Incremental indexing and retrieval mechanism for scalable and robust shape matching

Shehzad Khalid *

Department of Computer and Software Engineering, Bahria University, Islamabad, 44000, Pakistan

Abstract

Techniques for efficient and effective content-based image matching are becoming increasingly important with the widespread increase in digital image capturing systems. The fundamental ingredient of content-based image retrieval is the selection of appropriate features to describe the content of the image. Shape of an object, represented by its contour, is the most important visual feature that is thought to be used by humans to determine the similarity of objects. The selected feature and its distance measure must be robust to different distortions such as noise, articulation, scale and rotation. Existing approaches provides invariance to these distortions at the cost of either the accuracy due to poor discrimination ability or the efficiency. In this paper, we present an effective representation of shape, using its boundary information, that is robust to arbitrary distortions and affine transformation. The contour representation of shape is converted into time series and is modeled using orthogonal basis function representations. Shape matching is then carried out in the chosen coefficient feature space. Encoding contour representation of shapes in this manner leads to efficiency gains over existing approaches such as structural shape representation and techniques that use discrete point-based flow vectors to represent the contour. The efficiency of shape matching is further improved by indexing the shape descriptors using hierarchical indexing structure. A novel distributed beam search based technique is proposed that operates on the indexing structure and ensures no false dismissal for a given k-NN query. Experimental evaluation demonstrates that the proposed shape representation and matching mechanism is robust, efficient and scalable to very large shape datasets.

Key words: Shape matching, dimensionality reduction, affine invariant matching, indexing and retrieval, shape distance, pruning.
PACS:

* Corresponding author.
Email address: shehzad_khalid@hotmail.com (Shehzad Khalid).

1 Introduction

The current prevalence of digital image capturing system has prompted much research activity aimed at the development of sophisticated techniques that enables content-based image search. A fundamental feature that determines the content-based similarity of images is the similarity of object's shape in the image. Shape matching is an important problem with its application in diverse domains such as medicine, engineering, industry and security. Shape matching generally looks for effective and perceptually important shape representation and distance measures that are invariant to many distortions including noise, rotation, articulation, scale, jag etc. Rotation invariance is relatively difficult to handle as compared to noise, translation and other distortions. Shape matching using any representation scheme normally gives good results if the two shapes are rotationally aligned. Much of the earlier research on shape matching achieve rotation invariance at the cost of accuracy [1] or efficiency [2][3][4][5]. Our ultimate goal is to generate efficient and accurate rotation-invariant features to compare shapes that are represented using closed planar contours.

Much of the earlier research in shape matching and recognition have focused only on the accuracy of shape matching and retrieval systems. Efficiency requirement is often ignored in the evaluation of shape matching techniques which is equally important as accuracy. An accurate but inefficient retrieval system will not be appreciated due to online retrieval demands. Low computation complexity is an important characteristic of a good shape descriptor. We target efficiency in two ways: decreasing one to one shape matching complexity and allowing indexing.

Efficient shape matching can be achieved using low dimensional feature space representation of shapes. The selected dimensional reduction technique should transform the high dimensional image to compact feature space while retaining most of the characteristic structure of the original data. Related work within the data mining community on approximation schemes for indexing time series data is highly relevant to the parameterisation of contour-based representation of shapes. Shapes can be converted into time series by calculating distance of every point from the shape-centroid. This makes the indexing techniques for time series applicable to contour-based shape matching. For example, Discrete Fourier Transforms (DFT) [6][7], Discrete Wavelet Transforms (DWT) [8], Adaptive Piecewise Constant Approximations (APCA) [9], and Chebyshev polynomials [10] have been used to conduct similarity search in time series data. Sequential matching of query with all shapes in dataset using compact features may retrieve results in reasonable time for small to medium size datasets (number of samples < 10000). However, for large datasets (number of samples $\gg 10000$), even efficient matching using compact features is

too slow. One of the important challenges is to devise search mechanism that is scalable to very large datasets.

The technique proposed in this paper achieves efficient retrieval in large datasets by generating hierarchical indexing structure using compact feature representation of shapes. We apply time series modeling of contour-based shape representation to the problem of shape matching. Contours are modeled using function approximation techniques and matching of shape is carried out in the coefficient subspace. An efficient matching technique is proposed that can match shapes which are not rotationally aligned. The proposed technique is also invariant to noise and different affine transformations. An incremental indexing structure is proposed that enables extremely efficient retrieval. The tree basically defines a hierarchical quantization obtained using recursive clustering of shapes in the dataset. The proposed indexing structure supports online insertion and deletion of shape samples from dataset which is critical for incremental indexing. A novel distributed beam search based retrieval algorithm is proposed that ensures no false dismissal for a k-NN query. This results in exact indexing and retrieval of shapes which will result the same answer for k-NN query as the sequential search.

The remainder of the paper is organized as follows: We review some relevant background material in section 2. In section 3 we present some function approximation approaches to contour-based shape representation. Section 4 addresses the issue of rotation invariant shape matching without compromising on efficiency and accuracy. In section 5, a hierarchical structure is proposed for incremental indexing of shape datasets. A novel approach, referred to as Distributed Beam Search (DBS), for searching using proposed indexing structure is also presented. Experiments have been performed to provide a comprehensive evaluation of our proposed technique and its comparison with existing approaches. These experiments are reported in section 6. The paper concludes with a discussion and proposals for further work.

2 Background and related work

Shape descriptors are known to be useful candidates for content-based image indexing and retrieval schemes. Previous work has sought to represent shapes through shape context, shape signature, integral invariants, curvature, moments etc. Broadly speaking, these shape representation and matching techniques are classified into two classes: 1) contour-based that only exploit contour information and 2) region-based that incorporate all the pixels within the shape to generate shape descriptor.

The contour-based approaches are more common in literature as studies on

human perception have shown that humans can recognize and discriminate shapes mainly by their contour features. We therefore consider only contour-based approaches here. Most of the contour-based shape representation generates a global representation of contour. Some of the global representations include shape context [11][12][13], shape signature [14], integral invariants [15] and differential invariants [16][17]. The shape matching using global approximation is a simple process that requires parameter distance such as euclidean distance [2] or original points-space distance such as DTW [2], hausdorff distance [18] and correspondence-based shape matching [11][12][13]. Point-space distances are not suitable for online shape matching due to high computation costs associated with 2-D mapping of shapes. Some approaches [5][19]segments the contour into pieces and generate a piecewise representation of shapes. Piecewise approaches differ in the segmentation criteria to break contours into pieces and the modeling mechanism used to represent contour segments. Some of the piecewise approaches include polygon decomposition [5], smooth curve decomposition [19] and curvature decomposition [20]. The advantage of piecewise approaches is its ability to support partial matching and as a result dealing with the problem of partial occlusion. However, this merit of piecewise approaches comes with the disadvantage of complex and inefficient matching. Piecewise approaches do not capture global features of shape which is extremely important for shape recognition and discrimination.

Rotation invariance is critical for accurate shape matching and is hard to achieve as compared with invariance to other distortions [2][21]. There exists a variety of techniques that has been used for rotation invariant shape matching. Some approaches [22][23] make use of rotation invariant features including features associated to curvature and centroid distances, ratio of perimeter to area, circularity, convexity, histogram of distances between every pair of points on the contour etc. These approaches achieve rotation invariance by compromising on the accuracy. This is expected as selection of only rotation invariant features may result in discarding other features that may be sensitive to rotation but is important for fine discrimination between different shapes. Approaches using 1D time series representation of shapes has also been proposed [2][24][25][26][27]. Some of these approaches [26] achieve rotation invariance by selecting very few starting point to obtain 1D time series representation of 2D shape. Normally shapes are aligned with respect to the major axis. However, such alignment are very unreliable specially when there is no well defined major-axis and slight articulation in shape may have significant impact on rotation alignment. Researchers recognize the risks associated with shape alignment using one or very few points on contour. Some researchers proposed to use brute-force search over all possible rotation to identify the true alignment of shapes [2][24][25]. We need to shift one contour n times ($n \gg 100$) where n is the number of points on the contour. The matching of one shape is carried out with n different alignments of the other shape which jeopardises the efficiency requirement of content-based image search and retrieval systems.

The contribution of this paper is to show that a low dimensional coefficient-based contour encoding scheme can be used more efficiently for image searching than previous approaches that rely on high dimensional representation of shapes. The parameter subspace representation of contours is also robust to the presence of different levels of noise and other distortions. An efficient mechanism to achieve rotation invariance in shape matching without compromising on efficiency and accuracy is proposed.

3 Shape representation

In this section, we present our global contour-based shape representation scheme based on time series representation. Formally, the object contour *C(O)* is defined by the point sequence:

$$C(O) = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \tag{1}$$

where (x_1, y_1) represents the location of the left-top point on the contour and n is the number of points on the contour.

The 2-D raw point-based feature vector C is then converted into 1-D centroid distance CD based time series by mapping each point on the contour to the distance between the point and the shape's centroid as:

$$CD_t = \{\sqrt{((x_t - x_c)^2 + (y_t - y_c)^2)}\} \qquad t = 1, 2, 3, ..., n \tag{2}$$

where

$$x_c = \frac{1}{n}\sum_{t=1}^{n} x_t\,, \quad y_c = \frac{1}{n}\sum_{t=1}^{n} y_t \tag{3}$$

The scale variance in shape representation is taken care of by normalizing the distance vector CD with respect to the standard deviation σ of the centroid distances as:

$$CD_t = \frac{CD_t}{\sigma} \qquad t = 1, 2, 3, ..., n \tag{4}$$

The process of conversion of shape into 1-D time series representation is depicted graphically in Figure 1. A sample shape of fighter plane is shown in Figure 1(a). Figure 1(b) presents the projection of plane in C space with 'o' marker highlighting the starting (left-top) point on the contour. The process of generating 1-D time series from 2-D contour is highlighted in Figure 1(c).

The distances of every point on the contour to the centroid is represented as 1-D time series in *CD* space. The projection of shape in *CD* space is presented in Figure 1(d).

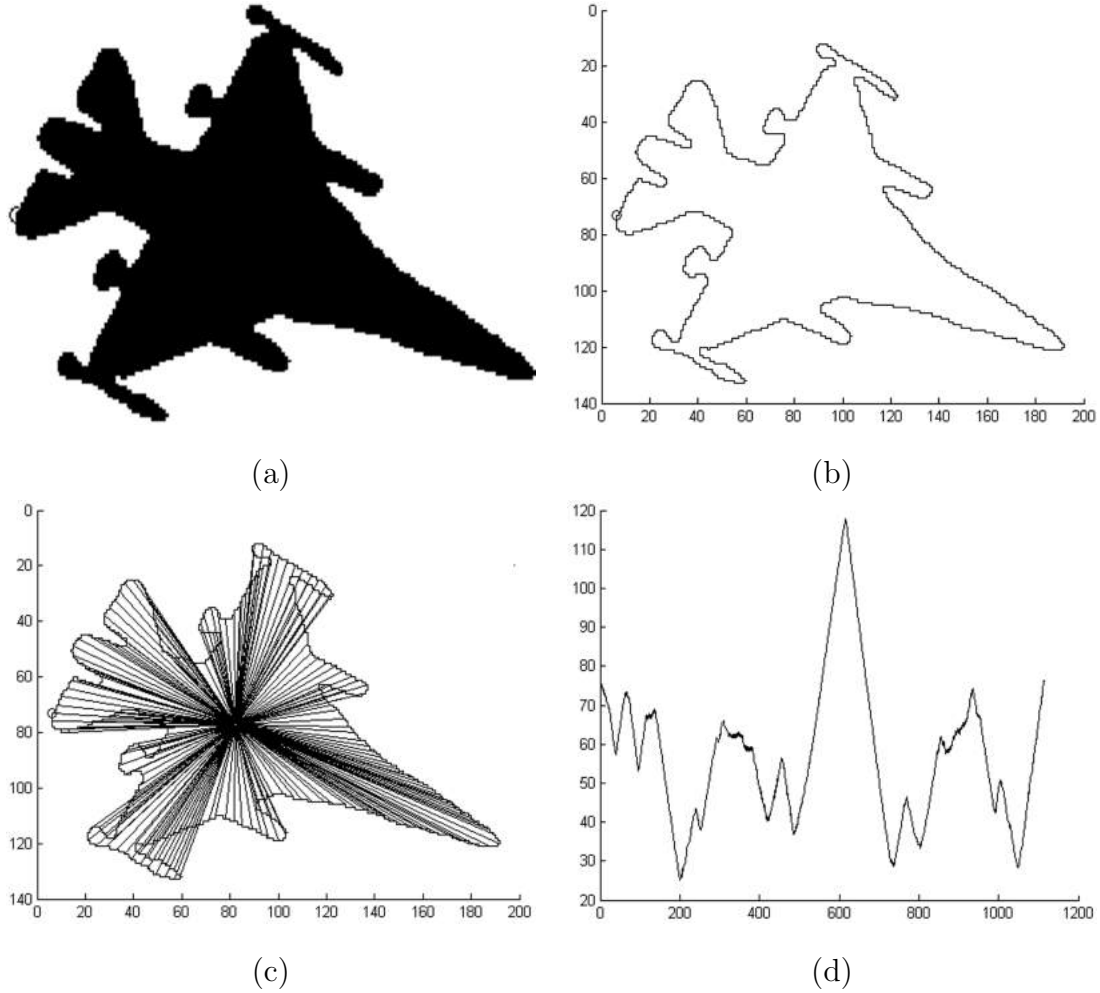


Fig. 1. Extracting 1D time-series representation of contour-based shapes. (a) Projection of shape in image plane. (b) Projection in C space with 'o' marker specifying the starting point (c) Mapping of contour from 2D C-space to 1D CD-space (d) Projection in CD-space.

For small objects with relatively simple shapes ($n < 100$) it is feasible to work on raw centroid distance vector CD. However for realistic scenarios, $n >> 100$ and this renders direct manipulation of point sequence impractical for retrieval purposes. The key to implementing efficient trajectory matching is dimensionality reduction. The idea is to determine a feature extraction function F that reduces the dimensionality of the data from n to m such that $m \ll n$. Similarity search and retrieval is then conducted in the reduced feature space. It is important that the feature space captures the most salient characteristics of the raw distance vector space. Influential work within the data mining community on indexing techniques for time series is highly relevant to the parameterisation of CD-space representation of shapes. For example, Discrete Fourier

Transform (DFT) [6], Discrete Wavelet Transform (DWT) [8], Adaptive Piecewise Constant Approximation (APCA) [9], and Chebyshev polynomials [10] have been used to index time series and perform similarity retrieval.

We have employed Discrete Fourier Transform (DFT) to model the projection of shape in CD based time series representation. The n-point DFT of $\{CD_i\}$, defined as a sequence $\{\overline{CD_f}\}$ of n complex numbers ($f = 1, ..., n$), is given as:

$$\overline{CD_f} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} CD_i exp(-j2\pi fi/n) \quad f = 1, 2, ..., n \tag{5}$$

where j is the imaginary unit $j = \sqrt{-1}$, and $\overline{CD_f}$ are complex numbers with the exception of $\overline{CD_0}$ which is real. Typically, the DFT sequence is truncated after m terms, $f = 0, ..., m-1$. In this case, the feature vector consists of $2m-1$ entries (from real and imaginary parts). More formally, let a_i and $\widehat{a_i}$ be the real and imaginary part of $\overline{CD}$. Shapes can be represented in the coefficient feature space by a $2m-1$ dimensional vector of DFT coefficients $\mathbf{F}_{DFT}$, where

$$\mathbf{F}_{DFT} = [a_0, a_1, \widehat{a}_1, ..., a_{m-1}, \widehat{a}_{m-1}] \tag{6}$$

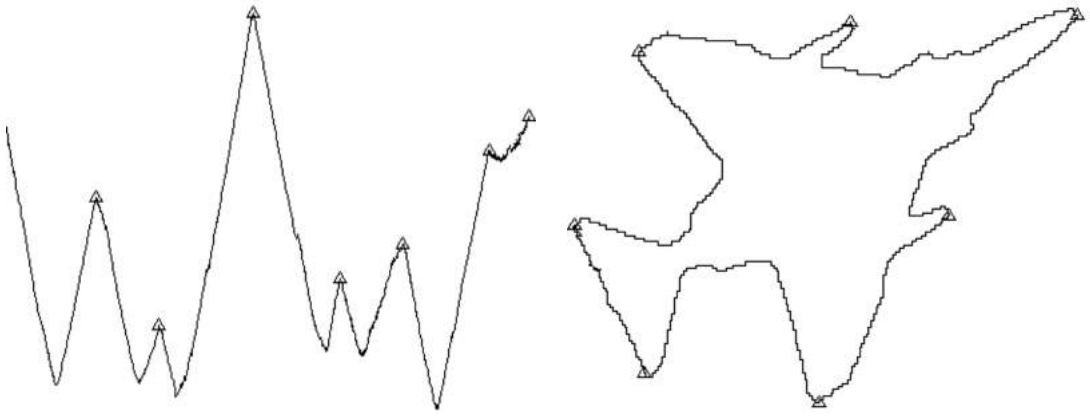
The DFT based projection of 2D shapes results in consistent size (m) feature vector irrespective of the number of points on the shape contour. The distance between two shapes is measured by calculating Euclidean distance between the DFT-based feature space representation of shapes. The selection of DFT for coefficient feature space representation of contour as compared to its competitive descriptors such as Chebyshev (CS)[10] and Piecewise Aggregate Approximation (PAA) [28] is justified through experimental evaluation as presented in section6.

4 Rotation Invariant Shape Representation

The mechanism specified in section 3 generates a contour distance (CD) based time series representation of shapes. The distance measure can than be calculated using DFT-based feature space representation of contours. This method produces good results if CD-based time series representation of two shapes are rotation aligned. However, this method can produce extremely poor results if the two shapes are not rotation aligned. Existing techniques [2][3][4][24][25] solve this problem by keeping one shape fixed and rotating the other shape. The distance between the shapes is the minimum of the distances of the fixed shape with all possible rotations of the other shape. This minimum distance is then assumed to be the true distance between the shapes. This brute force

approach, though effective, is extremely inefficient. Let n ($n >> 100$) be the length of *CD*-based representation of contour, then we need to calculate n shape distances to achieve a rotational invariant shape matching. This cripples our one of the primary objective of efficient shape matching.

To overcome inefficiency in rotation invariant shape matching, we propose a Critical-point based approach for Rotational Invariant Shape Matching (CRISM). Instead of calculating the distance of fixed shape with all possible rotations of other shape, we rotate both the shapes along the selected number of critical points on the contour and use them as starting points to convert contour into *CD*-based time series representation. The critical points in the contours are extracted by identifying local maximas in *CD*-space. The process of identifying critical points on the contour is highlighted in Figure 2. Figure 2(a) presents the *CD*-space representation of shape with critical points highlighted using '$_{\triangle}$' marker. Figure 2(b) presents the 2D contour-based representation of shape with '$_{\triangle}$' markers superimposed on critical points.



(a) (b)

Fig. 2. Identification of critical points using local-maxima heuristic. Critical points are highlighted using '$_{\triangle}$' marker (a) in 1D *CD*-space representation (b) in 2D contour-space representation

The contour of the fixed shape is extracted by starting from the left-top point on the contour, as specified in eq. (1). The centroid distance of left-top point on a contour represents one of the local maxima in the CD space. The fixed shape therefore has a default alignment w.r.t. one of the local maxima.

The distance between the fixed shape is then calculated with a set of only critical rotations of the other shape. The critical rotations of the shapes that gives the minimum distance value will result into correct alignment of two shapes and will return the rotation invariant distance. Comparison of proposed critical-point alignment with computationally expensive brute-force alignment is presented in Figure 3. For different pair of shapes, the alignment obtained using brute-force alignment is no better than the alignment obtained

using critical-point approach. Brute force alignment requires shifting one of the curves n times where n is the number of points on the curve. On the other hand, critical-point alignment requires shifting the curve nc times where nc is the number of critical points on the curve. As $nc \ll n$, we have managed to achieve efficient rotation invariance without compromising the accuracy of rotation invariance shape matching.

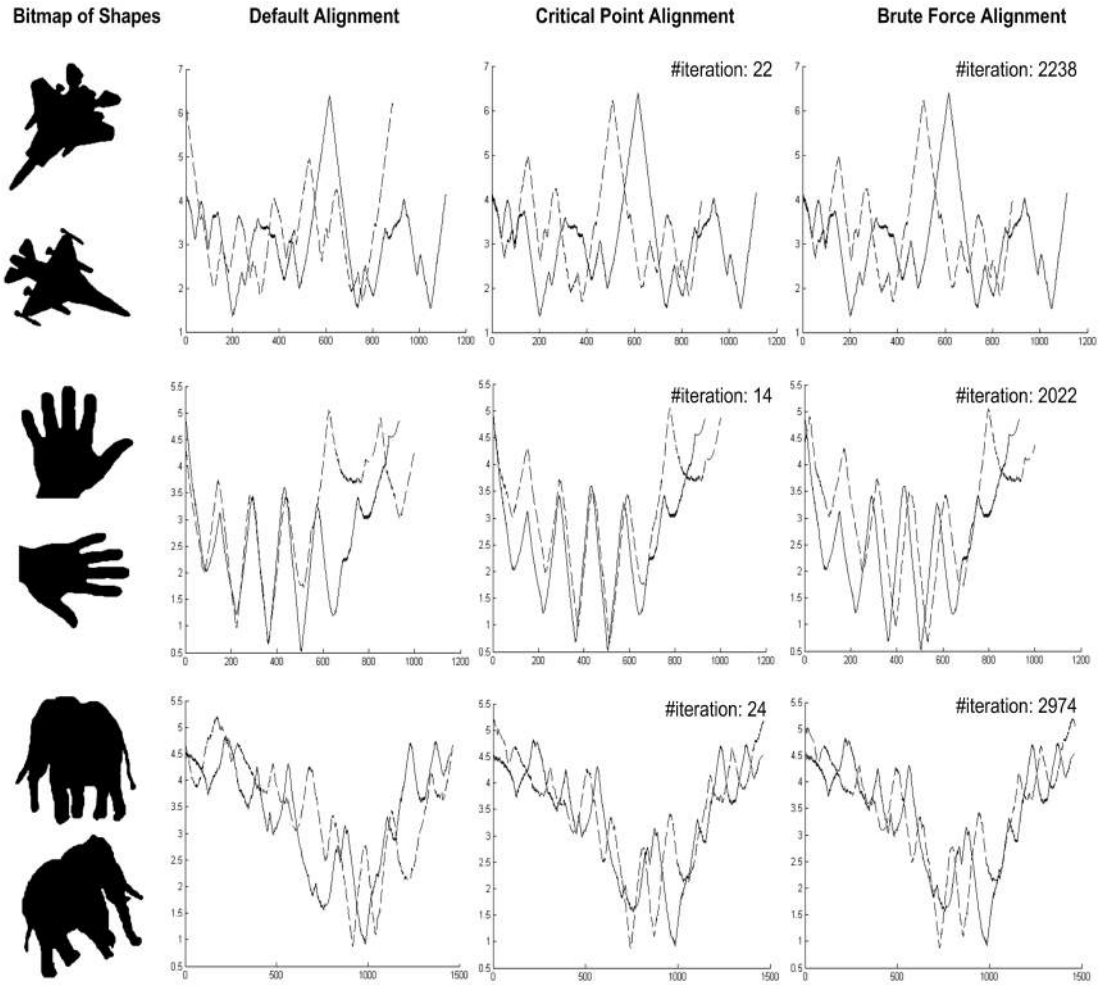


Fig. 3. Comparison of rotational invariant shape alignment using critical-point and brute-force alignment.

More formally, suppose we have a time series A representing a shape in CD-space as:

$$A = a_1, a_2, a_3, \ldots, a_p \tag{7}$$

where p is the length of time series A. We identify a set of nc_A critical points $C = \{c_1, c_2, .., c_{nc_A}\}$ using time series A. We achieve rotation invariance by

expanding B into a matrix $\mathbf{A}$ of nc_A time series as:

$$\mathbf{A} = \begin{bmatrix} a_{c_1}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_1-1} \\ a_{c_2}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_2-1} \\ \vdots \\ a_{c_{nc_A}}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_{nc_A}-1} \end{bmatrix} \tag{8}$$

Each row in matrix $\mathbf{A}$ is a time series representing contour in CD-space aligned w.r.t. one of the critical point. To make our distance measure invariant to mirror images, we pad our matrix $\mathbf{A}$ with the reverse of all the time series as:

$$\mathbf{A} = \begin{bmatrix} a_{c_1}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_1-1} \\ a_{c_2}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_2-1} \\ \vdots \\ a_{c_{nc_A}}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_{nc_A}-1} \\ a_{c_1-1}, \cdots, a_1, a_q, a_{q-1}, \cdots, a_{c_1} \\ a_{c_2-1}, \cdots, a_1, a_q, a_{q-1}, \cdots, a_{c_2} \\ \vdots \\ a_{c_{nc_A}-1}, \cdots, a_1, a_q, a_{q-1}, \cdots, a_{c_{nc_A}} \end{bmatrix} \tag{9}$$

The proposed framework for rotational invariant shape matching also supports partial rotational invariance where we only want to allow limited rotation in shape matching. This can be achieved by rotating one of the shape along limited number of critical points. This can be specified as:

$$\mathbf{A} = \begin{bmatrix} a_{c_1}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_1-1} \\ a_{c_2}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_2-1} \\ \vdots \\ a_{c_i}, \cdots, a_{q-1}, a_q, a_1, \cdots, a_{c_i-1} \\ a_{c_{nc_A-i+1}-1}, \cdots, a_1, a_q, a_{q-1}, \cdots, a_{c_{nc_A-i+1}} \\ a_{c_{nc_A-i+2}-1}, \cdots, a_1, a_q, a_{q-1}, \cdots, a_{c_{nc_A-i+2}} \\ \vdots \\ a_{c_{nc_A}-1}, \cdots, a_1, a_q, a_{q-1}, \cdots, a_{c_{nc_A}} \end{bmatrix} \tag{10}$$

where $i < (\frac{nc_A}{2})$.

Let B be the time series representing some other shape in CD-space, $C = \{c_1, c_2, .., c_{nc_A}\}$ be the set of nc_B critical points identified using time series B and $\mathbf{B}$ be a feature matrix of B as obtained using eq. (10) using corresponding critical points for B. The rotational invariant distance between shapes A and B can then be specified as:

$$RID(\mathbf{A}, \mathbf{B}) = \min_{1 \leq i \leq nc} \min_{1 \leq j \leq nc} (DIST(DFT(A_{c_j}), DFT(B_{c_i}))) \tag{11}$$

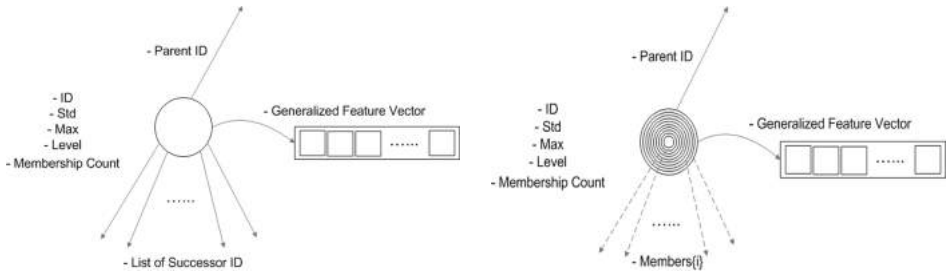
where $DFT(.)$ is DFT based dimensionality reduction function and $DIST(.,.)$ is the distance measure. The distance function commonly used for matching coefficient feature space representation of shape is euclidean distance. However, DTW can also be used to match PAA based coefficient feature space representation of shapes. For euclidean distance, the time complexity of querying a shape database using CRISM algorithm is $O(2 * nc_A * nc_B * m * N)$ where nc_A and nc_B are the number of critical points, m is the size of feature vector used for shape representation and N is the number of samples in a shape dataset. However, the number of critical points rarely exceeds 15 even for complex shapes and the value of m is on the order of 8 to 32.

5 Hierarchical Tree-Based Indexing and Retrieval

Given a very large shape dataset, it is very difficult to meet online retrieval demands even in coefficient feature space representation due to sequential matching of query shape with all shapes in the dataset. To solve this problem, we wish to quickly filter distant shapes and identify a short list of candidate shapes that includes the k-nearest neighbours of the query. Shape matching should than be carried out with the identified subset of candidate shapes. In this section, we propose a hierarchical tree based indexing and retrieval technique that address this issue. The tree basically defines a hierarchical quantization obtained using recursive clustering of shapes in the dataset. The algorithms for major operations on the tree including creation, updation and search are presented in following subsections.

5.1 Building Hierarchical Index Structure

The proposed algorithm for creating shape-database index is a tree-based indexing mechanism that takes the entire set of shapes and attempts to determine b^* groupings in the dataset. Each grouping in represented by a node data



(a) (b)

Fig. 4. Depiction of node data structures in the indexed tree for (a) non-leaf nodes (b) leaf nodes.

structure in the tree. Each node has a parent, a state represented by a generalized feature vector representation of the shapes and other bookkeeping fields. The node data structure for non-leaf and leaf nodes are depicted in Fig. 4. Details of different bookkeeping fields in node data structures are presented in Table 1. If the membership count of any node exceeds a membership threshold, the node split occurs. This is done by identifying further b^* groupings from the subset of shape represented by the node. The process is repeated until the membership count of none of the node exceeds the membership threshold.

The identification of hierarchical groupings to generate tree-based indexing structure is based on iterative SOM-based clustering algorithm which comprises the following steps:

(1) Initialize SOM network with the b^* output nodes.
(2) Estimate mean (μ) and covariance (Σ) of DFT-based coefficient feature space representation of shapes in DB using their default alignment. Initialize weight vectors W_i (where $1 \leq i \leq \#_{output}$) from the $PDF\ N(\mu, \Sigma)$.
(3) Sequentially input feature vectors from DB and identify k Nearest Weights (k-NW) to input feature vector using:

$$k - NM(\mathbf{A}, \mathbf{W}, k) = \{\mathbf{C} \in \mathbf{W} | \forall R \in \mathbf{C}, S \in \mathbf{W} - \mathbf{C}, \\ RID(\mathrm{R}, \mathbf{A}) \leq RID(S, \mathbf{A}) \wedge |\mathbf{C}| = k\} \quad (12)$$

where $\mathbf{W}$ is the set of all weight vectors, C is the set of k closest weight vectors and $RID(R, \mathbf{A}) = \min_{1 \leq i \leq nc}(DIST(R, A_{c_i}))$. The value of k determines the number of output nodes that are nearest to $\mathbf{A}$ and will be updated in the specific iteration of learning process. For a given training cycle t, $k = \delta(t)$ where $\delta(t)$ is a neighborhood size function.
(4) Train SOM network by adjusting a subset of weight vectors (C) using

$$W_c(t+1) = W_c(t) + \alpha(t)\zeta(j)(F - W_c(t)) \quad \forall W_c \in C \quad (13)$$

where W_c is the weight vector representation of output node c, j is the order of closeness of W_c to F $(1 \leq j \leq k)$, $\zeta(j,k) = exp(-(j-1)^2/2k^2)$

Bookkeeping field	Symbol	Description
ID	ID	Identity number to uniquely identify the node.
Parent ID	$parent\_ID$	Identity number of the parent node which later be used for bottom-up parsing of tree.
Level	l	Depth level of the node in the tree.
Generalized Feature Vector	$\overline{W}$	Stores the descriptor feature vector representation of the group (represented by the node) that is closest to the group center.
List of Successor ID	$successors$	Bookkeeping field of non-leaf nodes that stores the list of pointers (represented by solid arrows) of the child nodes which will later be used for top-down parsing of tree.
List of members shape	$members\{i\}$	Bookkeeping field of leaf nodes that stores the list of pointers (represented by dashed arrows) of the shape IDs indexed by the leaf node. Within the node, members are further grouped into $i = \{1, .., p\}$ bins (represented by rings in Figure 4b) w.r.t. the distance of sample from the generalized feature vector to improve pruning power. Bin 1 is closest and bin p is furthest from node center.
Max statistic	max	Specifies the maximum of the distance of all the member shape samples from the generalized feature vector of the node.
STD statistic	std	Specifies the standard deviation of the distance of all member shape samples from the generalized feature vector of the node.
Membership count	$count$	Total number of members indexed with the node of the tree.

Table 1
Description of bookkeeping fields in the node data structure.

is a membership function, $\alpha(t)$ is the learning rate of SOM and t is the training cycle index.

(5) Decrease the learning rate $\alpha(t)$ and neighborhood size $\delta(t)$ exponentially over time.

(6) Repeat steps 4-6 for all the training iterations. The resultant weight vectors are the cluster center representation for the b^* groupings in a given set of shapes.

(7) Generate b^* nodes to represent b^* groupings and set:

$$\begin{aligned} \Gamma_i.\overline{W} &= W_i \\ \Gamma_i.max &= \max_{\forall \mathbf{A} \in \Gamma_i}(RID(\Gamma_i.\overline{W}, \mathbf{A})) \qquad for \;\; i = 1, .., k \\ \Gamma_i.std &= \textstyle\sum_{\forall \mathbf{A} \in \Gamma_i}(RID(\Gamma_i.\overline{W}, \mathbf{A})/|\Gamma_i|) \end{aligned} \tag{14}$$

where Γ is the node structure and $|\Gamma_i|$ is the membership count of Γ_i. Add these nodes in the index tree as the successor (child nodes) of the parent node.

(8) Terminate the algorithm if the indexing process is stable. The tree-based indexing structure is considered stable if the membership count of none of the identified grouping is greater than a cluster membership threshold κ. The validation process can be specified as:

$$\widehat{\mathbf{\Gamma}} = \{\Gamma_i \in \mathbf{\Gamma} || \Gamma_i| > \kappa\} \quad \forall i \tag{15}$$

If the indexing process is unstable ($\widehat{\mathbf{\Gamma}} \neq \{\}$), the subset of shapes represented by cluster $\Gamma \in \widehat{\mathbf{\Gamma}}$ is treated as another set of shapes. The algorithm goes to step (1) and attempts to identify further b* groupings in the given set of shapes.

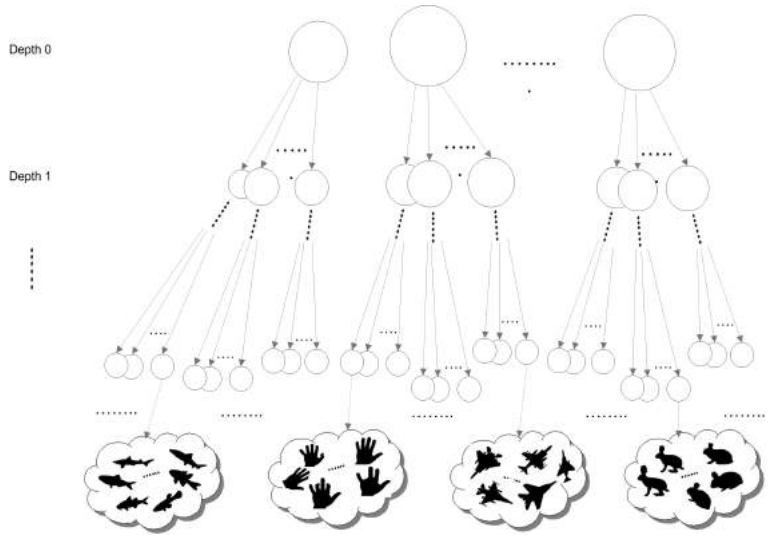


Fig. 5. Depiction of a hierarchical tree-based indexing structure generated using proposed indexing mechanism.

Depiction of hierarchical tree-based indexing structure, generated using the proposed indexing algorithm, for indexing of large shape datasets is presented in Figure 5. Blob area of each node represents its membership count. Solid arrows represents the link from the parent to the child node of the tree where as dotted arrow represents the link from the leaf node to its indexed subset of shapes in the dataset. The proposed top-down approach for hierarchical index structure in more efficient than the alternative bottom-up approach. The time

complexity of generating hierarchical index structure using top-down approach is $O(l * b^* * N)$ instead of $O(b^{*l} * N)$ for bottom-up approach where N is the number of training iterations.

5.2 Updating Hierarchical Index Structure

One of the important operation on the indexing structure is updation which is critical for incremental indexing of shapes. The main aim of updation is to insert or remove shapes from the indexed dataset without the requirement of re-building the index structure all over again which is computationally expensive for gigantic datasets. The proposed hierarchical tree-based index structure is flexible to any updates caused by insertion or deletion of shape samples in the database.

5.2.1 Insertion Algorithm

The algorithm for insertion of new shapes in the indexed dataset and updation of hierarchical tree-based index structure comprises the following steps

(1) Generate matrix-based feature vector representation ($\mathbf{A}$) of shape to be inserted in the indexed dataset
(2) Initialize candidate search list $\mathbf{\Gamma}$ with the nodes present at level 0 of the tree.
(3) Classify the sample shape to one of the nodes from $\mathbf{\Gamma}$. This is done by calculating the rotational invariant distance (RID) between the sample shape and the weight vectors associated to each node in the search list and identify the closest node as:

$$c = arg\ min_{\,k}\ \ RID(\Gamma_k.\overline{W}, \mathbf{A}) \qquad \forall k \tag{16}$$

where $\Gamma_k.\overline{W}$ is the weight vector associate to node Γ_k
(4) Check the validity of classification process. The classification of sample shape to node Γ_c is considered to be valid if:

$$RID(\Gamma_c.\overline{W}, \mathbf{A}) \leq max(\Gamma_c.max, 3 * \Gamma_c.std) \tag{17}$$

(5) If the condition specified in eq. (20) is satisfied and Γ_c is a non-leaf node, set candidate list $\mathbf{\Gamma}$ to the successors (children nodes) of Γ_c. Go to step (3).
(6) If the condition specified in eq. (20) is satisfied and Γ_c is a leaf node, index the shape with the ID of Γ_c and update the max and std statistic assocaited to Γ_c . If the membership count of Γ_c gets greater than κ, Γ_c becomes unstable and is split into b^* child nodes by using the algorithm

specified in section 5.1. Subset of shapes from DB that are indexed with Γ_c are used for sub-tree generation.

(7) If the condition specified in eq. (20) is not satisfied, generate a new node Γ_{new} at the depth level equivalent to $\Gamma_c.l$. Set $\Gamma_{new}.parent\_ID = \Gamma_{new}.parent\_ID$ and set $\Gamma_{new}.\overline{W}$ to the feature vector associated to the default alignment of the shape. The shape is than indexed with the ID of Γ_{new} and is inserted into the database.

5.2.2 Deletion Algorithm

The algorithm for removing shapes from the dataset and updation of hierarchical tree-based index structure comprises the following steps

(1) Remove the shape ID from the *members* field of the leaf node with which the shape is indexed.
(2) If the leaf node contains more members, recursively update the book-keeping fields (max, std and membership count) of the leaf-node and its parent nodes.
(3) If the leaf node does not contain any member, remove the leaf node and update the successor list of its parent node. If the successor list of parent node gets empty, remove the node and update is corresponding parent. Repeat this process till the successor list of the parent node is a not empty list.

5.3 Retrieval Algorithm

This section presents a retrieval algorithm to search k-NN of the query shape using index tree structure while ensuring no false negatives. The basic idea of the algorithm is to exploit multiple paths instead of moving along single branch of the tree in order to exploit all possible nodes that may contain the desired result of query shapes. Parsing the index tree using single branch will give correct results if the query shape is closer to the generalized feature vector representation of the selected nodes along the tree-based indexing structure. However, if the query shape lies at the boundary of different groupings of shapes indexed by different nodes, the desired results can not be achieved. This phenomena is highlighted in Figure 6. Figure 6(a) and Figure 6(b) depicts the problem of single-branch parsing and its possible solution using multi-branch parsing in tree-based indexing space and 2D simulated space respectively. Single-branch parsing selects only one node at each level of the tree where as the desired query result may contain shapes that are indexed by different nodes. This will result in the presence of false negatives in the candidate search list thus effecting the search results. The problem can be solved by parsing

through multiple branches to select all nodes that may contain subset of shapes of the desired result. Figure 6(b) presents the simulation of this phenomena in 2D space for ease of visualization and understanding. Each point in the plot is representing a shape in 2D space and shape indexed with same node id is represented with similar marker. Points enclosed in dashed ellipse represents the search results for the query represented by '+' marker. The proposed

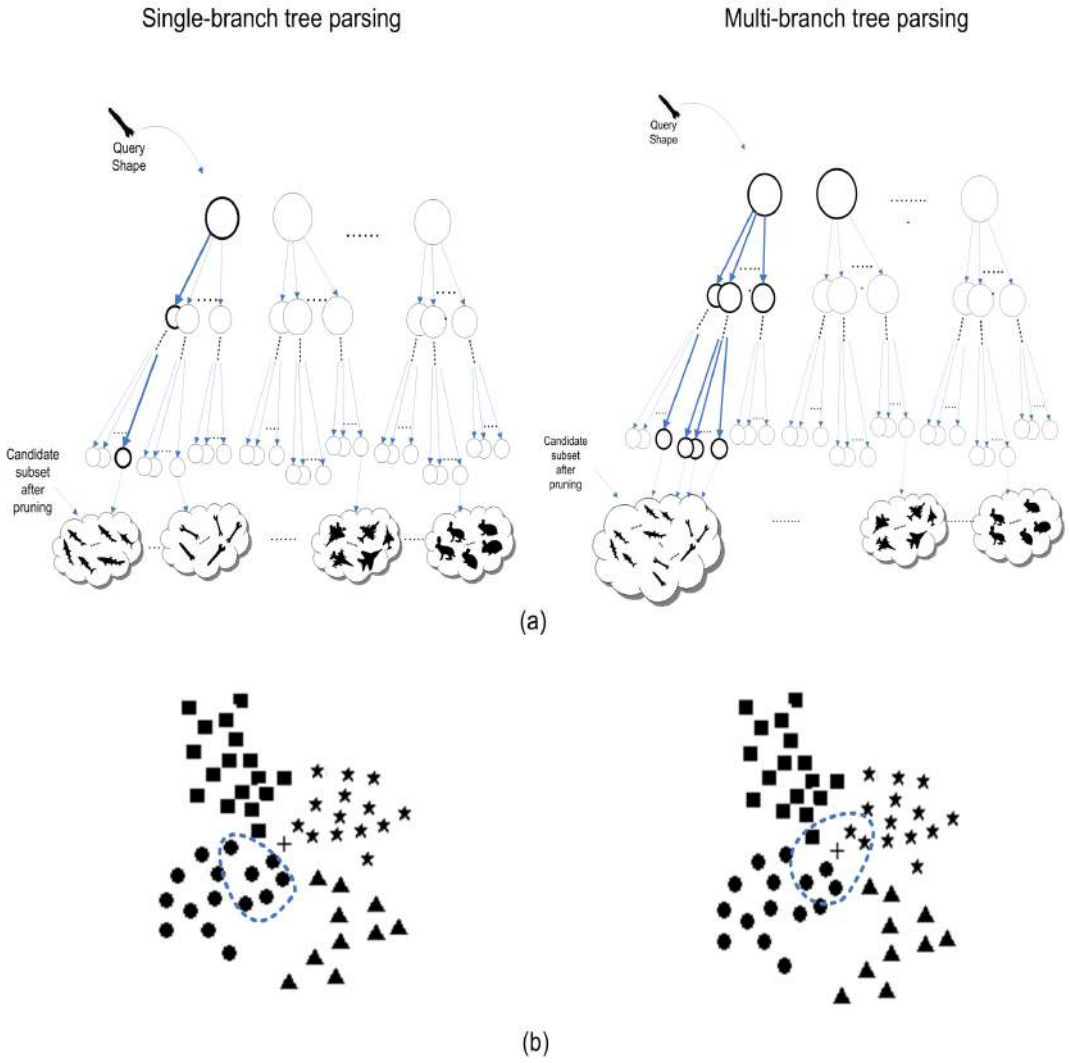


Fig. 6. Depiction of k-NN query using single-branch and multi-branch parsing of hierarchical index structure in (a) tree-based indexing space (b) simulated 2D space.

retrieval approach, referred to as Distributed Beam Search (DBS), make use of this multi-branch parsing method. The algorithm dynamically checks the eligibility of nodes to be included in search at different level of indexed tree structure. The algorithm for k-NN retrieval using proposed DBS technique comprises the following steps:

(1) Generate matrix-based feature vector representation ($\mathbf{A}$) of the query shape.

(2) Initialize candidate search list $\mathbf{\Gamma}$ with the nodes present at depth 0 of the tree.
(3) Sort $\mathbf{\Gamma}$ in ascending order with respect to the distance of generalized feature vector representation of all nodes $\Gamma \in \mathbf{\Gamma}$ from query $\mathbf{A}$.
(4) Identify subset of nearest nodes from query shape as:

$$\mathbf{\Gamma}_\imath = \{\{\Gamma_1, \Gamma_2, ..., \Gamma_p\} \in \mathbf{\Gamma}\} | \sum_{i=1}^{p} \Gamma_i.count \geq k \wedge \sum_{i=1}^{p-1} \Gamma_i.count < k\} \quad (18)$$

where p is the index of most distant node from query shape in $\mathbf{\Gamma}_\imath$.
(5) Identify second subset of nodes from $\mathbf{\Gamma}$ to ensure no false negatives as:

$$\mathbf{\Gamma}_\jmath = \{\Gamma_i \in \mathbf{\Gamma} \| (RID(\Gamma_i.\overline{W}, \mathbf{A}) - \Gamma_i.max) \leq (RID(\Gamma_p.\overline{W}, \mathbf{A}) + \Gamma_p.max)\} \ \ \forall i (19)$$

(6) Set $\mathbf{\Gamma} = \mathbf{\Gamma}_\imath \cup \mathbf{\Gamma}_\jmath$.
(7) If there are non-leaf nodes in $\mathbf{\Gamma}$, replace them with their child nodes. Repeat steps 3-7 till there are no leaf nodes in $\mathbf{\Gamma}$.
(8) Retrieve the feature vector representation of shapes that are indexed by nodes present in $\mathbf{\Gamma}$ satisfying the following condition:

$$RID(\Gamma_i.\overline{W}, \mathbf{A}) \leq (RID(\Gamma_p.\overline{W}, \mathbf{A}) + \Gamma_p.max)\} \ \ \forall i \in \mathbf{\Gamma} \quad (20)$$

(9) For the nodes that do not satisfy condition specified in eq. (23), retrieve only subset of shapes that belong to the outer rings of the nodes whose distance from query sample $\mathbf{A}$ is less than $(RID(\Gamma_p.\overline{W}, \mathbf{A}) + \Gamma_p.max)$. The process of retrieval of shapes indexed by nodes in $\mathbf{\Gamma}$ is highlighted in Fig. 7. The query sample in state space is represented by '+' marker. The blob area of each node represents the max statistics and the ring represents the bins within each node based on the distance from node center. The samples lying in shaded region of the state space is selected for sequential matching with query shape.
(10) Let DB_{pruned} be the set of samples retrieved in steps (8) and (9), identify k-NN to query $\mathbf{A}$ using:

$$k - NN(\mathbf{A}, DB_{pruned}, k) = \{\mathbf{C} \in DB_{pruned} | \forall R \in \mathbf{C}, S \in DB_{pruned} - \mathbf{C}, RID(R, \mathbf{A}) \leq RID(S, \mathbf{A}) \wedge |\mathbf{C}| = k\} (21)$$

where $\mathbf{R}$ and $\mathbf{S}$ are matrix-based feature vector representation of shapes in DB_{pruned}.

6 Experimental Results

This section analyzes the performance of proposed approach of shape matching with respect to the accuracy and efficiency requirement of shape search

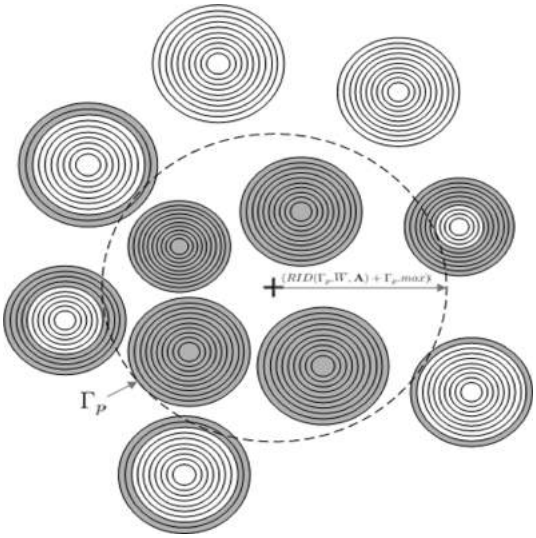


Fig. 7. Depiction of pruning with proposed DBS using ring-based structure of leaf nodes.

from very large shape databases. The experiment has been conducted on variety of shape datasets including silhouette dataset [35], projectile points and hetrogenous dataset [2], chicken pieces dataset [], MixedBag dataset [] and Diatoms dataset [].

6.1 Experiment 1: Performance evaluation of dimensionality reduction techniques

The purpose of this experiment is to investigate the robustness of DFT-based dimensionality reduction technique (DR) to the real life problem of noise and other distortions in shapes as compared to its competitive techniques including Chebyshev (CS) and Piecewise Aggregate Approximation (PAA). The experiment has been conducted on noisy shapes from silhouette dataset [35]. We have randomly selected six samples from each of the shape categories that are present in the dataset.The evaluation metrics used for the comparison of various DR techniques and distance measures are Exact Retrieval Accuracy (ERA) and Class Retrieval Accuracy (CRA). In the context of the current experimental evaluation, ERA can be defined as the ratio of the number of 1-NN queries that retrieve the desired result to the total number of queries. CRA can be defined as the ratio of the number of correct closest matches in 6-NN queries that retrieve the desired result to the total number of retrievals.

The silhouette dataset [35] provides ground truth (i.e. manually labelled) shapes and so we try to simulate the effects of noise. A dataset is corrupted by moving all points on the contour in the normal direction by a certain distance d which determines the amount of noise that is induced in the shape. The

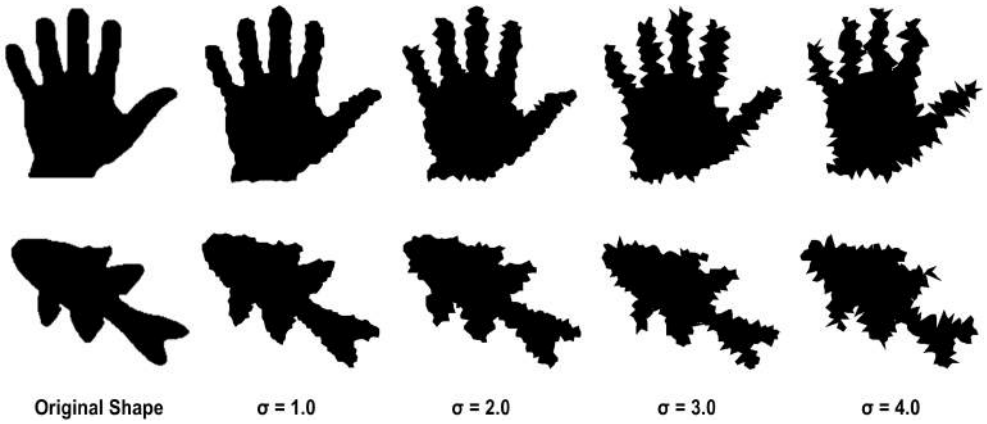


Fig. 8. Effect of simulated noise using increasing value of σ on sample shapes.

value of d is generated from a zero-mean gaussian distribution and standard deviation of σ. If $\mathbf{S}$ represents the original data, a noise corrupted dataset $\mathbf{S_C}$ is produced by adding the term $N[0, \sigma]$ to each (x, y) coordinate on the contour in the normal direction. We set $\sigma = \{1.0, 2.0, 3.0, 4.0\}$ to simulate different noise levels. Simulation of different level of noise on two shapes is presented in Fig. 4. Coefficient feature vector of contours in $\mathbf{S}$ and $\mathbf{S_C}$ are generated separately using DFT, CS and PAA. Each corrupted trajectory in $\mathbf{S_C}$ is then selected as an example query Q_C and we search for a set of k nearest matches in the original dataset $\mathbf{S}$. This is defined as:

$$k - NN(Q_C, \mathbf{S}, k) = \{\mathbf{R} \in \mathbf{S} | \forall A \in \mathbf{R}, B \in \mathbf{S} - \mathbf{R}, \\ RID(A, Q_C) \leq RID(B, Q_C) \wedge |\mathbf{R}| = k\} \tag{22}$$

For ERA-based evaluation metric, we set $k = 1$ in eq. (20). A set of rankings $\forall Q_C \in \mathbf{S_C}$ is produced. The closest match to Q_C should be its corresponding uncorrupted version in $\mathbf{S}$ which produces a rank value of unity. For ease of comparison we record the proportion of times (as a percentage) the query shape is ranked correctly as unity when taken over all S_C. For CRA-based evaluation metric, we set $k = 6$ in eq. (20) as there are six members in each shape class. A set of rankings $\forall Q_C \in \mathbf{S_C}$ is produced. A rank value for each 6-NN query is calculated as a percentage of the correct closest matches in nearest six matches. ERA percentage is then the average rank values for $\forall Q_C \in \mathbf{S_C}$.

For comparison of distance metrics, we have selected euclidean distance (ED) and DTW as $DIST(.,.)$ function in eq. (19) for rotational invariant shape matching. We have used DTW as a distance metric for only PAA based feature vector representation of shapes as it can not handle feature space obtained using transformed global approximations. ED can work with both the original and transformed feature subspace and has been used as a distance metric for CS, DFT and PAA based representation. The experiment is repeated for different number of coefficients ($\#_{features}$)in CS, PAA and DFT

and for various values of σ. The results using ERA metrics are summarised in Fig. 5. The figure consists of four separate graphs for the standard deviation of $\sigma \in \{1.0, 2.0, 3.0, 4.0\}$. In all cases, varying the size of the feature vector is shown on the x-axis and the y-axis shows the average retrieval accuracies obtained using different dimensionality reduction schemes and distance measures. The experiment is repeated with the CRA metric and the results obtained are shown in Fig. 6.

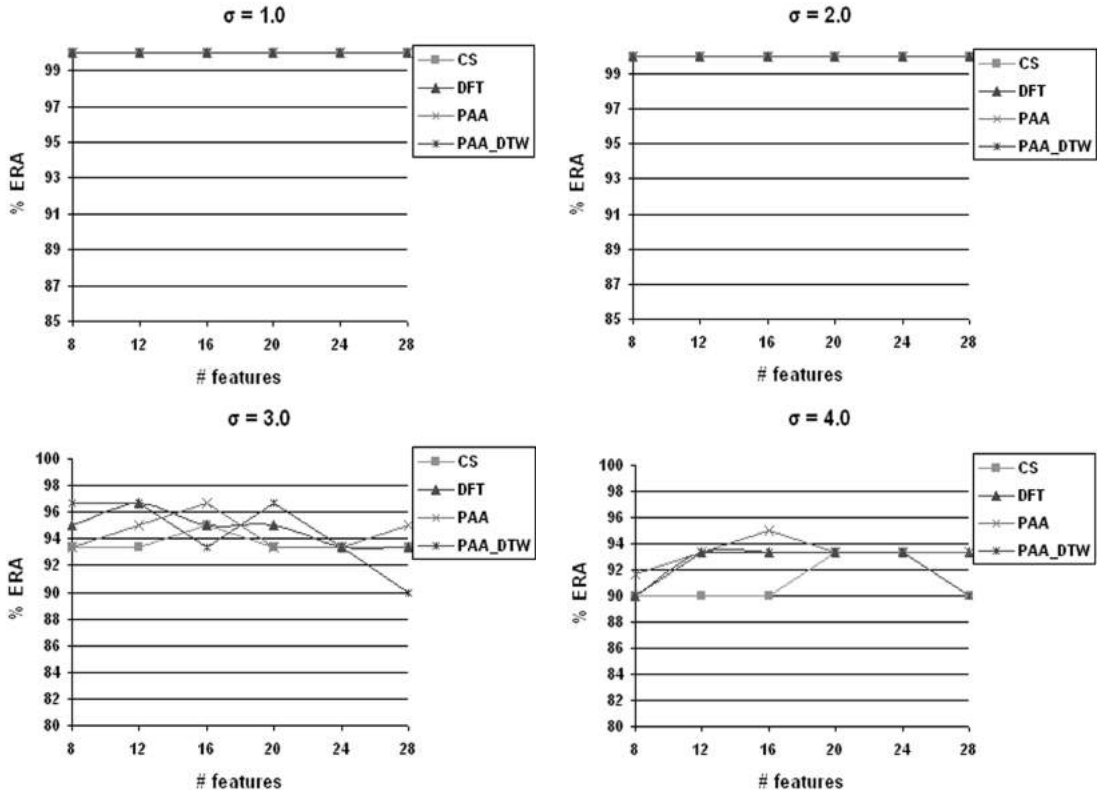


Fig. 9. Effect of different level of noise on exact retrieval accuracy (ERA) using different dimensionality reduction techniques and distance measures.

For small amounts of noise (i.e. $\sigma <= 2.0$), the choice of dimensionality reduction scheme or number of coefficients does not appear to be too critical for ERA-based evaluation. For higher noise levels (i.e. $2.0 < \sigma < 4.0$), it is apparent that PAA and DFT performs better than CS. For PAA, this is explained by the fact that PAA is an averaging scheme. As the noise induced is uniform on either side of the points on the contour, averaging over an interval has a cancelling effect of the noise. For DFT, good retrieval accuracy is explained by the property of distance preserving transform. CS does not give good retrieval accuracies and its performance degrades with increasing noise levels as compared to PAA and DFT. For CRA-based evaluation as presented in Fig. 6, DFT performs consistently better than other dimensionality reduction techniques followed by PAA . It can also be observed from Fig. 5 that PAA using ED gives better retrieval accuracy as compared to PAA

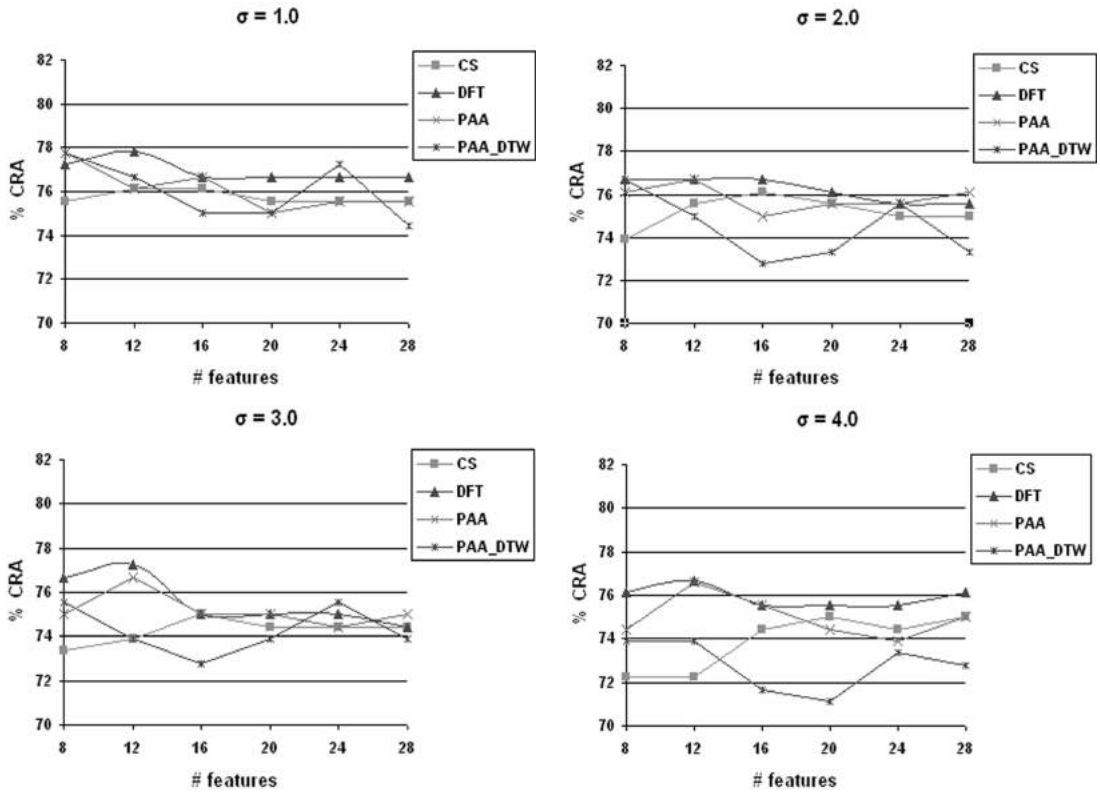


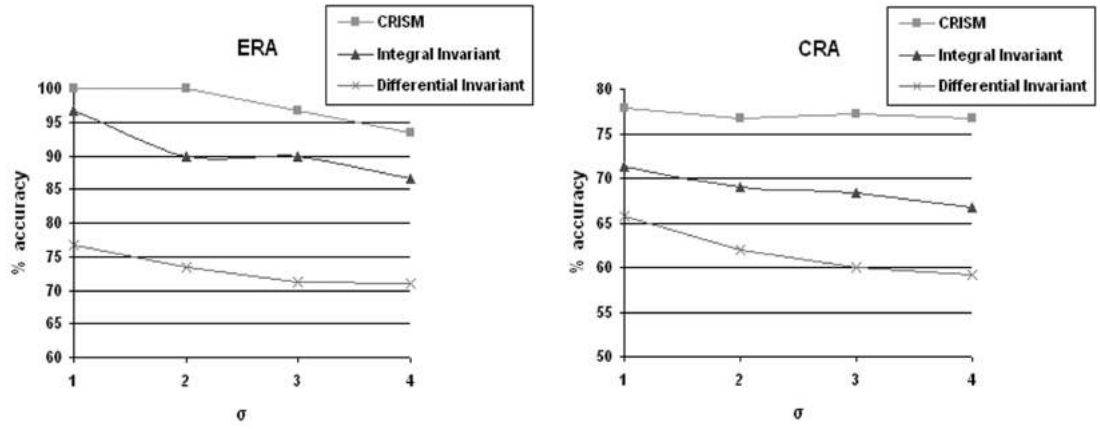
Fig. 10. Effect of different level of noise on class retrieval accuracy (CRA) using different dimensionality reduction techniques and distance measures.

using DTW (PAA_{DTW}). Although DTW is useful in situations where time series have local time shifting but this benefit comes at the cost of increasing sensitivity to noise. DTW caters for time shifting by duplicating elements to ensure different parts of the time series can correspond. This aspect makes DTW noise-sensitive as outliers may be duplicated as well resulting into the degradation of performance of DTW in the presence of noise.

Overall, the experiment demonstrates that DFT is a better choice for feature space representation of shape contours as compared to its competitive techniques. It is fairly robust to varying amounts of noise and other distortions as compared to PAA and CS specifically when evaluated with respect to class retrieval accuracy (CRA). A comparison of different distance measures in the presence of noise and other distortions, showed that ED in feature space is a better choice for similarity search than DTW. This justifies our selection of using DFT as a feature space representation and ED as a similarity metric in contour-based shape matching.

6.2 Comparison of CRISM with Competitive Techniques

The purpose of this experiment is to compare the performance of proposed CRISM-based approach for shape matching with competitive techniques. To establish a base case, we have implemented two different systems for comparison including integral invariants and differential invariants. The experiment has been conducted on silhouette dataset. For the proposed approach, shapes from the silhouette dataset are modeled using DFT-based coefficient feature vector. We assume $m = 5$ in eq. (8) and (9) based on the comparative evaluation in experiment 1. The implementation of integral-invariants for shape modeling and matching is based on the adaptation of local-area integral invariants [15]. The implementation of differnetial invariants is based on the adaptation of curvature invariants [16][17]. Comparative evaluation is provided in terms of ERA and CRA metrics. Average ERA and CRA based retrieval accuracies for varying amount of noise in silhouette dataset are presented in Fig. 8. The results from Fig. 8 shows that CRISM gives the highest retrieval accuracies in the presence of different noise levels followed by integral invariants. Differential invariants performs worst in the presence of noisy shapes and its performance deteriorates with increasing noise levels.



(a) (b)

Fig. 11. Effect of different level of simulated noise on shape retrieval accuracy using (a) ERA metric (b) CRA metric

To highlight the robustness of CRISM to noise and other distortions as compared to integral and differential invariants, the results of retrieving noisy shapes from a subset of silhouette dataset are presented in Fig. 8. Noisy query shapes are presented in the left column and the dataset is presented in the top row. The diagonal entries show the distance of noisy query shape with its corresponding non-noisy shape and the non-diagonal entries presents the distance of noisy shapes with other samples. The bold number in each row represents the best match for the query shape. Ideally, the diagonal entries should

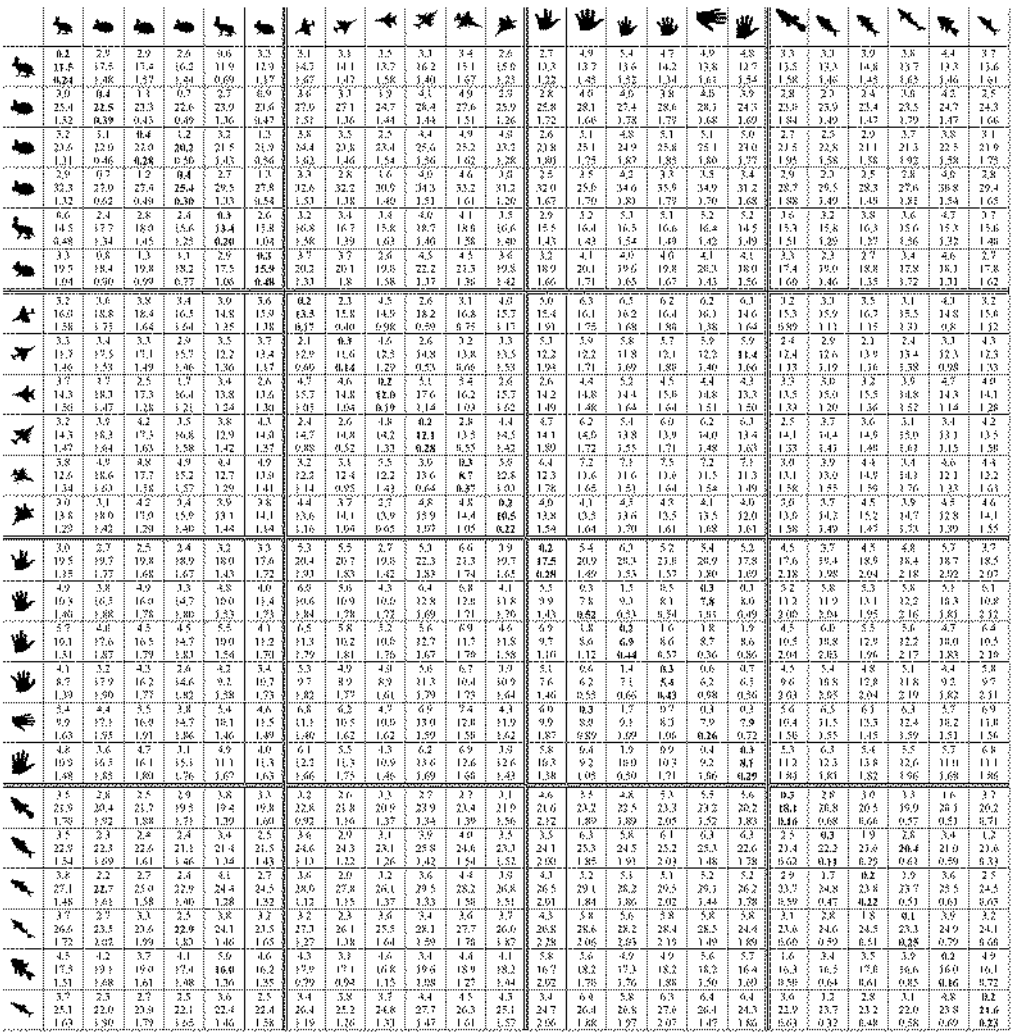


Fig. 12. Shape distances between noisy shapes (left column) and original shapes (top row). The bottom, center and top number in each cell is representing distances calculated using CRISM, differential invariant and integral invariant repectively.

represent the best match (minimum distance) for the noisy query sample. The distances between two shapes as computed using CRISM, differential invariants and integral invariants are represented by the bottom, central and top number respectively in each cell. It is obvious from Fig. 8 that the proposed approach for shape matching always retrieve the correct closest match for different noisy shapes. On the other hand, integral invariants does not give 100% ERA as it can not discriminate between different hand shapes. Differential invariants give the worst result due to its sensitivity to noise. The proposed shape matching approach and integral invariants also exhibits better CRA as compared to differential invariants.

Shape distances, as presented in Fig. 8, are now presented in graphical format in Fig. 9-11 for ease of visualization. The graphical representation of the per-

formance of CRISM, integral-invariant and differential-invariant are presented in Fig. 9, Fig. 10 and Fig. 11 respectively. Lower gray levels represent low distances and vice versa. Ideally, we want to have a a block diagonal structure with low gray levels along the diagonal. CRISM presents the best block diagonal structure followed by integral-invariant. Differential invariant presents high distances on the diagonals and does not present a block diagonal structure. Based on these results, we see that the proposed shape matching approach is more robust to the presence of noise in shapes as compared to competitive techniques. Fig. 12 presents the retrieval results for some of the query shapes to highlight the effectiveness of the proposed approach as compared to integral invariant and differential invariant. The results retrieved using CRISM, differential and integral invariant are presented in bottom, center and top row respectively corresponding to each query.

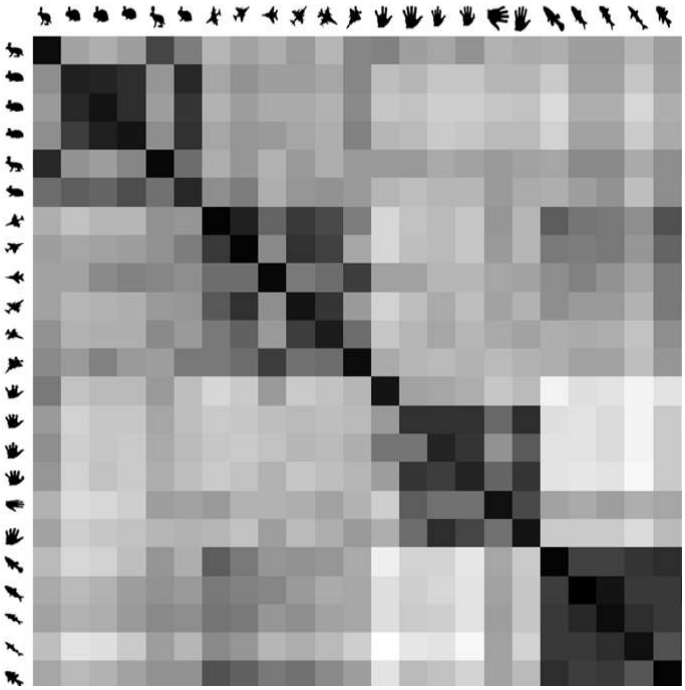


Fig. 13. Shape distances calculated using proposed approach. Higher gray levels (lighter shades) represent high distances and vice versa.

We now demonstrate the performance of CRISM using other publicly available datasets including Chicken, Diatom and MixedBag datasets. This will allow us to compare directly to variety of published work that have reported the performance of their shape matching approaches used identical experiment. Comparative evaluation is provided in terms of 1-NN classification accuracies obtained using leave-one-out cross-validation. Table 2 shows the classification accuracies of CRISM using different datasets.

Experiment on chicken dataset enable us to compare directly with to [36] and [2] who report the classification accuracy of 79.5% and 80.04% respectively. The shape matching approach specified in [36] takes around a minute

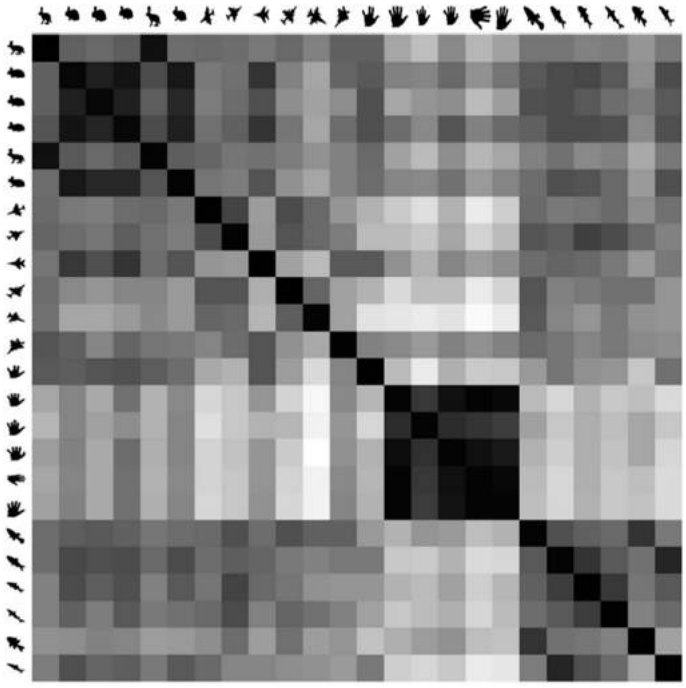


Fig. 14. Shape distances calculated using integral invariant approach. Higher gray levels (lighter shades) represent high distances and vice versa.

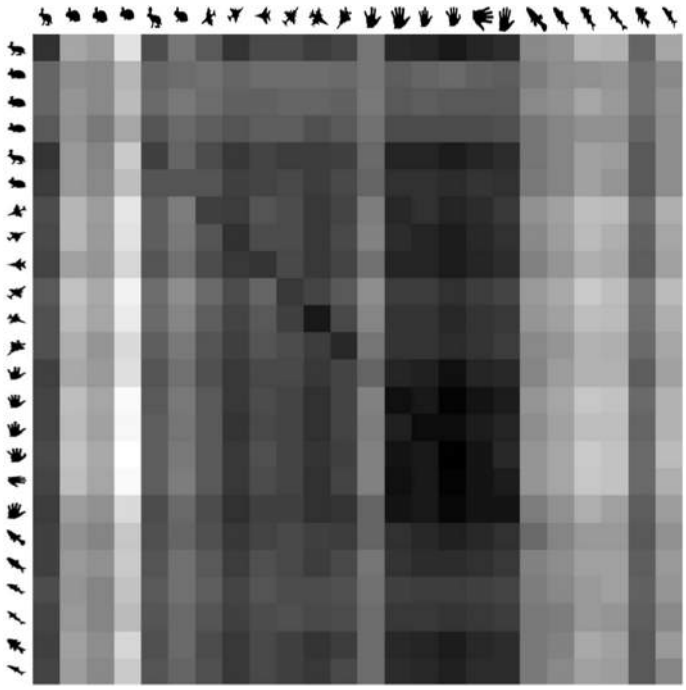


Fig. 15. Shape distances calculated using differential invariant approach. Higher gray levels (lighter shades) represent high distances and vice versa.

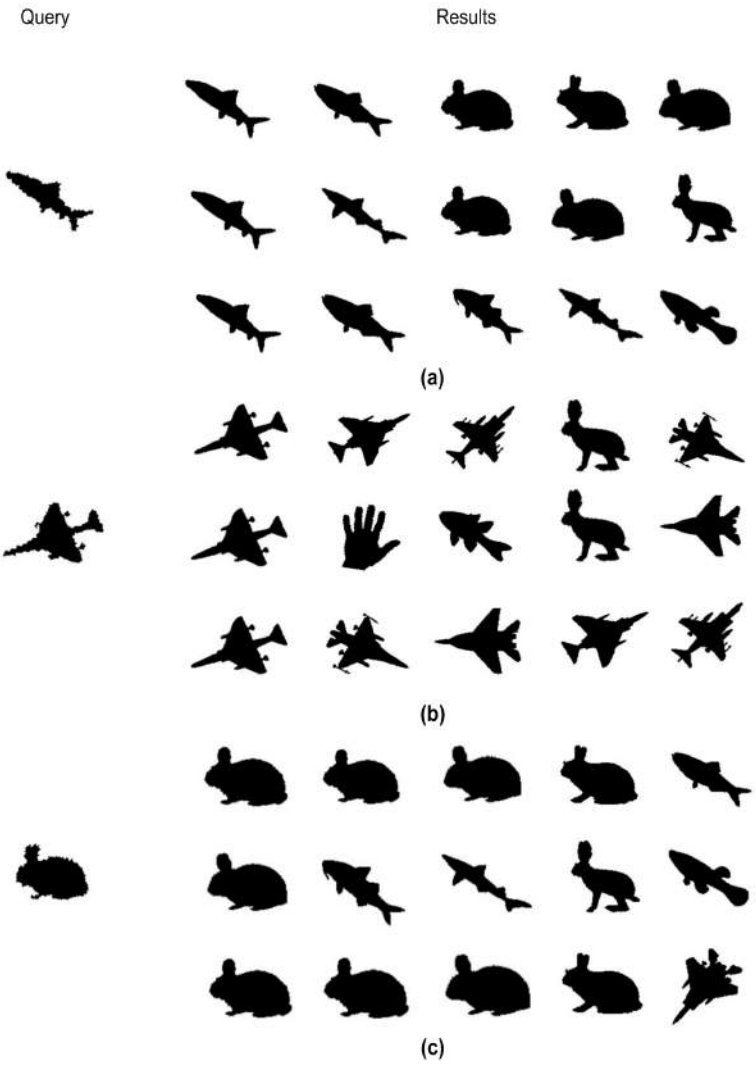


Fig. 16. Illustrative results for 5-NN queries on silhouette dataset (a) Query 1 (b) Query 2 (c) Query 3.

Dataset	**# of Instances**	**# of classes**	**% Classification Accuracy**
Diatom	781	37	72.73
Chicken	446	5	80.71
MixedBag	160	9	95.625

Table 2
Classification accuracies of CRISM using several publicly available shape datasets.

for one-one shape matching whereas [2] reported an average time of 0.0039 seconds. On the other hand, shape matching using CRISM takes and average time of 2.709×10^{-4} seconds. Similarly, [2] reported the classification accuracy of 72.47% for Diatom datasets whereas CRISM achieved the classification accuracy of 72.73% although CRISM-based shape matching is 2 orders of mag-

nitude faster than the shape matching approach proposed in [2].

7 Discussion and conclusions

In this paper, we have presented a detailed discussion on shape matching in the presence of noise and other distortions such as articulation and rotation. A CRISM algorithm has been proposed that exploits the contour information for shape matching. Contours are converted into normalized centroid distance based time series and is modeled using orthogonal basis coefficient feature space representation. We have compared the performance of different dimensionality reduction techniques. DFT has been selected as dimensionality reduction mechanism as it gives overall best results in real time situation where shapes are susceptible to high level of noise. Mapping shape contours from point sequences to DFT coefficient feature space improves retrieval efficiencies. A critical-point based approach to support efficient rotation-invariant shape matching is presented. The proposed algorithm is robust to affine transformations and other arbitrary distortions.

Experimental results are presented to show that CRISP-based shape matching gives better retrieval accuracies than competitive techniques such as differential invariants and integral invariants. Shape matching, using proposed approach, is robust to the presence of high levels of noise until the presence of very high level of noise results in total change in shape structure. The proposed approach demonstrates good discrimination capability than the competitors as reflected in the results presented in Figs. 7-11.

References

[1] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape Distributions. ACM Transactions on Graphics, vol. 21, no. 4, October, 2002, pp. 807-832.

[2] E. Keogh, L. Wei, X. Xi, S-H. Lee, M. Vlachos, LB-Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures, VLDB, 2006.

[3] T. Adamek, N. E. OConnor, A multiscale representation method for nonrigid shapes with a single closed contour, IEEE Circuits and Systems for Video Technology, vol. 14, no. 5, 2004, pp. 742-753.

[4] T. Adamek, N. E. OConnor, Efficient contour-based shape representation and matching, Multimedia Information Retrieval, 2003, pp. 138-143.

[5] E. Attalla, P. Siy, Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching, Pattern Recognition, vol. 38, no. 12, pp. 2229-2241, 2005.

[6] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast Sub-sequence Matching in Time-Series Databases, Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, 1994, pp. 419-429.

[7] R. Agarwal, C. Faloutsos, A. Swami, Efficient Similarity Search in Sequence Databases, 4th International Conference of Foundations of Data Organization and Algorithms, Evanston, Illinois, USA, October 1993, pp. 69-84.

[8] K. Chan, A. Fu, Efficient time series matching by wavelets, Proc. of International Conference on Data Engineering, Sydney, March 1999, pp. 126-133.

[9] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrota, Locally adaptive dimensionality reduction for indexing large time series databases, Proc. ACM SIGMOD Conference, 2001, pp. 151-162.

[10] Y. Cai, R. Ng, Indexing Spatio-Temporal Trajectories with Chebyshev Polynomials, ACM SIGMOD/PODS Conference, France, June 13-18, 2004, pp. 599-610.

[11] G. Mori, S. Belongie, J. Malik, Efficient Shape Matching using Shape Contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 11, pp. 1832-1837, November 2005.

[12] S. Belongie, J. Malik, J. Puzicha, Shape Matching and Object Recognition using Shape Contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 24, pp. 509-522, April 2002.

[13] S. Belongie, J. Malik, J. Puzicha, Matching shapes, Proceedings of Eighth IEEE International Conference on Computer Vision, Vol. I, Vancouver, Canada, pp. 454461, July 2001.

[14] E. R. Davies, Machine Vision: Theory, Algorithms, Practicalities, Academic Press, New York, pp. 171191, 1997.

[15] S. Manay, D. Cremers, B-W. Hong, A. J. Yezi, S. Soatto, Integral Invariants for Shape Matching, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 10, pp. 1602-1618, October 2006.

[16] A. M. Bruckstein, R. J. Holt, A. N. Netravali, T. J. Richardson, Invariant Signatures for Planar Shape Recognition under Partial Occlusion, Journal of Computer Vision, Graphics, and Image Processing, vol. 58, no. 1, pp. 49-65, 1993.

[17] M. Boutin, Numerically Invariant Signature Curves, Journal of Computer Vision, vol. 40, no. 3, pp. 235-248, 2000.

[18] D. Chetverikov, Y. Khenokh, Matching for shape defect detection, Lecture Notes in Computer Science, Vol. 1689, Springer, Berlin, pp. 367374, 1999. E.R. Davies, Machine Vision: Theory, Algorithms, E.R.

[19] S. Berretti, A.D. Bimbo, P. Pala, Retrieval by shape similarity with perceptual distance and e¡ective indexing, IEEE Transaction on Multimedia, vol. 2, no. 4, pp. 225239, 2000.

[20] G. Dudek, J.K. Tsotsos, Shape representation and recognition from multiscale curvature, Journal of Computer Vision and Image Understanding, vol. 68, no. 2, pp. 170189, 1997.

[21] D. Li, S. Simske, Shape Retrieval Based on Distance Ratio Distribution, HP Tech Report HPL-2002-251, 2002.

[22] A. Cardone, S. K. Gupta, M. Karnik, A survey of shape similarity assessment algorithms for product design and manufacturing applications, ASME Journal of Computing and Information Science in Engineering, vol. 3, no. 2, pp. 109-118, 2003.

[23] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape Distributions, ACM Transactions on Graphics, vol. 21, no. 4, pp. 807-832, October, 2002.

[24] T. Adamek, N. E. OConnor, A multiscale representation method for nonrigid shapes with a single closed contour, IEEE Circuits and Systems for Video Technology, vol. 14, no. 5, pp. 742-753, 2004.

[25] Z. Wang, Z. Chi, D. Feng, Q. Wang, Leaf Image Retrieval with Shape Features, In Proceedings of the 4th International Conference on Advances in Visual Information Systems, pp 477- 487, 2000.

[26] B. Bhanu, X. Zhou, Face recognition from face profile using dynamic time warping, In Proceedings of International Conference on Pattern Recognition, pp. 499-502, 2004.

[27] X. Wang, L. Ye, E. Keogh, C. Shelton. Annotating Historical Archives of Images, 8th ACM/IEEE-CS Joint Conference on Digital libraries, pp. 341-350, 2008.

[28] E. J. Keogh, M. J. Pazzani, A simple dimensionality reduction technique for fast similarity search in large time series databases, Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000, pp. 122-133.

[29] F. I. Bashir, A. A. Khokhar, D. Schonfield, A hybrid system for affine-invariant trajectory retrieval, In Proc. MIR'04, pp. 235-242, 2004.

[30] S. Khalid, A. Naftel, Evaluation of matching metrics for trajectory based indexing and retrieval of video clips, Proceedings of IEEE WACV, Colorado, USA, January 2005, pp. 242-249.

[31] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrota, Dimensionality reduction for fast similarity search in large time series databases, Journal of Knowledge and Information Systems, 2000.

[32] E. J. Keogh, Exact indexing of dynamic time warping, In Proc. Of 28th International Conference on Very Large Data Bases, pp. 406-417, 2002.

[33] S. Kim, S. park, W. Chu, An indexed based approach for similarity search supporting time warping in large sequence databases, In Proceedings of 17th International Conference on Data Engineering, pp. 607-614, 2001.

[34] E. J. Keogh, M. J. Pazzani, Scaling up dynamic time warping for data mining applications, In Proceddings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, USA., pp. 285-289, August 2000.

[35] http://www.lems.brown.edu/vision/researchAreas/SIID/silhouette-database.tar.gz

[36] R. A. Mollineda, E. Vidal, F. Casacuberta, Cyclic Sequence Alignments: Approximate Versus Optimal Techniques In International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), vol. 16, no. 3, pp. 291-299, 2002.