Recognizing 3D Continuous Letter Trajectory Gesture Using Dynamic Time Warping

Jingren Tang, Hong Cheng(✉), and Lu Yang

Center for Robotics, School of Automation Engineering,
University of Electronic Science and Technology of China, Chengdu, China
hcheng@uestc.edu.cn

Abstract. Letter trajectory gesture recognition is widely used in Human Computer Interaction. Many approaches for letter trajectory gesture recognition have been proposed in the past several years. Most of the traditional approaches detect letters based on the beginning/end points provided by the user. It causes low writing speed and uncomfortable writing experience. Moreover, traditional Dynamic Time Warping cannot classify the letters which have the familiar trajectory. In this paper, we combine Dynamic Time Warping with structured points of letters to overcome those problems. The main contribution of this paper is that we introduce the structured points information of letters in Time Warping process to detect letters from hand trajectories. Based on this, we can successfully recognize the letter from the weak inter-class feature and the continuous trajectory without beginning point and end point given by the user. Furthermore, we can handle the self-contained trajectory based on the complexity of letters. We evaluate this system in our gesture dataset, and it shows that the proposed approach can significantly outperform the traditional begin-end gesture approach.

Keywords: Letter trajectory gestures · Dynamic time warping · Human computer interaction

1 Introduction

Vision-Based 3D gesture interaction approach has drawn much attention in recent years thanks to the emerging techniques of 3D sensors [1,2,13]. It is a natural and efficient way of human computer interaction (HCI). Moreover, it provides an attractive, user-friendly alternative that using an interface device (keyboard, mouse and other controller)[8] without physical contact.

The trajectory gesture is one kind of most important gestures. In this work, we use this information to build a system and detect letter from hand motion trajectory. The Microsoft Kinect is a 3D sensor which is wildly used now. Also we use it to get RGB data and depth data. The information of beginning point and end point in trajectory is very important to detect letter. In traditional approach, those points are marked by user, which cause writing speed slow and uncomfortable experience. Bhuyan *et al.* [8] proposed a novel continuous hand

Y.-S. Ho et al. (Eds.): PCM 2015, Part II, LNCS 9315, pp. 191–200, 2015.
DOI: 10.1007/978-3-319-24078-7_19

gesture recognition approach by using new features including writing speed. They assumed that the writing speed will slow down at beginning point and end point. According to this assumption, their system will not work if the writing speed of user remains constant. So, the same issue occurred as bad experience and poor efficiency. Furthermore, how to classify the letter from continuous trajectory is a difficult issue. We use normalized vector from different frame as the feature to classify letter at first. However, lots of vector features are similar. For example the vector feature of letter b and letter p are similar and it is hard to distinguish them with this approach.

In this paper, we propose a novel approach for 3D continuous letter trajectory gesture recognition without writing speed restriction and marked points. Series approaches have been proposed for gesture recognition such as Dynamic Time Warping (DTW) [9], Hidden Markov Model (HMM) [3], Finite State Machines (FSM) [10], *etc.* We use the improved Dynamic Time Warping to recognize letters from continuous trajectory for it's high accuracy and easy to be trained with few samples. Also, lots of improved DTW have been proposed such as Multidimensional dynamic time warping (MD-DTW) [4], memory efficient Dynamic Time Warping (MES-DTW) [12] *etc.* The main contribution of this paper is that we recognize letter from continuous trajectory by combining structured points with Dynamic Time Warping algorithm, which uses a natural way to recognize letter without low speed restriction. Furthermore, we handle the self-contained issue between letters which is based on the complexity of letters. This approach is evaluated on new data set and the experiment results show good performance.

The rest of the paper is organized as follows: Section 2 introduces the state of art of trajectory gesture recognition. Section 3 presents the detail of DTW with structured points approach and the solution of self-contained issue. Section 4 designs the experiment and shows the results. Section 5 gives the conclusions of this paper.

2 Related Work

The phases of continuous trajectory gesture recognition include hand location, hand tracking and extracting, classification. In this paper, we mainly concentrate on classification.

The approaches to improve the performance of classification can be carried out in two ways. The first is to improve the classifier. Kristensson *et al.* proposed a approach by using probabilistic algorithm to incrementally predict users intended gestures [11]. Though it has high accuracy, they use a zoning technique which means they detect the distance between user and Kinect. Once the distance below a threshold they define the input zone. The beginning point can be detected easily by the input zone. And they use two hands to select the gesture from some similar results. The approach mentioned above restrict the users hands in the input zone and user has to select gesture from results, for those reasons the writing speed is limited and the user gets uncomfortable experience. Cheng *et al.* proposed Windowed Dynamic Time Warping (WDTW) to classify

trajectory gesture [6]. They clustered general gestures into a set of strokes then use the parameterized searching window to recognize the gesture. However, the length of the window cannot be find by a certain process or formula. Lichtenauer *et al.* propose a approach to recognize sign language by combining statistical DTW and independent classification [7], they separated the time warping and classification to satisfy conflicting similar modeling demands, by doing so, the features which without distinction can be abandoned to simplify calculation and enhance robustness.

The second way is to improve features. Bhuyan *et al.* proposed a novel set of features for continuous hand gesture recognition [8], they use the velocity of hand motion trajectory as the new feature and use it to detect begin-end point. This feature also works for distinguishing intentional movements from unintentional movements. Though it is an effective feature to classify trajectory, it assumed that the velocity of hand motion would be decrease when the user beginning and finish writing. So the restriction of velocity slows down the writing speed.

3 The Proposed Continuous Letter Trajectory Recognition System

We use the 3D camera (Microsoft Kinect) to locate the hand center point, then we get the hand motion trajectory. The system support user write in air and then give the output. Once the trajectory have been gained, we use the motion vector from different frame as feature. The scale of letter is unfixed for the variant distance between user and camera, we use motion vector and then normalize it, the normalized vector is calculated as

$$\boldsymbol{n} = \frac{(x_t - x_{t-1}, y_t - y_{t-1})}{\|(x_t - x_{t-1}, y_t - y_{t-1})\|_2}. \tag{1}$$

where x_t, y_t are the points in current frame and x_{t-1}, y_{t-1} are the points in last frame. Subtract x_{t-1}, y_{t-1} from x_t, y_t we can get the motion vector, and then $\boldsymbol{n}$ has been calculated by normalize the vector. In fact, $\boldsymbol{n}$ can be shown as

$$\boldsymbol{n} = (\cos\theta, \sin\theta). \tag{2}$$

So, the feature reflects angle between new hand point and last hand point as shown in Fig. 1(a) and the hand trajectory as shown in Fig. 1(b).

3.1 Traditional Dynamic Time Warping Algorithm

Dynamic Time Warping algorithm is wildly used as a matching algorithm for it is easy to be trained and high accurate. With those advantages, lots of improved Dynamic Time Warping approaches have been proposed, the traditional DTW and improved DTW will be introduced. And we will give more details about DTW with structured points in this section.

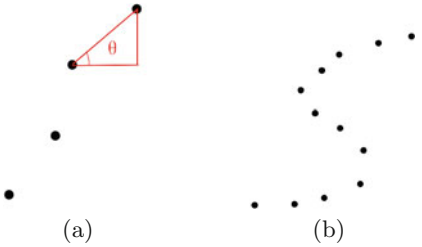

Fig. 1. An illustration of a letter trajectory gesture: (a) The reflection of a point angle; (b) Letter 's' trajectory gesture

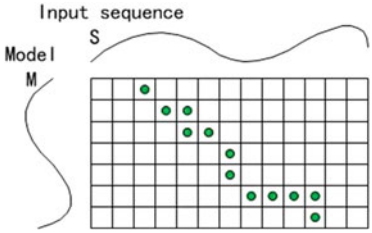

Fig. 2. The DTW algorithm

Assume that the trajectory model $M = \{m_1, m_2 ... m_n\}$ and the trajectory segment $S = \{s_1, s_2 ... s_m\}$ in long input stream, as shown in Fig. 2. The similarity of model vector and input vector should be calculated by similarity measure. We use the Euclidean distance to measure the similarity. Then we can get the similarity matrix $G_{n\times\infty}$. To find the optimized path the restrictions in DTW algorithm should be followed which means the next point $G(i, j)$ in path should be selected from neighbour points $G(i-1, j), G(i, j-1)$ or $G(i-1, j-1)$. This restriction simplify the algorithm and make it more reasonable, the final similarity is calculated by

$$\omega(P_{(i,j)}) = d(P_{(i,j)}) + \min(\omega(P_{(i-1,j-1)}), \omega(P_{(i-1,j)}), \omega(P_{(i,j-1)})). \quad (3)$$

Where $P_{(i,j)}$ is the location in similarity matrix $G_{n\times\infty}$, and $d(P_{(i,j)})$ is the Euclidean distance at (i, j) , and $\omega(P_{(i,j)})$ is accumulated Euclidean distance.

Once we detect the last row value which is smaller than threshold in the similarity matrix $G_{n\times\infty}$, the gesture segment in input stream match with the model trajectory. Thats means we can detect M from S while

$$\omega(P_{(n,j)}) < \alpha, \qquad j \in [0, \infty), \quad (4)$$

α is the threshold which is subject to different gestures, and it can be learned by using the leave-one-out cross validation strategy [9]. Now, we detect the same segment between model trajectory and input sequence, and this is the typical dynamic time warping algorithm.

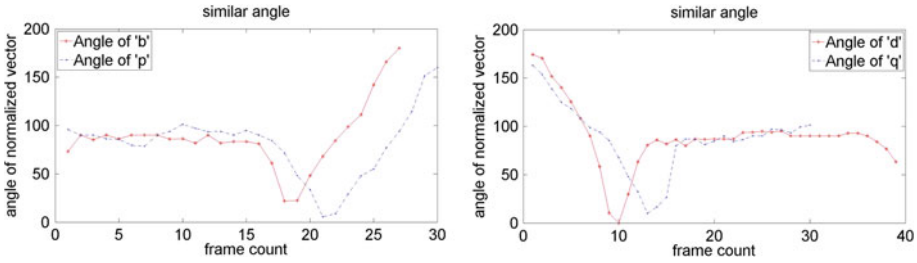

Fig. 3. Similar vectors

3.2 Dynamic Time Warping with Structured Points

In this subsection, we will give specific detail about the novel approach. As for features of trajectory, motion vector is the simple, visualized and efficient one. So, lots of approaches use this feature for it's advantages. To overcome the influence of different velocity, the motion vectors should be normalized. However, the issues occured while we use the normalized motion vectors. The motion vectors are similar of some letters which are hard to classify such as letter b and letter p, shown as in Fig. 3. To improve the performance of classification, we need find more information in DTW process. The DTW with structured points framework is shown as Fig. 4. The red points are beginning point and end point, the green points are turn points. All of them are points on the optimal path. The number of turn points is unconstant, it's depends on the structure of letter.

Next, the detail of how to find the structured points which include the beginning point, end point and turn point will be provided. We detect the similar trajectory while the final cost of optimal path below the threshold value, we find the end point which is the last point of path at the same time. To find the beginning point, we must record the direction of every point which means that we need record the next point of (i, j) is $(i + 1, j)$, $(i, j + 1)$ or $(i + 1, j + 1)$. After we detected the end point, we can find the beginning point by backtracking which use the direction data.

To find the turn point, we use the formula

$$\theta_t = \arccos(\delta_{x(t)}) \pm \arccos(\delta_{x(t-1)}) \tag{5}$$

to calculate the θ_t which is the angle between two vector. $\delta_{x(t)}$ is the normalized motion vector in x axis. To get the index of turn point in the cost matrix, every point of it should be calculated. θs of different letters are shown as Fig. 5. Then, we can detect the turn point only if

$$(\theta_i - \theta_{i-1}) * (\theta_i - \theta_{i+1}) \leq \tau, \tag{6}$$

where τ is threshold to detect the turn point. Now, we find the structured points of letters, the sample of them is shown as Fig. 6.

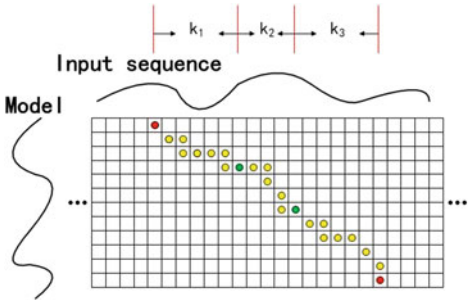

Fig. 4. The framework of DTW with structured points approach

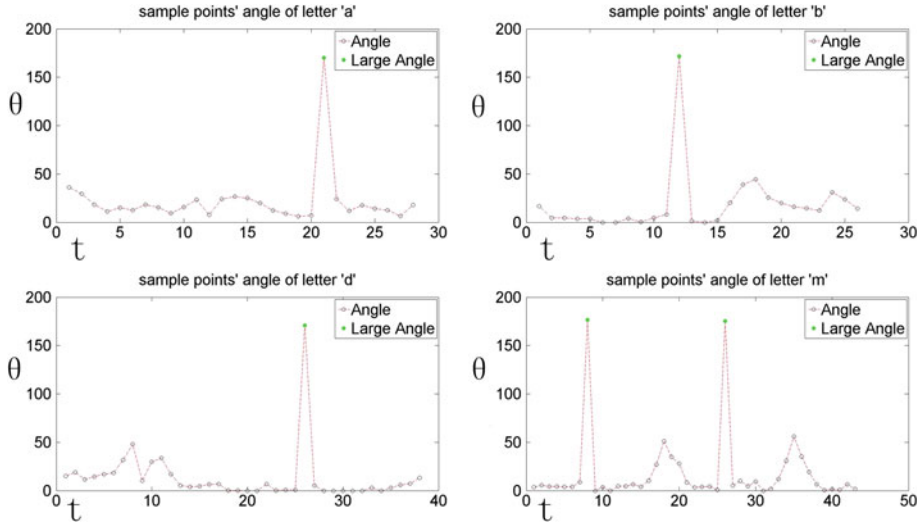

Fig. 5. Θs of different letters

As mentioned above, we can use the structured points to classify the trajectories which have the similar vector. The unit of cost matrix should include the distance data which can express as $(\delta_{x(t)}, \delta_{y(t)}; l(t))$. Define k_i as the distance structured points which is shown in Fig. 4. Once we detect the structured point, we calculate k_i using

$$k_i = (\sum_{t=l}^{m} \delta_{(x(t)\times l(t))}, \sum_{t=l}^{m} \delta_{(y(t)\times l(t))}). \tag{7}$$

l is the index of latest calculated structured point, m is the index of latest structured point, in this way we can reduce the calculation. Assume that the distant data of model is $k_i^{'}$, then we can calculate the similarity of them $||k_i^{'} - k_i||_2$. After getting the similarity, we change the cost value of optimal path according the similarity. Now we can classify the letters which have similar vectors. Actually,

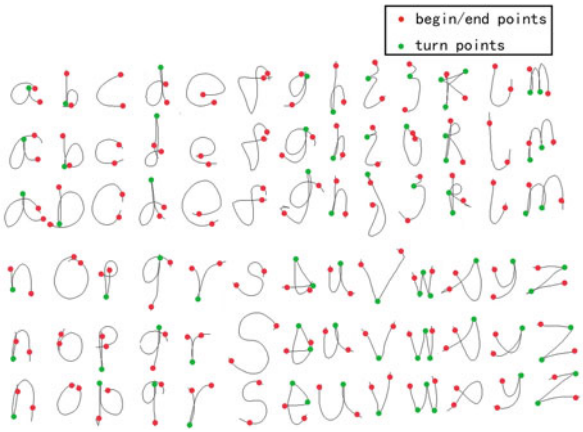

Fig. 6. Structured points of 26 letters

this approach use the relative location of points in trajectories to improve the performance of classifier.

3.3 Determine the Output Letter

Though we have handled the weak inter-class feature issue, there is another problem exist. Think about all letters, we will raise the question: the output is letter d or c while the input trajectory is d. Obviously, letter c contain with letter d, as shown in Fig. 7. The red is the common part between letter d and c.

Fig. 7. Self-Contained issue

One approach to handle this problem is that using the speed of movements to locate the beginning/end point and extract intentional movements [8]. However, this approach restrict writing speed and it does not work if all states of trajectory are same which including speed, location, acceleration, depth and so on. In this case we propose a rule-based approach which determine the output letter by letters' complexity for isolated letter detection. Actually, we find that the complex letters always contain with the simple letters, so, we will choose the more complex letter in output buffer as the system output. Moreover, we should discriminate whether two letters in the input sequence is contained or not. For example, the input buffer is c, d, c while user write d and c, the first c in the buffer is the contain part and another c is isolate letter which should be

output.So, we record the location of each point in the frame, then, the sample points location of letter in the buffer should be compared with each other, if one letter contain with another, they share the same location data in common part.

Finally, we find out whether the letter in the output buffer contain with each other, then we will choose the more complex letter as the output if one letter contain with another. We can know that the segment 1 have multi-outputs. Also, we can confirm that the optimal path of contained letters in similarity matrix are similar. So, whole process in Dynamic Time Warping as shown in Fig. 8.

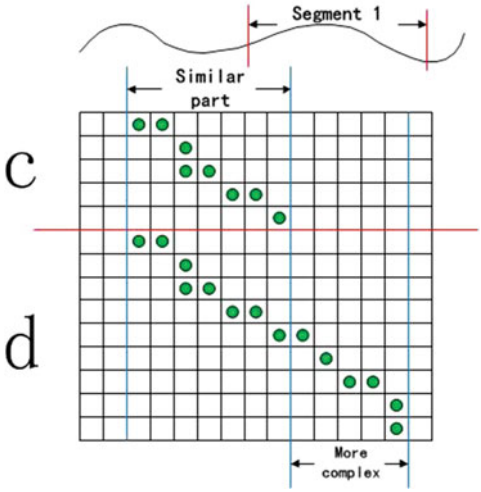

Fig. 8. An illustration of determining the output letter

4 Experimental Results and Analysis

Dataset: We have designed a new data set of 26 lowercase letters trajectory using Kinect devices. We record both RGB and depth clips, and get the hand trajectory points by NITE at the same time. Each letter is performed 10 times continuously by 5 volunteers. There are 1300 samples in total. Setup: we used 26 samples for training and the rest data for testing. The results are obtained by averaging 10 different trials to evaluate the performance of our approach. Assume that each letter is performed times and then we use correct detection rate as follow:

$$cRate = \frac{\sum_{i=1}^{m} C_i^2}{\sum_{i=1}^{m}(O_i + NC_i)}. \tag{8}$$

Where C_i means correct match and O_i is the number of all outputs, NC_i refer to the outputs without correct letter, in addition, the value of C_i and NC_i can

be only 1 or 0. Then we average of 10 different trials to obtain the final results. Results:

Table 1. Using DTW with motion vector only

Char	cRate	Char	cRate	Char	cRate	Char	cRate	Char	cRate
a	40.4 %	*g*	28.5 %	*m*	24.8 %	*s*	53.4 %	*y*	88.4 %
b	32.5 %	*h*	32.4 %	*n*	51.9 %	*t*	51.0 %	*z*	67.2 %
c	81.0 %	*i*	72.1 %	*o*	92.7 %	*u*	77.6 %		
d	45.5 %	*j*	50.4 %	*p*	30.7 %	*v*	83.4 %		
e	50.4 %	*k*	15.2 %	*q*	54.7 %	*w*	78.1 %		
f	64.8 %	*l*	93.6 %	*r*	75.3 %	*x*	69.5 %		

Table 2. Our approach

Char	cRate	Char	cRate	Char	cRate	Char	cRate	Char	cRate
a	83.5 %	*g*	61.4 %	*m*	53.8 %	*s*	83.9 %	*y*	90.9 %
b	81.3 %	*h*	87.4 %	*n*	90.4 %	*t*	78.5 %	*z*	66.5 %
c	81.6 %	*i*	84.5 %	*o*	93.4 %	*u*	78.5 %		
d	52.9 %	*j*	56.9 %	*p*	66.7 %	*v*	92.2 %		
e	86.8 %	*k*	53.8 %	*q*	63.8 %	*w*	91.7 %		
f	63.7 %	*l*	94.0 %	*r*	91.3 %	*x*	81.8 %		

We can conclude from the results, the performance of system which combining DTW with structured points of letters and have the self-contained solution is better than the system which using DTW and motion vector only. Note that the recognition rate of some letters in Table 1 is extremely low, because the letters contain with lots of other letters and the system cannot separate them. In addition, writing habits differ from person to person, thus causing some letters are hard to be recognized (Table 2).

5 Conclusions

One critical issue in continuous gesture recognition research is that how to find the effective approach to get the correct classification. And another issue is that the letter usually contain with each other. In this paper, we combine Dynamic Time Warping with structured points of letters to get the correct classification for 3D continuous hand trajectory gesture recognition. Moreover, we propose the novel approach to overcome the self-contained problem between letters which

use the complexity of letters. The evaluation shows that the approach improves performance compares with classical DTW.

Acknowledgment. This work was partially supported by NSFC (No.61305033, 61273256), Fundamental Research Funds for the Central Universities (ZYGX2013J088, ZYGX2014 Z009) and SRF for ROCS, SEM.

References

1. Chaudhary, A., Raheja, J.L., Das, K.: Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey. arXiv:1303.2292 (2013)
2. Kurakin, A., Zhang, Z., Liu, Z.: A real time system for dynamic hand gesture recognition with a depth sensor. In: EUSIPCO. IEEE (2012)
3. Gehrig, D., Kuehne, H., Woerner, A.: Hmm-based human motion recognition with optical flow data. In: HR. IEEE (2009)
4. Ten Holt, G.A., Reinders, M.J.T., Hendriks, E.A.: Multi-dimensional dynamic time warping for gesture recognition (2007)
5. Cheng, H., Zhongjun, D., Liu, Z.: Image-to-class dynamic time warping for 3D hand gesture recognition. In: ICME. IEEE (2013)
6. Cheng, H., Luo, J., Chen, X.: A windowed dynamic time warping approach for 3D continuous hand gesture recognition. In: ICME. IEEE (2014)
7. Lichtenauer, J.F., Hendriks, E., Reinders, M.J.T.: Sign language recognition by combining statistical DTW and independent classification. Pattern Anal. Mach. Intell. **30**(11), 2040–2046 (2008)
8. Bhuyan, M.K., Kumar, D.A., MacDorman, K.F.: A novel set of features for continuous hand gesture recognition. J. Multimodal User Interfaces **8**(4), 333–343 (2014)
9. Reyes, M., Dominguez, G., Escalera, S.: Featureweighting in dynamic timewarping for gesture recognition in depth data. In: ICCV Workshops. IEEE (2011)
10. Hong, P., Turk, M., Huang, T.S.: Gesture modeling and recognition using finite state machines. In: AFGR. IEEE (2000)
11. Kristensson, P.O., Nicholson, T., Quigley, A.: Continuous recognition of one-handed and two-handed gestures using 3D full-body motion tracking sensors. In: IUI. ACM (2012)
12. Anguera, X., Ferrarons, M.: Memory efficient subsequence DTW for query-by-example spoken term detection. In: ICME. IEEE (2013)
13. Ren, Z., Yuan, J., Meng, J.: Robust part-based hand gesture recognition using kinect sensor. Multimedia **15**(5), 1110–1120 (2013)