Similarity Retrieval of Symbolic Images with Multiple Instances of Iconic Objects: A Novel Approach

P. Punitha
Dept. of Studies in Computer Science
University of Mysore, Mysore
punithaswamy@yahoo.com;

D. S. Guru
Dept. of Studies in Computer Science
University of Mysore, Mysore
guruds@mailcity.com

Abstract

In this paper, a problem associated with archival and retrieval of symbolic images with multiple instances of iconic objects, in/from symbolic image database (SID) invariant to image transformations is addressed. An efficient methodology to retrieve images similar to a query image at two levels of similarity from the SID is presented. The presented model takes O(log n) search time in worst case, where, n is the total number of symbolic images stored in the SID.

1. Introduction

Retrieval of images from a symbolic image database (SID), based on similarity is an important task in image database applications. The ability of a SID to store images is not remarkable unless it supports efficient retrieval of images having a high degree of similarity with the query image in less time, irrespective of the size of the SID. The representation scheme must also support the retrieval process in retrieving similar images, irrespective of the transformation through which the query image has undergone. In addition, the SID is expected to take care of multiple instances of iconic objects.

Similarity retrieval through the perception of spatial relationships has received considerable attention of many researchers. Indeed, perception of spatial relationships among the components of a symbolic image preserves the reality being embedded in the respective physical image. The object oriented search started with the introduction of 2D string [4]. Based on the 2D string representation, many algorithms were proposed to represent/retrieve symbolic images in/from a SID [8],[7]. However, these string based representation schemes are not invariant to image transformations, especially to rotation and in addition, the string matching process takes non-deterministic-polynomial time complexity during the process of retrieval. In order to reduce the search time and to avoid string matching, hash oriented methodologies for similarity retrieval were explored [2],[13],[10],[11]. Although, hash oriented algorithms take care of multiple instances of iconic objects to some extent, they require O(m^2) search time in the worst case for retrieval of symbolic images, where m is the number of iconic objects and construction of hash table takes exponential time complexity. [3] and [14] have proposed an exact match and similarity retrieval schemes by the use of the 9DLT matrix[1]. But, the models based on 9DLT matrices are not robust to take care of image transformations especially, rotation.

[15] introduced an image retrieval scheme based on object's orientation spatial relationship, which preserves the information of partial and total orientation relationships among m objects present in the image by a matrix of order $m \times m$. Though, this method is invariant to image transformations, even for considerably small number of images, the space complexity is very high and hence retrieval takes longer time. Also, the method ignores the situations of multiple instances of iconic objects, stating that similarity retrieval problem is NP-hard when multiplicity of iconic objects is allowed in picture matching.

[9] discovered 2D-R string representation based on 2D-C string mechanism [6] as a rotation invariant concept. However, the method is sensitive to the image centre.

The method proposed by [12], is the only method, which takes care of multiple instances of iconic objects at least to some extent. Although, the method allows multiple instances of objects in images during representation, it entails only one instance of each object in the query image. In addition, the method is not invariant to image transformations.

Thus, it is clear from the above discussion that all the aforementioned methods appear to be efficient in one or the other sense. But, the models that are robust and simple seem to be inefficient, in addition to being not invariant to image transformations and the methods, invariant to image transformations are either sensitive to the reference point or not effective from the point of view of storage requirement. The model, which takes care of multiple instances of the iconic objects to some extent, is not invariant to image transformations and entails single instance of the iconic objects in a query image. The problem therefore is to devise an efficient methodology, which is not only smart enough to take care of image transformations, but can rejuvenate to take care of multiple instances of iconic objects and be acceptable for real pragmatic situations.

In view of this, a problem associated with archival and retrieval of symbolic images with multiple instances of iconic objects, in/from SID, invariant to image transformations is addressed in this paper. The proposed model preserves triangular spatial relationship (TSR) [5] among the components in a symbolic image by the use of quadruples. A SID is created through construction of B-tree. Since B-tree does not support multi-dimensional data (4-D data in our case), a distinct and unique key is computed for each distinct quadruple and then the computed keys are stored in B-tree as representatives of the corresponding quadruples. Each key value stored in B-tree is then attached with a list of images, which have the corresponding quadruple as one of their associated quadruples along with the frequency of the quadruple in the corresponding image. An efficient methodology to retrieve similar images from the SID, and then to order them according to weighted similarity is presented. The presented model takes $O(\log_r n)$ search time in worst case, where, n is the total number of symbolic images stored in the B-tree and r is the order/rank of the B-tree.

2. Overview of Triangular Spatial Relationship (TSR)

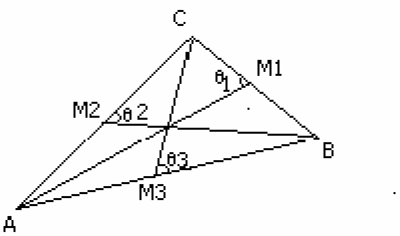

Fig. 1 Triangular spatial relationship

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 as shown in Fig.1. Let M_1, M_2, and M_3 be the midpoints of the sides of the triangle as shown in Fig.1. Let θ_1, θ_2, and θ_3 be the smaller angles subtended at M_1, M_2, and M_3 respectively and are shown in Fig.1. The TSR among the components A, B and C is represented by a set of quadruples $\{(L_a, L_b, L_c, \theta_3), (L_a, L_c, L_b, \theta_2), (L_b, L_a, L_c, \theta_3), (L_b, L_c, L_a, \theta_1), (L_c, L_a, L_b, \theta_2), (L_c, L_b, L_a, \theta_1)\}$. Since, representing all six possible quadruples for every three non-collinear components is unwieldy, it is recommended to choose only one of these six quadruples, based on a set of defined criteria [5]. The TSR is claimed to be invariant to image transformations. For more details on TSR the readers are directed to refer [5].

3. The Proposed Model

This section describes a novel scheme to represent and retrieve symbolic images with multiple instances of iconic objects.

3.1 Creation of SID

In order to make the representation scheme invariant to image transformations, the triangular spatial relationship existing among all components present in a symbolic image to be stored in the SID is perceived and preserved by the use of quadruples. Thus, the problem of symbolic image representation is reduced to the problem of storing those quadruples such that the retrieval task becomes effective and efficient. Hence, it is advised to store the quadruples in the database through creation of B-tree, an efficient multilevel indexing structure. However, since B-tree doesn't support storage of multi-dimensional data such as quadruples, a unique and distinct number called key is generated for each of the distinct quadruples to be stored in B-tree and then the generated keys are stored in the B-tree as representatives of the corresponding quadruples.

If (L_a, L_b, L_c, θ) is a quadruple to be stored in B-tree then the corresponding key is computed as

$$k = D_\theta (L_a-1)m^2 + D_\theta(L_b-1)m + D_\theta(L_c-1) + (C_\theta-1) \quad \ldots (1)$$

where, m is the number of distinct iconic objects in the SID, D_θ is the number of slices/classes, the continuous interval type θ domain [0…90] is split into and C_θ is the class number to which a specific value of θ belongs. The discretization of the continuous domain of θ is suggested to take care of the possible errors that can occur during the computation of θ value due to the limitation of the computing system in handling floating point numbers.

It can be noticed that the associated keys for any two different quadruples are distinct and unique. If N is the total number of distinct keys generated due to all n symbolic images, then all these N keys are stored in a B-tree. Each key value is then attached with a list of image indices and frequencies. The image indices that are attached to a key k are the indices of images, which have k as one of the keys in their corresponding key set and the frequency is the number of times k is present in the corresponding key set.

Therefore, following is the algorithm proposed for representing symbolic images in SID.

Algorithm: Creation of SID
Input: $S_1, S_2, S_3, \ldots, S_n$ - Set of symbolic images
Output: Symbolic Image Database (B-tree)
Method:
Step 1: For each symbolic image S_i do
    Total keys $T_i=0$, Total distinct keys $U_i=0$
    For each three non-collinear components do
      i. Apply TSR and compute the TSR key k as explained in section 3.1 and increment T_i.
      ii. Search for k in the B-tree
        If k is present
          Search for i in the attached list of image indices
          If i is present
            Increment its frequency
          Else
            Update the attached list by inserting i with frequency 1
          If end
        Else
          Insert k into the B-tree and attach the list with image index i and frequency 1 and increment U_i
      If end
    For end
  For end
Algorithm ends.

3.2 Retrieval of similar symbolic images

Similarity retrieval of images is in fact a fuzzy issue. Here, the task is to retrieve if not an exact, at least a nearest match to the given query image. To achieve this, in this paper, two levels of similarity retrieval are proposed. The query image could be, exactly similar to a model image or completely contained in the model image. Sometimes, the query image itself may completely contain a model image or simply have a sort of overlap with a model image. However, in any case, since each symbolic image in the SID is just a collection of TSR keys, the similarity retrieval problem reduces to the problem of subset matching between the TSR keys of a given query image and TSR keys of every image stored in the database. Instead of searching through all images one by one, the proposed novel B-tree based image representation scheme allows us to identify as quickly as possible, the similar images.

Let Q be a query image with T_Q number of total TSR quadruples. Consequently, T_Q TSR keys are computed. For each distinct key in the query key set, its frequency is noted. Let U_Q be the number of distinct TSR keys present in Q. The B-tree is then accessed through, in search of each distinct TSR key of Q and the list of image indices with frequencies attached to the key is extracted. For each image index i, in each extracted list (list corresponding to a key k), the difference d in frequencies of k in S_i and Q (frequency of k in S_i – frequency of k in Q) is noted. Thus, for each TSR key of Q present in the B-tree, a list of pairs, *(i, d)* is obtained. If there exists a symbolic image S_i, consisting of all keys of Q, then i is present in all lists of pairs obtained. Hence, the symbolic images analogous to the image indices obtained through the intersection of the lists of pairs based on i, are classified as first level similar images. The remaining images are grouped as second level similar images since at least a TSR key of Q is being missed in them. For each extracted image index i, the minimum (d_{min}) and the maximum (d_{max}) of the differences in frequencies due to all keys for which i is extracted are noted from the lists of pairs. The differences in frequencies of keys can thus be represented by an interval $[d_{min}, d_{max}]$. If a symbolic image S_i in the database is exactly similar to Q, then it is certainly classified as a first level similar image. In addition, the number of distinct keys in Q and S_i are equal and the frequency difference interval becomes $[d_{min}, d_{max}] = [0, 0]$. Incase, if Q is completely contained in S_i, no doubt S_i is at the first level of similarity, but will also have either total keys or total distinct keys, or both, greater than that of Q and hence the interval $[d_{min}, d_{max}] = [\geq 0, \geq 0]$. On the other hand, S_i being in the first level of similarity is completely contained in Q if Q has all keys of S_i and in addition has the interval $[d_{min}, d_{max}] = [<0, \leq 0]$. S_i though being in the first level of similarity is said to just overlap with the Q, if at least one key of Q has a greater frequency than that of S_i and S_i has additional (new keys not present in Q) keys. Thus, the first level images are classified into four categories with exact matched images being most preferred, symbolic images completely containing the query given the next preferred class, symbolic images completely containing Q being the next preferred and symbolic images with a sort of overlap with Q the least preferred amongst the first level similar images.

The images in the second level of similarity can also be further grouped into two categories. Here also, a symbolic image S_i can be completely contained in Q, if all keys of S_i are present in Q and Q has additional keys in addition to the interval $[d_{min}, d_{max}]=[\leq 0, 0]$. Symbolic images completely contained in Q, are categorized to be in a higher class amongst the images at second level of similarity. For any other case, S_i is said to have a sort of overlap with Q. Two images are said to have the least possible overlap when both the images have at least three iconic objects in common with the same spatial scattering.

The following algorithm is therefore, devised for retrieving similar symbolic images from SID for a given query symbolic image.

Algorithm : Proposed retrieval scheme
Input : Q, a symbolic query image
Output : List of indices of symbolic images similar to Q
Method :

Step1: Apply TSR and compute the TSR keys as explained in section 3.1. and note down T_Q and U_Q.

Step2: Compute the frequencies of each distinct key.

Step3: For each TSR key k of Q with frequency f_{k_Q} do

Search for k in B-tree

If k is present

Extract the list of image indices and frequencies, (i, f_{k_i}) and compute the list of pairs (*i, d*), where $d = f_{k_i} - f_{k_Q}$

Else

Return null list

If end

For end

Step4: For each extracted image index i do

Find the minimum d_{min} and maximum d_{max} of all d's associated with i to form the interval $[d_{min}, d_{max}]$.

For end

Step5: Classify the retrieved images into two levels of similarity and their categories as explained in section 3.2.

Algorithm ends.

4. A complete illustration with an example:

In this section we illustrate the proposed scheme for symbolic image representation in SID and demonstrate how the proposed similarity retrieval takes place.

4.1 Representation of symbolic images in SID

Consider $n=6$ symbolic images shown in Fig.2 with multiple occurrences of iconic objects. Let {*1, 2, 4, 5, 7*} be the set of symbolic labels of 5 distinct iconic objects. As explained in section 3.1, the concept of TSR is employed and the TSR existing among the components of each image is preserved by the use of quadruples. Since I_1 has got only one distinct component with 4 instances and no three of them are collinear, the total number of quadruples generated is $^4C_3 = 4$ and they are, {*(4,4,4, 90.00), (4, 4, 4,90.00), (4, 4, 4, 90.00), (4, 4, 4, 90.00)*}.

Now the 4th component, which is a real value, of each quadruple is mapped onto its class index (as suggested in section 3.1). In this example, the θ-domain [0°…90°] is into *18* classes of size 5°. The above quadruples thus become, {*(4, 4, 4, 18), (4, 4, 4, 18), (4, 4, 4, 18), (4, 4, 4, 18)*}. Similarly, the sets of quadruples preserving TSR among the components of the remaining symbolic images are; $\boldsymbol{I_2}$ = *{(4, 4, 2, 17), (4, 4, 4, 18), (4, 4, 2, 16), (4, 4, 2, 18), (4, 2, 2, 8), (4, 4, 2, 14), (4, 4, 2, 6), (4, 2, 2, 2), (4, 2, 2, 3)}*. $\boldsymbol{I_3}$ = *{(5, 4, 4, 12), (4, 4, 2, 13), (4, 4, 2, 13), (5, 4, 2, 15), (5, 4, 2, 2), (4, 2, 2, 18), (5, 4, 2, 1), (5, 4, 2, 12), (4, 2, 2, 18), (5, 2, 2, 13)}*. $\boldsymbol{I_4}$ = *{(4, 4, 2, 17), (4, 4, 4, 18), (4, 4, 4, 18), (4, 4, 2, 18), (4, 4, 2, 13), (4, 4, 4, 18), (4, 4, 2, 6), (4, 4, 2, 18), (4, 4, 4, 18), (4, 4, 2,17)}*. $\boldsymbol{I_5}$ = *{(4, 4, 2, 17), (4, 4, 2, 16), (4, 4, 4, 18), (7, 4, 4, 13), (4, 2, 2, 8), (4, 4, 2, 13), (7, 4, 2, 17), (4, 4, 2, 18), (7, 4, 2, 16), (7, 4, 4, 14), (4, 2, 2, 2), (4, 4, 2, 18), (7, 4, 2, 2), (4, 4, 2,16), (7, 4, 2, 2), (7, 4, 4, 18), (4, 2, 2, 16), (7, 2, 2, 3), (7, 4, 2, 15), (7, 4, 2, 10)}* and $\boldsymbol{I_6}$ = *{(4, 2, 1, 13), (4, 2, 1, 14), (4, 4, 1, 13), (7, 4, 1,13), (2, 2, 1,18), (4, 2, 1, 2), (7, 2, 1, 15), (4, 2, 1, 5), (7, 2, 1, 10), (7, 4, 1, 18), (4, 2, 2, 2), (4, 4, 2,18), (7, 4, 2, 2), (4, 4, 2, 16), (7, 4, 2, 2), (7, 4, 4, 18), (4, 2, 2, 16), (7, 2, 2, 3), (7, 4, 2, 15), (7, 4, 2,10)}*

For each distinct quadruple, a distinct and unique key is computed and its frequency in the corresponding image is noted. Hence, each symbolic image can be described by a set of TSR keys and their frequencies as given below.

TSR Key Set of $\boldsymbol{I_1}$ = {*3095 (4)*}; **TSR Key Set of $\boldsymbol{I_2}$** = *{3058(1), 3095(1), 3057(1), 3059(1), 2797(1), 3055(1), 3047(1), 2791(1), 2792(1)}*; **TSR Key Set of $\boldsymbol{I_3}$** = *{3971(1), 3054(2), 3938(1), 3925(1), 2807(2), 3924(1), 3935(1), 3684(1)}*; **TSR Key Set of $\boldsymbol{I_4}$** = *{3058(2), 3095(4), 3059(2), 3054(1), 3047(1)}*; **TSR Key Set of $\boldsymbol{I_5}$** = *{3058(1), 3057(2), 3095(1), 5736(1), 2797(1), 3054(1), 5704(1), 3059(2), 5703(1), 5737(1), 2791(1), 5689(2), 5741(1), 2805(1), 5438(1), 5702(1), 5697(1)}* and **TSR Key Set of $\boldsymbol{I_6}$** = *{2784(1), 2785(1), 3036(1), 5682(1), 1025(1), 2773(1), 5432(1), 2776(1), 5427(1), 5687(1), 2791(1), 3059(1), 5689(2), 3057(1), 5741(1), 2805(1), 5438(1), 5702(1), 5697(1)}*.

A B-tree of order *r*=4 is constructed to store the distinct keys. For each distinct key *k*, a list of indices of images, the key sets of which contain *k* is worked out along with its frequency in the key set and then attached to the B-tree for later retrieval. For example, for the key *k*=3095, the list of the image indices obtained are, *1*, *2, 4* and *5* with frequencies *4, 1, 4* and *1* respectively. The entire B-tree constructed for the example is as shown in Fig.3

4.2 Retrieval of similar symbolic images from SID

Consider a query image Q_1 (Fig. 4) consisting 3 distinct iconic objects with multiple instances of the iconic objects 2 and 4. The TSR existing among the components of Q_1 is perceived and the TSR keys are computed. The set of the computed TSR keys *{3971(1), 3054(2), 3938(1), 3925(1), 2807(2), 3924(1), 3935(1), 3684(1)}* describes the query image Q_1.

In order to retrieve the images similar to Q_1, access through the B-tree in search of each key *k* of Q_1 and extract the lists of image indices and frequencies attached to *k*. For the key 3971, the extracted list consists of only the image index 3 with frequency 1 i.e., (3, 1). The difference in frequency of the key 3971 in the image corresponding to the index 3 is 0 and is represented by a pair (*i, d*) = (*3, 0*). Similarly the lists of pairs obtained for the remaining keys are as follows.

For k=3054, ((3, 0), (4, -1), (5, -1)); For k=3938, ((3, 0)); For k=3925, ((3, 0)); For k=2807, ((3, 0)); for F=3924, ((3, 0)); For k=3935, ((3, 0)); For k=3684, ((3, 0)); When all these lists are subjected to intersection, the resultant is only the image index *3* and thus the symbolic image corresponding to *3* is classified as a first level similar image. The symbolic images corresponding to the index *4* and *5* are classified into second level similar images. The minimum difference in frequencies d_{min} for the image with index *3* is *0* as it is the minimum difference for all keys. The maximum difference d_{max} is also *0* as itself is the maximum. Hence, the image with index *3* has the interval [0, 0]. Being in the first level of similarity and having the interval [0, 0], since the number of distinct keys in Q_1 (U_{Q1}=8) and I_3 (T_3=8) corresponding to the index *3*are equal, Q_1 and I_3 are exactly similar.

On the other hand, the image indices *4* and *5* are extracted for only the key 3054. Hence, the difference is -1, itself will be the minimum and maximum for both. Thus, the interval is [-1, -1]. Since, both the images are at the second level of similarity, and the interval happens to be [-1, -1] the images I_4 and I_5 are classified as the images having a overlap. Since I_4 and I_5 are extracted for only one key, one can notice *Q* and I_4, I_5 have only three common objects with same triangular spatial relationship.

An experiment, with each symbolic image shown in Fig. 2 and Fig. 4 as query image is performed. The weight ordering of the similar images is shown in Table-1.

Table 1: Similarity ordering of images

Query	First level similar images				Second level similar images	
	Exact image	Containing query	Contained in query	Having overlap with query	Contained in query	Having overlap with query
I_1	I_1	I_4		I_2, I_5	-	-
I_2	I_2	-	-	-	-	I_1,I_4,I_5,I_6
I_3	I_3	-	-	-	-	I_4,I_5
I_4	I_4	-	-	-	I_1	I_2,I_3,I_5,I_6
I_5	I_5	-	-	-	-	I_1,I_2,I_3,I_4,I_6
I_6	I_6	-	-	-	-	I_2,I_4,I_5
Q_1	I_3	-	-	-	-	I_4,I_5
Q_2	-	-	I_1	I_2,I_4,I_5		

5. Discussion and conclusion

In this paper, we have made an attempt in devising a novel method to take care of multiple instances of iconic objects in an image. The proposed model is invariant to image transformations and the retrieval scheme requires $O(U_Q \log_r n)$ search time where U_Q is the number of distinct TSR keys in the query image and n is the total number of symbolic images stored in the B-tree. The designed model can be adapted easily for dynamic image database without any additional computational burden. Ranking of each extracted image, based on a proximity measure is our future goal.

References

[1] Chang C. C., 1991, Spatial Match Retrieval of Symbolic Pictures, Information Science and Engineering 7, 405-422.

[2] Chang C. C and Lee S. Y., 1991, Retrieval of Similar Pictures on Pictorial Database, Pattern Recognition, 24(7), 675-680.

[3] Chang C. C and Wu T. C., 1995, An exact Match Retrieval Scheme Based upon Principal Component Analysis, Pattern Recognition Letters 16, 465-470.

[4] Chang S. K., Shi Q. Y and Yan C. W., 1987 Iconic Indexing by 2D-Strings, IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5), 413-418.

[5] Guru D. S and Nagabhushan P., 2001, Triangular Spatial Relationship: A New Approach for Spatial Knowledge Representation, Pattern Recognition Letters 22, 999-1006.

[6] Lee S. Y and Hsu F. J., 1990, Picture Algebra for Spatial Knowledge Representation for Image Database Systems, Pattern Recognition, 23(10), 1077-1087.

[7] Lee S. Y and Hsu F. J., 1992, Spatial Reasoning and Similarity retrieval of Images Using 2D C-String Knowledge Representation, Pattern Recognition, 25(3), 305-318.

[8] Lee S. Y., Shan M. K and Yang W. P., 1989, Similarity Retrieval of Iconic Image Databases, Pattern Recognition, 22(6), 675-682.

[9] Petraglia G., Sebillo M., Tucci M and Tortora G., 1996, A Normalized Index for Image Databases, In Intelligent Image Database Systems, ed. Chang S. K., Jungert E and Tortora G., World Scientific, Singapore.

[10] Sabharwal C. L and Bhatia S. K., 1995, Perfect Hash Table Algorithm for Image Databases Using Negative Associated Values, Pattern Recognition, 28(7), 1091-1101.

[11] Sabharwal C. L and Bhatia S. K., 1997, Image Databases and Near Perfect Hash table, Pattern Recognition, 30(11), 1867-1876.

[12] Soffer A and Samet H., 1996, Handling multiple instances of symbols in pictorial queries by image similarity, Proc. 1st Int. Worshop on Image databases and Multi media search, Amstedam, 51-58.

[13] Wu T. C and Chang C. C., (1994), Application of Geometric Hashing to Iconic Database Retrieval, Pattern Recognition Letters 15, 871-876.

[14] Zhou X. M and Ang C.H., 1997, Retrieving Similar Pictures from Pictorial Database by an Improved Hashing Table, Pattern Recognition Letters 18, 751-758.

[15] Zhou X. M., Ang C. H and Ling T. W., 2001, Image Retrieval Based on Object's Orientation Spatial Relationship, Pattern Recognition Letters 22, 469-477.

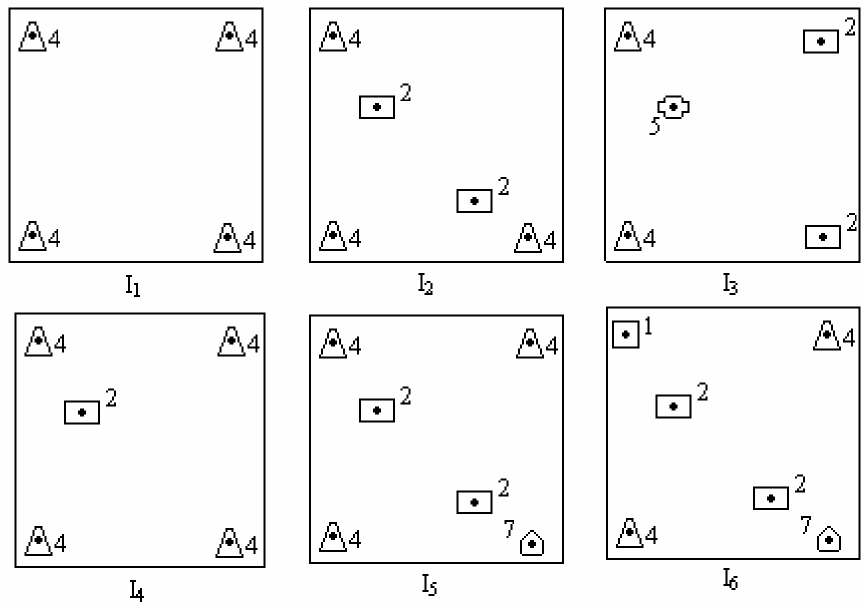

Fig. 2 Symbolic images as example

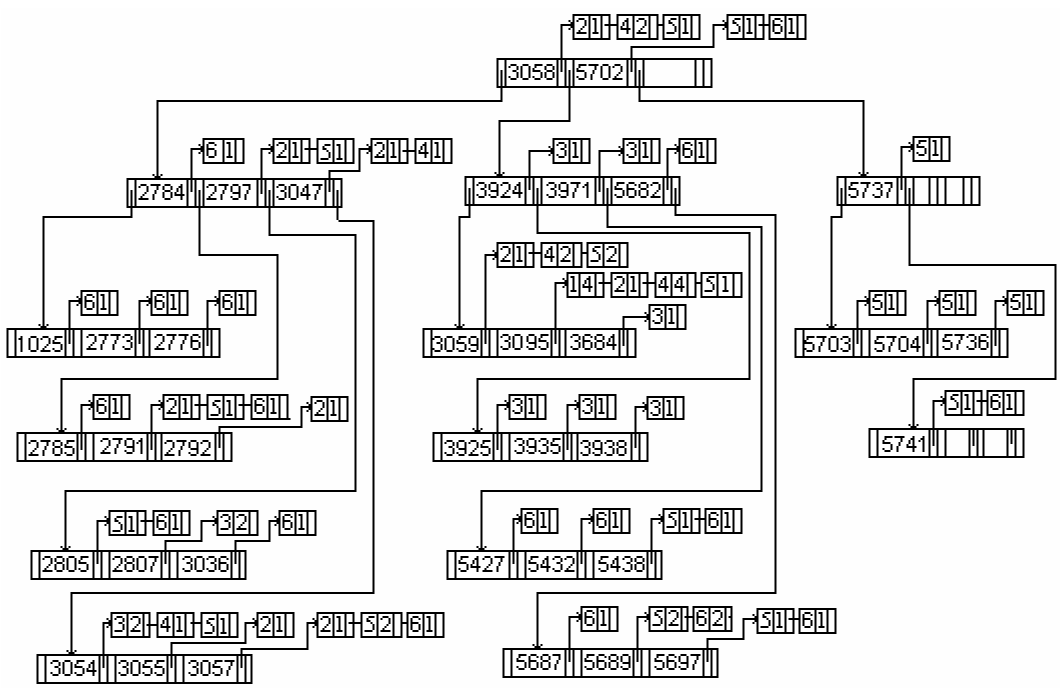

Fig. 3 B-tree representation

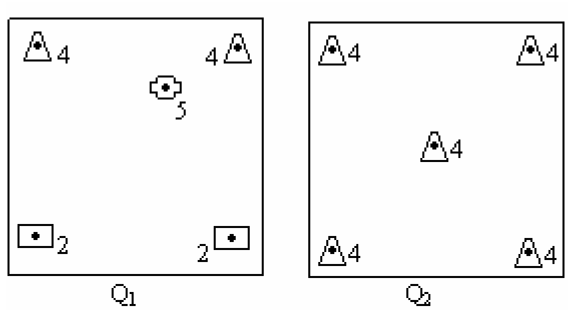

Fig. 4 Query images