2010 International Conference on Pattern Recognition

TrajAlign: A Method for Precise Matching of 3-D Trajectories

Zeyar Aung Kelvin Sim Wee Siong Ng
Institute for Infocomm Research
*A*STAR (Agency for Science, Technology and Research)*
1 Fusionopolis Way, #21-01 Connexis (South Tower), Singapore 138632
Republic of Singapore
Email: zeyarag@gmail.com, {shsim, wsng}@i2r.a-star.edu.sg

***Abstract*—Matching two 3-D trajectories is an important task in a number of applications. The trajectory matching problem can be solved by aligning the two trajectories and taking the alignment score as their similarity measurement. In this paper, we propose a new method called "TrajAlign" (Trajectory Alignment). It aligns two trajectories by means of aligning their representative distance matrices. Experimental results show that our method is significantly more precise than the existing state-of-the-art methods. While the existing methods can provide correct answers in only up to 67% of the test cases, TrajAlign can offer correct results in 79% (i.e. 12% more) of the test cases, TrajAlign is also computationally inexpensive, and can be used practically for applications that demand efficiency.**

***Keywords*-3-D trajectory matching; trajectory alignment; distance matrix; AUSLAN dataset;**

I. Introduction

3-D Trajectory matching is an important task in a wide range of application areas such as geographic information system, video tracking, mobile computing, location-based services, and sign language recognition. Given two trajectories in 3-D space, we have to determine how similar they are. A number of methods such as [1], [2], [3], [4], [5] have been proposed to solve the 3-D trajectory matching problem.

However, all the existing methods still have limitations in terms of precision. For the clustering test on the widely used Australian Sign Language (AUSLAN) dataset [6], the best of the existing methods can provide correct results in only 67% of the cases. Thus, our objective is to develop a new 3-D trajectory matching method to offer a better precision.

We try to solve the trajectory matching problem by aligning the two trajectories and taking the alignment score a measurement of their similarity. To achieve this, we propose a new method called "TrajAlign" (**Traj**ectory **Alig**nment). The proposed method aligns two trajectories by means of aligning the distance matrices representing them. It employs two levels of dynamic programming followed by a refinement procedure. TrajAlign is adapted and extensively modified from the MatAlign algorithm by Z. Aung and K.L. Tan (2006) [7], which is used for comparing 3-D protein structures.

Although TrajAlign utilizes dynamic programming, it is significantly distinct from the classical dynamic time warping (DTW) approach [8] which is also based on dynamic programming. DTW is designed to handle 1-D sequence data or 2-D/3-D trajectory data with the same relative position and orientation. On the other hand, employing the powerful distance matrix-based alignment mechanism, TrajAlign can cope with 3-D trajectories with different relative positions and orientations.

The clustering test using the AUSLAN dataset reveals that TrajAlign can offer considerably more precise results than the existing state-of-the-art trajectory matching methods. While the existing methods can provide the correct answers in only up to 67% of the test cases, our method can offer the correct results in 79% (i.e. 12% more) of the test cases.

II. Related Works

A number of methods to match trajectory data have been previously proposed. Dynamic time warping (DTW) [8] is a classical approach to match a pair of 1-D trajectories or 2-D/3-D trajectories within the same spatial framework (position and orientation). Piecewise dynamic time warping (PDTW) [1] is an attempt to speedup DTW by using higher level presentations of trajectories.

Vlachos *et al.* proposed a method to match trajectories by finding the longest common sub-sequence (LCSS) in them [2]. Rodriguez *et al.* developed a curve similarity algorithm based on relative shape signatures [3]. Croitoru *et al.* put forward a trajectory matching technique using pose normalization [4]. Recently, Tsumoto and Hirano proposed a trajectory comparison method by segmenting trajectories and calculating dissimilarities of the segments [5].

We will compare the performances of the above methods with that of our proposed TrajAlign method in Section V.

III. Definitions

A. 3-D Trajectory

A 3-D trajectory is an ordered set of 3-D positional points that an object takes place at the defined adjacent time points. A trajectory P can be expressed as $\{p_1, p_2, \ldots, p_{|P|}\}$ where p_i is the positional point of the object at time point t_i ($t_1 < t_2 < \ldots < t_n$), and $|P|$ is the length of P (i.e. the number of positional/time points in P).

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.930

The position point p_i is the (x, y, z) coordinate of the object in 3-D space which is defined as:

$$p_i \equiv [p_i.x,\ p_i.y,\ p_i.z] \quad (\text{where } 1 \leq i \leq |P|)$$

B. Trajectory Alignment

Given two trajectories P and Q, the objective of an alignment is to find the matching point pairs $\{\langle p_{a_1}, q_{b_1}\rangle, \langle p_{a_2}, q_{b_2}\rangle, \ldots, \langle p_{a_N}, q_{b_N}\rangle\}$ in which an ordered set of selected points from P (denoted as $P_{Al} \equiv \{p_{a_1}, p_{a_2}, \ldots, p_{a_N}\}$ where $1 \leq a_1 < a_2 < \ldots < a_n \leq |P|$) *optimally* matches an ordered set of selected points from Q (denoted as $Q_{Al} \equiv \{q_{b_1}, q_{b_2}, \ldots, q_{b_N}\}$ where $1 \leq b_1 < b_2 < \ldots < b_n \leq |Q|$) when superimposed onto each other. Here, $N = |P_{Al}| = |Q_{Al}|$ is the length of the alignment.

Fitness is one of the important criteria to measure the optimality of an alignment. It is conventionally measured in terms of the root mean square deviation (RMSD) (denoted as Δ) of the superimposed points.

$$\Delta(P_{Al}, Q_{Al}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_{Al}[i] - (\mathbf{R}\cdot Q_{Al}[i] + \mathbf{T}))^2} \tag{1}$$

where $\mathbf{R}$ and $\mathbf{T}$ are the rotation and the translation matrices for superimposition respectively. The smaller the value of RMSD, the fitter the alignment is.

However, RMSD (fitness) alone cannot determine the optimality of a matching. The alignment length is another important criterion. If we have an alignment where the RMSD is very small but the alignment length is very short as well, such an alignment cannot be considered as optimal.

A number of objective functions are available to ensure the optimality of an alignment by balancing the RMSD and the length of the alignment [7] . Among the available objective functions, we choose a simple yet effective function called *alignment score* [9] (denoted as S) here.

$$S = \frac{3 \times N}{1 + \Delta} \tag{2}$$

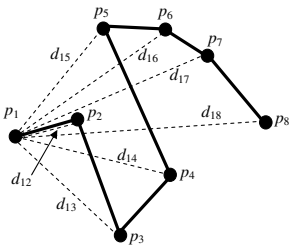
The larger the value of S, the better the alignment is.

C. Distance Matrix

A 3-D trajectory P with $|P|$ points can be represented as a $|P|\times|P|$ matrix M_P. Each cell $M_P[i, j]$ of the matrix is the Euclidean distance d_{ij} between the points p_i and p_j ($1 \leq i, j \leq |P|$) in that trajectory.

$$M_P[i, j] = d_{ij} = \sqrt{(p_i.x - p_j.x)^2 + (p_i.y - p_j.y)^2 + (p_i.z - p_j.z)^2} \tag{3}$$

The distance matrix is invariant of rotation and translation. That is, the two trajectories of the same shape are represented by the same distance matrix regardless of their different orientations and relative positions in 3-D space. An example of a trajectory and the distance matrix presenting it are demonstrated in Figure 1.



(a)

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
p_1	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}
p_2	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}	d_{27}	d_{28}
p_3	d_{31}	d_{32}	d_{33}	d_{34}	d_{35}	d_{36}	d_{37}	d_{38}
p_4	d_{41}	d_{42}	d_{43}	d_{44}	d_{45}	d_{46}	d_{47}	d_{48}
p_5	d_{51}	d_{52}	d_{53}	d_{54}	d_{55}	d_{56}	d_{57}	d_{58}
p_6	d_{61}	d_{62}	d_{63}	d_{64}	d_{65}	d_{66}	d_{67}	d_{68}
p_7	d_{71}	d_{72}	d_{73}	d_{74}	d_{75}	d_{76}	d_{77}	d_{78}
p_8	d_{81}	d_{82}	d_{83}	d_{84}	d_{85}	d_{86}	d_{87}	d_{88}

(b)

Figure 1. Examples of (a) A trajectory. (b) The trajectory's distance matrix.

IV. The TrajAlign Method

The trajectory matching problem can be effectively solved by aligning the two given trajectories. The major challenge for the alignment problem here is to find the matching point pairs in the two trajectories. Once the matching point pairs have been determined, the superimposition and the RMSD calculation can be easily done in linear time [10].

If we try to align (find the matching point pairs of) the two trajectories in their original 3-D space, we have to take care of their relative positions and orientations, and need to perform the relatively expensive operations of rotation and translations a number of times. Thus, our strategy is to convert the 3-D trajectories into 2-D distance matrices, which are independent of the original trajectories' positions and orientations, and align these matrices instead. To this end, we propose a method to align the matrices effectively and efficiently. We name our method as "TrajAlign" (Trajectory Alignment).

The basic steps of TrajAlign can be briefly described as follows:

- **Step 1.** Convert the two input trajectories P and Q into their respective distance matrices M_P and M_Q.
- **Step 2.** Align every row from M_P with every row from M_Q using Needleman and Wunsch's dynamic programming algorithm [11]. There will be a total of $|P|\times|Q|$ row-row matching (alignment) scores. Store these row-row matching scores in a $|P|\times|Q|$ matrix SM.
- **Step 3.** Perform Needleman and Wunsch's dynamic programming again on the score matrix SM, and extract the alignment path (matching pairs of points) as the initial alignment.
- **Step 4.** Superimpose the initially aligned trajectory Q_{Al} onto its counterpart P_{Al} by using the least-squares fitting method [10]. The superimposed Q_{Al} will be called Q'_{Al}.

- **Step 5.** Calculate RMSD (Δ) and alignment score (S) of P_{Al} and Q'_{Al}.

- **Step 6.** Remove the farthest point pair from the superimposed aligned trajectories P_{Al} and Q'_{Al} forming their new shorter versions. If the alignment length N becomes equal to the minimum length threshold, stop the process and report the current score S as the result.

- **Step 7.** Calculate the new RMSD (Δ_{new}) and the new alignment score (S_{new}) of the new P_{Al} and Q'_{Al}. If the new score S_{new} is greater than the old score S (i.e. there is still a room the improve the score), assign S_{new} as S and go back to Step 6. Otherwise, stop the process and report the old score S as the result.

In addition to the above basic steps, we incorporate the mechanisms of (1) alignment within a band, (2) weighting row-row matching scores, and (3) generating multiple initial alignment seeds — with a view to enhancing both the precision and the speed of TrajAlign.

As a matter of fact, TrajAlign is based on the MatAlign algorithm [7] which is used for aligning 3-D protein structures. However, we have extensively revised the original algorithm to suit our task of trajectory alignment. In particular, we have made substantial modifications in the row-row scoring scheme (in Step 2), the initial alignment extraction procedure (in Step 3), and the alignment refinement process (in Steps 6 and 7) in developing our new method.

V. Experimental Results and Discussions

In order to evaluate the relative performance of TrajAlign in an objective manner, we conduct the same experiment using the same test data as in the previous works [1], [2], [4], [5].

In achieve this, we use the Australian Sign Language (AUSLAN) dataset [6] from University of California, Irvine Knowledge Discovery in Databases Archive (UCI KDD). AUSLAN contains trajectory data on the 3-D (x, y, z) positions of the signers' hands when performing the sign language. Form the dataset, we obtain our test data as follows [4], [12].

The three signers: 'Adam2', 'Andrew2', and 'John2' are chosen. (For simplicity, they will be referred to as Signers 1, 2, and 3 respectively.) For each signer, the 10 signs: 'cold', 'crazy', 'eat', 'forget', 'happy', 'innocent', 'later', 'lose', 'Norway', and 'spend' are selected.

For each sign, the first 5 recordings are taken into account. For example, for the sign 'cold', its first 5 recordings ('cold0' ... 'cold4') are used. Thus, in total, we have $3{\times}10{\times}5 = 150$ recordings. An example of a recording of a sign is demonstrated in Figure 2.

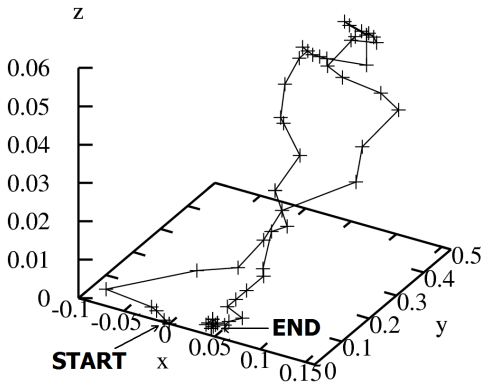


Figure 2. Example of a recording (trajectory) of the sign 'cold' in AUSLAN.

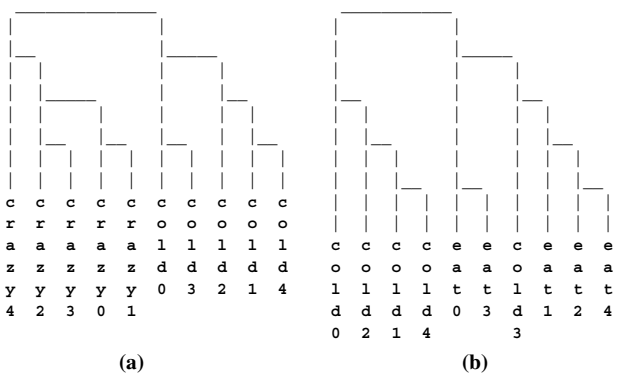


Figure 3. (a) Example of a correct clustering. The highest level divides the recordings into their correct clusters. (b) Example of an incorrect clustering. The highest level fails to divide the recordings into their proper clusters.

For each signer, we select a pair of signs at a time form the pool of 10 signs. In total, we have $\binom{10}{2} = 45$ distinct pairs of signs. For a particular pair of signs, we take the 5 recordings from each sign in the pair. Thus, we have a total of $2{\times}5 = 10$ recordings for each sign pair. Then, we perform an average distance-based hierarchical clustering [13] on these 10 recordings, and observe the resultant dendrogram.

The clustering is considered to be correct if the highest level of the dendrogram divides the recordings into two distinct clusters — each containing all the 5 recordings belonging to a single sign [1], [2], [4], [5]. (The dendrogram's lower level arrangements are deemed not very important as these can be quite subjective.) Figure 3 shows examples of correct and incorrect clusterings respectively.

It should be noted that to generate a correct clustering is not a trivial task. There are $34,459,425$ possible ways to cluster 10 recordings, and only $11,025$ of them correctly separate the two signs at the dendrogram's highest level. Thus, the chance to get a correct answer by random guessing is only 0.031% [1].

In Table I, we compare the clustering performances of the seven methods: dynamic time warping (DTW) [8] (as re-

Table I
COMPARISON OF CLUSTERING PERFORMANCES OF DIFFERENT METHODS.

Signer	Number of Correct Clusterings (out of 45)						
	DTW [8] (avg.)	PDTW [1] (avg.)	LCSS [2] (avg.)	RSS [3]	PN [4]	SB [5]	TrajAlign
1	22	23	21	15	23.0	nr*	35
2				12	20.4	26	36
3				15	23.9	34	36
Avg.	22	23	21	14	22.4	30†	35.7

*nr = not reported
†Average accuracy of the 10 selected signers reported in [5] is 30.6.

ported in [1]), piecewise dynamic time warping (PDTW) [1], longest common sub-sequence (LCSS) [2], relative shape signatures (RSS) [3] (as reported in [4]), pose normalization (PN) [4], segment-based (SB) [5], and our proposed TrajAlign.

We can observe that TrajAlign is clearly more precise than the existing methods. While the SB method [5], which is the best performer among the existing ones, can only provide correct clusterings in an average of $30/45 = 67\%$ of the cases, TrajAlign can offer correct clusterings in an average of $35.7/45 = 79\%$ (i.e. 12% more) of the cases!

Our proposed TrajAlign method is also computationally efficient. The average running time for all the 45 clustering processes for a signer is 138.67 seconds on a modest Pentium D 3.2GHz machine with 2GB main memory running Windows XP 32-bit Edition. Since each clustering process involves $10{\times}9/2 = 45$ trajectory matchings, the average running cost for a single trajectory matching task is merely $138.67/(45{\times}45) = 0.19$ seconds.

VI. CONCLUSION

In this paper, we have proposed a 3-D trajectory matching method called TrajAlign based on the alignment of trajectories' distance matrices. Experimental results show that our method is significantly more precise than the existing methods, and is also computationally efficient. As such, it can be practically used in many application areas: geographic information system, video tracking, mobile computing, location-based services, sign language recognition, etc. As a future work, we plan to develop an algorithm for multiple alignment of 3-D trajectories in addition to the current pairwise alignment.

REFERENCES

[1] E. Keogh and M. Pazzani, "Scaling up dynamic time warping for data mining applications," in *Proc. 6th ACM SIGKDD Intl Conf. on Knowledge Discovery and Data Mining (KDD'00)*, 2000, pp. 285–289.

[2] M. Vlachos, D. Gunopoulos, and G. Kollios, "Discovering similar multidimensional trajectories," in *Proc. 18th Intl Conf. on Data Engineering (ICDE'02)*, 2002, pp. 673–684.

[3] W. Rodriguez, M. Last, A. Kandel, and H. Bunke, "3-dimensional curve similarity using string matching," *Robotics & Autonomous Sys.*, vol. 49, pp. 165–172, 2004.

[4] A. Croitoru, P. Agouris, and A. Stefanidis, "3D trajectory matching by pose normalization," in *Proc. 13th Annl ACM Intl Wksp on Geographic Information Systems (GIS'05)*, 2005, pp. 153–162.

[5] S. Tsumoto and S. Hirano, "Multidimensional trajectory mining and its application to medicine," in *Proc. 2009 IEEE/ICME Intl Conf. on Complex Medical Engineering (CME'09)*, 2009, pp. 1–6.

[6] University of California, Irvine Knowledge Discovery in Databases Archive (UCI KDD). (1999) Australian Sign Language (AUSLAN) Dataset. [Online]. Available: http://kdd.ics.uci.edu/databases/auslan/auslan.data.html

[7] Z. Aung and K. L. Tan, "MatAlign: precise protein structure comparison by matrix alignment," *J. Bioinfo. & Comput. Biol.*, vol. 4, pp. 1197–1216, 2006.

[8] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, pp. 43–49, 1978.

[9] N. N. Alexandrov and D. Fischer, "Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures," *Proteins: Struct., Funct., and Genetics*, vol. 25, pp. 354–365, 1996.

[10] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp. 698–700, 1987.

[11] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, pp. 443–453, 1970.

[12] A. Croitoru, "Personal communications," 2009.

[13] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann Publishers, 2006.