Cyclic Sequence Comparison Using Dynamic Warping

Nafiz Arica

Department of Computer Engineering, Turkish Naval Academy
34942, Tuzla, Istanbul, Turkey
narica@dho.edu.tr

Abstract. In this study, we propose a new dynamic warping algorithm for cyclic sequence comparison, which approximate the optimal solution efficiently. The comparison of two sequences, whose starting points are known, is performed by finding the optimal correspondence between their elements, which minimize the distance. If the sequences are cyclic and their starting points are not known, the alignment computation must determine the amount of cyclic shift for the optimal solution. However, this process increases the complexity of the algorithm and may be cumbersome especially for large databases. Instead of finding the optimal solution, the proposed algorithm finds the approximate distance at once and decreases the time complexity substantially. The algorithm is tested in boundary based shape similarity problem. The experiments performed on MPEG-7 Shape database, show that the proposed method performs better than the classical cyclic string comparison methods in the literature and gives very similar results with the optimal solution.

1 Introduction

The similarity measurement of two patterns represented by sequences is used in various pattern recognition applications such as speech recognition and molecular biology. Given two or more sequences, comparison of them is a process by which one attempts to measure the extent to which they differ. The calculation of distance is performed by aligning one sequence with the other according to a cost function. The alignment defines a set of elementary operations that transforms one sequence into the other. Each individual operation is qualified by associating a cost. The alignment which minimizes the total cost of elementary operations determines the distance between the sequences [1], [2].

The comparison of sequences, which consist of items in finite length alphabets as in text processing, is called string matching. The elementary operations in string matching are substitution, deletion and insertion of elements. There are also other operations used in various applications. For instance, the sequences from infinite alphabets, such as speech signal obtained by sampling from continuous functions of time, are generally compared by using dynamic warping (DW). The elementary operations in DW, which differs from the string matching, are compression and expansion.

No matter what the elementary operations are used in alignment process, the similarity of two sequences is generally calculated by dynamic programming approach. Given two sequences $\boldsymbol{A}$ and $\boldsymbol{B}$ with lengths N and M respectively, the alignment is performed by constructing a minimum distance table of dimensions $(N \times M)$ with a time and space complexity of $O(NM)$.

Many patterns such as the closed contour of objects are cyclic in nature. In order to align two cyclic sequences optimally using dynamic programming, the starting elements of them are to be matched. In other words, the amount of cyclic shift of sequences must be included in the definition of optimal alignment. However, this process increases the time complexity of the algorithm so that it may not be practical to search for the optimal solution, especially for the database applications, which require large number of sequence comparisons. For this reason, instead of finding the exact distance, the suboptimal solutions which measures the distance between the sequences approximately may be more appropriate.

The approximate solutions to the problem of cyclic sequence comparison, has been investigated thoroughly in the literature when the sequences are from a finite alphabet [6], [7]. DW has also been widely and successfully applied to the comparison of sequences from infinite alphabets. In this study, our motivation is whether we can extend DW in approximate cyclic comparison. For this purpose, we develop a new dynamic warping algorithm, which finds the approximate distance between two cyclic sequences whose items are from infinite alphabets. The proposed algorithm combines the approaches in classical DW and approximate cyclic string matching algorithms. It finds the approximate solution with a time and space complexity of $O(NM)$. The performance of the proposed algorithm is tested on contour based shape similarity problem. The shape contours are represented using a set of sequences, and the distance between them, are approximately measured using the proposed algorithm. The experiments performed on MPEG-7 Shape Database show that the proposed algorithm outperforms the available approximate string matching algorithms and gives almost the same results with the algorithm which finds the exact distance by determining the starting elements.

The paper is organized as follows. The formal definition of cyclic sequence comparison and an overview of available methods in the literature are given in section 2. The proposed cyclic DW algorithm is described in section 3. The experiments on shape similarity are discussed in section 4. Finally, the last section concludes the paper and discusses future studies.

2 Cyclic Sequence Comparison Background

Given two cyclic sequences, the exact distance between them is calculated by aligning their elements optimally. This requires to find the starting elements of the sequences. For this reason, the alignment computation must determine the amount of cyclic shift in order to find the best match.

Mathematically speaking, let us denote two cyclic sequences as $\boldsymbol{A}$ and $\boldsymbol{B}$ with the elements A_i, $i=1,...,N$ and B_j, $j=1,...,M$ respectively. If we denote the shifted version of $\boldsymbol{A}$ as $\boldsymbol{A'}$, then;

$$A' \equiv A \Leftrightarrow A' = \sigma^k(A), \quad for \;\; 1 \le k \le N \tag{1}$$

where

$$\sigma^k(A) = A_{k+1}A_{k+2}...A_N A_1 A_2 ... A_k . \tag{2}$$

The cyclic distance, D_C between $\boldsymbol{A}$ ve $\boldsymbol{B}$ is defined as

$$D_C(A,B) = \min\left\{D\left(\sigma^k(A), \sigma^l(B)\right), 1 \le k \le N, 1 \le l \le M\right\}. \tag{3}$$

The easiest method of solving cyclic sequence comparison problem is to shift any of the sequences one item at a time and recompute the alignment. The optimal alignment is then found by the cyclic shift which results with a minimum distance [1], [2]. This can be formulated as follows;

$$D_C(A,B) = \min_{1 \le l \le M} \left\{ D\left(A, \sigma^l(B)\right) \right\}. \tag{4}$$

The minimum value among the shifted distances is taken as the exact distance between the sequences. However, shifting the elements of any sequence at each time, makes the complexity of the algorithm $O(MN^2)$. There are also other studies for optimal solution to cyclic sequence comparison in the literature [3], [4], [5], [9]. A Divide and Conquer method is presented in [3] to efficiently compute the optimal cyclic alignment with a computational complexity *O(MNlogN)* in the worst case. Another algorithm is introduced in [4], which uses a channeling technique to reduce the complexity of each alignment and a shift elimination technique to reduce the number of alignments carried out. The computation complexity of this algorithm is also *O(MNlogN)*. In [5], the optimal solution is found by a guided search that discards candidate cyclic shifts as suboptimal on the basis of bounds on the corresponding alignment costs, which results in a data dependent computation complexity that varies between *O(MN)* and $O(MN^2)$.

Searching for strict optimality is not practical and efficient in the applications which require large amount of sequence comparison. Therefore, in practical problems, it is worth to find a suboptimal solution by approximate distance measures, rather that exact solution. The approximate techniques may serve as realistic alternatives to the optimal matching. The approximate solutions in the literature mainly focus on the cyclic string matching. These approaches double one of the sequences and then find the subsequence therein that best resembles the other sequence. A lower bound estimation of cyclic distance is computed in [6] by working on an edit graph which is defined by a quadratic set of nodes of *(M+1)* rows and *(2N+1)* columns and a set of arcs. In this method, the horizontal arcs correspond to insertions, diagonal arcs to substitution and vertical ones to deletions. This approach builds partial edit sequences between *A* and the doubled *B* (the concatenation of *B* with itself) and takes the minimum weighted sequences as its approximate value with a complexity of *O(MN)*. The extensions to this approach are proposed in [7]. Similarly, another approximate approach is developed in [8] for partial shape matching problem.

3 Cyclic Dynamic Warping

Let us start by briefly summarizing the classical dynamic warping algorithm, used for the sequences matching, when starting elements are known in advance. The DW algorithm compares two sequences whose elements are sampled from a continuous domain by finding an optimal match between their elements. The optimal match allows stretching and compression of the sequences.

In order to align two sequences, $\boldsymbol{A}=A_1,...,A_N$ and $\boldsymbol{B}=B_1,...,B_M$ using DW, we construct and *N-by-M* table, where each element *(i,j)* contains the distance between the points A_i and B_j . The goal is to find a path through the table, which minimizes the sum of the local distances of the points, starting from *(1,1)* and ending at *(N,M)*. This path is called warping path;

$$W = w_1, w_2, ..., w_K \tag{5}$$

and it is subject to several constraints;

- *Boundary Conditions* : This requires the warping path to start at $w_1=(1,1)$ and finish at $w_K=(N,M)$.
- *Continuity* : Given $w_k=(a,b)$, this constraint requires $w_{k-1}=(c,d)$, where

$$a - c \leq 1 \quad and \quad b - d \leq 1 \tag{6}$$

- *Monotonicity* : Given $w_k=(a,b)$ and $w_{k-1}=(c,d)$, this constraint insures that

$$a - c \geq 0 \quad and \quad b - d \geq 0 \tag{7}$$

The warping path on the DW table is found by dynamic programming algorithm, which accumulates the partial distances between the sequences. As we discuss in the previous section, if the sequences to be compared are cyclic, all the shifted versions of sequences must be considered in order to find the exact distance between the sequences.

In order to avoid the computational burden of matching all shifted versions of the sequences we propose the following method: Given two cyclic sequences $\boldsymbol{A}$ and $\boldsymbol{B}$, as the first step, the sequence $\boldsymbol{B}^2$ is built by concatenating $\boldsymbol{B}$ with itself, resulting in a doubled sequence. A cyclic minimum distance table with *N* rows and *2M* columns is then constructed as shown in Figure-1. In this table, there are more than one warping paths different from the classical DW. The warping paths start from the first *M* entries of the first row and end at various points of last *M* elements in the last row. The warping path with the minimum accumulated distance is selected as the solution to the problem. The goal of this algorithm is to find a subsequence of $\boldsymbol{B}^2$ with length $\boldsymbol{M}$, which is most similar to $\boldsymbol{A}$.

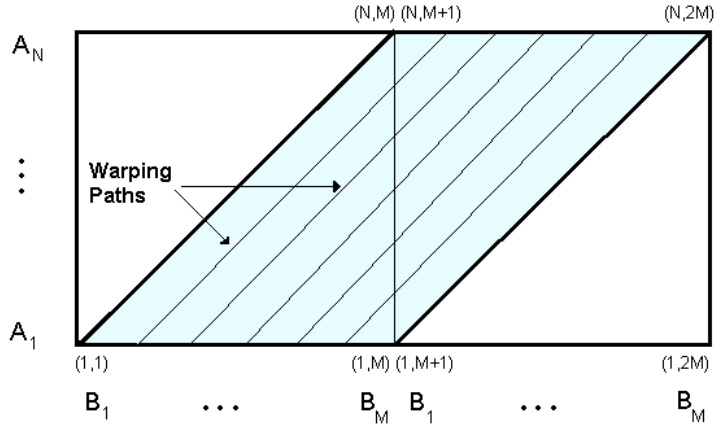

Fig. 1. Cyclic Minimum Distance Table

Let us define the entries of cyclic minimum distance table *D(i,j)* as the total distance from some point in the first row to the entry *(i,j)*. The value of *D(i,j)* is evaluated as;

$$D(i,j) = d(A_i, B_j) + \min\begin{cases} D(i-1, j-1) + P(i-1, j-1), \\ D(i-1, j) + P(i-1, j), \\ D(i, j-1) + P(i, j-1), \end{cases} \quad (\mathbf{8})$$

for i=2,...,N and j=2,...2M-1.
The boundary conditions are ;

$$D(1, j) = d(A_1, B_j) \quad (\mathbf{9})$$

for j=1,...,2M and

$$D(i,1) = d(A_i, B_1) + D(i-1,1) \quad (\mathbf{10})$$

for i=2,...,N .

In the above recurrence relation, the function *P( )* is called penalty function and used for controlling the warping paths throughout the table. The function takes the coordinates of the previous entry, which the warping path can move and returns a penalty value or zero.

Ideally, the warping path starting from k^{th} entry of the first row, is expected to end at the $(M+k)^{th}$ entry of the last row. In other words, the warping paths should proceed with the slope of *(N/M)*, that we call *ideal slope* of the paths in cyclic minimum distance table. The function penalizes the paths that move away from the ideal slope.

In order to make the calculations efficiently, first an ideal path table, *S* is constructed simply by drawing ideal paths from the first *M* entries of the first row to the corresponding entries of the last row. The entries on the same ideal path are numbered with the same value.

After the construction of ideal path table, the calculation of penalty function *P(k,l)* for the entry D(i,j) in the cyclic minimum distance table can be achieved by the following equation;

$$P(k,l) = \begin{cases} 0 & if \quad S(k,l) == S(i,j) \\ penalty & otherwise \end{cases} \tag{11}$$

Note that, if the warping paths are not controlled during the computation, the length of subsequence of ***B*** may tend to go far away from *M*, the length of ***B***. This leads to a partial matching of ***B*** against ***A***. As a matter of fact, this result is a lower bound of the cyclic distance between the sequences, as in the case of string matching [7]. For this purpose, the proposed algorithm enforces the paths to proceed with the ideal slope and penalizes the other paths that move away from the ideal slope.

Finally, the values at *D(N,j)* for *j= M+1,...,2M* contain the total distances of the paths through the table, from each starting point in ***B*** running from the first point to the last point of ***A***. The path with the lowest *D(N,j)* is the minimum distance path in the table. The total distance of this path is taken as cyclic distance between the sequences as follows;

$$D_C(A,B) = \min_{M+1 \le j \le 2M} \{D(N,j)\}. \tag{12}$$

Illustration of the cyclic DW is shown in figure-2. In the example, the distance between the sequences, *A=[0 4 3 5 3 2]* and *B=[5 2 2 6 1 0]* is calculated as *7*. For simplicity, the penalty function returns 0 for this particular example.

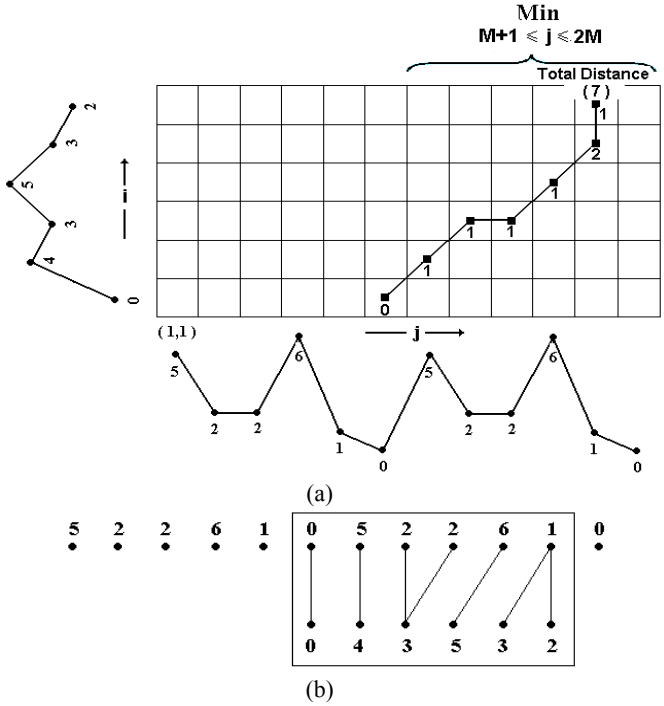

Fig 2. An example of cyclic DW (a) Warping Path, (b) Matching between the elements.

4 Experiments

The performance of the proposed algorithm is compared with both the exact distance algorithm and the other cyclic string matching algorithm in the literature. In the experiments, the contour based shape similarity problem is selected as the application area. MPEG-7 Shape Data Set is used as the test data. The shape boundaries are represented as cyclic sequences using Beam Angle Statistics (BAS) method, which is described in [1]. BAS represents a closed contour by a varying length cyclic sequence. Therefore, the output of BAS is a set of cyclic sequences representing the shape boundaries in a shape database.

The main part of MPEG-7 data is Part B. The total number of shapes in the database is 1400; 70 classes of various shapes, each class with 20 shapes. Each image is used as query and the number of similar shapes, which belong to the same class, is counted in the top 40 matches. Since the maximum number of correct matches for single query is 20, the total number of correct matches is 28000.

In the experiments, the exact cyclic distance is calculated by keeping one of the sequences fixed and shifting the other sequence, one item at a time. The classical DW is recomputed over and over again. The minimum distance is taken as the exact distance between the sequences [1]. The Bunke and Buhler algorithm proposed in [6], which is also considered as the basic approximate cyclic string matching algorithm, is another method used in the experiments. The proposed algorithm is also compared with the study proposed for partial shape matching [8]. The results of tests are depicted in table -1. In order to make the comparisons fair, the penalty values in all the algorithms are taken as 50. In the same way, the penalty function in our algorithm returns 50 for the arcs out of the ideal warping paths.

Table 1. Results of Cyclic Sequence Comparison Algorithms tested on Shape Similarity Problem (%).

Algorithm	**Length of Sequence**				
	10	**20**	**30**	**40**	**50**
Exact Distance	62.15	75.94	79.74	81.19	81.85
Cyclic DTW	**59.85**	**74.72**	**79.30**	**80.56**	**81.26**
Cyclic String Match	51.95	67.76	75.02	78.17	79.70
Partial Shape Matching	54.90	70.22	75.72	77.26	77.79

As it is shown in the table, the results of proposed algorithm are almost the same as the ones achieved in optimal solution which gives the exact distance. In addition, for this particular data set cyclic DW algorithm outperforms the other approximate cyclic string matching algorithms in the literature.

5 Conclusion

In this study, a DW algorithm is proposed for the cyclic sequence comparison. Determining the amount of cyclic shift for the optimal solution increases the computa-

tional complexity. The proposed algorithm decreases the time complexity significantly by abandoning the strict optimality. For this purpose, firstly, one of the sequences is concatenated with itself resulting in a doubled sequence. Then the subsequence on this doubled sequence, which is most similar to the other sequence, is found. The experiments performed on MPEG-7 Shape database show that the proposed algorithm gives satisfactory results, when it is used for contour based shape similarity problem.

Cyclic sequence comparison is one of the most important problems in shape based image retrieval applications. The nature of closed boundary represents a cyclic pattern. To ensure a consistent description of shapes, a unique starting point must, therefore be defined for each shape. Since this task is impractical to achieve, the alignment computation must determine the amount of cyclic shift. However, the computation of optimal solution to cyclic alignment increases the complexity of shape description, the complexity of whose representation is already high. For this purpose, we consider that the proposed algorithm provide significant contribution especially to the computation of shape similarities.

References

1. Arica N., Yarman-Vural F. T, BAS: A Perceptual Shape Descriptor Based On The Beam Angle Statistics, Pattern Recognition Letters, vol: 24/9-10, (2003) 1627-1639.
2. Sankoff D., Kruskal J., Time Warps, String Edits and Macromolecules, CLSI Publications, 1999.
3. Maes M., On A Cyclic String-To-String Correction Problem, Information Processing Letters, 35 (2), 73-78, 1990.
4. Gregor J., Thomason M. G., Efficient Dynamic Programming Alignment Of Cyclic Strings By Shift Elimination, Pattern Recognition, 29 (7), 1179-1185, 1996.
5. Gregor J., Thomason M. G., Dynamic Programming Alignment Of Sequences Representing Cyclic Patterns, IEEE Trans. Pattern Analysis and Machine Intelligence, 15 (2), 129-135, 1993.
6. Bunke H., Buhler U., Applications Of Approximate String Matching To 2-D Shape Recognition, Pattern Recognition, 26 (12), 1797-1812, 1993.
7. Mollineda, R. A., Vidal E., Casacuberta F., Cyclic Sequence Alignments: Approximate Versus Optimal Techniques, International Journal Of Pattern Recognition and Artificial Intelligence, 16 (3), 291-299, 2002.
8. Gorman J. W., Mitchell O. R., Kuhl F, P., Partial Shape Recognition Using Dynamic Programming, IEEE Trans. Pattern Analysis and Machine Intelligence, 10 (2), 257-266, 1988.
9. Tell D., Carlsson S., Combining appearance and topology for wide baseline matching, in *Proc. 7th European Conference on Computer Vision* (P. Johansen, ed.), vol. 2350 of *Lecture Notes in Computer Science*, (Copenhagen, Denmark), pp. 68--72, Springer Verlag, Berlin, May 2002.