Multimed Tools Appl (2006) 29: 191–208
DOI 10.1007/s11042-006-0010-x

MBC-based shape retrieval: basics, optimizations, and open problems

Maytham Safar · Cyrus Shahabi

Published online: 1 June 2006

Abstract Shape of an object is an important feature for image and multimedia similarity retrievals. In our previous studies we introduced a new *boundary-based* technique (*MBC-based*) for shape retrieval and compared its performance to other techniques. In this study, we describe in detail the basics of our *MBC-based* shape representation techniques, and we show how they support different query types. In addition, we describe two original optimization techniques that can further improve the performance of our *MBC-based* methods in several aspects, and show that they are also applicable to other applications (e.g., pattern recognition techniques). Finally, we define open problems in the area (e.g., partial similarity) and provide some hints on how to approach those problems.

Keywords Multimedia systems · Multimedia digital libraries · Image content · Image indexing · Shape representation · Shape similarity

1 Introduction

Several applications in the areas of CAD/CAM, computer graphics, and multimedia require to store and access large databases. A major data type stored and managed by these applications is the representation of two dimensional (2D) objects. Objects contain many features (e.g., color, texture, shape, etc.) that have meaningful semantics. Among those, shape is an important feature that conforms with the way human beings interpret and interact with real world objects. The shape representation of objects can be used for their indexing, retrieval, and as a similarity measure. Shape queries can hence be utilized to perform a quick search on the database to find a set of

M. Safar (✉)
Computer Engineering Department, Kuwait University, Safat, Kuwait
e-mail: maytham@eng.kuniv.edu.kw

C. Shahabi
Integrated Media Systems Center, Department of Computer Science,
University of Southern California, Los Angeles, CA 90089-0781, USA
e-mail: shahabi@usc.edu

similar objects, or objects satisfying some spatial relationship, efficiently and independent of the database size.

In this study, we first present, in detail, the basics of a new *boundary-based* (termed *MBC-based*) approach to shape representation and similarity measure that was proposed by the authors in [22]. We start by describing how our techniques for similarity shape retrieval depend on extracting features (e.g., angle sequences and touch points) from an object's minimum bounding circle (*MBC*). Then, we show how our proposed *MBC-based* method could be used for efficient retrieval of 2D objects by utilizing three different index structures based on features that are extracted from the object's *MBC*. Next, besides similarity retrieval, we describe the support of other query types (e.g., spatial queries such as topological and direction queries) using our *MBC* features [16]. Finally, we describe the effectiveness of one of our *boundary-based* techniques (*MBC-TPVAS*) as compared to three different orthogonal shape representation methods (i.e., Fourier descriptors method—*FD* [21], grid-based method—*GB* [19], and Delaunay triangulation method—*DT* [24, 25]. We compare their performance in terms of accuracy, storage and computation requirements, and robustness under various scenarios of "uncertainty" (e.g., presence of noise in the database).

Second, we describe two original optimization techniques for our MBC-based method that are also applicable to other applications (e.g., pattern recognition techniques). First, we propose a simple and more efficient algorithm to identify the minimum bounding circle (*MBC*) of an object. The technique relies on the fact that only a small subset (*S*) of the points of an object uniquely determines the *MBC* of the object. Hence, to speed up the *MBC* algorithm, we first find *S*. Then, we compute *MBC* of the set *S*. Second, we propose a new technique to improve the response time of *MBC-based* methods for match queries under a specified rotation angle (*SRA*). The technique relies on utilizing the symmetry property of the phase values of the Fourier transform (*FT*) of a real sequence (*AS*) under different rotations. Finally, we show that the fundamental observations utilized by our techniques can be adapted and extended to be applicable to other frameworks and application domains. For example, our optimized *MBC* computation algorithm can be used in *GIS* systems that sometimes maintain *MBCs* as approximations for spatial objects.

Finally, we define some open problems in the area and provide some hints on how to approach those problems. For example, we investigate match queries under other transformations such as mirroring, non-uniform scaling, and shearing.

The remainder of this paper is organized as follows. Section 2 describes *MBC-based* shape description techniques. In Section 3, we propose a technique to enhance the performance of *MBC* computation algorithm. In addition, we describe the impact of using different starting points (i.e., rotating/shifting an angle sequence) on the phase values of discrete Fourier transform (*DFT*) of a real sequence. Section 4 illustrates some experimental results. Finally, Section 5 concludes the paper, and Section 6 provides some insight on future work and directions and lists some open problems in the area.

2 MBC-based shape retrieval

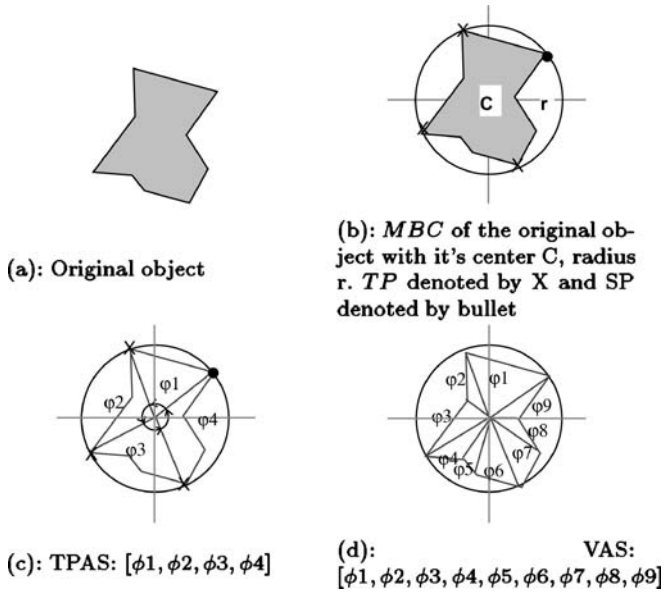
In this section, we present a detailed description of the *MBC* features, *MBC-based* shape representation, *MBC-based* index structures, the support for different query types, and present its effectiveness compared to other methods.

2.1 MBC features and shape representation

MBC-based shape representation methods (*MBC-TPAS*, *MBC-VAS*, and *MBC-TPVAS*) are shape descriptor methods that are based on features extracted from the object's minimum bounding circle (*MBC*). Those features are the center coordinates of *MBC*, the radius of *MBC*, the set of touch points on *MBC* (*TP*s), the touch points angle sequence (*TPAS*), the vertex angle sequence (*VAS*), and the start point of the angle sequence (*SP*). Figure 1 depicts an example of an object with its six identified features. In figure 1b the *MBC* of an object is computed and its touch points are identified as the vertices that lay on its *MBC* (denoted by X). Figure 1c and d show the touch points and vertex angle sequences, respectively. With the *MBC-based* methods, the shape descriptor of an object relies on a subset of the six *MBC* features. Each method starts by obtaining a feature function of the object called shape signature $f(k)$. To obtain $f(k)$, four steps are required. The minimum bounding circles of the objects are first computed. Second, *MBC* features such as angle sequences (*TPAS*, and *VAS*) and the number of touch points (*TP*) are identified. Third, the object's shape signature is identified as the sequence combination of *TP*, *TPAS* and/or *VAS*. Finally, to fix and reduce the dimensionality of the shape signatures, *DFT* of the signatures is used as proposed in [1, 4]. *DFT* of the shape signatures preserves all the details available about the original signatures. In addition, only some significant components from the transformed vectors are used for shape representation, index key, and shape similarity computation.

2.2 Index structures

In our study [22], we presented a *multi-step* query processing method that efficiently supports the retrieval of 2D objects. We showed how our proposed *MBC-based*



(a): Original object

(b): MBC of the original object with it's center C, radius r. TP denoted by X and SP denoted by bullet

(c): TPAS: $[\phi 1, \phi 2, \phi 3, \phi 4]$

(d): VAS: $[\phi 1, \phi 2, \phi 3, \phi 4, \phi 5, \phi 6, \phi 7, \phi 8, \phi 9]$

Fig. 1 MBC features

methods could be used for efficient retrieval of 2D objects by utilizing three different index structures based on features that are extracted from the objects *MBC*. We proposed three types of index structures on different subsets of *MBC* features. The first structure, I_{TPAS}, indexes the objects based on their number of vertices (m), number of touch points (n_{TP}), and their *TPAS*s. The second structure, I_{VAS}, uses *VAS* of the objects for indexing purpose. Finally, the third index structure, I_{TPVAS}, is a hybrid of the first two structures, where it uses the information provided by both the vertices of the objects (*VAS* and m) and their *TP*s. The index only helps to identify a set of candidates that satisfy the query. We might have false hits but no false drops. This set is then filtered in two steps in order to eliminate all the false hits. In the first step, we use a subset of *MBC* features (i.e., *SP*, r, and C), to identify rotated, scaled and/or translated candidates. Finally, in the last step, the final candidates can be compared to the query object (O_q) by employing any known algorithm proposed in the area of computational geometry. However, we continue utilizing *MBC* features for the final comparisons. Towards this end, we employ our *feature-based* comparison algorithms, termed *PCEM*, *PCRST*, and *PCSIM*, which are used for *EM*, *RST*, and *SIM* queries, respectively. *EM*, *RST*, and *SIM* are three different match search queries on a database. *EM* is searching for exactly identical objects. *RST* is searching for identical objects that are within a specified or unspecified rotation, scaling, and translation. Finally, *SIM* is searching for similar objects per our definition of similarity.

2.3 Spatial queries

Besides similarity retrieval, we describe the support of spatial queries (i.e., topological and direction queries) using our *MBC* features. Our primary motivation for using *MBC* is that we showed the interesting properties of *MBC* to improve the efficiency of similarity queries for the retrieval of 2D objects by shape. Therefore, in order to utilize the same approximation for other spatial queries, we investigated the usefulness and feasibility of *MBC*s for such queries. We identified a subset of *MBC* features that could be utilized to support such queries. As a result, there is no need to maintain two approximations (i.e., minimum bounding rectangle (*MBR*) and *MBC*) per object for efficient support of different queries (i.e., similarity and spatial). In addition, we argued that depending on the application, using *MBC* approximations can be more beneficial than using *MBR*s. First, circles are insensitive to orientation and hence an object's *MBC* is unique and invariant to translation and rotation. This property suits the requirements of the topological relations in which *MBC*s stay invariant under topological transformations such as translation, rotation and scaling [14]. Second, in [12] an indexing technique (*Sphere\tree*) was proposed for circles that could be used to efficiently support *MBC*-based spatial queries (see [16] for further details). Third, circles occupy less storage space than rectangles. Finally, we showed that *MBC* is a successful approximation for topological overlap queries (important for intersection joins). We showed that the performance of overlap queries can be improved by distinguishing among nine different types of overlap between the *MBC*s. Then, we determined the cases in which the relation between the actual objects can be determined efficiently from the relation between their corresponding *MBC*s.

2.4 Related work and performance evaluation

In this section, we focus on the related work in the database area and discuss their performance compared to our technique. We compared four boundary-based methods for shape representation and retrieval: Fourier descriptors method (*FD*) [6, 15, 21], grid-based method (*GB*) [8, 19], Delaunay triangulation method (*DT*) [24, 25], and one of the *MBC*-based methods (*MBCTPVAS*). *DT* is a histogram-based approach used to classify shape features of objects. For each object, high-curvature points, called corner points, are used as the feature points of the object. Then, a Delaunay triangulation of these feature points is constructed. Subsequently, a feature point histogram is obtained by the discretization of the angles produced by this triangulation into a set of bins and counting the number of times each discrete angle occurs in the object. *FD* obtains the object representation in the frequency domain as complex coefficients of the Fourier series expansion of the object's shape signature. The shape signature of the object is based on shape radii (or centroid distances). It computes the distance between points uniformly sampled along the object boundary and its centroid. Finally, the *DFT* of the shape signature is obtained. The last approach for shape representation is *GB*. With this method, objects are first normalized for rotation and scale. Then, the object is mapped on a grid of fixed cell size. Subsequently, the grid is scanned and a 1 or 0 is assigned to the cells depending on whether the number of pixels in the cell that are inside the object are greater than or less than a predetermined threshold. Finally, a unique binary number is obtained as shape representation by reading the 1's and 0's assigned to the cells from left to right and top to bottom.

In [17, 18], we tested the efficacy of our *boundary−based* technique (*MBCTPVAS*) in terms of: 1) retrieval accuracy, 2) storage cost for their indices, 3) computation cost of generating the shape signature, 4) computation cost of computing the similarity between two different shape signatures, 5) sensitivity to the presence of noise in the database, and 6) sensitivity to the alternative ways of identifying the boundary points of a shape. Our results demonstrated the superiority of *MBCTPVAS* over other shape representation techniques using our evaluation metrics. Our studies proved that the similarity retrieval accuracy of our method (*MBCTPVAS*) is as good as other methods, while it had the lowest computation cost of generating the shape signatures of the objects. Moreover, it had low storage requirement, and a comparable computation cost of computing the similarity between two shape signatures. Furthermore, *MBCTPVAS* showed to have better accuracy than others in the presence of noise, and is less sensitive than others to the number of vertices used to represent shape. Finally, the results showed that *MBCTPVAS* had direct support for other query types (such as *SRST* and spatial queries), while others lack to support them (refer to [18] for further details).

3 Optimization techniques

In our previous studies we introduced a new *boundary−based* technique for shape retrieval and compared its performance to other techniques. Our results showed the superiority of our *MBC−based* techniques in terms of robustness, retrieval accuracy, storage requirement, and computation cost. Our current experiments and analysis

show that we can further improve the performance of our method in several aspects. First, we propose a simple and more efficient algorithm to identify the minimum bounding circle (*MBC*) of an object. It operates only on a subset of the object's vertices instead of all the vertices. Second, we propose a new technique to improve the response time of *MBC−based* methods for *SRA* match queries. The technique relies on utilizing the symmetry property of the phase values of the *DFT* of a real sequence under different rotations.

3.1 MBC computation

We start by describing the algorithm used to find the minimum bounding circle (*MBC*) of an object. *MBC* of any finite set of points (P) in a 2D plane is its smallest enclosing circle. The set of points of P can be divided into two sets P_{in} and P_{on} (touch points); where P_{in} is the set of all points that are contained by the *MBC* but are not laid on its boundary and P_{on} is the set of all points that are contained by the *MBC* and are laid on its boundary.[1] To compute the *MBC* of an object we use a simple randomized algorithm that computes the smallest enclosing disk (circle) of a set of n points in the plane in expected $O(n)$ (linear time). The algorithm is based on Seidel's Linear Programming, that solves a Linear Program with n constraints and d variables in expected $O(n)$ time, provided d is constant (for details on *MBC* algorithm see [26]). In general, for a set of n points, the algorithm computes minimum disk containing P in an incremental fashion. It starts with the empty set and adding the points in P one after another while maintaining the smallest enclosing disk of the points considered so far. The expected complexity of the algorithm is independent from the input distribution; it averages over random choices made by the algorithm. In theory, there is no input that should force the algorithm to perform poorly. However, in practical situations, the algorithm sometimes does perform poorly. This is because when the algorithm computes the smallest enclosing disk for x points ($x \leq n$) from the smallest enclosing disk for $x-1$ points, it randomly selects a point in $P-(x-1)$ and checks if this point leads to a larger enclosing disk for the x points. We observed that most the time, the algorithm picks up points that are already covered by the smallest enclosing disk of a previous set of points. In [9], another deterministic linear time algorithm for computing smallest enclosing balls was proposed. However, their algorithm is not nearly as easy to describe and to implement as Seidel's algorithm, and it highly depends on the constant d.

3.1.1 Optimized MBC computation

In this section, we propose a technique to enhance the performance of the *MBC* computation algorithm. It reduces the number of points from which the randomized selection part of the original algorithm can select. To this end, we identify a subset of P that we know would contribute to the computation of the *MBC* of the points in P, ignoring all the other points. In [26], a lemma states that if P and R are finite point sets in the plane, and P is nonempty, then if there exists a disk containing P with R on its boundary, the minimum bounding disk of P and R are unique.

[1] To eliminate numerical errors, we use a circle with a wide boundary rather than its line contour.

Therefore, we can conclude that the set of points P_{in} does not affect the *MBC* algorithm and that P_{on} is the set that uniquely determines $MBC(P)$. In fact, there are at most three points (p) in P_{on} such that $MBC(P)$ is not equal to $MBC(P-p)$; i.e., $MBC(P)$ is determined uniquely by a maximum of three points in P_{on}.

Using the previous observations, we propose a new technique to speed up the *MBC* computation algorithm by identifying all those points that may belong to P_{in} and compute $MBC(P-P_{in})$. Note that previously, given P, we cannot exactly determine P_{on} and P_{in} without actually computing $MBC(P)$. Therefore, we propose an approximation algorithm to find P_{in}. Towards this end, given P, we first compute $MBR(P)$ (in $O(n)$), and identify the largest diagonal of $MBR(P)$ (*maxR*). Second, we compute a reduced (scaled down) *MBR* (*RedMBR*(P)) with the largest diagonal (*RedR*) equal to *maxR* reduced by a factor *RF* (i.e., $RedR = RF \times maxR$). Note that the reduction factor value should be ≤ 1 (i.e., $RF \leq 1$). Third, we consider the points inside *RedMBR*(P) to be P_{in}. Finally, we compute the $MBC(P-P_{in})$. Figure 2 embodies a graphical representation of the algorithm. Our approximation algorithm to find P_{in} is highly dependent on the reduction factor *RF*. That is, for some values of *RF*, P_{in} may contain some points that belong to P_{on}. Hence, we might find an approximated $MBC(P)$. For example, in figure 3a we show an example of a set of points (P) to which we applied two different values of *RF*. *RedMBR*1 contains two points (the black vertices) that should belong to P_{on}, therefore, the calculated *MBC* is an approximation of the original *MBC*. However, *RedMBR*2 does not contain the two black vertices, hence, the algorithm finds the exact *MBC*. Therefore, we need to find the optimal values of *RF* that would produce the exact *MBC*.

Given P and $MBC(P)$, the largest bounding rectangle of the points in P could be the circumscribed rectangle of $MBC(P)$ (*CirMBR*), and their smallest bounding rectangle that could fit inside $MBC(P)$ is the inscribed rectangle of $MBC(P)$ (*InsMBR*) (see figure 3b for further details). *InsMBR* can be considered as the largest (and hence optimal) *MBR* that approximates P_{in}, and the points inside *InsMBR* (s) are a subset of P_{in} (in figure 3b, the stripped area represents s, and the gray area represents $P_{in}-s$). The smallest reduction factor (*RF*) for which $MBC(P)$ is always exact should satisfy the following relation: $1-\frac{r}{D} \leq RF \leq 1 \equiv 1-\frac{r}{\sqrt{2}\times r} \leq RF \leq 1 \rightarrow 1-\frac{1}{\sqrt{2}} \leq RF \leq 1$. In general $maxR \geq r$ (as shown in figure 4), however, $MBR(P) \leq CirMBR$. Hence, by constraining *RF* to be $\geq 1-\frac{1}{\sqrt{2}}$ and ≤ 1, we guarantee that *RedMBR* is always a subset of *InsMBR*. For example, in figure 4 the

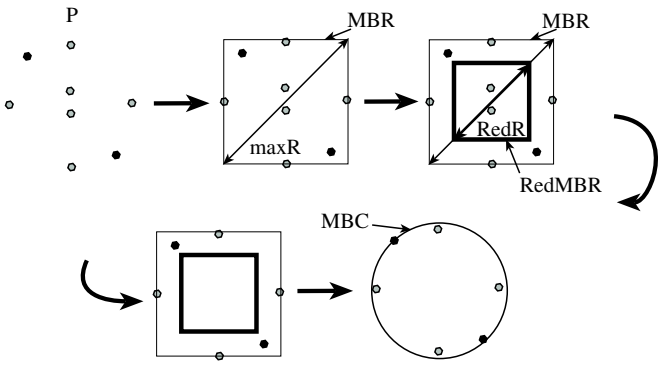


Fig. 2 Optimized MBC

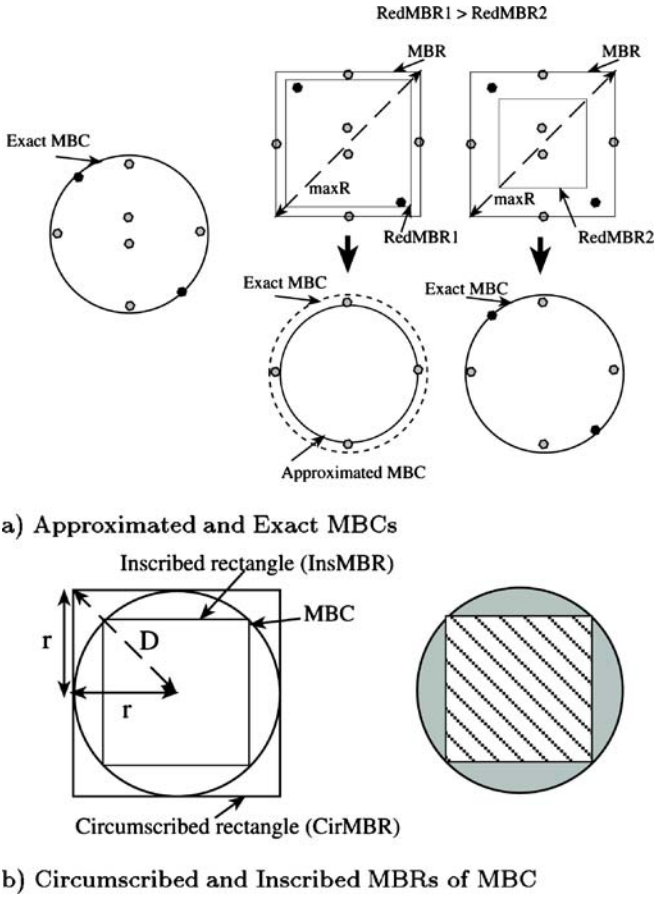


Fig. 3 MBC and its MBRs

points inside the gray area belong to *InsMBR*. However, to guarantee that we compute the exact $MBC(P)$, those points were not included inside *RedMBR* (approximation of P_{in}).

3.2 DFT phase values trends

For similarity search, a shape representation technique should be capable of supporting different types of similarity search queries such as: 1) exact match, 2) exact match regardless (invariant) of size and orientation 3) exact match under a specified rotation angle (*SRA*), scaling factor, translation vector, or any combination of the three. The support for query types under a specified size and/or orientation is essential in some application domains. For example, searching for similar tumor shapes in a medical image database [7]. A tumor is represented by a set of 2D images, each corresponding to a slice cut through its 3D representation. A method for retrieving similar tumor shapes would help to discover correlations between tumor shape and certain diseases. In addition to the shape, the size and location of the tumor would help in the identification of patients with similar health histories.

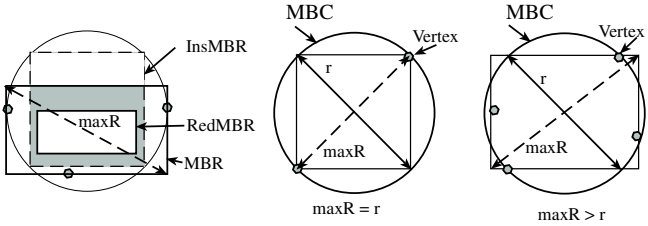


Fig. 4 maxR

With *MBC−based* approaches, a subset of the *MBC* features (i.e., radius r, center C, and start point SP) is utilized with a special index structure (e.g., I_{TPAS}) to support match queries under a specified size and/or orientation. Our previous analytical studies showed that the algorithm to find similar objects in a database under a specified rotation angle (*SRA*) incurred the highest cost ($O(n_{SP}N^2)$); where N is the number of vertices of an object and n_{SP} is the number of start points (see [23] for further details). This is due to the fact that rotating objects would lead to different *SP*s, hence resulting in various angle sequences (*AS*s). As a consequence, given a query object, we have to submit n_{SP} queries to the database to find the candidate objects for match (each query is answered using our I_{TPAS} index structure in $O(N^2)$). In the worst case, SP could be any point in the original object, hence $n_{SP} = N$ which leads to a total complexity of $O(N^3)$. To reduce the query processing complexity, we need to reduce the number of queries submitted (i.e., reduce the number of *SPs* that we have to investigate). Therefore, in Section 3.2.1 we study the impact of using different starting points (*SPs*) on the phase values of the *DFT* of a real sequence. In addition, we show that instead of submitting n_{SP} queries for *SRA* search, we submit only *one* query using the circular shift property of the *DFT* of a real sequence. Another solution would be to store n_{SP} different representations for each object in our database. Then submit one *SRA* search query. However, this approach would require much more additional storage space (n_{SP} times the space required for our approach).

We first start with some definitions and notations that we will use in following section. Consider an Ndimensional real sequence $X = (x_0, \ldots, x_{N-1})$. The *DFT* of X is the N- dimensional complex sequence $\hat{X} = (\hat{x}_0, \ldots, \hat{x}_{N-1})$ given by $\hat{X} = F(X)$ where

$$\hat{x}_f = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} x_t \, exp\left(\frac{-j2\pi f^* t}{N}\right); \; for \; f = 0, 1, \ldots, N-1$$

where j is the imaginary unit ($j = \sqrt{-1}$) of a complex number. In general, a complex number $a = \alpha + j\beta$ can be defined as the set $\| a \|$ (i.e., the magnitude of a) and $\angle a$ (i.e., the phase of a). In our discussion hereon, we assume that N is even for simplicity, although our observations extend to the case of arbitrary N. It is well known that Fast Fourier Transform (*FFT*) can be used to calculate the *DFT* coefficients in $O(N \log N)$ arithmetic operations. Some fundamental properties of the *DFT* are: 1) if we start with an N-dimensional nonnegative real sequence X, then $\hat{X}$ is a complex sequence with the exception of $\hat{x}_0$, which is a nonnegative real number ($\hat{x}_0 = \frac{\sum_{t=0}^{N-1} x_t}{\sqrt{N}}$), 2) the phase values for $\hat{x}_0$ and $\hat{x}_{\lceil \frac{N}{2} \rceil}$ are always equal to zero regardless of the real sequence, 3) $\hat{x}_i$ and $\hat{x}_{N-i}$ are conjugate complex numbers for $i = 1, 2, \ldots, \lceil \frac{N}{2} \rceil$, hence, $\| \hat{x}_i \| = \| \hat{x}_{N-i} \|$ (i.e., their magnitudes are equal), and $\angle \hat{x}_i = -\angle \hat{x}_{N-i}$ (i.e., the absolute value of their

phases are equal). Figure 5 shows the symmetry of the magnitude (represented by $M_i s$) and phase values (represented by $P_i s$) of the *DFT* of a real sequence, and 4) when applying a linear shift to a periodic sequence (or a circular shift of a sequence), we observe the circular shift property of the *DFT* [13] as follows: $\vec{X[m]} = \exp^{-j(\frac{2\pi m}{N})n} X[m]$ (for $0 \leq m \leq N-1$); where $\| a \| = \sqrt{\alpha^2 + \beta^2}$ and

$$\angle a = \begin{cases} arctan(\frac{\beta}{\alpha}) : if \alpha \geq 0 \\ arctan(\frac{\beta}{\alpha}) + \pi : if \alpha < 0 \ and \ \beta \geq 0 \\ arctan(\frac{\beta}{\alpha}) - \pi : if \alpha < 0 \ and \ \beta < 0 \end{cases}$$

3.2.1 Circular shift property of Fourier transform

In this section, we investigate how to improve the response time of *MBC−based* methods for *SRA* match queries by utilizing the circular shift property of the *DFT* of a real sequence. In addition, from our following analytical models, we will show that only *one* single phase value (second phase value of a sequence) is used to enhance the query response time of *SRA* match queries instead of all the phase values for different rotations/shifting. Hence, instead of submitting n_{SP} queries, we end up submitting only *one* query which reduces the total complexity for *SRA* queries to $O(N^2)$. To this end, we study the impact of using different starting points (i.e., rotating an object or shifting the angle sequence) on the second phase value of the *DFT* of a real sequence. If we consider an *N*-dimensional real sequence $X = (x_0, \ldots, x_{N-1})$, then the rotated (shifted) sequence of X is the *N*-dimensional sequence $\vec{X} = (x_n, \ldots, x_{N-2}, x_{N-1}, x_0, x_1, \ldots, x_{n-1})$; where n is the number of rotations. From the circular shift property of *DFT* we can conclude that the magnitude values of a *DFT* does not change with different start points, and that the symmetry property of the phase values of the transformed sequence is still observed. However, the phase values change depending on the number of rotations. Our study shows that by applying a transformation to the Fourier components with negative phase values we can observe a new trend in the phase values of the *DFT* of a sequence. Each Fourier component is a vector in complex plane. Therefore, if we replace the

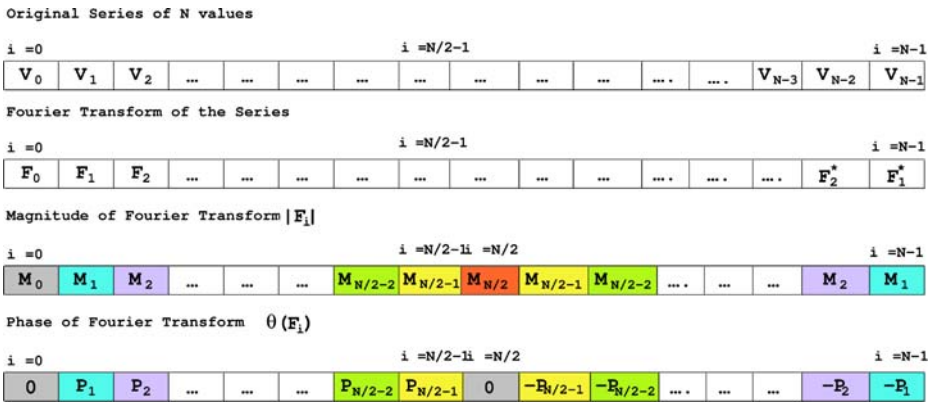


Fig. 5 Symmetry in Fourier transform

#Rotations	Position =1						i =N/2−1	i =N/2		
j =0	0	$P_{1,2}$	$P_{1,3}$	...	...	...	$P_{1,N/2-1}$	$P_{1,N/2}$	0	...
	0	$P_{2,2}$	$P_{2,3}$	...	...	...	$P_{2,N/2-1}$	$P_{2,N/2}$	0	...
	...	...	...				...	...	...	
j =N/2−1	0	$P_{N/2,2}$	$P_{N/2,3}$	...	...	...	$P_{N/2,N/2-1}$	$P_{N/2,N/2}$	0	...
j =N/2	0	$P_{1,2}$	$P_{1,3}$	...	...	...	$P_{1,N/2-1}$	$P_{1,N/2}$	0	...
	0	$P_{2,2}$	$P_{2,3}$	...	...	...	$P_{2,N/2-1}$	$P_{2,N/2}$	0	...
	...	...	...				...	...	...	
j =N−1	0	$P_{N/2,2}$	$P_{N/2,3}$	...	...	...	$P_{N/2,N/2-1}$	$P_{N/2,N/2}$	0	...

Fig. 6 Phase trend with rotation/shifting

vectors with negative phase values with vectors with the same magnitude but in the opposite direction (same effect as adding a π to the phase values) we observe the new trend. Figure 6 shows the phase values ($P_{i,j}s$: where i represents the ith phase value, and j represents the number rotations) of FT of a real sequence for different number of rotations/shifting ($0 \leq$ *number of rotations* $\leq N-1$) after adding π to the negative values. From figure 6 we observe that after $\frac{N}{2}$ rotations of the original sequence, the phase values repeat, i.e., $\angle\hat{X}_i = \angle\hat{X}_{\frac{N}{2}+i}$ $i = 0, 1, \ldots, \frac{N}{2}-1$; where i is the number of times the original sequence (X) is rotated. To be able to answer *SRA* query we need to submit n_{SP} queries, each corresponding to a phase value of the DFT of a real sequence of size N for different rotation values of $0, \ldots, n_{SP}$. However, using our observations about the symmetry of the phase values under rotation, we conclude that only $\frac{n_{SP}}{2}$ queries are required to answer an *SRA* query.

In addition, from our following analytical models, we will show that we can further enhance the query response time of *SRA* match queries. This is accomplished by reducing the number of queries to answer an *SRA* from $\frac{n_{SP}}{2}$ to only *one* query by only storing a single phase value (second phase value of a sequence) of all the phase values for different rotations/shifting.

We define a function $\delta_{k,i,j}$ as follows:

$$\delta_{k,i,j} = \begin{cases} \angle\hat{x}_k^j - \angle\hat{x}_k^i : if \angle\hat{x}_k^j - \angle\hat{x}_k^i \geq 0 \\ \angle\hat{x}_k^j - \angle\hat{x}_k^i + \pi : if\ \angle\hat{x}_k^j - \angle\hat{x}_k^i < 0 \end{cases}$$

for $k, i, j = 1, \ldots, N$; where $\angle\hat{x}_k^i$ and $\angle\hat{x}_k^j$ are the phase values of the kth component of $\hat{X}$ after $i-1$ and $j-1$ rotations of the real sequence X, respectively. In figure 7 we show the value of the function $\delta_{k,i,j}$ for $k = 2, j = 1, \ldots, N$, and $i = 1, \ldots, (\frac{N}{2}+1)$. For $i = (\frac{N}{2}+1), \ldots, N$ the values of $\delta_{k,i,j}$ are the same as for $i = 1, \ldots, \frac{N}{2}$ but were not included in the table to save space. If we define δ as follows: given two real sequences $X1$ and $X2$ with the second phase values of their $DFT \angle\hat{X1}_1$ and $\angle\hat{X2}_1$, then:

$$\delta = \begin{cases} \dfrac{\angle\hat{X}1_1 - \angle\hat{X}2_1}{\frac{2\times\pi}{N}} : if\ \angle\hat{X}1_1 - \angle\hat{X}2_1 \geq 0 \\ \dfrac{\angle\hat{X}1_1 - \angle\hat{X}2_1}{\frac{2\times\pi}{N}} + \dfrac{N}{2} : if\ \angle\hat{X}1_1 - \angle\hat{X}2_1 < 0 \end{cases}$$

then, from figure 7, we can conclude the following: if we have two similar real sequences $X1$ and $X2$ (i.e., the Euclidean distance between the magnitude values of their DFT ($\hat{X}1$ and $\hat{X}2$) is close to zero, then using the phase values $\angle\hat{X}1_1$ and $\angle\hat{X}2_1$ we can tell if $X1$ and $X2$ are exactly the same or a rotated version of each other under a specific rotation value according to the following condition:

- if $\delta = 0 \longrightarrow X1$ and $X2$ are exactly the same, or rotated clockwise $\frac{N}{2}$ times
- if $\delta = n \longrightarrow$ same real sequences but shifted to the left n times, or $n + \frac{N}{2}$

where $1 \leq n \leq \frac{N}{2} - 1$. To generalize our observations, if $F1$ is the second Fourier coefficient of $\hat{X}$, then second Fourier coefficient of $\hat{X}$ after k shifts can be computed as $F1 \times e^{jk\theta}$ (see figure 8).

4 Performance

In several applications, an important criterion for testing the efficacy of the shape retrieval methods is that for each query object the relevant items (similar shapes) in the database be retrieved efficiently. Therefore, in our previous studies [17, 18], experiments were designed to measure the effectiveness of the similarity retrieval of the four different shape representation methods (i.e., FD, GB, $M-DT$, and MBC-$TPVAS$) in terms accuracy, storage and computation requirements, and robustness under various scenarios of "uncertainty" such as: 1) the presence of noise in the database, 2) parameters used to identify a shape and its vertices, and 3) differences between shape representation techniques and human perception in defining similarity. However, in this study, we test some optimization techniques that can be used to enhance the performance of MBC-based methods.

4.1 Optimization techniques

Originally, to find the MBC of an object we used a simple randomized algorithm that computed the smallest enclosing disk (circle) of a set of n points in the plane in expected $O(n)$ (linear time) [26]. While the expected complexity of the algorithm was independent from the input distribution, in practical situations the algorithm sometimes performed poorly. To address this problem, we proposed a technique to enhance the performance of the MBC computation algorithm that relies on the fact that only a small subset S of the points of an object uniquely determines the MBC of the object. Hence, to speed up the MBC algorithm, we first found the MBR of the

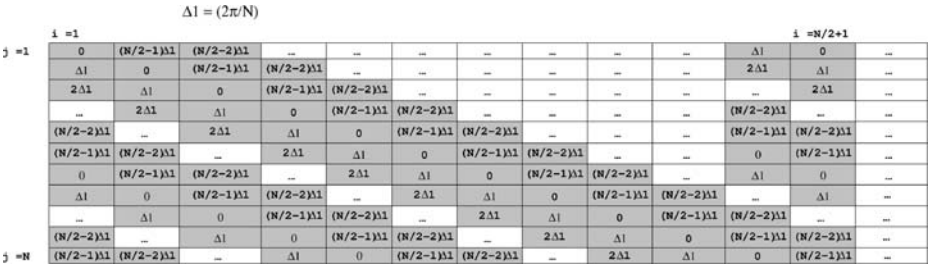


Fig. 7 Phase difference trend

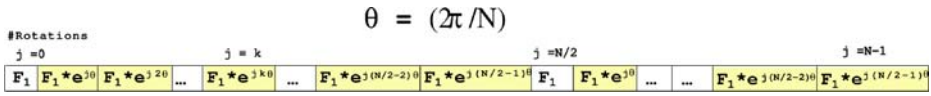


Fig. 8 Trend in second phase value

points. Then, we applied a reduction factor to produce a smaller *MBR*. The points that did not belong to the smaller *MBR* were considered as the points that belonged to *S*. Finally, we computed $MBC(S)$.

To evaluate the performance of our optimized *MBC* computation algorithm we conducted two experiments using real and synthetic data. In the first experiment, we used a database of 100 real objects (contours of fish images) with an average number of points in an object equal to 100. We varied the reduction factor *RF* from 2% to 98% with an increment of 2. Figure 9a illustrates the results where the *X*-axis is percentage of reduction factor (*RF*), and the *Y*-axis is the percentage of reduction in the number of vertices and percentage of reduction in the execution time of the optimized *MBC* computation algorithm as compared to the original *MBC* algorithm. The percentage of reduction in the number of vertices measures the quality of our proposed algorithm in identifying the points of an object that will not contribute in computing $MBC(P)$. That is, it computes $\frac{P-RedMBR}{P}$. The percentage of reduction in the execution time of *MBC* computation algorithm measures the time saved by not considering the points in *RedMBR* in the computation of its *MBC*. That is, $\frac{Time(MBC(P))-Time(OptimizedMBC(P))}{Time(MBC(P))}$. From the results in figure 9a we can conclude that our proposed algorithm reduces the execution time of $MBC(P)$ significantly (up to 90% reduction for $RF = 4\%$). As the value of *RF* increases, the percentage of reduction in the number of vertices increases, and hence the execution time decreases. As a consequence, the maximum reduction in the execution time of $MBC(P)$ for values of $RF \geq$ *optimal value of RF* was 60%. However, we were able to find the exact *MBC* even for cases where $RF <$ *optimal value of RF* (i.e., $RF < \frac{1}{\sqrt{2}}$). The smallest value of *RF* without producing approximate *MBC*s was 4%.

In the second experiment, we used a database of 100 synthetic objects with an average number of points in an object equal to 135. The synthetic data were gen-

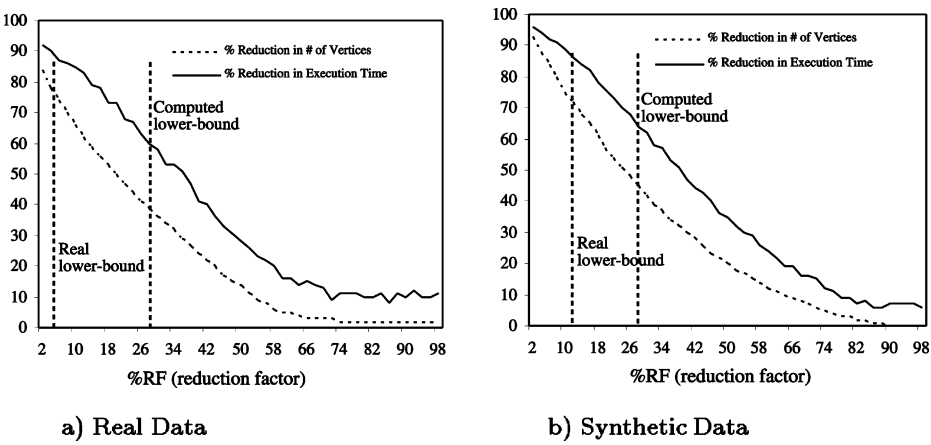


Fig. 9 Performance improvement

erated with more points inside P_{in} than the real data. We varied the reduction factor RF from 2% to 98% with an increment of 2. Figure 9b illustrates the results where the X-axis is percentage of reduction factor (RF), and the Y-axis is the percentage of reduction in the number of vertices and percentage of reduction in the execution time of MBC computation. From the results in figure 9b, we can conclude that the optimized MBC computation algorithm performed marginally better than the case where we were using real data (about 5% increase in performance). This is due to the fact that the real data had fewer points inside $InsMBR$ and hence smaller values of reduction in the number of vertices. The smallest value of RF without producing approximate MBCs was 12%.

Our experimental results were consistent with our analytical observations. In all our experiments the proposed method reduced the execution time of the MBC computation algorithm. In addition, even for values of $RF < \frac{1}{\sqrt{2}}$, we still sometimes compute the exact MBC.

5 Conclusion

In this study, we first presented a detailed description of our *boundary-based* (termed *MBC-based*) approaches to shape representation and similarity measure. We described how features extracted from an objects minimum bounding circle (MBC) (e.g., angle sequences and touch points) can be used for: 1) representing 2D objects, 2) similarity measure, 3) efficient retrieval of 2D objects, 4) building index structures, and 5) the support of other query types (e.g., spatial queries). Second, we showed the effectiveness and the superiority of *MBC-based* methods as compared to different shape-based representation techniques.

We also described two original optimization techniques for our *MBC-based* methods. Our new experiments and analytical models showed that we can further improve the performance of our methods in several aspects. First, our experiments showed that the algorithm to find the minimum bounding circle sometimes does perform poorly, despite the fact that it has an expected complexity of $O(n)$ (linear time). To address this problem, we proposed a simple and efficient technique to enhance the performance of the MBC computation algorithm. Our analysis showed that only a small subset S of the points of an object uniquely determines the MBC of the object. To identify the points in S, we first compute the MBR of the object. Then, we identify a smaller enclosed MBR ($RedMBR$) with a specified reduction factor RF. Finally, S can be determined by removing all the points that are inside the $RedMBR$ and apply MBC algorithm to S. To select a proper value for RF, we computed the *optimal value* of RF (larger than or equal to $1-\frac{1}{\sqrt{2}}$) that would guarantee to produce the exact MBC of an object. We evaluated our new algorithm over real and synthetic data. The results showed that our proposed algorithm reduces the execution time of the MBC computation algorithm significantly (up to 90% reduction). Moreover, we were able to find the exact MBC even for cases where $RF <$ *optimal value of* RF (i.e., $RF < 1-\frac{1}{\sqrt{2}}$). Second, our analytical studies in [22] showed that the algorithm to find similar objects in a database under a specified rotation angle incurred the highest cost $O(n_{SP}N^2)$. This is due to the fact that, using different SP (due to the rotation of the object) generates different angle sequences (AS). As a consequence, given a query object, we have to submit n_{SP} queries to the database to find the candidate objects for match. Each query differs in

its *SP*, and hence differs in its phase values but not its magnitude values. Therefore, in this paper we proposed a new technique to improve *MBC-based* support for match queries under a specified rotation angle by utilizing the circular shift property of the *DFT* of a real sequence. Towards this end, we studied the impact of using different starting points (i.e., rotating an object or shifting the angle sequence) on the second phase value of the *DFT* of a real sequence. Our studies showed that the magnitude values of *FT* do not change with different start points. However, the phase values of *FT* change depending on the number of rotations. In addition, we observed a symmetry property of the phase values of the transformed sequence that would reduce the number of queries to only *one* by storing a single phase value (second phase value of a sequence) instead of all the phase values for different rotations/shifting. Furthermore, using a δ function we can determine if two similar real sequences $X1$ and $X2$ (i.e., the Euclidean distance between the magnitude values of their *DFT* is close to zero), are exactly the same or a rotated version of each other under a specific rotation value.

Finally, in Section 6 we will define some open problems in the area and provide some hints on how to approach these problems. For example, we will investigate how to support match queries under other transformations such as mirroring, non-uniform scaling, and shearing.

6 Open problems and future work

We intend to extend this work in several directions. First, we plan to investigate match queries under other transformations such as mirroring, non-uniform scaling, and shearing. For example, shearing and non-uniform scaling may occur either due to distortion of the objects or by having different camera view points in animation and structured video applications. The basic idea is to use a parametrization of the object's boundary which is robust with respect to an affine transformation. Second, since polygons are very limited in approximating objects, we intend to identify better features to represent objects with curved parts, holes, and open curves or lines. Those features to be used to build index structures for similarity match queries. Towards this end, a shape could be represented by sample points obtained from tracing its boundary. Moreover, curves could be handled by using quantization at uniform angles. In addition, holes could be considered as sub-parts of the objects. Hence, we could employ the same proposed shape representation techniques to represent them. However, it is important to identify and maintain the spatial relations (e.g., location, distance, orientation) between the holes of the same object. Finally, we can employ *UNLFT* that can handle open curves and lines. Third, we have noticed that humans consider shapes to be similar if their general descriptions are the same or if they have similar parts, and they do not pay attention to some specific details. In addition, partial similarity match is not supported by all the systems. Therefore, we intend to extend this work to utilize our features to support partial similarity match queries. Towards this end, we can use *FT* of the polygon sections instead of the polygon itself, where each polygon section is represented by an edge sequence of fixed length. Fourth, we plan to study relationships between more than two objects. For example, define topological relations between three objects given that their *MBC*/*MBS* are overlapping. Fifth, we plan to design a topological-direction model for the specification of the spatio–temporal relation-

ships among objects (e.g., objects in a video sequence). Sixth, we want to extend this work to support three dimensional objects. Our preliminary investigations show that analogous to *MBC* features for 2D objects, we can extract features from 3D objects by using their minimum bounding spheres. In addition, by using projection along three faces, we can reduce the 3D problem to three 2D problems. Finally, the optimized *MBC* computation algorithm and the trends in the phase values of the *DFT* of real sequences could be adapted and extended to other frameworks and application domains. For example, in *GIS* systems, *MBC* is sometimes used as an approximation for handling complex spatial objects to improve the performance of approximation-based query processing. The optimized *MBC* computation algorithm can be used to speed up the process of generating an object's approximation. Another application domain could be pattern matching, in which we try to find the occurrences of some pattern (P) in a sequence when also rotations of P are allowed. Using partial matching of the *DFT* of the original sequence, we can detect the occurrence of P in the sequence. In addition, using the phase value of the *DFT* of the sequence we can determine if P is rotated/shifted as well as the exact number of shifts.

Acknowledgments This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC), IIS-0082826 (ITR), IIS-0238560 (CAREER) and IIS-0307908, and unrestricted cash gifts from Okawa Foundation and Microsoft. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Agrawal R, Faloutsos C, Swami A (1993) Efficient similarity search in sequence databases. FODO
2. Bebis GN, Papadourakis GM (1992) Object recognition using invariant object boundary representations and neural network models. Pattern Recognit 25:25–44
3. Chung C, Lee S, Chun S, Kim D, Lee J (2000) Similarity search for multidimensional data sequences. To appear in the proceedings of the 16th international conference on data engineering (ICDE), San Diego, California, February 29–March 3
4. Faloutsos C, Ranganathan M, Manolopoulos Y (1994) Fast subsequence matching in time-series databases. SIGMOD
5. Gary J, Mehrotra R (1995) Similar-shape retrieval in shape data management. IEEE, Computer September
6. Gonzalez RC, Wintz P (1987) Digital image processing, 2nd edn. Addison-Wesley, Reading, Massachusetts
7. Korn F, Sidiropoulos N, Faloutsos C, Siegel E, Protopapas Z (1996) Fast nearest neighbor search in medical image databases. Proc. 22nd VLDB conference, Mumbai, India, pp 215–226
8. Lu G, Sajjanhar A (1992) Region-based shape representation and similarity measure suitable for content-based image retrieval. Multimedia Syst 7(2):165–174, Springer
9. Megiddo N (1984) Linear-time algorithms for linear programming in R^3 and related problems. SIAM Journal of Computation 12:108–116
10. Mehrotra R, Gary JE (1995) Image retrieval using color and shape. Second Asian conference on computer vision, 5–8 December, Singapore. Springer, Berlin Heidelberg New York, pp II-529–II-533
11. Mokhtarian F, Abbasi S, Kitter J (1996) Efficient and robust retrieval by shape content through curvature scale space. Proceedings of international workshop on image database and multimedia search, Amesterdam, Netherlands, pp 35–42
12. Oösterom PV, Claassen E (1990) Orientation insensitive indexing methods for geometric objects. 4th international symposium on spatial data handling, Zurich, Switzerland, pp 1016–1029

13. Oppenheim AV, Schafer RW (1975) Digital signal processing. Prentice Hall, Englewood Cliffs, New Jersey
14. Papadias D, Theodoridis Y, Sellis T, Egenhofer MJ (1995) Topological relations in the world of minimum bounding rectangles: a study with R-trees. Proceedings of ACM SIGMOD international conference on management of data
15. Pitas I (1993) Digital image processing algorithms. Prentice Hall, Englewood Cliffs, New Jersey
16. Safar M, Shahabi C (1999) 2D Topological and direction relations in the world of minimum bounding circles. In the proceedings of IEEE 1999 international database engineering and applications symposium (IDEAS), Montreal, Canada, pp 239–247, August 2–4
17. Safar M, Shahabi C (2001) A framework to evaluate the effectiveness of shape representation techniques. In Journal of Applied Systems Studies, special issue on Distributed Multimedia Systems and Applications, November
18. Safar M. Shahabi C, Sun X (2000) Image retrieval by shape: a comparative study. In the proc. of IEEE international conference on multimedia and exposition (ICME), New York, July 30–August 2
19. Sajjanhar A, Lu G (1997) A grid based shape indexing and retrieval method. Special Issue of Australian Computer Journal on Multimedia Storage and Archiving Systems 29(4):131–140, November
20. Sajjanhar A, Lu G (1998) A comparison of techniques for shape retrieval. International conference on computational intelligence and multimedia applications, Monash University, Gippsland Campus, pp 854–859, 9–11 Feb
21. Sajjanhar A, Lu G, Wright J (1997) An experimental study of moment invariants and Fourier descriptors for shape based image retrieval. Proceedings of the second Australia document computing symposium, Melbourne, Australia, pp 46–54, April 5
22. Shahabi C, Safar M (1999) Efficient retrieval and spatial querying of 2D objects. In the proceedings of IEEE international conference on multimedia computing and systems (ICMCS), pp 611–617, June 7–11, Florence, Italy
23. Shahabi C, Safar M, Hezhi A (1999) Multiple index structures for efficient retrieval of 2D objects. Technical Report, USC-TR-99-694, University of Southern California, 1999
24. Tao Y, Grosky WI (1999) Delaunay triangulation for image object indexing: a novel method for shape representation. Proceedings of the seventh SPIE symposium on storage and retrieval for image and video databases, San Jose, California, pp 631–642, January
25. Tao Y, Grosky WI (1999) Object-based image retrieval using point feature maps. Proceedings of the international conference on database semantics (DS-8), Rotorua, New Zealand, pp 59–73, January
26. Welzl E (1991) Smallest enclosing disks (balls and ellipsoids). New results and new trends in computer science. Lect Notes Comput Sci 555:359–370

Cyrus Shahabi is currently an Associate Professor and the Director of the Information Laboratory (InfoLAB) at the Computer Science Department and also a Research Area Director at the NSF's Integrated Media Systems Center (IMSC) at the University of Southern California. He received his B.S. degree in Computer Engineering from Sharif University of Technology in 1989 and his M.S. and Ph.D. degree in Computer Science from the University of Southern California in 1993 and 1996, respectively. He has two books and more than hundred articles, book chapters, and conference papers in the areas of databases, GIS and multimedia. Dr. Shahabi's current research interests include Geospatial and Multidimensional Data Analysis, Peer-to-Peer Systems and Streaming Architectures. He is currently an associate editor of the IEEE Transactions on Parallel and Distributed Systems (TPDS) and on the editorial board of ACM Computers in Entertainment magazine. He is also in the steering committee of IEEE NetDB and ACM GIS. He serves on many conference program committees such as ACM SIGKDD 2006, IEEE ICDE 2006, ACM CIKM 2005, SSTD 2005 and ACM SIGMOD 2004. Dr. Shahabi is the recipient of the 2002 National Science Foundation CAREER Award and 2003 Presidential Early Career Awards for Scientists and Engineers (PECASE). In 2001, he also received an award from the Okawa Foundations.

Maytham Safar is currently an Assistant Professor at the Computer Engineering Department at Kuwait University. He received his Ph.D. degree in Computer Science from the University of Southern California in 2000. He has one book and more than 25 articles, book chapters, and conference/journal papers in the areas of databases and multimedia. Dr. Safar's current research interests include Peer-to-Peer Networks, Spatial Databases, Multidimensional Databases, and Geographic Information Systems. He served on many conferences as a reviewer and/or a scientific program committee member such as ICDCS, EURASIA-ICT, WWW/Internet, ICWI, ICME, AINA, WEBIST, IPSI, HPC&S, ICICT, i-Society, ET-WBC, ICDIM and iiWAS. He also served as a member on the editorial board or a reviewer for many journals such as IEEE TMJ, ACM Computing Reviews, JDIM, MTAP, IEEE TPAMI, and ACM MSJ.