. First the methods like pixel based methods (Chock, Cardenas, & Klinger, 1984), R-tree based methods (Guttman, 1984), and vector based methods were proposed. These methods were not intelligent enough and were not suitable for complex operations.

Then a method called 2D-string (Chang, Shi, & Yan, 1987) was proposed. In 2D-string method, image elements are projected on the x and y axis, in this method a series of relational operators are used to represent the spatial relationships between two image symbols along the x and y axis. As the following, according to 2D-string method, different types of methods such as 2D-C string (Lee & Hsu, 1990) and 2D-Z string (Lee & Chiu, 2003) were proposed, but these methods were not efficient enough and consume a lot of memory. After that another method called 2D-H string (Chang & Li, 1988) was proposed. This method had more ability to work with the complex images, but generally the problem with methods based on 2D-string is that they have high time complexity (their time complexity is NP) and they are not able to retrieve transformed images specially rotated ones.
Then other methods based on hashing were proposed (Chang & Lee, 1991), (Sabharwal & Bhatia, 1995), (Zhou & Ang, 1997). Hash based methods have low time complexity, but the problem with these methods is that when a new image is inserted in the database, hash table must be reconstructed which has exponential time complexity, so these methods are not suitable to be used in dynamic databases.
To be able to use hash based methods in dynamic databases Mughal et al. have proposed a method called 3D hash function (Mughal et al., 2007). They have used 2D-string to represent image spatial relationships and a method like hashing to store these relationships. 3D hash function decreases the insertion time complexity but this method doesn't support exact match retrieval. In (Chang & Wu, 1995) an exact match retrieval method is proposed but this method doesn't have the proper accuracy and false positive may occur.
Punitha & Guru in (Punitha & Guru, 2008) have proposed a method based on TSR and B-tree. In their method TSR is used to represent image spatial relationships and B-tree is used as data structure. Time complexity of this method is O(log n). Yazdi et al. in (Yazdi et al., 2008) have proposed a method based on TSR and linear hashing. They also have used TSR to represent image spatial relationships and have stored these relationships by use of linear hashing. Time complexity of this method is $O(L_C.k')$ where L_C is the average number of nodes in link list and k' is the number of available quadruples in image.

In this paper we use TSR to represent the spatial relationships existing among image elements and a kind of binary neural network called CMM is used as data structure. This neural network is so fast and flexible that is proper to be used in large databases that consist of many images. This proposed method has low time complexity and is able to do exact match retrieval and partial match retrieval. This method is also able to retrieve the transformed images.

In this paper first we introduce CMM in section 2. In section 3 TSR is described. In section 4 we explain how CMM is used to store and retrieve symbolic images. In section 5 we analyze time complexity of this method both theatrically and practically. In section 6 our proposed method is compared with other existing methods and finally section 7 concludes the paper.

2 CMM

CMM is a kind of binary neural network that uses binary inputs, outputs and weights. In this neural network hebbian learning is used. Despite other neural networks the process of learning in this neural network is so fast (Austin, 1996).
For training the network we apply input and output vectors simultaneously to CMM and the CMM will be set to 1 where input and output are both 1. Network training is shown in Fig. 1.

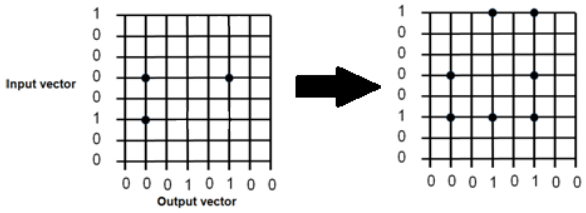

Figure 1: Training the neural network.

The recall process is like the learning process. First the input pattern is applied to network, then the rows are summed and the activation of the network will be thresholded to obtain the output vector. Three kinds of thresholds may be applied: Willshaw, L-Max and Perceptron. In Fig. 2 the recall process is shown. Here Willshaw threshold is used. In the Willshaw threshold, threshold value is constant for all columns and is equal to the number of 1s available in input vector.

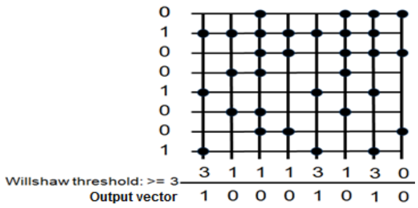

Figure 2: Recall process using Willshaw threshold. The input pattern has three bit 1 so the CMM is thresholded at three.

3 TSR

A TSR is formally defined by connecting three non-collinear components in a symbolic image as follows (Guru & Nagabhushan, 2001): Let *A*, *B* and *C* be any three non-collinear components of a symbolic image. Let L_a, L_b and L_c be the labels of *A*, *B* and *C* respectively. Connecting the centroids of these components mutually forms a triangle with M_1, M_2 and M_3 as the midpoints of the sides of the triangle and θ_1, θ_2 and θ_3 as the smaller angles subtended at M_1, M_2 and M_3 respectively. Fig. 3 shows an example. The TSR among the components *A*, *B* and *C* is represented by a set of quadruples:

$$\{(L_a, L_b, L_c, \theta_3), (L_a, L_c, L_b, \theta_2), (L_b, L_a, L_c, \theta_3), (L_b, L_c, L_a, \theta_1), (L_c, L_a, L_b\ \theta_2), (L_c, L_b, L_a, \theta_1)\}.$$

This representation is unwieldy, as there are six possible quadruples for every three non-collinear components. We can preserve spatial relationships among the three components by just a quadruple (L_i, L_j, L_k, θ) satisfying the condition stated in (Punitha & Guru, 2008) instead of six quadruples.

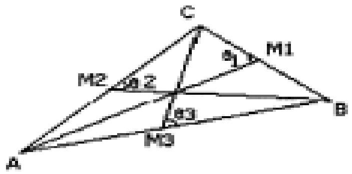

Figure 3: Triangular spatial Relationship.

4 Using the binary neural network for storage and retrieval of symbolic images

Let $\{I_1, I_2, \ldots, I_n\}$ be a set of N symbolic images and Q^n be the number of extracted quadruples from nth symbolic image by the TSR method ($1 \leq n \leq N$). Qth quadruple of nth symbolic image is represented by the binary vector I_q^n in which $|I_q^n|$ is constant for all n, q ($1 \leq q \leq Q^n$). Suppose that the nth symbolic image is shown by binary vector O^n where $|O^n|$ is constant for all n. O^n and I_q^n are the binary vectors that have only one bit 1. Let *TS* be a set of training samples that are shown as ordered pairs (I_q^n, Q^n). Then CMM is defined to be a $|I_q^n| \times |Q^n|$ matrix with entries being 0 or 1 and is trained by the *TS* set.

4.1 Inserting image in database

Suppose we want to insert N symbolic images named $I_1, I_2, I_3, \ldots, I_n$ in image database. Then to insert images in database we should first extract the quadruples of the each image by the use of TSR method and convert each quadruple into binary form to obtain I_q^n. We should convert the name of each image into binary form to obtain O^n. Each binary vector should have only one bit 1 to avoid false positive. The relationship between each image and it's extracted quadruples should be stored in neural network. To do this we should train the neural network by these samples:

$$TS = \left\{ (I_1^n, O^n), (I_2^n, O^n), \ldots\ldots, (I_q^n, O^n) \right\}$$

In other word to train the neural network, vector I_q^n must be applied as input and vector O^n must be applied as the output of the neural network and the neural network should also adjust its weights according to the method explained in section 2. In addition we should create an auxiliary table with N rows and two columns named I_n and Q^n ($1 \leq n \leq N$). We call this table T.

With regards to explanations above to store an image in database, the neural network should be trained by Q^n samples. In order to decrease the number of training iterations and to increase the speed of training the network, we can superimpose the I_q^n binary vectors for each image n ($1 \leq n \leq Q^n$) to create a single vector called S_n. Now we can train the neural network with this single sample to store the *nth* image in database.

By superimposing input vectors we can store association between an image and its quadruples in the network in single epoch (instead of Q^n iterations). Here the proposed algorithm to insert the images in database is presented:

Proposed inserting algorithm:
Step1: Extract the quadruples of image n.
Step2: Convert quadruples of image n to binary form using orthogonal vector (with one bit set) to obtain I_q^n.
Step3: Convert the image id (I_n) to binary from to obtain O^n.
Step 4: Superimpose the I_q^n $(1 \leq q \leq Q^n)$ vectors to obtain single vector S_n for image n.
Step 5: Train the CMM with single training sample TS= (S_n, O^n).
Step 6: Add I_n, Q^n to the auxiliary table T.
End of proposed inserting algorithm.

4.2 Image Retrieval from the Database

Mainly we have three retrieval methods in database: Exact Match Retrieval, Similarity Retrieval and Sub-image Retrieval (Hsieh & Hsu, 2008). The proposed method in this paper has the capability to do these three types of retrieval.

Let i be the query image. First the quadruples of the image i should be extracted by TSR method. Suppose that Q^i be the number of extracted quadruples of image i. Now each of the quadruples should be converted into binary form to obtain I_q^i ($1 \leq q \leq Q^i$). Then all of these binary vectors should be superimposed to obtain single vector S_i. To retrieve the sub-images, vector S_i should be applied as the input to the neural network and then the activation of the neural network should be thresholded with the Willshaw threshold. After the thresholding, a binary vector is obtained that represent sub-images of the query image.

To do similarity retrieval, the above steps should be done to retrieve sub-image but the different is that in similarity retrieval, network should be thresholded at x ($1 \leq x \leq Q^i$). The closer the value of x to Q^i, the more similar the retrieved images to the query image and vice versa.
The database should be able to measure similarity of the retrieved images. To achieve this Eq. 1 is proposed. Let i be the query image and Q^i be the number of extracted quadruples of image i. Let r be a retrieved image and Q^r be the number of its extracted quadruples. Let A be the number of the common quadruples between these two images, so the similarity of images r and i is calculated by the following formula:

$$Similarity(r,i) = \frac{Q^r - A}{A} \tag{1}$$

The closer the result of the above formula to zero, the more similar the two images are, and the less closer to zero, the less similar they are. In this deduction when the number of the common quadruples are more it causes that from one hand the denominator of the deduction increase and from another hand the numerator of the deduction decrease and the deduction comes closer to zero. If the number of the common quadruples are less, it causes that the denominator of deduction decreases and the numerator of deduction increases and so the value of deduction increases and gets away from the zero. So by the use of this formula the database can measure the similarity of the two images and sort the retrieved images according to their similarities. Here the proposed algorithm to retrieve the images is presented:

Proposed retrieval algorithm:
Step1: Extract the quadruples of the query image i.
Step 2: Convert quadruples to binary vectors I_q^i.
Step 3: Superimpose the binary vectors to obtain a single vector S_i.
Step 4: According to retrieval type select appropriate threshold.
Step 5: Obtain output vector using the selected threshold in step 4.
Step 6: Convert the output vector to image id.
End of Proposed retrieval algorithm.

5 Time Complexity analyze

In this section time complexity of insertion and retrieval will be analyzed both theoretically and practically.

5.1 Insertion time complexity

Let *m* be the number of extracted quadruples of image to be inserted in database, so to insert an image, the neural network should be trained by binary vector S_i. In training process, *m* bits from the network change into 1, so the time complexity of image insertion becomes O(*m*). As *m* is a constant and small value and its value is not dependant to the number of images available in database, so the insertion time complexity becomes O(1) (Cormen et al, 2001).

5.2 Retrieval Time complexity

To search an image, the binary vectors of the extracted quadruples in the query image should be computed. Then computed vectors should be applied to the neural network as the input. The neural network can compute the output vector in one pass. If *m* be the number of extracted quadruples from the query image, then the number of 1s available in input vector is *m* and so *m* vectors should be added together. Therefore the retrieval time complexity is O(*m*). As *m* is a small and constant value and its value is not dependant to the number of images available in database, so the retrieval time complexity is O(1).

To analyze the retrieval time complexity of this method practically, a database is made by the use of MatLab software. In the diagram shown in Fig. 4, the retrieval time of an image versus the number of images available in a database is shown. At first the database consists of one image and in each phase one image is added to the database. After adding the new image in each phase, one image is chosen randomly and its retrieval time is measured. As it is observed the retrieval time is not dependant to the number of images available in database.

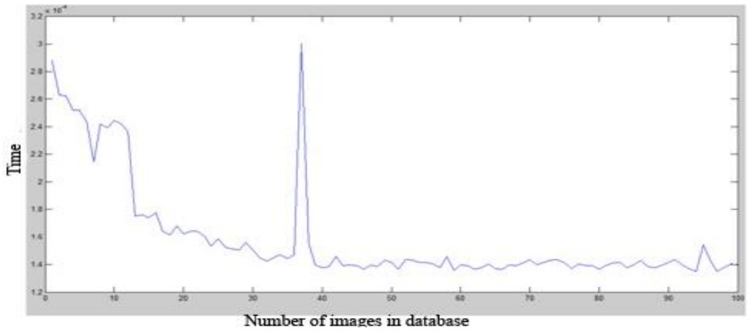

Figure 4: Retrieval time versus number of images available in database.

6 Comparison with other Methods

A summary of retrieval time complexity of existing methods is shown in table 1. As it is seen the 2D-string based methods have NP time Complexity, therefore they are expensive methods and they don’t have the capability to be used in large databases. Hash based methods have the O(1) time complexity to retrieve the image, but these methods are not suitable to be used in dynamic databases because in dynamic databases, the image insertion occurs frequently and while the image is inserting, the hash table should be reconstructed and the time complexity of reconstructing the hash table is exponential. So these methods are not suitable to be used in dynamic databases and they are just suitable for static databases.

Among the methods shown in table 1, those which use the tree as the data structure have logarithmic time complexity like the proposed by Wu and Cheng (Wu & Cheng, 1997), but the problem with this method is that it is not invariant to image transformation, it means that if the query image transforms it can't retrieve the similar image anymore.

The other method which has the logarithmic time complexity is the proposed method by Chang and Wu (Chang & Wu, 1995), but the problem with this method is that it has only the ability of retrieving exactly similar images and don't support other kinds of retrieval.

The suggested method by Punitha & Guru (Punitha & Guru, 2008), is more efficient than other methods because it is invariant to image transformation and is capable to do different types of retrieval and it has the logarithmic time complexity.
The proposed method by Mughal et al. (Mughal et al., 2007), has the time complexity of O(1) but their method is not capable of exact match retrieval. The suggested method by Yazdi et al. (Yazdi et al., 2008) has the time complexity of O($L_c.K'$) where K' is the number of extracted quadruples of the query image and L_c is the average number of the nodes available in link list. As $L_c.K'$ is a constant and small value, so the time complexity of this method is O(1).
The proposed method in this paper is invariant to image transformation. This method has the capability of different types of storage and retrieval and has the time complexity of O(1) which is better than other existing methods.

Method	***Data structure***	***Invariant to image transformation***	***Suitable for dynamic database***	***Time complexity***
(Lee & Hsu, 1992)	*2D-C string*	*no*	*no*	*NP*
(Petraglia et al., 2001)	*2D-C string*	*no*	*no*	*NP*
(Lee & Chiu, 2003)	*2D-Z string*	*no*	*no*	*NP*
(Chang & Wu, 1995)	*9DLT matrix + PCA*	*no*	*no*	*O(log n)*
(Zhou & Ang, 1997)	*9DLT Matrix +hashing*	*no*	*no*	*O(n)*
(Wu & Cheng, 1997)	*G-tree*	*no*	*yes*	*O(log n)*
(Punitha & Guru, 2008)	*TSR + B-tree*	*yes*	*yes*	$O(\log_r n)$
(Mughal et al., 2007)	*2D-string+3D hash function*	*no*	*yes*	*O(1)*
(Yazdi et al., 2008)	*TSR+ linear hashing*	*yes*	*yes*	*O(1)*
Proposed method	*TSR+CMM*	*yes*	*yes*	*O(1)*

Table 1. Comparison between existing methods.

7 Conclusion

In this paper a new method for symbolic image storage and retrieval is proposed. In this method the spatial relationships between the available symbols in image are extracted by the use of TSR method. The spatial relationships of each image are represents by a set of quadruples. The association between each image and its extracted quadruples should be stored in database.
To store this association, a kind of binary neural network is used. Because of using binary inputs and weights, this neural network is very fast and flexible and has very high performance. Unlike other neural networks, this neural network does not need a long time to train. Because of using TSR method to represent the spatial relationship between existing elements in image, this method is invariant to image transformation. Using binary neural network makes this method capable to do different types of retrieval. Also we can do three kinds of retrieval in one pass.
By using the proposed formula, the database can measure the images similarities and sort them according to their similarities. This method is suitable to be used in dynamic databases because of its low time complexity (O(1)).

References

Austin, J. (1996). Distributed associative memories for high speed symbolic reasoning. *Fuzzy Sets and Systems - Special issue on connectionist and hybrid connectionist systems for approximate reasoning , 82* (2), 223-233.

Chang, C., & Lee, S. (1991). Retrieval of similar pictures on pictorial database. *Pattern Recognition , 24* (7), 675–680.

Chang, C.-C., & Wu, T.-C. (1995). An exact match retrieval scheme based upon principal component analysis. *Pattern Recognition Letters , 16* (5), 465–470.

Chang, S. K., Shi, Q. Y., & Yan, C. W. (1987). Iconic Indexing by 2-D Strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence , PAMI-9* (3), 413-428.

Chang, S., & Li, Y. (1988). Representation of multi resolution symbolic and binary pictures using 2D-H strings. *IEEE Workshop on Languages for Automation: Symbiotic and Intelligent Robots*, (pp. 190–195). College Park, MD.

Chock, M., Cardenas, A. F., & Klinger, A. (1984). Database Structure and Manipulation Capabilities of a Picture Database Management System (PICDMS). *IEEE Transactions on Pattern Analysis and Machine Intelligence , PAMI-6* (4), 484 - 492.

Cormen, T. H., Stein, C., Rivest, R. L., & Leiserson, C. E. (2001). *Introduction to Algorithms* (2 ed.). MIT Press.

Guru, D., & Nagabhushan, P. (2001). Triangular spatial relationship: a new approach for spatial knowledge representation. *Pattern Recognition Letters , 22* (9), 999–1006.

Guttman, A. (1984). R-trees: a dynamic index structure for spatial searching. *SIGMOD '84 Proceedings of the 1984 ACM SIGMOD international conference on Management of data* , (pp. 47-57). New York, NY, USA .

Hsieh, S.-M., & Hsu, C.-C. (2008). Graph-based representation for similarity retrieval. *Data & Knowledge Engineering , 65* (3), 401-418.

Lee, A. J., & Chiu, H.-P. (2003). 2D Z-string: a new spatial knowledge representation for image databases. *Pattern Recognition Letters , 24* (16), 3015-3026.

Lee, S., & Hsu, F. (1990). 2D-C string: a new spatial knowledge representation for image database system. *Pattern recognition , 23* (10), 1077-1087.

Lee, S., & Hsu, F. (1992). Spatial reasoning and similarity retrieval of images using 2D C string knowledge representation. *Pattern Recognition , 25* (3), 305–318.

Mughal, M., Nawaz, M., Ahmad, F., Shahzad, S., A.K.Bhatt, & Mohsin, S. (2007). A 3D-Hash Function for Fast Image Indexing and Retrieval. *Computer Graphics, Imaging and Visualization. CGIV apos*, (pp. 341-348). Bangkok.

Petraglia, G., Sebillo, M., Tucci, M., & Tortora, G. (2001). Virtual images for similarity retrieval in image databases. *IEEE Transactions on Knowledge and Data Engineering , 13* (6), 951–967.

Punitha, P., & Guru, D. (2008). Symbolic image indexing and retrieval by spatial similarity: An approach based on B-tree. *Pattern Recognition , 41* (6), 2068-2085.

Sabharwal, C., & Bhatia, S. (1995). Perfect hash table algorithm for image databases using negative associated values. *Pattern Recognition , 28* (7), 1091–1101.

Wu, T., & Cheng, J. (1997). Retrieving similar pictures from iconic database using G-Tree. *Pattern Recognition Letter , 18* (6), 595–603.

Yazdi, M., Najafzade, K., & Moghaddam, M. (2008). A Fast Symbolic Image Indexing and Retrieval Method Based On TSR and Linear Hashing. *IEEE International Symposium on Signal Processing and Information Technology*, (pp. 469 - 473). Sarajevo.

Zhou, X., & Ang, C. (1997). Retrieving similar pictures from pictorial database by an improved hashing table. *Pattern Recognition Letter , 18* (9), 751–758.