Fast Shape Matching and Retrieval Based on Approximate Dynamic Space Warping

Naif Alajlan

Advanced Lab for Intelligent Systems Research
Department of Computer Engineering
King Saud University, P.O.Box 51178, Riyadh 11543, Kingdom of Saudi Arabia
(najlan@ksu.edu.sa)

Abstract: Dynamic space warping (DSW) has emerged as a very effective tool for matching shapes. However, a central computational difficulty associated with DSW arises when a boundary's starting point (or rotation angle) is unknown. In this paper, the HopDSW algorithm is proposed to speed up the starting point computation. Rather than performing an exhaustive search for the correct starting point as in classical approaches, the proposed algorithm operates in a coarse-to-fine manner. The coarse search is global and uses a hopping step to exclude points from the search. Then, the search is refined in the neighborhood of the solution of the coarse search. A criterion that governs selecting the hopping step parameter is given, which reduces the number of starting point computations by an order. For shape representation, triangle area signature (TAS) is computed from triangles formed by the boundary points. Experimental results on the MPEG-7 CE-1 database of 1400 shapes show that the proposed algorithm returns the solution of the exhaustive search with a high degree of accuracy and a considerable reduction in the number of computations.

Keywords: - Shape matching, Shape retrieval, Approximate dynamic space warping, Dynamic programming.

I. INTRODUCTION

Shape is an important visual information that is traditionally used in many automated tasks such as industrial visual inspection [10], trademark registration [8], and content-based image description and retrieval [13]. Over the last decade, the adaption of dynamic time warping (DTW) for two-dimensional shape matching and retrieval has received considerable interest. This trend is mainly motivated by the high effectiveness of the DTW algorithm in achieving intuitive correspondences between the boundary points of two shapes being matched. Most (if not all) shape matching techniques reported in the literature that achieved the highest retrieval rates are based on Dynamic Programming (DP). However, the main limitation of DP-based shape matching techniques is the high computational complexity, which is the main focus of this work. Other issues related to shape matching such as robustness to noise and partial occlusion are beyond the scope of this paper.

Unlike the Euclidean and other vector-based distances, which provide a one-to-one alignment between points of two sequences, a nonlinear alignment can be achieved via the DTW algorithm, as shown in Fig. 1. Note that the Euclidean distance maps rigidly each boundary point in shape *A* to its counterpart in shape *B*. On the other hand, nonlinear alignment is possible using the DTW algorithm where one point in shape *A* can be matched to one or more points in shape *B*. Such alignment compensates for nonrigid deformations that occur in shape images in practice such as articulation and partial occlusion. The DTW algorithm finds the optimal (least cost) warp path that aligns the points of two boundaries. This warp path is constrained to be bounded (i.e., starts and ends by aligning the end points of the two sequences), monotonically increasing, and continuous. These constraints ensure that every point in the two boundaries is used in the warp path, a more intuitive alignment is obtained, and an efficient recursive implementation of the algorithm is possible.

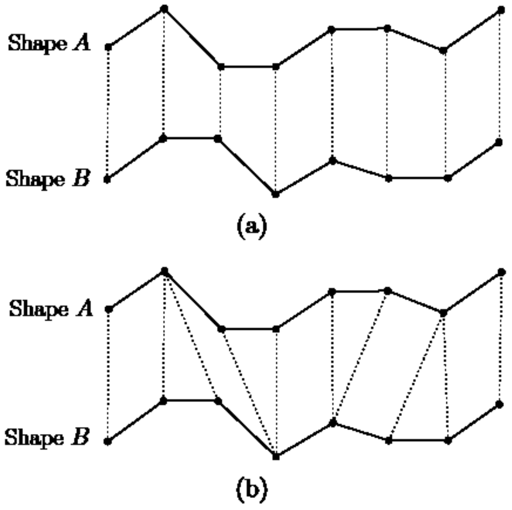


Fig. 1. An illustration of aligning the boundary points of two shapes *A* and *B* using (a) the Euclidean distance and (b) the DTW algorithm.

The basic DTW algorithm has quadratic time complexity since all pairwise distances between the points of two sequences have to be computed. For closed boundaries and rotation-dependent shape representations, where the starting point is unknown, the time complexity becomes cubic, since the basic DTW algorithm has to be executed for every possible starting point. Several approximations were proposed to reduce the computations as described in Section 2. In this paper, an approximate DSW algorithm, HopDSW, is proposed to speed up the basic DSW algorithm for planar shape matching and retrieval. Rather than searching all possible starting points for the optimal solution, only the points that most likely correspond to the optimal solution are searched in a coarse-to-fine fashion. The proposed approximate DSW algorithm works in two stages: coarse and refined. The coarse search for the minimum-cost starting point uses a constant-step hopping to avoid local minima solutions. Then, a refinement search in close proximity around the minimum-distance points returned by the coarse stage is performed. Our experiments show that selecting only the best point from the coarse search is sufficient to obtain a highly accurate solution. The proposed approach is applied to the triangle area representation (TAR) [3], which is sensitive to the starting point selection or shape rotation, as the shape signature. Experiments are conducted on the widely used MPEG-7 CE-1 shape database.

The rest of the paper is organized as follows. Section 2 reviews the related work in the literature. Then, Section 3 explains the proposed algorithm in more details. Experimental results are presented in Section 4 and, finally, Section 5 concludes the work of this paper.

II. RELATED WORK

The matching of one-dimensional sequences using dynamic programming was originated by the speech recognition community [5, 7, 18], where an optimal alignment between elements of two sequences is searched via DTW. Over the past decade, many researchers have applied DTW in the shape matching problem [1, 4, 12, 14]. This trend is mainly motivated by the desirable property of DTW in achieving nonrigid and more intuitive alignments. A particular computational difficulty occurs in matching shapes using DTW when the shape signature is sensitive to the starting point selection or the rotation angle. In [1, 3], a greedy search for the minimum-cost alignment is performed by executing the basic DTW algorithm for every possible starting point.

Several approximations were proposed from the speech recognition community to speed up the DTW algorithm. One of the most well-known approaches is the Sakoe-Chiba band [18], which imposes a constraint on the computation of the cost matrix to be limited within a fixed-width diagonal band. This band, besides reducing the computations, prevents matching farther points, which results in more meaningful alignments. For speech recognition, the Sakoe-Chiba band is set to approximately 10% of the sequence length. However, it is found that this choice is not suitable for the shape matching [1, 3] and the data mining applications [16], where the band width is set to less than 5%. This can be regarded to the fact that silence is allowed in speech, which requires longer warping.

Another class of approximate DTW methods is to reduce the size of the data representation itself and, thus, reducing the number of elements in the cost matrix. In [19], the DTW algorithm is applied to speech sequences at multiple sampling rates or resolutions, starting at the lowest resolution, and at each resolution level, the solution from the lower level is used to reduce the search space and obtain a more accurate solution.

Keogh et. al. proposed an iterative approach to the rotation invariance of the DTW algorithm [9]. Their method is based on discarding unnecessary computations of the DTW algorithm at starting points where the distance exceeds a predefined threshold. More specifically, lower and upper sequences are computed from a group of stored sequences, which results in an envelope or wedge. When a query sequence is presented, it is first compared to that wedge and the resulted distance is guaranteed to be less than or equal to the distance between the query and any member of the wedge's group; therefore, allowing early abandoning. However, when the area of the wedge is large, its distance to the query becomes very small and early abandoning is most likely not possible. To overcome this difficulty, the stored sequences, which may represent the same sequence at different starting points, are grouped based on their similarity to constitute multiple wedges. The authors reported impressive results where the main DTW is greatly sped-up. This method needs a parameter for optimal partitioning of the stored sequences in order to obtain thinner wedges.

III. THE PROPOSED APPROACH

In the following, we give a detailed description of the HopDSW, which is an algorithm that speeds up the basic DSW algorithm when the shape representation is not rotation invariant (or sensitive to the starting point selection). The proposed algorithm does not need parameter tuning and works for any rotation-variant shape representation that can be matched using the DSW algorithm. We first describe the triangle area signature (TAS) that is used for shape representation.

1. Triangle Area Signature (TAS)

The TAS is based on the area of the triangles formed by the points on the shape boundary, which has been used by many researchers as the basis for shape representations [6, 17]. In fact, the TAS is an abstracted version of the triangle area representation described in [3], which is considered as one of the most efficient boundary-based shape representations in the literature so far. The TAS is computed for an arbitrary closed boundary as described in the following.

Given a binary image containing a single shape, the boundary is extracted using the bug-following technique [15]. Then, each boundary point is represented by its x and y coordinates and separated parameterized boundary sequences x_n and y_n are obtained and re-sampled to N equidistant points. The curvature of the boundary point (x_n, y_n) is measured as:

$$TA(n,t_s)=\frac{1}{2}[x_n(y_{n+t_s}-y_{n-t_s})+x_{n+t_s}(y_{n-t_s}-y_n) \\ +x_{n-t_s}(y_n-y_{n+t_s})] \qquad (1)$$

where $TA(n,t_s)$ is the triangle area of point $n \in \langle 1, N \rangle$ at scale (or triangle side length) $t_s \in \langle 1, T_s \rangle$, (x_{n-t_s}, y_{n-t_s}), (x_n, y_n), and (x_{n+t_s}, y_{n+t_s}) are three consecutive boundary points. When the boundary is traversed in counter clock-wise direction, positive, negative and zero values of TA mean convex, concave and straight-line points, respectively. Fig. 2 demonstrates these three types of the triangle areas. The triangles at the edge points are formed by considering the periodicity of the closed boundary. Fig. 2 also shows the complete TA signature for the hammer shape when $t_s = 1$. By Increasing the length of the triangle sides, i.e., considering farther points, the function of eq. (1) represents longer variations along the boundary.

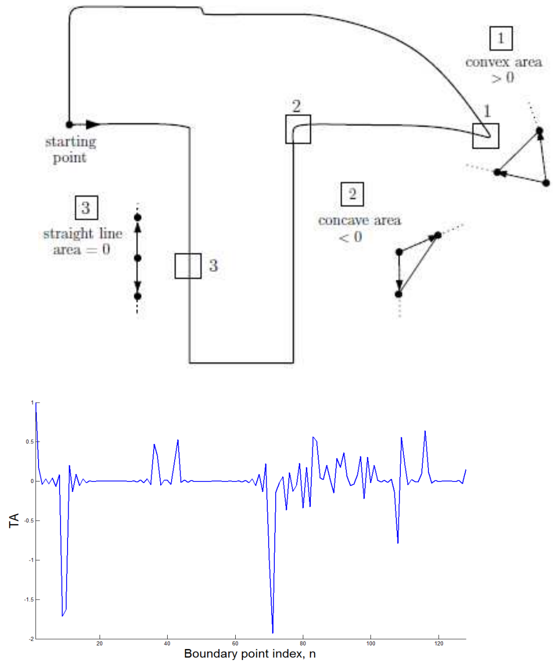


Fig. 2. Three different types of the triangle area values and the TA signature when $t_s = 1$ for the hammer shape.

For shape matching, there is a need to for normalizing eq. (1) to prevent the domination of large scales, which have large triangle areas. For this purpose, the signature is normalized locally at each scale or triangle side length:

$$TAN(n,t_s) = \frac{TA(n,t_s)}{\max\limits_{1 \le n \le N} TA(n,t_s)} \qquad (2)$$

Where $TAN(n,t_s)$ is a normalized version of $TA(n,t_s)$. Note that the triangle area exhibits an odd symmetry with respect to the triangle side length, where the center point is $N/2$ when N is even and does not exist when N is odd. Therefore, the number of scales (T_s) considered in this paper is equal to the floor value of $(N-1)/2$. Finally, the TAS is computed as:

$$TAS(n) = \frac{1}{T_s} \sum_{t_s=1}^{T_s} TAN(n,t_s) \qquad (3)$$

2. The HopDSW algorithm

The HopDSW algorithm is concerned with solving the rotation invariance or the starting point problem for boundary-based shape matching using the basic DSW. It should be noted that this algorithm works for any one-dimensional sequence, representing shapes, that is sensitive to the starting point selection. Instead of executing the basic DSW for all possible starting points, only points that most likely correspond to the minimum-cost solution are searched in a two-stage manner. In the following, a more detailed description of the proposed algorithm is given.

A pseudo code of the HopDSW algorithm is given in Algorithm 1. The algorithm accepts as inputs $TAS_A(n)$ and $TAS_B(n)$ of shapes A and B, where $n \in \langle 1, N \rangle$, and a hopping step $h_s \ge 1$. The hopping step controls the number of boundary points being skipped during the coarse search of the optimum starting point and the criterion for selecting h_s is discussed in the remainder of this section. In the coarse stage, a group of candidate starting points, with each consecutive pair is separated by $h_s - 1$ points, are executed using the basic DSW algorithm of Algorithm 2. Therefore, the number of these points equals $\lceil N / h_s \rceil$, where $\lceil\ \rceil$ is the ceiling function. The boundary point k that yields the minimum distance is passed to the fine stage where the search is resumed within the surrounding points up $h_s - 1$ points in each direction. Therefore, the number of starting points executed in this stage is $2(h_s - 1)$.

The basic DSW algorithm is described in the pseudo code of Algorithm 2. This algorithm starts by initializing a distance matrix, D_T, whose dimensions equal the lengths of the two sequences being matched. The Sakoe-Chiba band [18] of width w is used to restrict the warp path to remain within the w-width diagonal of D_T. Such restriction is useful in both reducing the number of computations and preventing the matching of farther points, which achieves more meaningful alignments. The distance d between two boundary points A_i and B_j is defined as:

$$d(A_i, B_j) = |TAS_A(i) - TAS_B(j)| \qquad (4)$$

The elements of the first row and column of D_T that lie within the w-width diagonal band are computed as the distances of the corresponding points. Then, the rest of the w-width diagonal elements are computed as:

$$D_T(i,j) = d(A_i, B_j) + \min \begin{cases} D_T(i-1, j) \\ D_T(i-1, j-1) \\ D_T(i, j-1) \end{cases} \qquad (5)$$

The distance between shapes A and B is taken as the value of $D_T(N,N)$, which corresponds to the least cost path between their TAS according to the selected starting point. To account for the shape flipping transformation, it is sufficient to flip one signature and repeat the algorithm. An illustrative example of the HopDSW search mechanism for the optimum starting point is shown in Fig. 3. The classic DSW includes all possible starting points in a greedy search as shown in panel (a) of the figure. Finding the optimum starting point, $n = 18$ in this example, is guaranteed; however, the basic DSW algorithm is executed 128 times. On the other hand, the coarse search of the HopDSW algorithm uses a hopping step $h_s = 8$ and returns $n = 17$ as the minimum distance point, as shown in panel (b). Then, the refinement stage, shown in panel (c), searches in points surrounding $n = 17$ and returns the optimal point, which is $n = 18$. Only 30 executions of the basic DSW algorithm are required by the HopDSW algorithm in this example. Our experiments have shown that the solution returned by this algorithm is the optimum with a high degree of accuracy. Besides, in many applications such as shape retrieval, an approximate solution is sufficient.

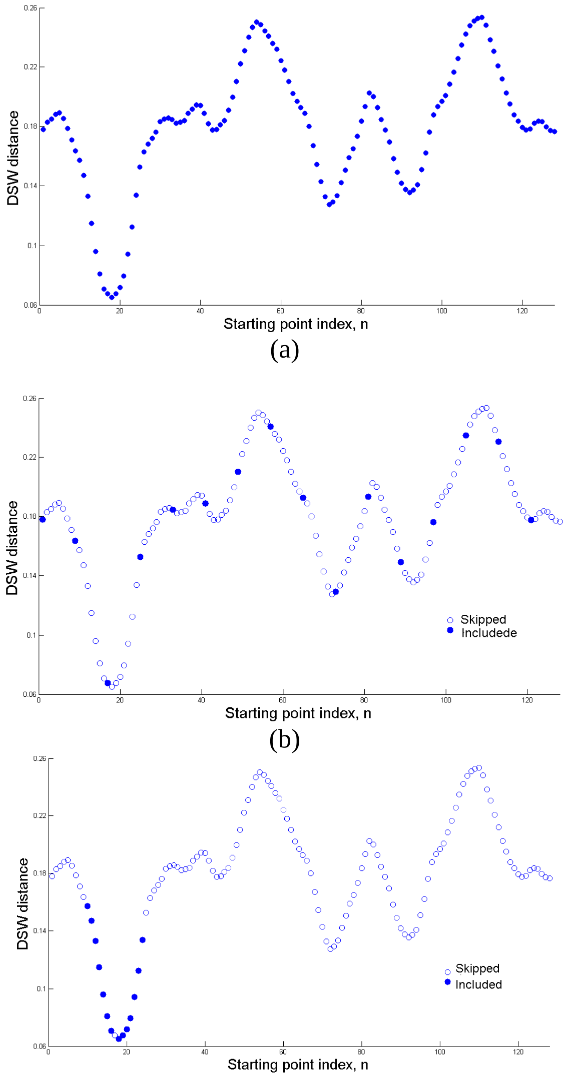
Algorithm 1: Approximate DSW (main algorithm):
dist = HopDSW(TAS$_A$, TAS$_B$, h$_s$)
Notation:
TAS$_A$ and *TAS$_B$* are TAS of shapes *A* and *B*, respectively.
N is the number of boundary points for each shape.
h_s is the hopping step.

```
1: $TAS_A \leftarrow [TAS_A\ TAS_A\ TAS_A]$ {to allow cyclic shifting}
2: $dist \leftarrow \infty$
{coarse search}
3: for $i = 1$ to $\lceil N/h_s \rceil$ do
4:     $c \leftarrow (i - 1) \times h_s + 1 + N$
5:     $d \leftarrow BasicDSW(TAS_A(c : c+N-1), TAS_B)$
6:     if $d < dist$ then
7:         $dist \leftarrow d$
8:         $k \leftarrow c$
9:     end if
10: end for
{fine search}
11: for $i = k - h_s + 1$ to $k + h_s - 1$ do
12:     $d \leftarrow BasicDSW(TAS_A(i : i + N-1), TAS_B)$
13:     if $d < dist$ then
14:         $dist \leftarrow d$
15:     end if
16: end for
17: Flip $TAS_A$ and repeat steps 3 to 16
18: return $dist$
```

Algorithm 2: *d = BasicDSW(TAS$_A$, TAS$_B$)*
Notation:
w is the width of the Sakoe-Chiba band. For N = 128, w is set to 3.
D$_T$ is an *N* × *N* distance matrix initialized as:

$$D_T(i,j) = \begin{cases} 0 & \max(1, i - w + 1) \le j \\ & \le \min(N, i + w - 1) \\ \infty & otherwise \end{cases}$$

```
1: for $i = 1$ to $w$ do
2:     $D_T(i,1) \leftarrow |TAS_A(i) - TAS_B(1)|$
3:     $D_T(1,i) \leftarrow |TAS_A(1) - TAS_B(i)|$
4: end for
5: for $i = 1$ to $N$ do
6:     for $j = max(1,i-w+1)$ to $min(N,i+w-1)$ do
7:         $p \leftarrow min[D_T(i-1, j)\ D_T(i-1, j-1)\ D_T(i, j-1)]$
8:         $D_T(i, j) \leftarrow |TAS_A(i) - TAS_B(j)| + p$
9:     end for
10: end for
11: return $d \leftarrow D_T(N,N)$
```



(a)

(b)

(c)

Fig. 3. The DSW distance versus the starting point of two MPEG-7 shapes returned by (a) the classic DSW algorithm, and (b) the coarse and (c) the fine stages of the HopDSW algorithm.

The selection of the hopping step parameter h_s is critical to the performance of the HopDSW algorithm. Let $F(h_s)$ represents the number of starting points executed by the HopDSW algorithm at a given h_s. Then, F is given by:

$$F(h_s) = \left\lceil \frac{N}{h_s} \right\rceil + 2(h_s - 1) \tag{6}$$

where $h_s \in \langle 1, \lfloor N/2 \rfloor - 1 \rangle$, and $\lceil\ \rceil$ and $\lfloor\ \rfloor$ are the ceiling and floor functions, respectively. As h_s increases, the number of starting points executed during the coarse search decreases and that of the fine search increases, represented by the second term of eq. (6). Note that the function of F is upper-bounded by N (when $h_s = 1$) and $h_s \geq \lfloor N/2 \rfloor$ is not considered since the value of F will exceed N in this case, which means unnecessary repetition of some starting points. Note also that the function of eq. (6) has a single minimum since the first and second terms are inversely monotonic. In this paper, h_s is chosen such that F is minimized. However, it is very difficult to analytically differentiate eq. (6) because the ceiling function is not continuous. Therefore, the minimum is computed numerically.

IV. RESULTS AND DISCUSSIONS

In this section, an empirical evaluation of our algorithm is presented. Two main tests were conducted. In the first, the effect of the hopping step h_s on the accuracy of the distance computation is investigated. The second test includes the shape retrieval application.

The well-known MPEG-7 CE-shape-1 database [11], which consists of 1400 images semantically classified into 70 classes with 20 shapes per class, is used here. This database contains a mixture of natural and artificial objects under various rigid and non-rigid deformations. In the following, the boundary of each shape is extracted and re-sampled to 128 equidistant points. Then, the TAS of each shape is computed according to eq. (3). To ensure availability, all datasets and algorithms used in this evaluation are placed online [2]. The codes are written using the Matlab © (version 7.0) software.

1. Accuracy of HopDSW

The aim of this experiment is to test the accuracy of the HopDSW algorithm. For this purpose, each shape is matched with all other shapes in the database using the HopDSW algorithm, which results in $K(K+1)/2$ distances, where K is the number of shapes in the database. These distances are compared with those obtained using the classic DSW algorithm. Fig. 4 shows the accuracy of the HopDSW algorithm as the number of shapes, K, increases. Note that in this case the number of distance computations ranges from about 5000 (for $K = 100$) to about a million (for $K = 1400$). In this figure, h_s is set to 8 according to eq. (6). Although the number of distance computations increased by more than two orders, there is no significant change in the accuracy, which suggests that the proposed algorithm has the ability to scale-up with the size of the shape database.

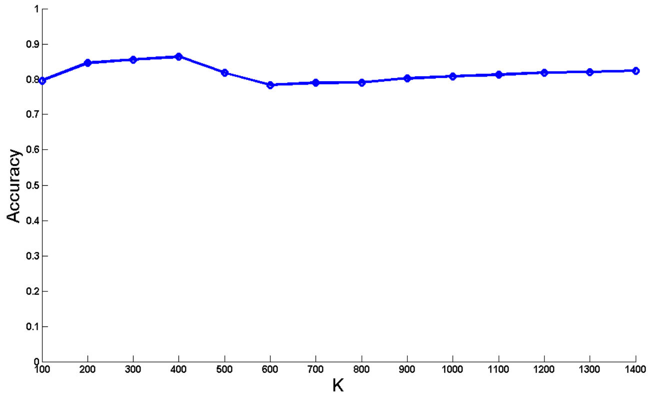


Fig. 4. The accuracy of the HopDSW algorithm, with h_s=8, versus the number of shapes, *K*, using the MPEG-7 CE-1 database. Note that there are *K(K+1)/2* distance computations at each point.

To illustrate the role of the hopping step parameter, Fig. 5 depicts the accuracy of the HopDSW algorithm at different values of h_s. Interestingly, the accuracy at h_s=8, which corresponds to the minimum number of starting point executions as given by eq. (6), is higher than other larger values of h_s, although these values include more starting points in the distance computation. The reason can be regarded to the fact that the additional starting points in the latter case come in favour of the local search for the optimum solution, which is represented by the second term in eq. (6). Whereas the former case more points are included in the global search represented by the first term of the same equation. This suggests that the choice of the parameter h_s that minimizes eq. (6) achieves a good balance for searching both globally and locally for the optimum solution.

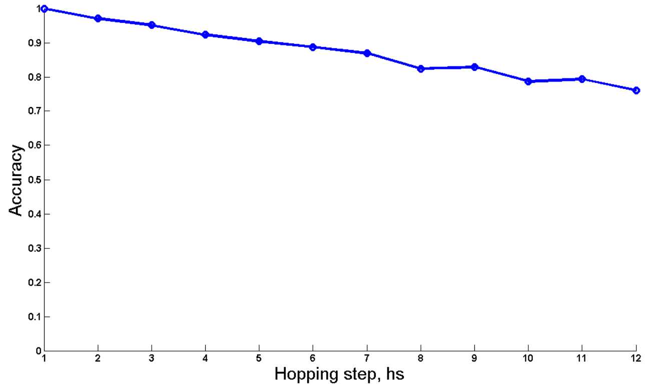


Fig. 5. The accuracy of the HopDSW algorithm versus the hopping step, h_s, using all 1400 shapes of the MPEG-7 CE-1 database.

2. Shape retrieval test

In shape retrieval, the aim is to acquire a ranked list of shapes according to their similarity with a query shape presented by a user. Therefore, obtaining an exact value of the distance is not a critical issue since approximate solutions are usually sufficient. It should also be emphasized that the work presented in this paper is not intended to enhance the effectiveness of shape retrieval, but rather, to reduce the number of starting point computations. However, shape retrieval results are reported here to demonstrate the trade-off between speed and accuracy of the proposed algorithm.

The MPEG-7 CE-1 part B test is conducted for both Algorithms. In this test, each of the 1400 shapes is considered as a query and the remaining shapes are ranked according to their similarity with the query. Then, the number of correct matches are counted among the first 40 retrieved shapes. The final precision is the average of all individual shape precisions. Table 1 shows the performance of the HopDSW algorithm at different values of h_s . It is obvious that the drop in the precision due to the hopping step is marginal. This suggests that the approximate distance returned by the HopDSW algorithm is either equal or very close to the exact distance returned by classic DSW algorithm ($h_s = 1$).

Table 1. The results of the MPEG-7 CE-shape-1 part B test using the HopDSW algorithm.

h_s	1	2	3	4	5	6
Precision	77.96	77.95	77.95	77.86	77.90	77.89
h_s	7	8	9	10	11	12
Precision	77.95	77.81	77.89	77.69	77.80	77.58

V. CONLCLUDING REMARKS

In this paper, a shape matching algorithm that approximates the DSW distance for arbitrary starting points is introduced. The algorithm performs a global search via hopping to locate a minimum-distance point followed by a local, refined search around the located point. The proposed algorithm achieves reduction by an order of the number of starting points as compared with the greedy search. Besides, experimental tests show that the proposed algorithm achieves a high degree of accuracy. This suggests that the basins of attraction for the minimum-cost starting point are quite large; therefore, there is no need to try every possible starting point. A desirable property of this algorithm is that it doesn't require any parameter setting by the user.

REFERENCES

[1] Adamek T and O'Connor N (2004), A multiscale representation method for nonrigid shapes with a single closed contour, IEEE Trans. on Circuits and Systems for Video Tech, 14(5):742–753.

[2] Alajlan N (2008), http://faculty.ksu.edu.sa/naifajlan/pages/activities.aspx.

[3] Alajlan N, Elrube I, Kamel MS, and Freeman H (2007), Shape retrieval using triangle-area representation and dynamic space warping, Pattern Recognition, 40:1911–1920.

[4] Bartolini I, Ciaccia P, and Patella M (2005), Warp: accurate retrieval of shapes using phase of fourier descriptors and time warping distance, IEEE Trans. Pattern Analysis and Machine Intelligence, 27(1):142–147.

[5] Deller J, Hansen J, and Proakis J (1999), Discrete-Time Processing of Speech Signals. Wiley-IEEE Press; Reprint edition.

[6] Ip HHS and Shen DG (1998), An affine-invariant active contour model (ai-snake) for model-based segmentation. Image and Vision Computing, 16(2):135–146.

[7] Itakura F (1975), Minimum prediction residual principle applied to speech recognition, IEEE Trans. Acoustics Speech Signal Process, 23:52-72.

[8] Jain AK and Vailaya A (1998), Shape-based retrieval: A case study with trademark image databases, Pattern Recognition, 31(9):1369–1390.

[9] Keogh E, Wei L, Xi X, Lee SH, and Vlachos M (2006), LB_Keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures, In Proceedings of the 32nd international conference on very large data bases, pages 882–893.

[10] Kunttu I and Lepisto L (2007), Shape-based retrieval of industrial surface defects using angular radius fourier descriptor, IET Image Processing, 1(2):231–236.

[11] Latecki LJ, Lakamper R, and Eckhardt U (2000), Shape descriptors for non-rigid shapes with a single closed contour, In IEEE Conf. on Computer Vision and Pattern Recognition, pages 424–429.

[12] Ling H and Jacobs D (2005), Using the inner distance for classification of articulated shapes, In IEEE International Conference on Computer Vision and Pattern Recognition, 2: 719–726.

[13] Pentland A, Picard R, and Sclaroff S (1996), Photobook: content-based manipulation of image databases, International Journal of Computer Vision, 18(3):233–254.

[14] Petrakis EGM, Diplaros A, and Milios E (2002), Matching and retrieval of distorted and occluded shapes using dynamic programming, IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(11):1501–1516.

[15] Pratt WK (1991), Digital Image Processing, John Wiley and sons Inc, second edition.

[16] Ratanamahatana C and Keogh E (2005), Three myths about dynamic time warping data mining, In International Conference on Data Mining, pages 506–510.

[17] Roh K and Kweon I (1998), 2-D object recognition using invariant contour descriptor and projective refinement, Pattern Recognition, 31(4):441–445.

[18] Sakoe H and Chiba S (1978), Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoustics, Speech, and Signal Processing, 26:43–49.

[19] Salvador S and Chan P (2004), FastDTW: Toward accurate dynamic time warping in linear time and space, In KDD Workshop on Mining Temporal and Sequential Data, pages 70–80.