An HMM Implementation for On-line Handwriting Recognition Based on Pen-Coordinate Feature and Pen-Direction Feature

Daiki Okumura, Seiichi Uchida, and Hiroaki Sakoe
Graduate School of Information Science and Electrical Engineering,
Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka-shi, 812-8581 Japan

Abstract

An on-line handwritten character recognition technique based on a new HMM is proposed. In the proposed HMM, not only pen-direction feature but also pen-coordinate feature are separately utilized for describing the shape variation of on-line characters accurately. Specifically speaking, the proposed HMM outputs a pen-coordinate feature at each inter-state transition and outputs a pen-direction feature at each intra-state transition, i.e., self-transition. Thus, each state of the proposed HMM can specify the starting position and the direction of a line segment by its incoming inter-state transition and intra-state transition, respectively. The results of recognition experiments on 10-stroke Chinese characters show that the proposed HMM outperforms the conventional HMM which does not use the pen-coordinate feature because of its non-stationarity.

1. Introduction

This paper concerns on-line character recognition and has two purposes. The first purpose is the proposal of a new HMM which utilizes both pen-direction feature and pen-coordinate feature. The second purpose is the proposal of a stroke-order free recognition framework for multi-stroke characters, such as Chinese characters, using the above HMM.

HMM has been employed in on-line character/string recognition [1, 2] because of its promising ability to model the geometric deformations of strokes and the variations of the number of sample points. The first issue on designing HMM is its topology and the left-to-right topology with self-transitions has been employed in general. Te remaining issues are: (i) the stroke segment assigned to a single state and (ii) output features. Those two issues are mutually related because the output features should stable within a stroke segment assigned to a single state.

The pen-direction feature has almost always been employed in conventional HMMs [3, 4, 5, 6, 7, 8]. If a stroke can be considered as a sequence of straight line segments, the pen-direction feature is stable within each line segment. Thus, for HMM with the pen-direction feature, each line segment naturally becomes the segment to be assigned to a single state and therefore the number of states is nearly equals to the number of line segments.

In contrast, the pen-coordinate feature has not always been employed although it is a more primal feature than the pen-direction feature. This will be because the pen-coordinate feature is not stable within a line segment. Thus, in order to use the pen-coordinate feature in the HMM framework, one should many many states ($\sim$ the number of sample points) [9, 10] at the cost of model complexity.

The proposed HMM utilizes both pen-direction feature and pen-coordinate feature while keeping the number of states at the number of line segments. The basic idea is a separative and alternate use of those two features. Specifically, the pen-direction features are observed at intra-state transitions and the pen-coordinate features are observed at inter-state transitions.

After reviewing the conventional HMM in Section 2, the proposed HMM for modeling a single stroke is formulated in Section 3. In Section 4, the proposed HMM is embedded into a stroke-order free multi-stroke character recognition framework, called cube search. In Section 5, the results of recognition experiments on single-stroke characters and multi-stroke characters are given.

2. Conventional HMM

Let $xy_1, \ldots, xy_t, \ldots, xy_T$ represent a single stroke, where $xy_t = (x_t, y_t)$ is the pen-coordinate feature at time t. A pen-direction feature θ_t is often defined by the angle of $xy_t - xy_{t-1}$. Figure 1 shows a typical conventional stroke HMM, hereafter called θ-HMM, where θ_t is observed at each state according to the distribution of the direction of the line segment. Each state is assigned to a line segment of the stroke since θ_t is stable within the segment.

As noted before, the pen-coordinate feature xy_t is not always employed in conventional stroke HMMs because it is not stable within line segment. In other words, one single

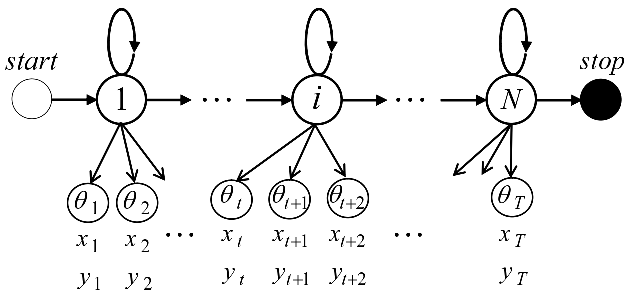

Figure 1. A conventional HMM (θ-HMM). At each state, a directional feature θ_t is observed and a pen-coordinate feature $xy_t = (x_t, y_t)$ is not.

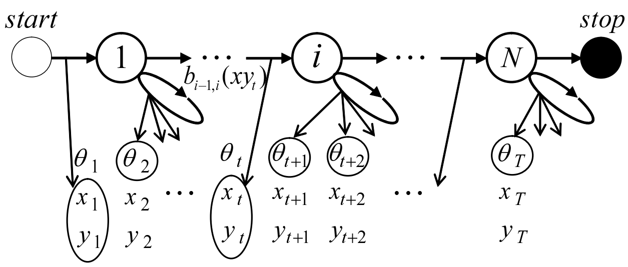

Figure 2. The proposed HMM ((xy/θ)-HMM). A pen-direction feature d_t and a pen-coordinate feature $xy_t = (x_t, y_t)$ are observed at each self-transition and inter-state transition, respectively.

state is not sufficient to represent all the variations of the pen-coordinate feature within a line segment.

The lack of the pen-coordinate feature, however, induces another problem that not only the position but also the length of a line segment can not be regulated by HMM. This problem will degrade the discrimination performance of HMM. For example, the discrimination between "1" and "7" are ambiguous without length. Furthermore, the problem becomes far more serious for stroke-order free recognition of multi-stroke characters. This is because in multi-stroke characters, the position and the length of each stroke are very important. For example, three-stroke Chinese characters "工" ("engineering") and "土" ("soil") can not be distinguished without the pen-coordinate feature.

3. The proposed HMM

Figure 2 shows the proposed HMM, hereafter called (xy/θ)-HMM. In (xy/θ)-HMM, not only pen-direction feature but also pen-coordinate feature are utilized with careful consideration for the stabilities of those features. Specifically, a pen-coordinate feature (x, y) is observed at

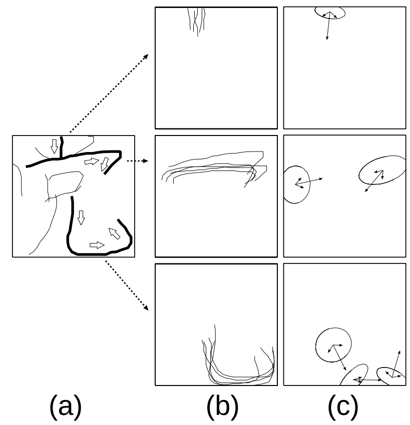

Figure 3. (a) A sample of "党" ("party"). (b) Variation of three strokes. (c) Output probabilities of (xy/θ)-HMM for the strokes. In (c), an ellipsoid represents the $\pm 2\sigma$-range of xy at the starting point of a line segment. Three arrows emerge from the mean of xy (i.e., the center of the ellipsoid). The center arrow among them represents the mean of θ and the remaining two arrows represent the $\pm 2\sigma$-range of θ.

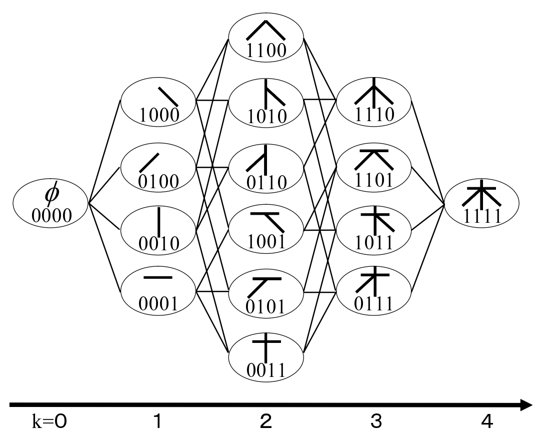

Figure 4. Cube graph for optimal stroke correspondence to the reference pattern of four-stroke character "木" ("tree"). This cube graph also becomes a probabilistic one by embedding a stroke HMM and a probability $p_{m,n}$ into each edge.

an inter-state transition and a pen-direction feature θ is observed at an intra-state transition (i.e., self-transition). Thus, each state specifies the starting position and the direction of a line segment by the its incoming inter-state transition and its self-transition, respectively.

The training of (xy/θ)-HMM can be done using the forward variable $\alpha_t(i)$ calculated by

$$\begin{aligned}\alpha_t(i) &= \alpha_{t-1}(i-1)a_{i-1,i}b_{i-1,i}(xy_t) \\ &\quad +\alpha_{t-1}(i)a_{i,i}b_{i,i}(\theta_t), \qquad (1)\end{aligned}$$

where $a_{i,j}$ is the transition probability from state i to j, $b_{i-1,i}(xy_t)$ is the observation probability of a pen-coordinate feature at the transition from state $i-1$ to i, and $b_{i,i}(\theta_t)$ is the observation probability of a pen-direction feature at the self-transition of state i. In this paper, $b_{i-1,i}(xy_t)$ and $b_{i,i}(\theta_t)$ are represented by two-dimensional and one-dimensional normal distributions, respectively. The backward variable can be calculated by the same manner.

The training of (xy/θ)-HMM can be done by the conventional Baum-Welch algorithm with the forward and backward variables calculated by the above manner. Figure 3 shows the training result of the (xy/θ)-HMM for a 10-stroke Chinese character, "党" ("party").

The value $\alpha_T(N)$ is a likelihood of the input data, where N is the number of states, i.e., the number of line segments forming a stroke. For recognizing an single-stroke on-line character pattern by the HMM, we can use $\alpha_T(N)$ as a discriminant function. (See Section 5.3.)

4. Extension to multi-stroke characters — cube search HMM

In this section, the above stroke HMM is embedded into a stroke-order free multi-stroke character recognition framework, called *cube search*. Note that we can use θ-HMM, (xy/θ)-HMM, and another HMM for the stroke HMM.

4.1. Cube search[11]

Cube search proposed in [11] is a technique for determining the optimal stroke-to-stroke matching between two K-stroke on-line Chinese character patterns, i.e., an input pattern $\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_K$ and a reference pattern $\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_K$. Each of $\boldsymbol{I}_k$ and $\boldsymbol{R}_k$ is a single stroke (often composed of $1 \sim 3$ line segments). Mathematically, cube search for K-stroke characters is formulated as an optimal path problem on a K-dimensional cube graph. Figure 4 shows the cube graph for the four-stroke Chinese character "木" ("tree"). Any node of the cube graph is indexed by a K-bit binary word. If the lth bit of a node of the kth *stage* of the cube graph is "1", $\boldsymbol{R}_l$ is matched to one of $\boldsymbol{I}_1, \ldots, \boldsymbol{I}_k$. Clearly, any node of the kth stage has k non-zero bits.

Consider a node m of the $(k-1)$th stage and a node n of the kth stage. If those two nodes are different only at the lth bit (e.g.,, "0101" and "1101"), two strokes $\boldsymbol{I}_k$ and $\boldsymbol{R}_l$ are matched by the transition from m to n and the similarity of those strokes is added as the weight $w_{m,n}$ of the *edge* between m and n. Finally, the optimal stroke-to-stroke matching is the maximum weight path which begins at the node "$00\ldots0$" and ends at the node "$11\ldots1$".

Note that the similarity is calculated in a deterministic way in [11], whereas it is calculated in a probabilistic way in this paper, as shown in the followings.

4.2. Embedding stroke HMMs to cube search

An HMM-based stroke-order free recognition framework, hereafter called cube search HMM, can be realized by using the following weight $w_{m,n}$ at the edge between m and n,

$$w_{m,n} = p_{m,n} L_l(\boldsymbol{I}_k), \qquad (2)$$

where $L_l(\boldsymbol{I}_k)$ is the likelihood of $\boldsymbol{I}_k$ by the stroke HMM of $\boldsymbol{R}_l$, that is, a similarity between $\boldsymbol{I}_k$ and $\boldsymbol{R}_l$. The probability $p_{m,n}$ represents an *a priori* transition probability from m to n and therefore represents the probability that the k stroke of the input pattern matches with the lth stroke of reference pattern. If $p_{m,n}$ is set appropriately, we can penalize rare stroke orders. The probability $p_{m,n}$ should satisfy $\sum_{n \in \mathcal{N}(m)} p_{mn} = 1$, where $\mathcal{N}(m)$ is the set of nodes succeeding to the node m.

A probabilistic stroke-order-invariant similarity between an input pattern and a reference pattern can be obtained as the accumulated similarity along with the optimal path searched by the Viterbi algorithm. Again, we can use any stroke HMM to provide the likelihood $L_l(\boldsymbol{I}_k)$.

4.3. Training of cube search HMM

In this paper, the following simple three-step algorithm is used for training the cube search HMM for category c:

[Step 1] The stroke HMM for $\boldsymbol{R}_l$ is individually trained using the lth strokes of training patterns of c written in their correct stroke order.

[Step 2] An initial cube search HMM is constructed by using the trained stroke HMMs and the probabilities $p_{m,n} = 1/\mathcal{N}(m)$.

[Step 3] The probability $p_{m,n}$ is trained using patterns written in various stroke orders whereas the parameters of the stroke HMMs are fixed. Simply speaking, if the edge between m and n is included in the optimal path for a training character, $p_{m,n}$ is increased.

5. Experimental results

5.1. Data sets

Recognition experiments were conducted on two data sets: *dataset-A* is a private on-line Chinese character data

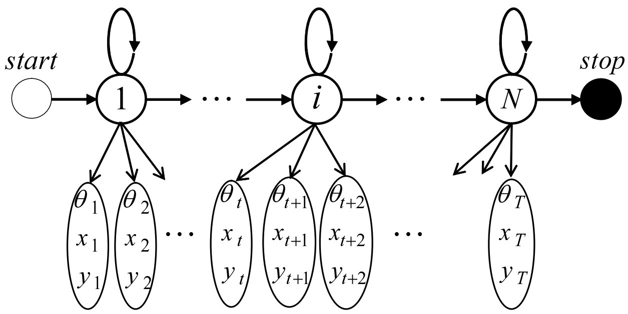

Figure 5. Another HMM ((xy, θ)-HMM) for comparative evaluation.

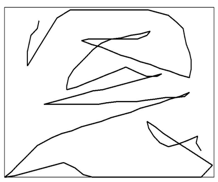

Figure 6. A single-stroke character created by connecting the 10 strokes of a character "恩" ("kindness"). This character was misrecognized as "息" ("breath") by θ-HMM and correctly recognized as "恩" by (xy/θ)-HMM.

set (30 writers) and *dataset-B* is the public on-line Chinese character data set called "HANDS-kuchibue_d-97-06-10" [12] (10 writers). Among the samples of those data sets, the samples belonging to 70 categories of 10-stroke Chinese characters subjected to the experiment. (Consequently, 1444 samples from dataset-A and 1051 samples from dataset-B were subjected.) All samples were linearly rescaled to be 128×128 and then resampled.

5.2. Another HMM for comparative evaluation

In addition to θ-HMM and (xy/θ)-HMM, another stroke HMM, hereafter called (xy, θ)-HMM (Fig. 5), was subjected to the experiment. In (xy, θ)-HMM, not only pen-direction feature but also pen-coordinate feature were observed at *every* state. Each state is assigned to a line segment (as well as the other two HMMs) and therefore the pen-coordinate feature is forcedly assumed to be stable within each line segment.

5.3. Recognition of single-stroke character by stroke HMM

In order to evaluate the basic performance of the proposed (xy/θ)-HMM of Section 3, a recognition experiment was conducted on single-stroke characters created by connecting all the strokes of original 10-stroke character samples written in their correct stroke orders. Figure 6 shows a single-stroke character of "恩" created in this manner. All HMMs were trained by using 1212 single-stroke characters from dataset-A, and tested by using 1212 and 767 single-stroke characters from dataset-A and dataset-B, respectively. The number of the states were the same for the three HMMs; N states were prepared for the single-stroke character composed of N line segments.

Table 1 shows that the proposed HMM could attain the highest recognition rates among the three HMMs. Note that the recognition rates tend to be high because all Chinese characters used here were written in their correct stroke orders. Figure 6 shows a character "恩", which is misrecognized as "息" by θ-HMM and correctly recognized by (xy/θ)-HMM. In this case, the difference in the pen-coordinate feature of the starting point was crucial for the correct recognition.

5.4. Recognition of multi-stroke character by cube search HMM

Another recognition experiment was performed to evaluate the performance the cube search HMM on recognizing multi-stroke characters with variable stroke orders. As the stroke HMM used in the cube search HMM, all the three stroke HMMs, i.e., (xy/θ)-HMM, θ-HMM, and (xy, θ)-HMM were examined. The number of states of a stroke was the same in the three HMMs and equal to the number of the line segment of the stroke. For example, two states ($N = 2$) were prepared for a "¬"-shaped stroke.

Cube search HMM for each character category was trained by the three-step algorithm of Section 4.3. For training individual stroke-HMMs, 1212 samples (with correct stroke order) of dataset-A were used. Then, for training $p_{m,n}$, 1444 samples of dataset-A were used.

Table 2 shows the recognition results. In the table, the column "fixed $p_{m,n}$" means the case that $p_{m,n}$ was fixed at $1/\mathcal{N}(m)$, namely, all the stroke order variations were assumed to have the same probability. On the other hand, the column "trained $p_{m,n}$" means the case that the probability $p_{m,n}$ was trained by the algorithm of Section 4.3.

Among three stroke HMMs, the proposed (xy/θ)-HMM outperforms other two HMMs in recognition accuracy (whether $p_{m,n}$ was trained or not). It should be emphasized that the performance of the conventional θ-HMM was very disappointing when it was used in the multi-stroke character recognition. This degradation was caused by the lack of the pen-coordinate feature. If the pen-coordinate feature is lacked, the similarity between two strokes becomes high even though their positions are totally different. Thus, its lack allows unnatural stroke-to-stroke matching results and often allows very high similarity between two different characters (Fig. 7). Although the use of trained $p_{m,n}$ re-

Table 1. Recognition rate of single-stroke characters (%).

test data	A (closed)	B (open)
θ-HMM	99.9	99.3
(xy, θ)-HMM	89.5	81.2
(xy/θ)-HMM	100.0	99.9

Table 2. Recognition rate of multi-stroke characters (%).

	fixed $p_{m,n}$		trained $p_{m,n}$	
test data	A (closed)	B (open)	A (closed)	B (open)
θ-HMM	59.0	53.1	92.6	90.5
(xy, θ)-HMM	94.5	85.6	96.7	91.2
(xy/θ)-HMM	97.7	93.9	98.5	97.4

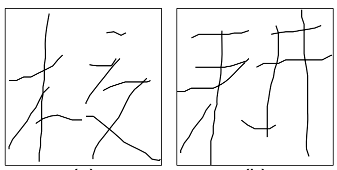

Figure 7. Multi-stroke characters misrecognized by cube search HMM based on θ-HMM and fixed $p_{m,n}$. Character (a) "校" ("school"), and (b) "耕" ("cultivate") were misrecognized as "料" ("stuff") and "特"("special"), respectively. Each character is misrecognized to a character with a totally different shape.

laxes this problem as shown in Table 2, the performance of θ-HMM is inferior to that of (xy/θ)-HMM.

The recognition rates attained by (xy, θ)-HMM are higher than those of θ-HMM and lower than those of the proposed HMM. In (xy, θ)-HMM, the positional variations of all points on a line segment are represented by a single normal distribution. Thus, the ability of (xy, θ)-HMM on compensating the positional variations is not sufficient. In fact, many misrecognized samples of (xy, θ)-HMM undergo the translations of strokes.

6. Conclusion

A novel HMM, called (xy/θ)-HMM, is proposed for online handwritten character recognition technique. In the proposed HMM, not only pen-direction feature but also pen-coordinate feature are utilized for describing the shape variation of on-line characters. Specifically speaking, the proposed HMM outputs a pen-coordinate feature at each inter-state transition and outputs a pen-direction feature at each intra-state transition, i.e., self-transition. In order to use the proposed HMM for multi-stroke character recognition, it is embedded in a stroke-order free recognition framework, called cube search. The results of experiments on 10-stroke Chinese characters show the usefulness of the proposed HMM especially for the recognition of multi-stroke characters.

Acknowledgment: The authors thank Prof. Masaki Nakagawa and his laboratory at Tokyo Univ. of Agriculture and Technology for providing us online character database, "HANDS-kuchibue_d-97-06-10". This work was supported in part by the Ministry of Public Management, Home Affairs, Posts and Telecommunications in Japan under Strategic Information and Communications R&D Promotion Programme (SCOPE).

References

[1] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: a comprehensive survey," IEEE Trans. PAMI, vol. 22, no. 1, pp. 63–84, 2000.

[2] C. -L. Liu, S. Jaeger, and M. Nakagawa, "Online recognition of Chinese characters: the state-of-the-art," IEEE Trans. PAMI, vol. 26, no. 2, pp. 198–213, 2004.

[3] R. Nag, K. H. Wong, and F. Fallside, "Script recognition using hidden Markov models," Proc. ICASSP, vol. 3 of 4, pp. 2071–2074, 1986.

[4] E. J. Bellegarda, J. R. Bellegarda, D. Nahamoo, and L. S. Nathan, "A fast statistical mixture algorithm for online handwriting recognition," IEEE Trans. PAMI, vol. 16, no. 12, pp. 1227–1233, 1994.

[5] T. Starner, J. Makhoul, R. Schwartz, and G. Chou, "Online cursive handwriting recognition using speech recognition methods," Proc. ICASSP, vol. 5 of 6, pp. 125–128, 1994.

[6] H. J. Kim, K. H .Kim, S. K. Kim, and J. K. Lee, "On-line recognition of handwritten Chinese characters based on hidden Markov models," Pattern Recognition, vol. 30, no. 9, pp. 1489–15000, 1997.

[7] T. Hasegawa, H. Yasuda, and T. Matsumoto, "Fast discrete HMM algorithm for on-line handwriting recognition," Proc. ICPR, vol. 4 of 4, pp. 535–538, 2000.

[8] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, "Substroke approach to HMM-based on-line Kanji handwriting recognition," Proc. ICDAR, pp. 491–495, 2001.

[9] K. C. Jung, S. M. Yoon, and H. J. Kim, "Continuous HMM applied to quantization of on-line Korean character spaces," Pattern Recognition Letters, vol. 21, no. 4, pp. 303–310, 2000.

[10] C. Bahlmann and H. Burkhardt, "The writer independent online handwriting recognition system *frog on hand* and cluster generative statistical dynamic time warping," IEEE Trans. PAMI, vol. 26, no. 3, pp. 299–310, 2004.

[11] J.-P. Shin and H. Sakoe, "Stroke correspondence search method for stroke-order and stroke-number free on-line character recognition — multilayer cube search —," IEICE Trans., vol. J82-D-II, no. 2, pp. 230–239, 1999. (In Japanese.)

[12] M. Nakagawa and K. Matsumoto, "Collection of on-line handwritten Japanese character pattern databases and their analyses," IJDAR, vol. 7, no. 1, pp. 69–81, 2004.