Air-Writing Recognition Using Reverse Time Ordered Stroke Context

Tsung-Hsien Tsai and Jun-Wei Hsieh

Department of Computer Science and Engineering, National Taiwan Ocean University
No.2, Beining Rd., Keelung 202, Taiwan, R. O. C.

ABSTRACT

A novel real-time recognition system is proposed to recognize finger air-writing characters without using any pen-starting-lift information. It presents a novel reverse time ordered stroke context to represent an air-writing trajectory in a backward way so that redundant starting-lift data can be effectively filtered out. Another two challenging problems often happen in the air-writing recognition system, *i.e.*, the multiplicity problem of writing and the confusion problem. The first one means a character is always written differently and the second one means different various characters own similar writing trajectory. To tackle them, a three-layer hierarchical structure to represent an air-writing character with different sampling rates is proposed. All the alphabets (including lowercase, capital, and digital letters) are recognized in this system. Performance evaluation shows that the proposed solution achieves quite higher recognition accuracy (more than 94.7%) even though no starting gesture is required.

Index Terms— Air-writing recognition, reverse time-order stroke representation, hierarchical classification

1. INTRODUCTION

Hand gestures can provide a more convenient way for home appliance control than a hand-held wireless controller. For example, simple "Up" and "Down" commands can be recognized to control the volumes or temperatures for appliances like TVs, music players, video players, or air conditioner systems. However, hand gestures themselves are simple and not expressive enough to input text for new or more complicated control applications. The 3D writing in the air will be a more useful way for human-computer interaction that allows users to type texts in a free space. Different from touch-screen handwriting, the in-air written character has no pen-starting-lift information, i.e., a character writing is always finished in one stroke without any starting and ending signals. The lack of a concrete anchoring or reference position for users to perform air-writing also makes motion data complex.

In the literature, there have been many frameworks proposed for gesture-based handwriting recognition. For example, in [1], Zhang et al. proposed a finger-writing-in-the-air system by combining skin, depth, and background information to segment and track fingertips and their positions via Kinect. In [2], Chiang et al. considered each writing trajectory as a one-stroke finger gesture and modified the DTW algorithm as a path finding algorithm to recognize all possible individual numerals. Moreover, Qu et al. [3] proposed a handwriting-based authentication system to allow users to write their passwords in 3D space via Kinect devices. Furthermore, Chu et al. [4]-[5] proposed a SVM-based hand-gesture recognition system to recognize numbers written in air via Kinect for home appliance control. Murata and Shin [6] used the dynamic programming technique to recognize characters written in air based on their intra-stroke information such as the XY position and the direction of each point. Schick, Morlock, and Amma [7] combined HMM and a 3D tracking technique to recognize characters handwritten in air. All the above methods need an initial gesture to start the writing action. In addition, they can recognize only a simple set of digital characters with a fixed writing style.

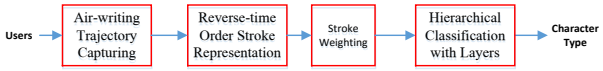


Fig. 1: Flowchart of the proposed air-writing recognition system.

Aiming at addressing the above challenges, this paper proposes a novel air-writing recognition system to recognize hand-written gestures without using any pen-starting-lift signal. Fig. 1 shows the flowchart of our proposed air-writing recognition system. To avoid using any pen-starting-lift signal, this paper proposes a novel reverse time-ordered shape context to represent an air-writing trajectory in a backward way. Then, with a weighting scheme, the path finding problem can be solved in real time via a dynamic time warping scheme. To tackle the multiplicity problem, one of common methods is to create multiple writing models in the dataset and thus also increase the matching complexity from $O(n)$ to $O(n \cdot m)$, where n is the number of character types and m is the number of used models. As to the confusion problem, it means different characters own similar writing trajectories. The two problems can be well tackled by introducing a new hierarchical classification scheme

which constructs a three-layer structure to represent an air-writing character with different sampling rates. The first layer is constructed for tackling the confusion problem in writing trajectories. The second and third layers are used for dealing with the multiplicity problem. The time complexities for each layer are $O(n)$, $O(n)$, and $O(m)$, respectively. Three major contributions of this work are:

1) A novel recognition system is proposed to recognize any air-hand-writing trajectories without using any pen-starting-lift information.
2) A new reverse time-order stroke descriptor is proposed and can effectively filter out unwanted redundant trajectory data.
3) A new hierarchical classification scheme is proposed to tackle both the multiplicity and confusion problems.

2. WRITING TRAJECTORY REPRESENTATION

2.1 Time-order Shape Context

Fig. 2: Detection results of 'true' turning points.

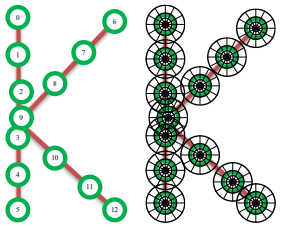
To represent an air-writing trajectory T, different turning points are first extracted by calculating their angles and choosing the minimum one. Fig. 2 shows the detection result of turning points. Then, given T, we can adopt the shape context [8] to characterize its shape. From T and its reference point r, we construct a vector histogram $\boldsymbol{H}_r^T = (h_r(1), ..., h_r(k), \ldots)$, in which $h_r(k)$ is the number of trajectory points in the kth bin when r is considered as the origin. Then, given two histograms, $\boldsymbol{H}_p^T$ and $\boldsymbol{H}_q^T$, their distance can be measured by a Chi-square distance:

$$\xi(\boldsymbol{H}_p^T, \boldsymbol{H}_q^T) = \sum_{k=1}^{K} \frac{(\boldsymbol{H}_p^T(k) - \boldsymbol{H}_q^T(k))^2}{\boldsymbol{H}_p^T(k) + \boldsymbol{H}_q^T(k) + 1}. \quad (1)$$

Assume T contains M points, *i.e.*, $T = \{p_0, p_1, ..., p_t, ..., p_{M-1}\}$, where p_t is written later than p_{t-1}. From each p_t, we can construct its corresponding shape context $\boldsymbol{H}_{p_t}^T$. Then, a Full Time-order Context (FTC) can be constructed to describe T, *i.e.*,

$$FTC(T) = \left\{\boldsymbol{H}_{p_0}^T, ..., \boldsymbol{H}_{p_t}^T, ..., \boldsymbol{H}_{p_{M-1}}^T\right\}. \quad (2)$$

As shown in Fig. 3, for each tracked point p_t of 'K', its corresponding shape context $\boldsymbol{H}_{p_t}^K$ is constructed (see (b)). The disadvantage of $FTC(T)$ is time-consuming in matching because all points on T are used. The complexity of $FTC(T)$ can be reduced if $\boldsymbol{H}_{p_t}^T$ is extracted only from the strokes of T.



(a) (b)

Fig. 3: Full time-order shape contexts of 'K'. (a) Time-order air-written points of 'K'. (b) Full time-order shape contexts of 'K'.

Assume there are $(N_T + 1)$ turning points extracted from T, *i.e.*, $C_T = \{c_0, c_1, ..., c_t, ..., c_{N_T}\}$. It is noticed that $N_T << M$. Two adjacent turning points c_t and c_{t+1} in C_T can form the tth stroke S_T^t in T. Let S_T denote the set of strokes collected from T, *i.e.*, $S_T = \{S_T^0, ..., S_T^t, ..., S_T^{N_T-1}\}$. Each stroke S_T^t can generate two shape contexts from its starting point c_t and ending one c_{t+1}. Then, we can create a Stroke-based Time-order Context (STC) to describe T:

$$STC(T) = \left\{\mathbb{H}^{U_T^0}, \mathbb{H}^{U_T^1}, ..., \mathbb{H}^{U_T^t}, ..., \mathbb{H}^T\right\}, \quad (3)$$

where $U_T^t = \bigcup_{i=0}^{t} S_T^i$ and $\mathbb{H}^{U_T^t} = \left\{\boldsymbol{H}_{c_0}^{U_T^t}, ..., \boldsymbol{H}_{c_{t+1}}^{U_T^t}\right\}$. Fig. 4 shows an example of the STC descriptor to describe 'K'.

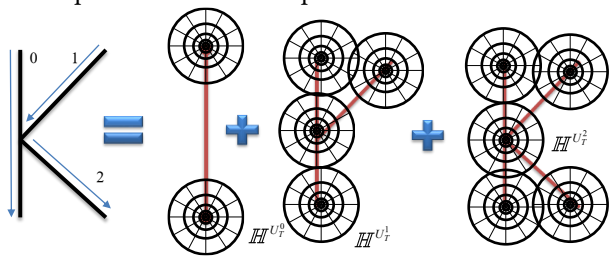


Fig. 4: Stroke-based time-order context (STC) for describing 'K'.

The STC descriptor can be further simplified if all its shape contexts are created only from the center of each stroke. Then, another new Central Stroke-based Time-order Context (CSTC) can be constructed for air-writing recognition. Then, the *CSTC* can be formed as follows:

$$CSTC(T) = \left\{\boldsymbol{H}^{U_T^0}, \boldsymbol{H}^{U_T^1}, ..., \boldsymbol{H}^{U_T^t}, ..., \boldsymbol{H}^T\right\}, \quad (4)$$

where $\boldsymbol{H}^{U_T^t}$ is the shape context with the center of U_T^t.

2.2 Reverse Time-Order Stroke Representation

In real conditions, a user often writes inputs from an unknown lifting position that results in some redundant strokes. For example, in Fig. 5, the first stroke in in Fig. 5 (formed by p_0 and p_1) is redundant for air-writing recognition. A novel reverse time order stroke representation is proposed in this paper to tackle the problem caused by redundant strokes. This representation stacks all turning

points and then pops them to get different strokes in a reverse time order.

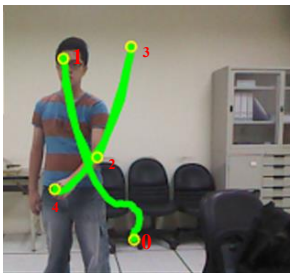


Fig. 5: Set of turning points extracted along a writing character '*y*'. All the turning points were indexed by time.

Let Ω_T denote the set of reverse time-order strokes of T and Ω_T^t be the tth version of Ω_T. Then, we have

$$\Omega_T = \bigcup_{t=0}^{N_T-1} S_T^t \text{ and } \Omega_T^t = \bigcup_{i=N_T-t-1}^{N_T-1} S_T^i .$$

With Ω_T^t, different evolved versions of T can be generated until a complete one. Fig. 6 shows an example of the reverse time order stroke representation of a '*y*' character (shown in Fig. 5). It is noticed that Ω_T^2 corresponds to the correct '*y*' character without including any redundant lifting strokes. Clearly, the correct version of the analyzed text will be evolutionally generated. In what follows, the sets $\{\Omega_T^t\}_{t=0,...,N_T-1}$ of reverse time-order stroke will be used to generate different trajectory contexts for recognizing T.

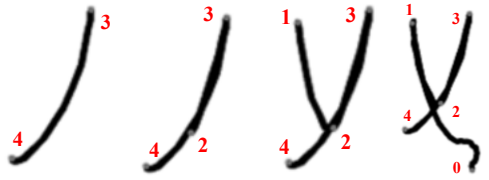


Fig. 6: Reverse time order stroke representation. (a): Ω_T^0. (b): Ω_T^1. (c): Ω_T^2. (d): Ω_T^3.

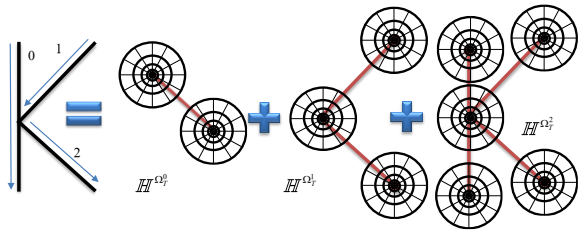


Fig. 7: Reverse stroke-based time-order context for 'K'.

The first reverse representation is similar to $FTO(T)$ (see Eq.(2)) but with a reverse time order. Then, a reverse Full Time-order Context (rFTC) can be created to describe T:

$$rFTC(T) = \left\{ \boldsymbol{H}_{p_{M-1}}^T, ..., \boldsymbol{H}_{p_t}^T, ..., \boldsymbol{H}_{p_0}^T \right\}. \quad (5)$$

If this reverse order is also adopted for constructing $STC(T)$, we can construct a reverse Stroke-based Time-order Context (*rSTC*) for air-writing recognition, *i.e.*,

$$rSTC(T) = \left\{ \mathcal{H}^{\Omega_T^0}, \mathcal{H}^{\Omega_T^1}, ..., \mathcal{H}^{\Omega_T^t}, ..., \mathcal{H}^T \right\}, \quad (6)$$

where $\mathcal{H}^{\Omega_T^t} = \left\{ \boldsymbol{H}_{c_{N_T}}^{\Omega_T^t}, ..., \boldsymbol{H}_{c_{N_T-t-1}}^{\Omega_T^t} \right\}$. Fig. 7 shows an example of this *rSTC* descriptor. Similarly, if the shape context is created only from the center of Ω_T^t, another new reverse Central Stroke-based Time-order Content (*rCSTC*) can be generated as follows:

$$r\text{CS}TC(T) = \left\{ \boldsymbol{H}^{\Omega_T^0}, \boldsymbol{H}^{\Omega_T^1}, ..., \boldsymbol{H}^{\Omega_T^t}, ..., \boldsymbol{H}^T \right\}, \quad (7)$$

where $\boldsymbol{H}^{\Omega_T^t}$ is the shape context with the center of Ω_T^t.

3. HIERARCHICAL AIR-WRITING RECOGNITION

The dynamical time warping technique is adopted in our system to measure the distance between two trajectories; that is, $DTW_X[t_i, t_j]$ with the descriptor X.

3.1 Reverse Representation with Weighting

The reverse time-order representation compares each stroke accordingly from the last one to the first one. Thus, the stroke written later is more important than the early one. For the tth stroke S_t, this paper enforces its importance W_t being proportional to t, i.e.,

$$W_t = \frac{2t}{N(N+1)}.$$

Then, $DTW_X[t_i, t_j]$ can be modified as follows:

$$\begin{aligned} DTW_X[t_i, t_j] = W_{t_i} cost_X[t_i, t_j] + \min(DTW_X[t_i - 1, t_j], \\ DTW_X[t_i, t_j - 1], DTW_X[t_i - 1, t_j - 1]). \end{aligned} \quad (8)$$

where $cost_X[t_i, t_j]$ is defined differently according to the used descriptor X based on Eq.(1).

3.2 Coarse-to-Fine Classification

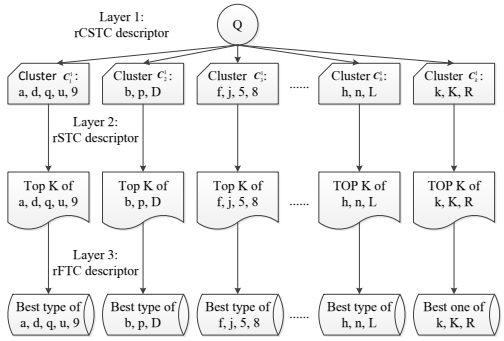


Fig. 8: Hierarchical structure for air-writing recognition.

This paper proposes a novel hierarchical classification scheme to recognize air-written texts more accurately. Fig. 8 shows the hierarchical structure for air-writing recognition. The first layer is constructed for tackling the confusion problem; the second and third layers are used for dealing with the problem of multiple styles in character writing. The feature used in the first layer should capture most of common characteristics of elements in the same category. It

should be simple, efficient, and representative enough to quickly filter out impossible candidates. Thus, the rCSTC descriptor is adopted in the first layer. Assume the kth cluster in the lth layer is denoted by $\mathbb{C}_k^l$ and the air-writing query is Q. The distance between Q and the kth cluster $\mathbb{C}_k^1$ in the first layer is defined as follows:

$$dist_{L1}(Q,\mathbb{C}_k^1)=\frac{1}{|\mathbb{C}_k^1|}\sum_{D\in\mathbb{C}_k^1} DTW_{rCSTC}[|T_Q|,|T_D|]. \quad (9)$$

Assume $\bar{\mathbb{C}}_Q^1$ is the best category of Q found in the first layer. Then, $\bar{\mathbb{C}}_Q^1$ can be obtained by the following equation:

$$\bar{\mathbb{C}}_Q^1=\arg\min_{\mathbb{C}_k^1} dist_{L1}(Q,\mathbb{C}_k^1). \quad (10)$$

For example, if Q is '*a*', the cluster $\bar{\mathbb{C}}_Q^1$ will be $\mathbb{C}_1^1$ as shown in Fig. 8. The second layer tries to tackle the problem of multiplicity. Assume that $\mathbb{C}_k^2$ is the kth category in the second layer; that is,

$$\mathbb{C}_k^2=\{\boldsymbol{T}_1,\boldsymbol{T}_2,...,\boldsymbol{T}_i,...,\boldsymbol{T}_{|\mathbb{C}_k^2|}\},$$

where $\boldsymbol{T}_i$ is the set of templates to record different writing styles of the ith character. The feature used in the second layer should be more powerful than the rCSTC descriptor to identify the correct type from other confused character types. Thus, the rSTC descriptor is chosen in this layer. Then, the distance between Q and $\boldsymbol{T}_i$ is defined as

$$dist_{L2}(Q,\boldsymbol{T}_i)=\frac{1}{|\boldsymbol{T}_i|}\sum_{D_{ij}\in\boldsymbol{T}_i} DTW_{rSTC}[|T_Q|,|T_{D_{ij}}|], \quad (11)$$

where D_{ij} records the jth writing template in $\boldsymbol{T}_i$. As described before, we assume that the best category of Q selected from the first layer based on Eq.(10) is $\bar{\mathbb{C}}_Q^1$. Furthermore, $\mathbb{C}_{\bar{\mathbb{C}}_Q^1}^2$ is supposed to be its corresponding set of writing templates constructed in the second layer. Then, the goal of the second layer is to find a collection of all the best K character types together from $\mathbb{C}_{\bar{\mathbb{C}}_Q^1}^2$. Let $\bar{\mathbb{C}}^2$ denote this collection which can be obtained as follows:

$$\bar{\mathbb{C}}^2=\mathbf{arg}\,\boldsymbol{minK}_{T_i\in\mathbb{C}_{\bar{\mathbb{C}}_Q^1}^2} dist_{L_2}(Q,\boldsymbol{T}_i), \quad (12)$$

where the function ***arg minK()*** returns the candidates which satisfies the best k minimum of $dist_{L_2}(Q,\boldsymbol{T}_i)$. To illustrate our idea more clearly, an example is given as follows. Assume that K is set to two in this paper, Q is '9', and $\mathbb{C}_{\bar{\mathbb{C}}_Q^1}^2=\{\boldsymbol{T}_a,\boldsymbol{T}_d,\boldsymbol{T}_q,\boldsymbol{T}_u,\boldsymbol{T}_9\}$. Only the best two types will be selected for further comparison; that is, $\bar{\mathbb{C}}^2=\{\boldsymbol{T}_q,\boldsymbol{T}_9\}$. In the final layer, the best type can be finely and more efficiently selected from $\bar{\mathbb{C}}^2$. Because K is smaller, we can use a time-consuming but more powerful descriptor to determine the best candidate; that is, the rFTC descriptor (see Eq.(5)). In the final layer, given a template $\boldsymbol{T}_i\in\bar{\mathbb{C}}^2$, we define the distance between Q and $\boldsymbol{T}_i$ as

$$dist_{L3}(Q,\boldsymbol{T}_i)=\frac{1}{|\boldsymbol{T}_i|}\sum_{D_{ij}\in\boldsymbol{T}_i} DTW_{rFTC}[|T_Q|,|T_{D_{ij}}|]. \quad (13)$$

Then, with $dist_{L3}(Q,\boldsymbol{T}_i)$, the best type of Q can be accurately determined from $\bar{\mathbb{C}}^2$ by the form:

$$\bar{\mathbb{C}}=\arg\min_{\boldsymbol{T}_i\in\bar{\mathbb{C}}^2} dist_{L3}(Q,\boldsymbol{T}_i). \quad (14)$$

4. EXPERIMENTAL RESULTS

To evaluate the performance of our method, a dataset which collects 1180 3D-signatures from 25 users over six months was used. The frame rate of our system is above 30 *fps*. Fig. 9 shows some results of our method to recognize different characters.

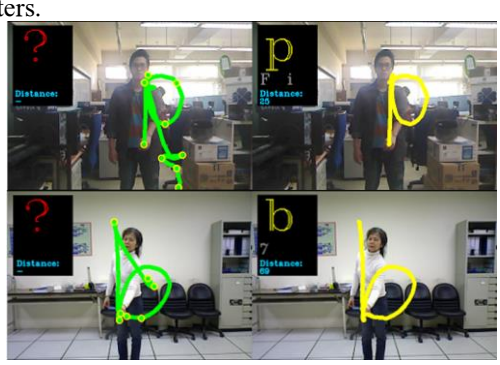


Fig. 9: Results of our proposed method to recognize different characters.

Table 1 shows the accuracy analyses of our method to recognize all alphabets written in air. The forward representation cannot effectively filter out redundant trajectories and thus got quite lower accuracies. Table 2 shows the accuracy comparisons among different methods [2], [4], [5], [6]. The four methods were designed for recognizing only "numbers". Thus, they performed poor in recognizing other non-number letters. Most of them need a starting signal to inform the writing action.

TABLE 1: ACCURACY OF OUR METHOD TO RECOGNIZE 62 LETTERS.

Letters	**A**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**
Accu (%)	88	100	86	96	98	98	98	80	88	92	92
Letters	**L**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**
Accu (%)	96	100	90	96	94	90	98	98	98	84	100
Letters	**W**	**x**	**y**	**z**	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Accu (%)	100	100	96	100	98	98	82	92	100	90	86
Letters	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**
Accu (%)	94	92	90	84	98	90	92	92	100	94	98
Letters	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**	**1**	**2**	**3**
Accu (%)	94	96	94	92	98	100	96	94	98	90	100
Letters	**4**	**5**	**6**	**7**	**8**	**9**	**0**	**Average**			
Accu (%)	98	100	100	98	100	98	70	94.2%			

TABLE 2: ACCURACY COMPARISONS AMONG METHODS [2], [4], [5], [6].

Methods	**Chiang[2]**	**Chu [4]**	**Huang[5]**	**Murata [6]**	**Proposed**
Digital	95.5%	90.8%	94.6%	95.0%	99.0%
Letters	79.8%	72.4%	73.5%	77.6%	96.1%
All	75.5%	63.7%	68.7%	71.5%	94.77%

5. REFERENCES

[1] X. Zhang , Z. C. Ye , L. W. Jin , Z. Y. Feng, and S. J. Xu , "A New Writing Experience: Finger Writing in the Air Using a Kinect Sensor," IEEE Multimedia, pp.85-93 , 2013.

[2] C.-C. Chiang, R.-H. Wang, and B.-R. Chen, "Recognizing Arbitrarily Connected and Superimposed Handwritten Numerals in Intangible Writing Interfaces," Pattern Recognition, 2016.

[3] C. Z. Qu, D. Y. Zhang and J. Tian, "Online Kinect Handwritten Digit Recognition Based on Dynamic Time Warping and Support Vector Machine," Journal of Information & Computational Science, vol. 12, no. 1, pp.413-422, Jan. 2015.

[4] T.-T. Chu and C.-Y. Su, "A Kinect-Based Handwritten Digit Recognition for TV Remote Controller," IEEE International Symposium on Intelligent Signal Processing and Communications Systems, pp.414-419, 2012.

[5] F.-A. Huang , Chung-Yen Su, and Tsai-Te Chu , "Kinect-Based Bid-Air Handwritten Digit Recognition using Multiple Segments and Scaled Coding," IEEE International Symposium on Intelligent Signal Processing and Communications Systems, pp. 694-697, Nov. 2013.

[6] T. Murata and J. Shin, "Hand Gesture and Character Recognition Based on Kinect Sensor," International Journal of Distributed Sensor Networks, 2014.

[7] A. Schick, D. Morlock, and C. Amma, "Vision-Based Handwriting Recognition for Unrestricted Text Input in Mid-Air," Proceedings of the 14th ACM international conference on Multimodal Interaction, pp.217-220, 2012.

[8] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 24, no. 4, pp.509-522, April 2002.