Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 2068–2085
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Symbolic image indexing and retrieval by spatial similarity: An approach based on B-tree

P. Punitha[a], D.S. Guru[b,*]

[a]Department of Computer Science, University of Glasgow, Glasgow, UK
[b]Department of Studies in Computer Science, University of Mysore, Manasagangothri, Mysore, Karnataka, India

Received 27 December 2006; received in revised form 13 June 2007; accepted 14 September 2007

Abstract

In this paper, the problem of indexing symbolic images based on spatial similarity is addressed. A model based on modified triangular spatial relationship (TSR) and B-tree is proposed. The model preserves TSR among the components in a symbolic image by the use of quadruples. A Symbolic Image Database (SID) is created through the construction of B-tree, an efficient multilevel indexing structure. A methodology to retrieve similar symbolic images for a given query image is also presented. The presented retrieval model has logarithmic search time complexity. The study made in this work reveals that the model bears various advantages when compared to other existing models and it could be extended towards dynamic databases. An extensive experimentation is conducted on various symbolic images and also on the ORL and YALE face databases. The results of the experimentation conducted have revealed that the proposed scheme outperforms the existing algorithms and is of practical relevance.
© 2007 Elsevier Ltd. All rights reserved.

Keywords: Symbolic image; Spatial relationship; Triangular spatial relationship; B-tree; Image indexing; Symbolic image database; Similarity retrieval; ORL face database; YALE face database

1. Introduction

Processing images representing information only at pixel level to have interactive response to high-level user queries is not economically viable if not technologically impossible in a database environment where the number of images tends to be large [1]. Therefore, given an image representing information at pixel level (lower level), various image processing and understanding techniques are used to identify the domain objects/components and their locations in the image along with the co-ordinates of their centroids. An image is then obtained by associating a unique name and a meaningful graphic icon with each domain object identified. Intuitively, a domain object is a semantic entity (with respect to an application) contained in an image. For example, in the interior design application, the various furniture and decorative items in an image constitute the domain objects [1]. An image of human face includes eyes, nose, mouth, ears and jaw as the domain objects [1]. As another example, minutiae points with class labels and additional information such as their orientation form domain objects in fingerprint images [2]. At the pixel or physical level, a domain object is defined as a subset of the image pixels. The set of objects thus identified form the group of generic components. These iconic objects are generic in the sense that all images can be described as different combinations of these iconic objects. Encoding each iconic object present in an image by the respective label produces the corresponding symbolic image/ iconic image/logical image [3]. The symbolic image can be regarded as an abstract physical image, while the physical image is the real image itself. Though this task of transforming a physical image into corresponding symbolic image is computationally expensive and difficult, it is performed only once at the time of image insertion into the image database. Moreover this task may be carried out in a semi-automated (human-assisted) scheme or in a fully automated scheme depending upon the domain and complexity of images [1,4]. However, the transformation of images into corresponding symbolic images is kept beyond the scope of this paper, as in itself is a challenging

* Corresponding author. Tel.: +91 821 2419557; fax: +91 821 2510789.
E-mail addresses: punithaswamy@dcs.gla.ac.uk (P. Punitha), dsg@compsci.uni-mysore.ac.in, guruds@lycos.com (D.S. Guru).

0031-3203/$30.00 © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.09.012

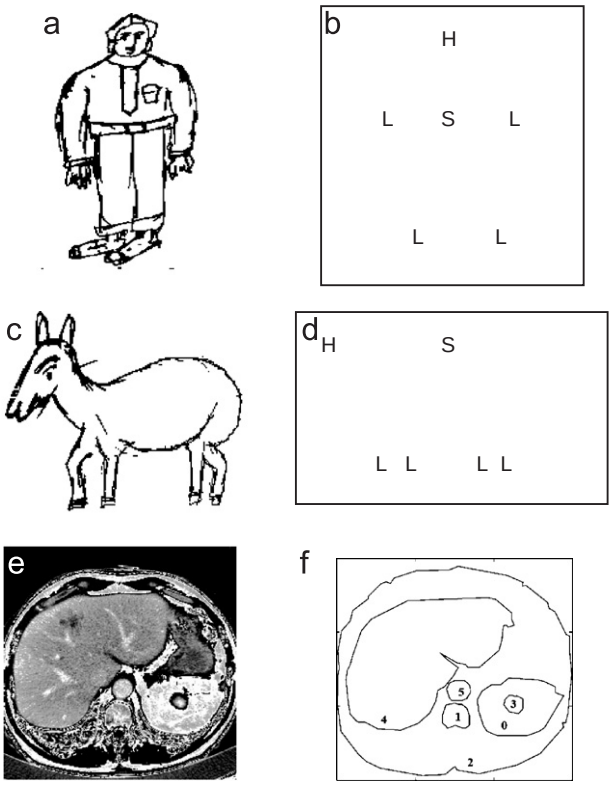

Fig. 1. (a–c) Three images. (d–f) Respective symbolic images (intermediate representation).

research topic. Nevertheless, a few methods to achieve the same can be found in literature; specifically few from our own group [3,5]. For example, images shown in Fig. 1(a–c) are three real images while the images shown in Fig. 1 (d–f) are their respective symbolic images.

An efficient image archival and retrieval system (IARS) is characterized by its ability to retrieve relevant images based on their visual and semantic contents rather than using simple attributes or keywords assigned to them [6]. Thus, it is necessary to support queries based on image semantics rather than mere-pixel-to-pixel matching [7]. An IARS should therefore allow adequate abstraction mechanisms for capturing higher level semantics of images in order to support content addressability as far as possible [7]. That is, for two images to be similar, not only the shape, color and texture properties of individual image regions must be similar, but also they must have the same arrangement (i.e., spatial relationships) in both the images. In fact, this is the strategy, which is generally being employed by human vision system most of the times. Thus, effective method of representing images depends on the perception of knowledge embedded in images in terms of objects/components (generally known as elements) present in them along with their topological relationships. The perception of spatial relationships existing among the significant elements of an image, helps in making the image database system more intelligent, fast and flexible [8,9]. Thus, the gist of the problem lies in developing such an efficient intelligent IARS.

In this paper, we propose a novel spatial knowledge based mechanism to represent and retrieve symbolic images in and from a symbolic image database (SID). A SID is a system, which consists of a large amount of image data and their related information represented by both symbolic and physical images [10]. In this paper, the problem of indexing and retrieval of symbolic images based on spatial similarity is addressed. A model based on triangular spatial relationship (TSR) [10], and B-tree is proposed. The proposed model preserves TSR among the components in a symbolic image by the use of quadruples. A SID is created through the construction of B-tree, an efficient multilevel indexing structure. A methodology to retrieve similar images for a given query image is also presented. The presented retrieval model has logarithmic retrieval time complexity. The accuracy of the proposed method is verified through several experiments. The suitability of the proposed method in face recognition is also explored and promising results have been obtained. The proposed method belongs to the class of those methods in which the spatial entities are (can be) assimilated to points (usually centroids). Handling topological relationships between objects, which is also a challenging area in the field of image retrieval, is kept beyond the scope of the current study. The study made in this work reveals that the proposed model bears various advantages when compared to other existing models and it could be easily extended towards dynamic databases.

The remaining part of the paper is organized as follows. In Section 2, a brief survey on the related work is given. An overview of TSR is given in Section 3. Section 4 proposes a modification to the TSR to accommodate even collinear objects. Section 5 proposes a novel indexing method of representing symbolic images by the use of TSR and the corresponding retrieval algorithm. A complete illustration of the proposed methodology is given in Section 6. The results of the experiments conducted to establish the suitability of the proposed methodology by considering some standard databases are given in Section 7. Sections 8 and 9 follow with discussions and conclusion, respectively.

2. Related work

Retrieval by spatial similarity has received considerable attention from many researchers and many methodologies have been proposed. A detailed survey can be found in Ref. [11]. The insinuated methodologies can be broadly classified into two categories: value oriented and object oriented models. The value oriented models viz., pixel based [12], quadtree based [13], R-tree based [14], which are shown to be insufficient to deal with complicated operations in an intelligent, fast and flexible manner [15], as they work on low level image features, while the other alternative object oriented models receive considerable attention. The object oriented methodologies can be further classified as string based, hash based, graph based, index based and matrix based approaches.

Chang et al. [16] introduced a data structure called 2D string. Many methodologies [9,15,17–24] based on the concept of 2D string can be found in literature. However, none of these image representation schemes are invariant to image transformations.

Using the concept of 2D string, in order to retrieve similar symbolic images from a SID, algorithms [25–28] based on longest common subsequence matching were proposed. Lee et al. [25] proposed three measures Type-0, Type-1 and Type-2 similarities, for image retrieval. However, Drakopoulos and Constantopoulos [29] reported that the subsequence matching fails for some instances. To overcome such false drops, Petrakis [30] devised an alternative algorithm. The process of string matching takes non-deterministic-polynomial time complexity thus making these algorithms inefficient in practice.

Chang and Lee [31] introduced a relative co-ordinates oriented symbolic string (RCOS string), but it is computationally expensive. Petraglia et al. [32] introduced the concept of virtual image. Virtual images though are invariant to translation and scaling, they are sensitive to rotation.

Petraglia et al. [33] discovered the 2D-R string representation based on the 2D-C string mechanism [21] as a rotation invariant concept. The method is sensitive to center of the image, as change in the center of an image results with different 2D-R string representations for the image. In addition, two dissimilar images are misjudged as similar and two similar images are misjudged as dissimilar [34].

To overcome the problem due of shift in the rotation center, Huang and Jean [34] proposed rotation invariant spatial knowledge representation called RS string. However, it is computationally expensive as the method entails the generation of all possible RS strings for matching purpose. Also, the objects in the query image are expected to be a subset of the objects in the database images. Petraglia et al. [32] also introduced the transformation invariant approach, the r-virtual image which is similar to virtual images. This is based on the preliminary work of 2D-R string. But it is sensitive to translation.

Gudivada [1] proposed ΘR-string and a similarity retrieval algorithm called $\mathrm{SIM_G}$ to recognize translation, scale and rotation variants and also the variants of the image generated by an arbitrary composition of translation, scale and rotation transformations. ΘR-string is obtained by ordering the objects in increasing order based on objects' angle of inclination with the center of the image subtended with respect to the positive X-axis. The objects are also associated with the information about the preceding and following objects. The $\mathrm{SIM_G}$ algorithm compares the ΘR-strings associated with the query and the database image and returns a similarity value which is comprised three factors, object factor: number of common objects, spatial factor: degree of importance of spatial relationship and scale factor: function of right and left neighbor distances. However, ΘR-string did not take into account the existence of multiple objects in images.

In Sciascio et al. [35] proposed $\mathrm{SIM_L}$ as an improvement to ΘR-string. The $\mathrm{SIM_L}$ extracts oriented angles between every pair of objects with respect to the line joining a third object. The algorithm then computes the similarity as a function of maximum difference between corresponding angles.

Gudivada and Raghavan [36] proposed two rotation invariant models to represent symbolic images. The spatial relationships in a symbolic image were represented as edges in a weighted graph. Two spatial similarity measures SIM, $\mathrm{SIM_R}$, were computed. Both of these algorithms being model based have quadratic time complexity in terms of total number of objects in both the database and the query image.

Contribution by Segupta and Boyer [37] is an hierarchical organization of graphs. However, query matching can become very time consuming. In addition, the method gives an approximate match and may miss an actual match at certain times. Similarly, the method of Messmer and Bunke [38] relies on the idea of computing distorted copies of the stored graphs which are represented in a decision tree. The degree of distortion has to be known in advance. The size of the decision tree increases exponentially with the size of the stored graphs and with the degree of distortion. Due to its large space requirements, this method can not be used in image databases.

The method $\mathrm{SIM_{DTC}}$, proposed by El-Kwae and Kabuka [39], used spatial orientation graph for representing directional spatial relationships. Each edge in the graph connects two objects in the image. The similarity was computed between two images as a function of common objects, closeness of directional spatial relationship and closeness of topological spatial relationships. The algorithm also uses rotation correction to obtain more accurate similarity. However, the retrieval can be inefficient as comparisons often involve time intensive operation such as graph matching [8,40]. Indexed search is many times faster than sequential scanning even for large answer sets (with more than 10,000 images) [41,42].

The method by Petrakis and Faloustsos [43] is based on the combined concept of attributed relational graphs (ARGs) [41,42] and R-trees. The method proposed by Korn et al. [44] is based on R-tree index structure.

The work of Papadias et al. [45] is to avoid the high computational complexity of sequential search for structural queries in image databases. Both the database images and the queries are mapped to regions in a multidimensional space and are indexed by R-trees. Except of its high complexity, this method treats only special types of relationships and object properties and it assumes that all images and queries are at a known scale and orientation. In addition, it cannot handle even translation.

The methods referred above require an expensive preprocessing step for building a tree index structure. They work well only for special forms of graphs. They are not designed for ranking of images based on their similarity. Their emphasis is on the design of efficient algorithms for computing the error correcting isomorphism between a query and a set of stored ARGs but not on similarity retrieval of images.

The method by El-Kwae and Kabuka [46] is a multilevel indexing technique based on bit signatures. Petrakis [42] emphasizes on showing how a spatial access method (SAM) (eg., R-tree) can be used to answer queries on data entities which are represented by multiple multidimensional points. The method for ARG matching finds the optimal solution but it has exponential time (and space) complexity in the worst case (i.e., tree matching can grow exponentially for large graphs). Polynomial time algorithms do exist but, either they are approximate or they require exponential space [47]. The method [40] proposes a hierarchical clustering of ARGs into classes of similar ARGs. The main advantage of this method is that it is well suited for

searching in image databases since it avoids matching of the query with stored ARG. The main disadvantage is the requirement of an expensive preprocessing step for building an ARG index structure. The method [43] is restricted to special kind of ARGs with a small number of nodes or requires labeled ARGs. In addition, it may require exponential space and do not guarantee low complexity matching.

Geometric methods use inherent geometric features in an image for its representation such as those based on the weighted center of mass (WCOM) [48]. In Ref. [48], each object is assigned a feature vector based on its area, length and the angle from its center to the image center of mass and neighboring objects. To avoid high dimensionality problem, only the first four significant objects were considered for indexing. This might lead to true dismissal, if any of the significant objects is not present in the query image.

Zhou et al. [49] introduced an image retrieval scheme based on objects' orientation spatial relationship. For an image with 'm' distinct components, the orientation spatial relationship is preserved in a matrix of order $m \times m$. Unlike other index based methods this is invariant to image transformations, but, even for considerably small number of images, the space complexity is very high and hence retrieval takes longer. The methods [50–54] are the other promising indexing methods for image retrieval based on hashing. The algorithm [55] is based on a slight modification of a perfect hash function defined by Cichelli [56]. This algorithm was shown to have several limitations [51].

Sabharwal and Bhatia [54] presented an enhancement to perfect hash table to allow insertion and deletion in hash table. However, the best known algorithm for the construction of the perfect hash table itself is of exponential time complexity [52]. Updating a static image database requires reconstruction of the entire hash table from the beginning, which is the real bottleneck problem.

Wu and Cheng [57] presented an efficient access method for retrieving similar images from iconic databases using G-tree [58]. The method preserves the merits of G-tree, a combined feature of both Grids and B-tree. But, since it is based on Grid tree, it fails to perceive invariant spatial relationships among iconic objects in an image and hence is not invariant to image transformations, even to translation, a simple unavoidable transformation.

It is evident from the above discussion that the index based models would be the better models than even any hash table based models. This is due to the fact that a hash table based technique suffers from computational bottleneck due to exponential time complexity of hash table construction [55].

Chang [59] formulated an indexing structure called 9DLT (nine-directional lower triangular) matrix to logically represent symbolic images with pair-wise spatial relationships among the components in a symbolic image with the help of nine directional codes. Chang and Wu [60] have proposed an exact match retrieval scheme based upon principal component analysis to retrieve symbolic images from SID by the use of the 9DLT matrix. However, two or more different images may be associated with the same principal component direction [3]. Zhou and Ang [53] suggested a better scheme based on the 9DLT matrix for retrieving symbolic images by an improved hashing table. The 9DLT matrix was represented by a set of triples. For each triple, a hashing value is found and stored.

Efforts were also made for spatial match retrieval using signature file techniques in combination with 2D strings [61] and 9DLT matrix [62]. Chang et al. [63] proposed an alternate similarity match retrieval scheme. The method is claimed to achieve better retrieval effectiveness than 9DLT based approaches. However, the scheme requires two different representations, one to support exact match retrieval and the other to support similarity retrieval. Chang and Yang [64] proposed a prime number based matrix strategy for indexing symbolic images by combining the advantages of 2D-C string and the 9DLT matrix. However, this model is not invariant to image transformations and since the spatial relationships existing among the components present in an image are represented by the use of a matrix of order $m \times m$, the memory requirement even for a small number of images is very high and hence encumbers retrieval. Unique-ID based [65], generalized prime number based [66] and bit pattern based [67] matrix strategies for more efficient iconic indexing of symbolic pictures were also proposed. However, the matrix based approaches, specifically, the 9DLT matrix based approaches suffer from considerable problems. A detail descriptions of the 9DLT matrix [59] and the 9DLT based exact match scheme [60] along with their problems to suit real pragmatic situations are discussed elaborately in Ref. [68].

Motivated to alleviate the problems of the 9DLT approach, altogether new concepts called TSR [10] and Direction of Reference [68,69] are proposed as improved, invariant spatial relationships to take care of image transformations. An exact match retrieval based on TSR was also proposed [70]. A mechanism of indexing symbolic images based on TSR was kindled in Ref. [71]. However, in Ref. [71] just an idea about how TSR can be exploited for creating indexing for fast retrieval of symbolic images is theoretically presented through a simple illustrative example.

In this paper, we pool previous ideas [10,71] together to realize a complete system for indexing symbolic images to support faster retrieval on large databases. Original TSR is modified to overcome its limitations in perceiving spatial relationship among collinear objects. Various existing models are studied and their complexities are presented. The proposed B-tree based model has been validated for its scalability and practicality on various databases including face databases. The comparative analysis made reveals that the proposed model outperforms the other existing models, in addition to supporting dynamic databases.

3. TSR: an overview

A TSR is formally defined by connecting three non-collinear components in a symbolic image as follows [10].

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,

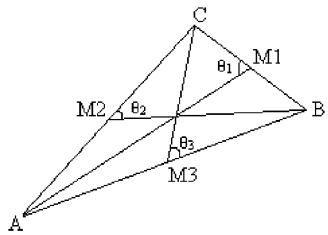

Fig. 2. Triangular spatial relationship.

and θ_3 as the smaller angles subtended at M_1, M_2, and M_3, respectively, and are shown in Fig. 2. 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. Thus, it was recommended to choose only one of those, which satisfies the following criteria.

If $(L_{i1}, L_{i2}, L_{i3}, \theta)$ is the quadruple to be chosen, then the labels L_{i1}, L_{i2}, and L_{i3} must satisfy one of the following conditions.

1. The labels L_{i1}, L_{i2}, and L_{i3} are distinct and $L_{i1} > L_{i2} > L_{i3}$.
2. $L_{i1} = L_{i2}$ and $L_{i3} < L_{i1}$.
3. $L_{i1} > L_{i2}$ and $L_{i2} = L_{i3}$ and $\mathrm{Dist}(\mathrm{Comp}(L_{i1}), \mathrm{Comp}(L_{i2})) \geqslant \mathrm{Dist}(\mathrm{Comp}(L_{i1}), \mathrm{Comp}(L_{i3}))$.
4. $L_{i1} = L_{i2} = L_{i3}$ and $\mathrm{Dist}(\mathrm{Comp}(L_{i1}), \mathrm{Comp}(L_{i2})) \geqslant M$, where

$$M = \mathrm{Max}(\mathrm{Dist}(\mathrm{Comp}(L_{i1}), \mathrm{Comp}(L_{i3})), \mathrm{Dist}(\mathrm{Comp}(L_{i2}), \mathrm{Comp}(L_{i3}))).$$

Here, $\mathrm{Dist}(A, B)$ is a function which computes the Euclidean distance between the midpoints of the components A and B. $\mathrm{Max}(a, b)$ is a function denoting the maximum among a and b and $\mathrm{Comp}(L)$ indicates the corresponding component (which is dependent on its position in the quadruple), the label of which is L.

It is guaranteed that, even if more than one possible permutation of the components satisfies the above-stated condition, the corresponding quadruples are one and the same.

In other words, the TSR among any three non-collinear components A, B and C is preserved by a quadruple $(L_{i1}, L_{i2}, L_{i3}, \theta)$, where the sequence of L's satisfies one of the above-stated conditions and θ is the smaller angle subtended at the midpoint of the components, the labels of which are L_{i1} and L_{i2}, due to the line joining that midpoint and the centroid of the remaining component, the label of which is L_{i3}. The θ is given by

$$\theta = \begin{cases} \theta_1 & \text{if } \theta_1 \leqslant 90^\circ, \\ 180 - \theta_1 & \text{otherwise} \end{cases} \tag{1}$$

here,

$$\theta_1 = \cos^{-1}((S_1^2 - S_2^2 - S_3^2)/(2 * S_2 * S_3)),$$

where

$$S_1 = \mathrm{Dist}(\mathrm{Comp}(L_{i1}), \mathrm{Comp}(L_{i3})),$$

$$S_2 = \mathrm{Dist}(\mathrm{Comp}(L_{i1}), \mathrm{Comp}(L_{i2}))/2,$$

$$S_3 = \mathrm{Dist}(\mathrm{Mid}(\mathrm{Comp}(L_{i1}), \mathrm{Comp}(L_{i2})), \mathrm{Comp}(L_{i3})).$$

Here, $\mathrm{Mid}(X, Y)$ denotes the midpoint of the line joining the centroids of X and Y.

The concept of TSR is proved to be invariant to image transformations viz., translation, rotation, scaling and flipping. More details can be found in Ref. [10].

4. Modified TSR

It shall be noted that the TSR is defined to be perceived among three components which are non-collinear. Such a restriction was imposed in their work [10] as the computation of θ value in case of collinear components is impossible. However, in general presence of three components in a collinear manner could even be an added knowledge for efficient retrieval. So, here we propose a modification to the concept of TSR, to take into account even the collinear components while representing images in a SID.

Since, the conditions stated in Section 3 for choosing unique quadruple out of six possible quadruples are independent of the θ value and are dependent only on the labels of the iconic objects and their mutual distances, we can always obtain such a unique sequence of labels of any three iconic objects/components irrespective of their collinear property. Once the unique sequence say L_{i1}, L_{i2}, L_{i3} is obtained, the θ value is computed as explained in the previous section if the components are non-collinear. In case, if they are collinear, we check if the component bearing the label L_{i3} lies in between the other two components. If it lies in between the other two components then the θ is assumed to be 90°, otherwise, it is one of the end points of the line joining all three components and the θ is assumed to be 0° (zero degree).

Thus, all in all, the modified TSR considers three components and preserves spatial relationship among them by a quadruple $(L_{i1}, L_{i2}, L_{i3}, \theta)$, which satisfies the conditions stated in the previous section, where θ is computed as follows:

$$\theta = \begin{cases} \theta \text{ computed by Eq. (1)} & \text{if the components are non-collinear,} \\ 90^\circ & \text{if the components are collinear and the } Comp(L_{i3}) \text{ comes in between } Comp(L_{i1}) \text{ and } Comp(L_{i2}), \\ 0^\circ & \text{otherwise.} \end{cases}$$

Based on this modified version of TSR, we propose, in the next section, a model for archival and retrieval of images. Unless otherwise it is specifically mentioned, the word TSR refers this modified version of TSR in the remaining part of the paper.

5. The proposed indexing scheme

This section describes a novel indexing scheme to represent symbolic images in a SID and also a corresponding retrieval scheme.

5.1. Representation of symbolic images in SID

Let $\{I_1, I_2, I_3, \ldots, I_n\}$ be a set of n symbolic images to be archived in a SID and let $L_1, L_2, L_3, \ldots, L_m$ be the labels of m distinct generic iconic objects called components. Therefore, each symbolic image I_i, $\forall i = 1, 2, 3, \ldots, n$ is said to contain $m_i \leqslant m$ number of components.

In order to make the representation scheme invariant to image transformations, we recommend to perceive the TSR existing among all components present in a symbolic image and then to preserve the TSR 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.

For this purpose, one may think of multidimensional indexing structures like multidimensional binary trees [72], KDB trees [73], Grids [74,75], G-tree [58], multiattribute [76] hashing and BD-tree [77]. Another technique for organizing multidimensional data is multiattribute hashing [76]. The BD-tree [77] guarantees a utilization of at least 50% while the G-tree makes no such guarantee and its absolute worst case performance can be very poor. In KDB trees handling deletions in hand is tougher and moreover if the structure is very dynamic, then the utilization can deteriorate over time.

Other hashing based methods such as those presented in Refs. [78,79] have similar shortcomings. SAMs like R-tree [13], SR-tree [80] require that the number of dimensions is known in advance and it is fixed. Any SAM (e.g., R-tree, G-tree, SR tree, R+-tree, R*-tree, Hilbert R-tree, compact KDB tree and G-tree), which is faster would make the method using it even faster. However, the point accessing methods, R-tree, R+-tree, R*-tree, Hilbert R-tree, compact KDB tree and G-tree, which cluster a point set into multidimensional regions concentrate on absolute position in multidimensional space rather than relative positions. Most of these tree structures are improvements of B-tree and were adopted to suit very high dimensional data. Nonetheless, Petrakis and Faloustsos [43] state that the performance of such trees (specially R-tree) can be improved by using dimensionality reduction. Thus, we use the same basic B-tree as the indexing structure, but is slightly modified for our requirement.

Still, B-tree does not support storage of multivalued data such as quadruples. Thus, 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 key K corresponding to the quadruple is computed as [71]

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

where D_θ is the number of slices/classes, the continuous interval type domain [0…90] associated with θ 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.

Splitting of the continuous interval type domain $[0 \ldots 90]$ associated with θ into D_θ classes is suggested because sometimes due to transformation of images there could be a variation in the fourth component of the quadruple, and during matching/retrieval there could be a mismatch between the quadruples representing the same components with the same spatial relationship. In order to take care of the variations, in the fourth component, we suggest splitting the continuous interval type domain $[0 \ldots 90]$ into D_θ classes.

It can be proved that the associated keys for any two different quadruples are distinct and unique. For each image to be stored in the SID, the quadruples are obtained and the unique keys are computed. Let T_i, $i = 1, 2, 3, \ldots, n$ be the total number of keys generated for the image I_i, $i = 1, 2, 3, \ldots, n$. The total number of keys computed for each image along with the image index is stored in an index table for later usage during retrieval. The quadruples and hence keys computed for all the images are not necessarily disjoint sets. There could be many keys in common between any two or more images depending on the components and their spatial scattering.

Let N represent the total number of distinct quadruples generated due to all n symbolic images and let $\{K_1, K_2, K_3, \ldots, K_N\}$ be the set of corresponding keys. All these N keys are stored in a B-tree. Each key value is then attached with a list of image indices. The image indices that are attached to a key value (K) are the indices of images, which have the key K as one of the keys in their corresponding key set.

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

Algorithm: Proposed representation scheme
Input: $I_1, I_2, I_3, \ldots, I_n$-Set of symbolic images
Output: Symbolic Image Database (B-tree)
Method:

(a) For each symbolic image I_i do
  i. Apply TSR and obtain a set of quadruples preserving TSR among the components present in I_i and count the total number of quadruples T_i.
  ii. For each obtained quadruple compute a unique key using Eq. (2) and update the list of image indices associated with that key by inserting i as a new index.
  For end
(b) Obtain the set of all keys computed in (a).
(c) Create a B-tree of rank r containing all the keys present in the set obtained in (b).
(d) Attach each key, in the B-tree, with the respective list of image indices.
(e) Construct an index table consisting of image index, i and the total number of keys T_i in the corresponding image I_i.

Proposed representation scheme ends.

The added advantage of the proposed model is that, it can be easily extended towards dynamic databases, as it is very easy to include new iconic objects' labels without re-labeling of existing iconic objects. The insertion of new iconic objects into the database, results with different TSR quadruples and hence distinct TSR keys, which can be easily inserted into the B-tree. In addition, insertion of new symbolic images is also simple as it is just the insertion of set of keys into the existing SID and updating the associated image lists.

The following section presents the method of retrieving symbolic images from SID based on the proposed B-tree indexing scheme.

5.2. Retrieval of symbolic images from SID

In similarity retrieval the task is to retrieve from a SID those images that are similar to a given query image. Since, each symbolic image is just a collection of TSR quadruples, the image matching problem becomes the problem of subset matching between the TSR quadruples of the given query image and all the TSR quadruples of all the images stored in the database. If T_Q is the number of quadruples generated for the query image and T_M is the average number of quadruples that each model image is associated with, then the conventional matching algorithm requires $O(nT_M T_Q)$ search time. 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 matched images as follows.

Let Q be a query image and let T_Q be the number of TSR quadruples of the query image Q and hence there are T_Q number of keys generated. The B-tree is accessed through in search of each key. For each key of Q present in the B-tree the list of image indices corresponding to that key is extracted from the SID (B-tree in our case). From the extracted lists, the list of indices of the desired images for the query image Q is computed by the intersection of all those extracted lists. Since, we retrieve only images which are common in all the extracted lists due to the various query keys, the retrieved images are the images which have all the query keys in them and hence have the query image as their sub-image. Accessing through the B-tree in search of a key requires $O(\log_r N)$ search time, where N is the total number of keys stored in the B-tree and r is the order of the tree and hence the proposed retrieval scheme requires $O(T_Q \log_r N)$ search time.

The core problem for similarity retrieval is determining a similarity metrics that is efficient to compute and capture the essential aspects of similarity between two symbolic images. To this aim we introduce a formula that measures the similarity between an image and a query in terms of number of iconic objects and their spatial relationships and expresses it as a value in the range [0, 1].

Thus a retrieval algorithm should be able to rank the retrieved images based on the degree of similarity or dissimilarity. Hence, once the list of images is obtained the degree of similarity between each retrieved image I_i and the query image Q is computed as follows.

Let T_Q and T_I be the total number of keys in the query image and the retrieved model image, respectively, then the similarity between Q and I, $SV(Q, I)$ is computed as

$$SV(Q, I) = \frac{T_Q}{T_I}. \tag{3}$$

If the number of keys in both the query and a retrieved image is one and the same then the similarity value is 1 (high) and there is one to one correspondence between the sets of keys as the image I is extracted through intersection. On the other hand, if I has more number of keys than Q, then the similarity decreases. Thus the retrieved images can be sorted on the basis of the similarity value computed. The actual value of similarity is not so important but it is just used to rank the retrieved images.

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 symbolic images containing Q
Method:

(a) Obtain a set of quadruples preserving TSR among the components of Q.
(b) Compute a unique key for each quadruple using Eq. (2).
(c) Access through B-tree in search of each key and extract the list of image indices associated with that key.
(d) Compute the list of indices of similar images through intersection of those extracted lists.
(e) For each image obtained in (d), compute the similarity value using Eq. (3), and display the images in decreasing sequence according to their similarity values.

Proposed retrieval scheme ends.

6. A complete illustration with an example

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

6.1. Representation of symbolic images in SID

Let us consider $n = 5$ symbolic images shown in Fig. 3. Let {1, 2, 3, 4, 5, 6, 7, 8} be the set of symbolic labels of 8 distinct iconic objects. As explained in Section 4, the concept of TSR is employed on the image S_1 and the TSR existing among the components of S_1 is preserved by the use of quadruples. Since S_1 has got four components, the number of quadruples generated is ${}^4C_3$=4 and they are, {(4, 3, 1, 53.857273), (3, 2, 1, 66.972889), (4, 2, 1, 48.272072), (4, 3, 2, 85.386888)}.

Now the fourth component, which is a real value, of each quadruple is mapped into its class index (as suggested in Section 5.1). In this example, the θ-domain [0° ... 90°] is split into 18 classes/intervals of size 5°. Thus, the quadruples become {(4, 3, 1, 11), (3, 2, 1, 14), (4, 2, 1, 10), (4, 3, 2, 18)}.

Similarly, the sets of quadruples preserving TSR among the components of the remaining symbolic images S_2, S_3, S_4 and S_5 are generated.

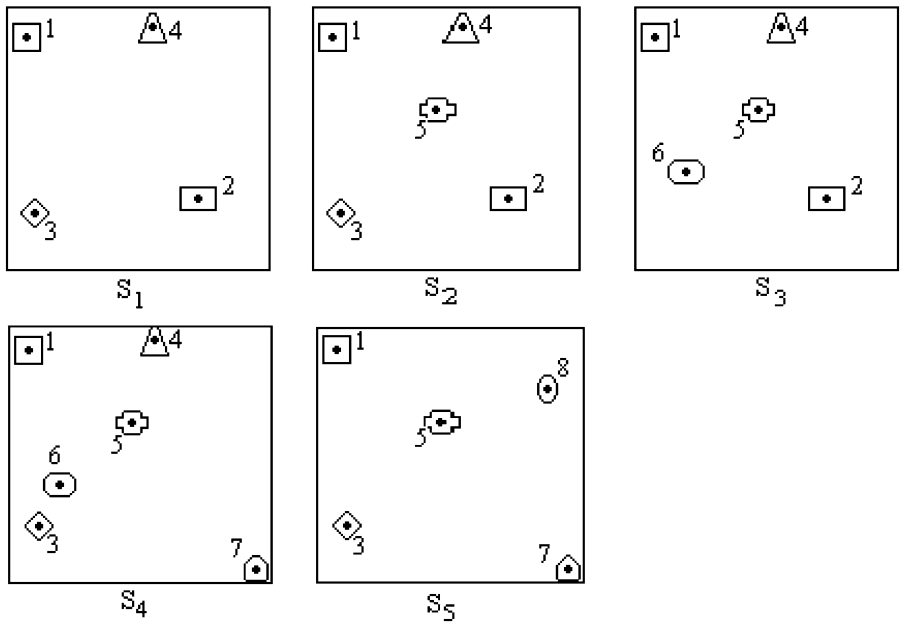

Fig. 3. Example symbolic images.

Table 1
The TSR quadruples and their corresponding TSR keys for the images shown in Fig. 3

Image	Quadruples	TSR keys
S_1	(4, 3, 1, 11) (3, 2, 1, 14) (4, 2, 1, 10) (4, 3, 2, 18)	3754 2461 3609 3779
S_2	(5, 3, 1, 17) (4, 3, 1, 11) (3, 2, 1, 14) (5, 4, 1, 18) (5, 2, 1, 3) (4, 2, 1, 10) (5, 4, 3, 5) (5, 3, 2, 14) (4, 3, 2, 18) (5, 4, 2, 8)	4912 3754 2461 5057 4754 3609 5080 4927 3779 5065
S_3	(6, 5, 1, 17) (6, 4, 1, 18) (6, 2, 1, 9) (5, 4, 1, 18) (5, 2, 1, 3) (4, 2, 1, 10) (6, 5, 4, 5) (6, 5, 2, 14) (6, 4, 2, 16) (5, 4, 2, 8)	6352 6209 5912 5057 4754 3609 6394 6367 6225 5065
S_4	(6, 3, 1, 6) (5, 3, 1, 17) (4, 3, 1, 11) (7, 3, 1, 12) (6, 5, 1, 17) (6, 4, 1, 18) (7, 6, 1, 9) (5, 4, 1, 18) (7, 5, 1, 3) (7, 4, 1, 12) (6, 5, 3, 3) (6, 4, 3, 1) (7, 6, 3, 16) (5, 4, 3, 5) (7, 5, 3, 16) (7, 4, 3, 17) (6, 5, 4, 5) (7, 6, 5, 11) (7, 6, 4, 14) (7, 5, 4, 6)	6053 4912 3754 7211 6352 6209 7640 5057 7490 7355 6374 6228 7683 5080 7539 7396 6394 7714 7699 7547
S_5	(5, 3, 1, 17) (8, 3, 1, 16) (7, 3, 1, 12) (8, 5, 1, 7) (7, 5, 1, 3) (8, 7, 1, 11) (8, 5, 3, 4) (7, 5, 3, 16) (8, 7, 3, 17) (8, 7, 5, 12)	4912 8367 7211 8646 7490 8938 8679 7539 8980 9011

For each distinct quadruple, a distinct and unique key is generated. Hence, each symbolic image can be described by a set of TSR keys. Table 1 presents the quadruples obtained for the images shown in Fig. 3 along with their respective TSR keys.

A B-tree of order $r = 4$ is constructed to store the distinct keys and each key in the B-tree is attached with its respective list of image indices. Fig. 4 is the B-tree constructed for the example.

In addition, an index table containing the total number of keys for each of the images to be stored is created. Fig. 5 shows the index table created for the example set shown in Fig. 3.

6.2. Retrieval of symbolic images from SID

Consider a query image consisting four iconic objects as shown in Fig. 6. According to the proposed retrieval algorithm (Section 5.2), the TSR existing among the components of Q is preserved by the set of quadruples {(5, 3, 1, 17), (4, 3, 1, 11), (5, 4, 1, 18), (5, 4, 3, 5)} with their fourth component being the respective θ-class index and the set of corresponding TSR keys is {4912, 3754, 5057, 5080}. This set, thus describes the query image Q.

In order to retrieve the images similar to Q, we access through the B-tree in search of each key obtained and then we extract the list of image indices attached to them. The lists of image indices extracted are:

corresponding to the key 4912: {2, 4, 5};
corresponding to the key 3754: {1, 2, 4};
corresponding to the key 5057: {2, 3, 4};
corresponding to the key 5080: {2, 4}.

Presence of a common image index in these extracted lists of image indices indicates that the image corresponding to such

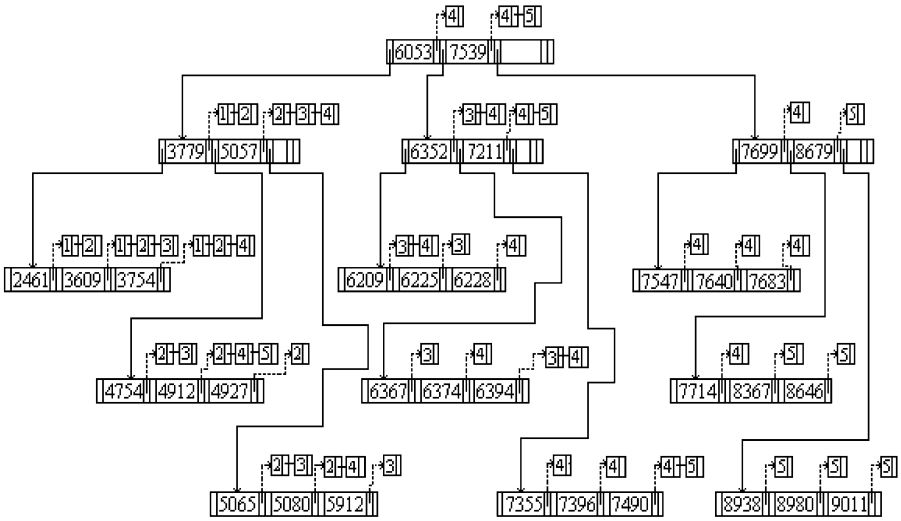

Fig. 4. A B-tree representation to the images shown in Fig. 3.

Image Index i	Total number of keys T_i
1	4
2	10
3	10
4	20
5	10

Fig. 5. The index table for the example images shown in Fig. 3.

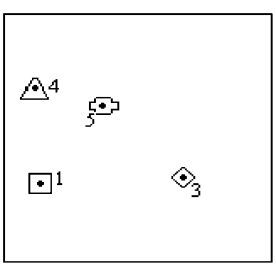

Fig. 6. Q a query image.

an index has got all the keys of the query image in its key set and hence, it is said to be a desired image containing the query image as the sub image. Therefore, the intersection of the extracted lists gives us the list of indices of the desired images.

So, only the images S_2 and S_4 that match the query image Q will be retrieved from the database. It can be noticed that the query image (Fig. 6) is a rotated and translated version of a sub-image of S_2 and S_4 of Fig. 3. Although the database can be very large and the corresponding B-tree can be very large too, we pull out only four lists of image indices (each corresponding to one key in the key set of the query image) from which the list of desired/similar images can be obtained.

After retrieving the desired images, we rank them based on the number of extra keys they have. We compute the similarity values $SV(Q, S_2)$ and $SV(Q, S_4)$ using Eq. (3)

$$SV(Q, S_2) = \frac{4}{10} = 0.4$$

and

$$SV(Q, S_4) = \frac{4}{20} = 0.2.$$

Thus, S_4 is less similar when compared to S_2 and hence the images are ordered as S_4 and S_2. Same thing is evident from Fig. 3 as the image S_4 has many more iconic objects than the image S_2, which are not in Q, the query image. We reiterate that the similarity values are used to only rank the images and the values themselves do not convey any semantics on similarity.

7. Experimental results

To uphold the effectiveness of the proposed methodology, we have conducted several experimentations on various test databases. We present here the results of only the experimentations conducted on the SIDs given in Refs. [35,36], and also the results of the experiment conducted on ORL real face image database.

7.1. Experimentation 1

In this experimentation, we have considered the set of images given in Ref. [36]. The TSR is employed on each image and the spatial relationship existing amongst the components is perceived and preserved in terms of quadruples. The θ domain [1..90°] is split into 18 classes of 5° intervals as explained in Section 5.1. The fourth components of the quadruples as suggested earlier are mapped onto their respective classes. A unique key is computed for each quadruple using Eq. (2). A total of 3420 keys were computed due to all 24 images. A B-tree of order $r = 4$ with five levels is constructed. An index table, (Table 2) consisting of the indices of all 24 images and their respective total number of keys is created.

Fig. 7 presents 24 basic query images. Each query image Q_i in Fig. 7 is formed by eliminating some iconic objects from the

image G_i of Gudivada and Raghavan [36]. Another set of 48 images are generated by subjecting each basic query image to arbitrary scaling of ±20 units. Randomly selected 48 images out of 72 images are subjected to arbitrary translation and another set of 48 images are generated. Now the query collection has a total of 120 images. For each of these 120 images, 152 arbitrarily rotated instances (±180°) are generated. The robustness of the proposed methodology in retrieving the desired images is tested by considering totally, 18,240 query images. It is observed that the desired images are retrieved from the SID for all the query images considered during experimentation. Table 3

Table 2
The index table consisting image index and the respective total number of keys for the images given in Ref. [36]

Image index	Total number of keys
G_1	10
G_2	10
G_3	10
G_4	10
G_5	4
G_6	4
G_7	4
G_8	4
G_9	84
G_{10}	84
G_{11}	84
G_{12}	84
G_{13}	84
G_{14}	84
G_{15}	286
G_{16}	286
G_{17}	286
G_{18}	286
G_{19}	286
G_{20}	286
G_{21}	286
G_{22}	286
G_{23}	286
G_{24}	286

Table 3
The images retrieved for the query images shown in Fig. 9

Query image	Images retrieved in sequence
Q_1	G_1 G_2 G_3 G_4
Q_2	G_1 G_2 G_3 G_4
Q_3	G_1 G_2 G_3 G_4
Q_4	G_1 G_2 G_3 G_4
Q_5	G_5
Q_6	G_6
Q_7	G_7
Q_8	G_8
Q_9	G_9
Q_{10}	G_{10}
Q_{11}	G_{11}
Q_{12}	G_{12}
Q_{13}	G_{13}
Q_{14}	G_{14}
Q_{15}	G_{15} G_{17} G_{18} G_{19} G_{20}
Q_{16}	G_{16} G_{21} G_{22} G_{23} G_{24}
Q_{17}	G_{17} G_{15} G_{18} G_{19} G_{20}
Q_{18}	G_{18} G_{15} G_{17} G_{19} G_{20}
Q_{19}	G_{19} G_{15} G_{17} G_{18} G_{20}
Q_{20}	G_{20} G_{15} G_{17} G_{18} G_{19}
Q_{21}	G_{21} G_{16} G_{22} G_{23} G_{24}
Q_{22}	G_{22} G_{16} G_{21} G_{23} G_{24}
Q_{23}	G_{23} G_{16} G_{21} G_{22} G_{24}
Q_{24}	G_{24} G_{16} G_{21} G_{22} G_{23}

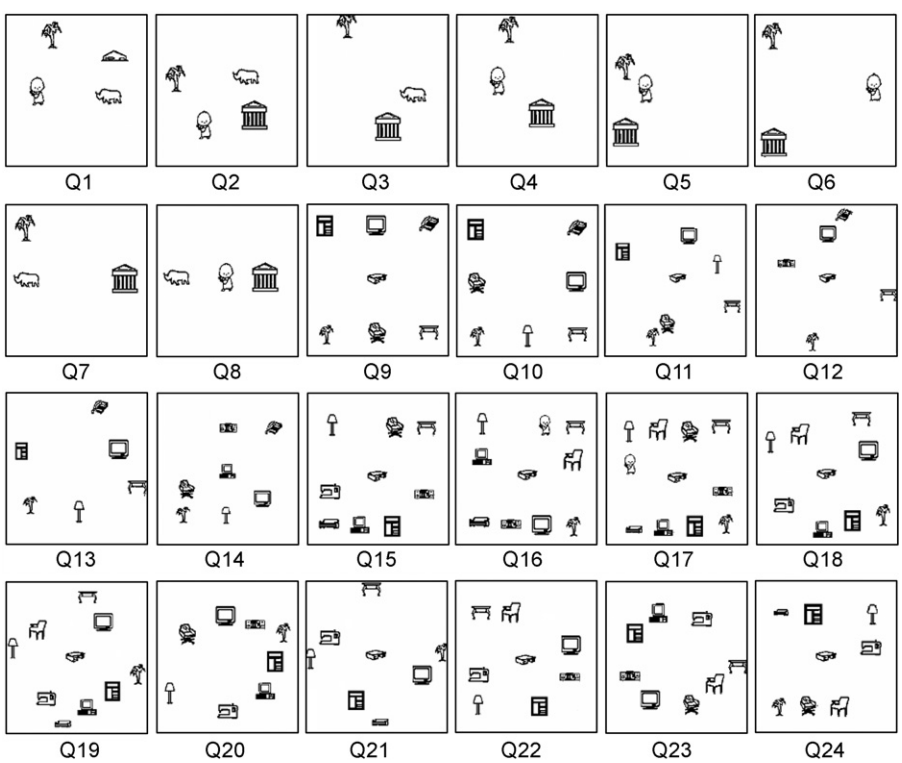

Fig. 7. Some query images.

Table 4
The images retrieved as the desired images for the images given in Ref. [36] when each image is given as query

Query	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}	G_{19}	G_{20}	G_{21}	G_{22}	G_{23}	G_{24}
G_1	√	√	√	√																				
G_2	√	√	√	√																				
G_3	√	√	√	√																				
G_4	√	√	√	√																				
G_5					√																			
G_6						√																		
G_7							√																	
G_8								√																
G_9									√															
G_{10}										√														
G_{11}											√													
G_{12}												√												
G_{13}													√											
G_{14}														√										
G_{15}															√		√	√	√	√				
G_{16}																√					√	√	√	√
G_{17}															√		√	√	√	√				
G_{18}															√		√	√	√	√				
G_{19}															√		√	√	√	√				
G_{20}															√		√	√	√	√				
G_{21}																√					√	√	√	√
G_{22}																√					√	√	√	√
G_{23}																√					√	√	√	√
G_{24}																√					√	√	√	√

shows the sequence of the images retrieved as the desired images for each of the query image shown in Fig. 7. Thus, the images retrieved for each query image Q_i are the members of the group for which the image G_i belongs, and hence, the retrieval is more effective. Table 4 shows the retrieved images when each image of Gudivada and Raghavan [36] is considered as a query image. The entry √ denotes that the image corresponding to the column index is retrieved when the image corresponding to the row index is given as the query image. It shall be noted that the expected (images of the same group) images are retrieved for each of the query image.

7.2. Experimentation 2

Another experiment has also been conducted on the set of 23 images given in Ref. [35]. In their work the images were grouped into five classes to study the behavior of their algorithm in handling rotational, translation, scaling and rotational/scaling/ translation multiple variants of images. A SID has been created by storing all the TSR keys generated for all the images along with their respective lists of image indices. Each image is subjected to the proposed retrieval algorithm as a query image and the retrieved images are noted down. Table 5 presents the images retrieved for each of the 23 images when given as a query image. The entry √ denotes that the image corresponding to the column index is retrieved when the image corresponding to the row index is given as the query image. It is observed that the expected images are retrieved for each of the images when given as query image.

The retrieval results for two specific query images, 12 and 20 are compared with the results provided in Ref. [35]. The other algorithms considered for comparison purpose are SIM_G [1] and SIM_{DTC} [39] which are reviewed briefly in Section 2. These algorithms are selected, as they are the only models available handling multiple rotation variants. In addition, these algorithms are the algorithms used for comparative study in Ref. [35] on the same collection of images. Tables 6 and 7 summarize the results of the experiments conducted, respectively, for the query images 12 and 20, on group 4 images (images 12–20). It has been stated that the method SIM_L outperforms the other approaches [35], in ranking the images according to spatial similarities. However, all the approaches rank the images by matching them one by one in sequence. It is clear from Tables 6 and 7 that the proposed B-tree based indexing scheme also has retrieved all and only the images which have been ranked higher by the SIM_L algorithm directly without matching the images in sequence. This experimentation has brought out, not only the effectiveness of the proposed algorithm but also its efficacy in retrieving the desired images without requiring sequential search in the database.

7.3. Experimentation 3

In order to explore the applicability of the proposed methodology in archival and retrieval of face images, we have conducted an experiment on the ORL face database (http://cam-orl.co.uk). The ORL face database consists of 400 gray level images of size 112×92. They are from 40 different individuals contributing 10 distinct images each. These face images are converted into symbolic form by manually annotating certain dominant points on the face images and labeling them uniquely. The process of manual annotation of

Table 5
The images retrieved as the desired images for the images given in Ref. [35] when each image is given as query

Query	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	√																						
2		√																					
3			√																				
4				√																			
5					√	√	√	√															
6					√	√	√	√															
7					√	√	√	√															
8					√	√	√	√															
9									√														
10										√													
11											√												
12												√						√	√	√			
13													√										
14														√									
15															√								
16																√							
17																	√						
18												√						√	√	√			
19												√						√	√	√			
20												√						√	√	√			
21																					√		
22																						√	
23																							√

Table 6
Results of applying SIM_G [1] SIM_{DTC} [39] and SIM_L [35] and the proposed indexing scheme to group 4 images given in Ref. [35] using the image no. 12 as the query image

Image number	12	13	14	15	16	17	18	19	20
SIM_G	100%	57%	43%	14%	55%	49%	100%	100%	0%
SIM_{DTC}	100%	93%	85%	86%	88%	88%	100%	100%	73%
SIM_L	100%	28%	28%	28%	28%	28%	100%	100%	100%
Proposed B-tree based indexing scheme	√	×	×	×	×	×	√	√	√

Table 7
Results of applying SIM_G [1], SIM_{DTC} [39] and SIM_L [35] and the proposed indexing scheme to group 4 images given in Ref. [35] using the image no. 20 as the query image

Image number	12	13	14	15	16	17	18	19	20
SIM_G	0%	14%	43%	43%	0%	0%	0%	0%	100%
SIM_{DTC}	73%	77%	85%	78%	80%	80%	80%	73%	100%
SIM_L	100%	28%	28%	28%	28%	28%	100%	100%	100%
Proposed B-tree based indexing scheme	√	×	×	×	×	×	√	√	√

a sample image taken from the ORL database and the subsequent conversion to an equivalent symbolic image is shown in Fig. 8.

Each dominant point considered on the face image is treated as an iconic object in the equivalent symbolic image. Iconic objects and their iconic labels are, outer right eyebrow (1), inner right eyebrow (2), inner left eyebrow (3), outer left eyebrow (4), outer corner of right eye (5), right eye pupil (6), inner corner of right eye (7), inner corner of left eye (8), left eye pupil (9), outer corner of right eye (10), right nostril (11), tip of the nose (12), left nostril (13), right mouth corner (14), center of upper lip (15), left mouth corner (16), center of lower lip (17), and chin (18).

Every face image is similarly transformed to its equivalent symbolic form with 18 distinct iconic objects. We have considered alternate five samples per class as database images and remaining as query images. Therefore there are 200 database images and 200 query images. For every query image there are five relevant images in the database. The relevant images in the database for any query vary in pose and appearance, thereby providing a natural way to test the robustness of the proposed algorithm with respect to image transformations and also

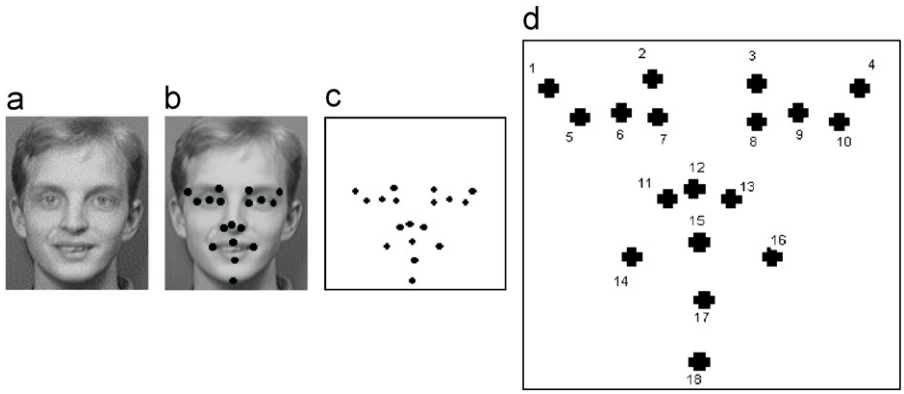

Fig. 8. (a) A sample image. (b) Manual annotation of the selected dominant points on the face. (c) Symbolic image of (a). (d) Symbolic image with labels.

Fig. 9. The five relevant images for the image given in Fig. 8(a).

distortions like non-uniform scaling and shearing to some extent. The five relevant images for the image given in Fig. 8(a) are shown in Fig. 9.

The ORL face database considered in this paper for experimentation is extremely homogenous as all the 18 distinct objects are present in all the images, and the overall spatial scatterings of these images are highly coherent. This now becomes an ideal platform to test the robustness of the proposed scheme with that of others, as handling a coherent homogenous dataset is indeed difficult.

A SID is created by storing the TSR keys of all the chosen 200 images (manual annotation for all the 200 images is done by a single user at the time of database creation). Each image from the query set is subjected to the proposed retrieval methodology. During querying system, the query images are annotated by different users so that there shall not be consistency in locating dominant points. A precision versus recall plot is obtained to estimate the retrieval accuracy. This is also benchmarked with the other precision versus recall plots of SIM_G [1], SIM_{DTC} [39] and SIM_L [35] given in Fig. 10.

Two sample queries are given in Figs. 11 and 12 to demonstrate the ability to retrieve relevant images from the database.

The graph plotted in Fig. 10 reveals that the proposed retrieval scheme outperforms SIM_G [1], SIM_{DTC} [39] and SIM_L [35] methods. SIM_L is capable of discriminating two nearly same images, however, as the degree of variance increases SIM_L fails to perceive the similarity between them. This is reflected in the sudden drop in precision of SIM_L with the increase in recall. This also indicates that SIM_L fails to retrieve all the relevant images accurately. SIM_G preserves the adjacency relationship between symbolic objects in an image and hence is able to perceive similarity between any two images,

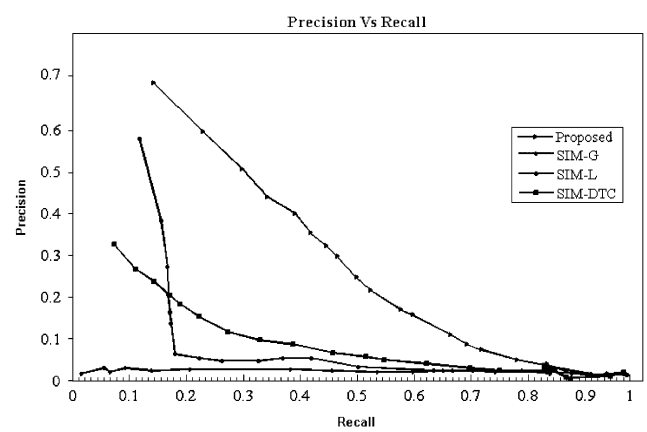

Fig. 10. Benchmarked precision versus recall graphs on ORL.

which are not necessarily identical. Therefore SIM_G has a better overall performance than SIM_L.

7.4. Experimentation 4

In order to corroborate the consistency of the proposed indexing scheme we also consider YALE (http://cvc.yale.edu/projects/yalefaces/yalefaces.html) face database. The YALE face database consists of 165 images of 15 different individuals, each contributing 11 distinct images. Faces were cropped and the images were resized to 112×92 pixels. These images were converted to their corresponding symbolic form as explained in experimentation 3. The variation in pose and moods reflected in the faces is varied. No two different images match closely as in the case of ORL where there was a higher degree of similarity between the faces of the same class.

Fig. 11. First five retrieved images from the indexed database for query image given in (a).

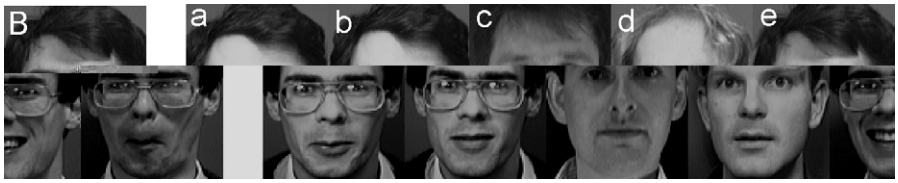

Fig. 12. First five retrieved images from the indexed database for query image given in (b).

Table 8
Average number of comparisons made by various methods for retrieval of similar images versus number of images in the database

No. of images in the database	SIM_G	SIM_{DTC}	SIM_L	Proposed method
50	35164.8	1170450	300370050	9086.32
100	77172.2	2340900	600740100	9125.37
150	107497.33	3511350	901110150	9000.27
200	142291.26	4681800	1201480200	9074.85

Fig. 13. Benchmarked precision v/s recall graphs on YALE.

Alternate images were treated as query and database samples and experiments were conducted. The obtained precision recall plot is shown in Fig. 13. It shall be noticed in Fig. 13 that the proposed scheme outperforms all the other three methods on this database also. The methods SIM_{DTC} and SIM_L have very low performance and SIM_G performs reasonably better than the other two methods. However, the performance of the proposed scheme is consistent and outperforms all the other existing spatial similarity algorithms. A study is also made on the time complexity of the spatial similarity algorithms (SIM_G, SIM_{DTC}, SIM_L) and our proposed methodology in retrieval of face images (ORL) based on the average number of comparisons taken by each methodology for a given set of images. The average number of comparisons taken by the different methodologies for a set of 50, 100, 150 and 200 database images (ORL images) is tabulated in Table 8.

The values in Table 8 reveals that the average number of comparisons required for retrieval of similar images by the existing algorithms (SIM_G, SIM_{DTC}, SIM_L) increase as the number of images in the database increases. However, the average number of comparisons required by the proposed algorithm is almost constant and is not influenced (drastically) by the increase in number of images in the database. This also reveals that the retrieval time remains almost constant even if the database grows larger, as the proposed method is index based in comparison to the other sequential search based algorithms (SIM_G, SIM_{DTC}, SIM_L). The pictorial representation of Table 8 is given in Fig. 14. The Y-axis plot of Fig. 14 is not uniformly scaled as it is difficult to have a discriminating plot for data varying from 9000.27 to 1201480200.

Fig. 14. Average number of comparisons versus number of images in the database.