HMM-BASED ON-LINE MULTI-STROKE SKETCH RECOGNITION

WEI JIANG[1], ZHENG-XING SUN[2]

[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093
[2]Department of Computer Science and Technology, Nanjing University, Nanjing 210093
E-MAIL: szx@nju.edu.cn

Abstract

This paper describes a new approach for on-line multi-stroke sketch recognition. The approach is based on Hidden Markov Model (HMM). Sketches are modeled to HMM chains, and strokes are mapped to different HMM states. The proposed approach introduces a new method to determine HMM state-number, based on which an adaptive HMM sketch recognizer is constructed. A combined feature based on curvature, velocity and geometrical character of stroke for sketch recognition is also proposed to improve recognition accuracy. Finally, the experiments prove the effectiveness and efficiency of the proposed approach.

Keywords:

Sketch Recognition; Multi-stroke; Adaptive Hidden Markov model

1. Introduction

Sketches help us convey ideas and guide our thinking process both by aiding short-term memory and by helping to make abstract problems more concrete. Most importantly, sketching is a natural input modality of increasing interest [1]. Recognizing their value, several researchers have paid attention to sketch recognition[2][3][4][5], either as a natural input modality[2][4] or to recognize complex sketchy objects[3][5]. The difficulty comes from that sketching is usually informal, inconsistent and ambiguous both in intra-person and inter-person settings in a given situation, and sketch recognition engine should automatically adapt to a particular user's sketching styles.

There have been a wide variety of sketch recognition techniques. Most of them are primitives-based, where the inputting patterns are first decomposed into basic geometric primitives (such as lines and curves) and then assembled into a graphical structure that encodes both the intrinsic attributes of the primitives and their relationships [6]. Sketching recognition is accordingly formulated as template-matching problem, for instance, the graph-isomorphism in graph-based method [7][8]. However, these approaches are highly sensitive to the stroke segmentation process, and their performance degrades drastically when applied to drawings that are heavily sketchy. In fact, stroke is natural representative of user's sketching styles. Stroke segmentation can lead to the lost of information about user's drawing styles. This is why the poor accuracy of traditional sketch recognition engines is always frustrating, especially for the newly added users even in the latest experimental systems [2][3][4][5].

To capture user's habit of sketching style, sketch recognition would be stroke-based and more complex, statistical approaches are required. Rubine[9] describes a trainable gesture recognizer for direct manipulation interfaces. A gesture is characterized by a set of 11 geometric and 2 dynamic attributes. Based on these attributes, a linear discriminant classifier is constructed whose weights are learned from the set of training seven examples. Because this method was developed exclusively for gesture-based interfaces, it is only applicable to single-stroke sketches and is sensitive to the drawing direction and orientation. Parametric methods [10] such as polygon, B-spline and Bezier curve fitting techniques have also been considered in shape representation and classification. A benefit of these approaches is that these methods are computationally efficient since only a few parameters are needed for shape description. Similar to the Rubine's method, however, these methods are mostly applicable to single-stroke sketches such as characters in handwritten text or gesture commands. In previous researches, we have developed a sketch recognition method based on SVM [11]. It can actively analyze the users' incremental data, and can largely reduce the workload of artificial labeling and the classifier's training time. While it has been proved both effective and efficient in our experiments, it can still deal with only single-stroke sketches since dimension of feature vectors of SVM must be fixed for all shapes.

In this paper, we will present our experiments in multi-stroke sketch recognition in terms of Hidden Markov models (HMM), inspired by its success in speech recognition [12] and handwriting recognition [13].

The rest of the paper is organized as follows: The HMM topology we selected is described firstly. An adaptive HMM approach to sketch recognition is discussed in succession. Then, we propose a combined feature for sketch recognition. Finally, some experiments and conclusion are given.

2. Selection of Hidden Markov model Topology

Hidden Markov model is one of the most successful stochastic modeling tools that have been used in the analysis of nonstationary time series [16]. HMM has been used with great success in stochastic modeling of speech for years. It has also been widely used in handwriting recognition in recent years [13][14][15].

In HMM, the observed pattern is viewed as the result of a stochastic process that is governed by a hidden stochastic model. Each stochastic model represents a different class pattern capable of producing the observed output. The goal is to identify the model that has the highest probability of generating the output. One aspect that distinguishes HMMs is their strong temporal organization; processes are considered to be the result of time-sequenced state transitions in the hidden model and expectation of a particular observation is dictated by the current state in the model and (usually) the previous state.

In on-line multi-stroke sketch recognition, drawing sketch, especially drawing multi-stroke sketch, can be regard as a time-sequenced process. Different users have different drawing styles. The input sketches for the same shape are quite different from user to user (e.g., when drawing a multi-stroke sketch, some users like to draw it in one sequence while others like to draw it in another) and even from time to time. Therefore, HMMs can be used to model different sketches and they can easily represent the user's drawing styles.

The HMM topology used in pattern recognition can be divided into two categories: chain topology and network topology. HMM chain has a simple structure. It is easy to implement and is widely used in recognizing simple symbol, for example: gesture recognition [9]. HMM network is constructed by grouping and interconnecting HMM chains and is largely used in recognizing handwritten characters [13][15]. To date, there has been no serious study or guidance in the use of HMM in sketch recognition, and it is the first time that we use HMM in sketch recognition. In this paper, we have selected the simple HMM chain topology because it has been shown to be successful in speech and handwriting recognition.

3. Adaptive HMM Approach

3.1. HMM State-number Determination

HMM needs enough free parameters to accommodate complexity of target patterns and to represent properties of the patterns. However, in practice, available training samples are usually limited, so it is usually difficult to obtain enough free parameters. In our approach, we focus on one design parameter: the number of states in HMM.

The number of HMM states is an important design parameter. For instance, a state could correspond to certain phonetic event in a sketch recognition system. Thus, in modeling complex patterns, the number of states should be increased accordingly. When there are insufficient numbers of states, the discrimination power of the HMM is reduced, since more than one signal should be modeled on one state. On the other hand, the excessive number of states can generate the over-fitting problem when the number of training samples is insufficient compared to that of the model parameters.

There are two approaches to determining HMM state-number used in on-line handwriting recognition. The first is using fixed state-number, which means using the same HMM state-number while training each category of samples. The second is using variable state-number, which means the handwritten characters are divided into subcomponents according to some given criterion (usually are divided by strokes). Each subcomponent is modeled by one single HMM state.

Neither of the two methods mentioned above is fit for on-line multi-stroke sketch recognition because sketch has its own characteristics compared with handwritten character.

First, the spatial relationships between strokes of one given sketch are more complex than that of the handwritten character. If we use fixed states number, we need to segment the sketch into subcomponents. The spatial relationships between strokes, which contain important sketching style information, will be broken, and the recognizer cannot capture enough information to represent user's sketching habits. Obviously, the recognition accuracy will be reduced.

Second, a number of standard character databases are present. In addition, the handwritten characters are some fixed, predefined, and well-known graphics objects among writers and readers, which have strict definition for strokes and stroke-sequence, so we can analyze all characters in the standard character databases and obtain the number of subcomponents which are often used in different characters. In sketch recognition, there is no such standard database, so we cannot analyze all of the sketches and enumerate all of the subcomponents which construct the sketches, and we cannot determine the states number according to the number of subcomponents.

As mentioned above, we must find a new approach to determine the number of HMM states in multi-stroke sketch recognition. Although the sketches drawn by different users are very different from each other, they are all drawn in strokes which are joined one by one. The stroke-number of one given sketch is different from each other among different sketching styles. Even if the numbers of strokes are same; the structure of each stroke will be different from each other. Stroke is natural representative of user's sketching styles. The recognition performance will upgrade if we make better use of the information contained in these strokes. In this paper, we proposed an adaptive HMM based on variable state number for the purpose of description of multi-stroke sketch. In this approach, the number of HMM states is determined by the structural decomposition of the target pattern. Sketch is structurally simplified as a sequence of strokes.

The main idea behind the proposed approach is to use one single HMM state to model each stroke. While collecting samples, the recognition system will automatically store the stroke-number of each sample (which is defined to be SNumber). Before we train the HMMs, we analyze the stored numbers and find out the maximum emergent number (which is defined to be TNumber) for each category of sketches. We consider TNumber to be the state-number of HMM, because the samples which correspond to TNumber are frequently drawn by user and they can represent the user's drawing habit. Then we train the HMM as follows.

If TNumber>SNumber, we segment the last stroke of the sketch into TNumber-SNumber+1 segments on average, and then model the remaining strokes and these segments to TNumber HMM states.

If TNumber<SNumber, we group the last SNumber-TNumber+1 strokes to one virtual stroke, and then model the remaining strokes and the virtual stroke to TNumber HMM states.

Using the proposed approach to determine HMM states number, the inner structure of HMM models is easily altered according to different users. Moreover, the approach does not need too much intervention by the user. After the user has been familiar with the input environment and the structure of sketches which are usually drawn, the user will draw one given sketch almost in the same style. Then the stroke-number will become equal to TNumber. Then the recognizer will seldom segment the sketch drawn by user. Compared with other approaches, our approach is fit for on-line multi-stroke sketch recognition. Our experiments show that the proposed method is both effective and efficient.

3.2. On-Line Multi-Stroke Sketch Recognition Based on Adaptive HMM

According to the characteristics of the drawing sketch, the position and structure of current stroke is usually depend on the previous stroke, and the position and structure of next stroke is depend on the current stroke. We assume that the stroke of one sketch drawn by user is only correlated with previous stroke and next stroke. Therefore, in our approach, we use a first-order left-to-right HMM (the topology is shown in Figure 1) to model each sketch. It is strictly causal: The current state depends only upon previous states. The experiments in handwriting recognition showed that this topology lead a high recognition accuracy.

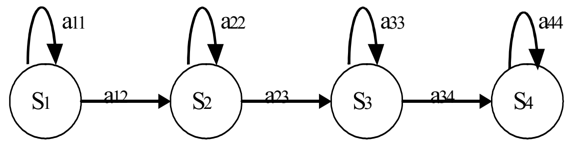


Figure 1. Adaptive HMM Topology

In training stage, we extract characteristic features from each stroke and use the method mentioned in the previous section to determine the HMM state-number and some other parameters. Models are trained using the well known iterative segmental training method based on Viterbi decoding. The transition probabilities indicate the relationships between strokes. The HMM chain can represent user's drawing habit very well.

In recognition stage, the recognizer calculates the probabilities using the trained HMM and returns the recognition results in the sequence of probabilities from high to low to user.

4. Features for Multi-Stroke Sketch Recognition

We consider that the features extracted for on-line multi-stroke sketch recognition must satisfy the following three criteria. First, the features must contain both geometric (spatial) and dynamic (temporal) features; Second, the features do not need to describe the sketch in detail; Third, the features must be able to represent the spatial relationships between strokes (not inner stroke). In this paper, we proposed a combined feature based on curvature, velocity and geometrical characteristics of stroke to satisfy the three criteria mentioned above.

In feature selection, Calhoun et al. [6] proposed a feature based on velocity and curvature which can be regard as a dynamic feature. Some researchers proposed a feature based on sketch centroid-radius which can be regard as a geometric feature.

They are both dynamic features and can describe the characteristics of sketch very well, so we combine them together in our approach. In addition, for multi-stroke sketch, the spatial relationships between strokes are also important characteristics of sketches. So we add a one-dimension vector composed of stroke-trend into our proposed combined feature to describe the spatial relationships between strokes. Finally, the combined feature is a seven-dimension vector; the components are shown in Table 1. First, we define some concepts, which are used in our approach as follows:

Centroid-radius: the distance between points of sketch and barycenter of sketch.

Point-velocity: the ratio of distance between current point and previous point to the time spent between drawing the two points.

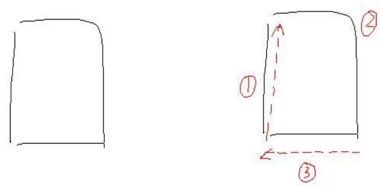
Point-curvature: the cosine of the corner angle constructed by the previous point, the current point and the next point.

Stroke-trend: the slope of virtual line, which is constructed by connecting the end-point of previous stroke and the start-point of current stroke.

Table 1. Components of Combined Feature

Feature	Feature Description	Feature Category	
f_1	mean of centroid- radius	global feature	geometric feature
f_2	standard deviation of centroid-radius	global feature	geometric feature
f_3	mean of point- velocity	global feature	dynamic feature
f_4	standard deviation of point-velocity	global feature	dynamic feature
f_5	mean of point- curvature	global feature	geometric feature
f_6	standard deviation of point-velocity	global feature	geometric feature
f_7	stroke-trend	global feature	geometric feature

During training and recognition stage, we extract combined feature from each stroke. The seventh feature is a variable vector according to the stroke type. If the stroke is not the last stroke of the sketch, the seventh feature is stroke-trend. On the contrary, the seventh feature is length ratio of the sketch to its closure. Figure 2 shows the structure of combined feature. Figure 2-(a) shows a sketch drawn by user. The sketch is drawn in three strokes, and the stroke-sequence is shown in Figure 2-(b). The dashed shows the stroke-trend and the elevations are defined to be α and β. The length ratio of the sketch to its closure is defined to be ε. Finally, the complete combined feature is shown in Figure 2-(c).



(a) Input sketch (b) Feature description

$$(f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, \tan\alpha)$$
$$(f_{21}, f_{22}, f_{23}, f_{24}, f_{25}, f_{26}, \tan\beta)$$
$$(f_{31}, f_{32}, f_{33}, f_{34}, f_{35}, f_{36}, \varepsilon)$$

(c) Feature extraction

Figure 2. Combined feature

5. Experiments and Evaluation

The purpose of our experiments is to evaluate the effectiveness and efficiency of the recognition approach we have proposed above. That is, the experiments should evaluate the recognition performance and the effectiveness of the adaptive HMM based on variable state-number. Thus, we design our experiments as follows: In order to evaluate our proposed state-number determination approach, we carry out the experiments with both variable state-number and fixed state-number. We also perform experiment to evaluate the performance of adaptive HMM.

By analyzing users' input strokes and some familiar graphics-based design software, we have set 9 classes of sketch which are most commonly used in the sketching process and are shown in Figure 3.

For the data collection, we collected two users' samples of these sketches. While using some present sketch recognition system, the users are forced to draw only one stroke for simple shapes. However, a successful system should not restrict the user's dawning styles. For comparison, we ask the first user to draw these sketches in one-stroke and the second user to draw these sketches freely. The numbers of each sample are shown in Table 2.

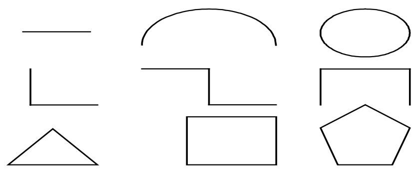


Figure 3. All Nine Classes of Sketches

Table 2. Sample Structure of User 1 and User 2

	Straight Line	Arc	Broken Line 1	Broken Line 2	Broken Line 3
User 1	801	800	800	800	800
User 2	811	820	798	851	802
	Triangle	Quadrangle	Pentagon	Ellipse	Total
User 1	800	800	800	801	7203
User 2	848	846	817	815	7408

Our experiment environment is Pentium 4 1.6G CPU, 256MB memory, Windows 2000, Visual C++ 6.0.

5.1. Comparison between Fixed State-number HMM and Variable State-number HMM

In the previous section of this paper, we consider that variable state-number HMM is better than fixed state-number HMM in sketch recognition. The experiment in this section will confirm the conclusion. Because sketches which have more number of strokes can easily lead to multi-drawing-style, we choose polygon samples drawn by User 2 for experiment. From the samples drawn by the second user, we obtain the drawing habits of the second user in forms of stroke-sequences, which are shown in Figure 4.

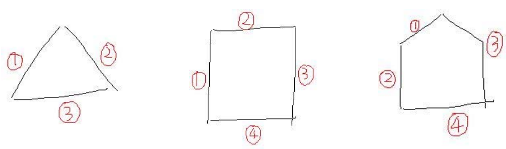


Figure 4. Polygon Drawn by User 2

In fixed state-number experiment, given the state-number to be 1, 2, 3 and 4, we obtain four different recognition precision. In variable state-number experiment, we use the state-number determining approach mentioned above to determine the state-number (In this experiment, the numbers of states are 3, 3 and 4) for each class.

Figure 5 shows the result of our experiments. As we can see, the red pentagram corresponds to triangle recognition (using variable state-number) precision, while the green one is for quadrangle and the blue one is for pentagon. The other points represent results for fixed state-number experiment.

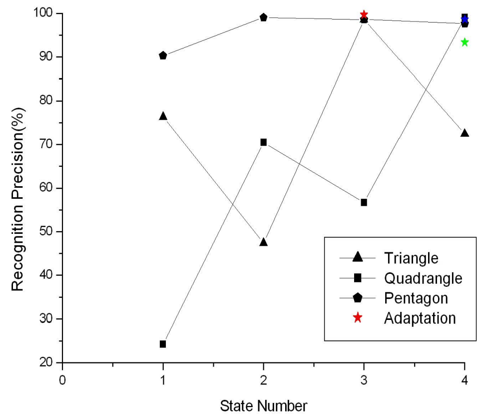


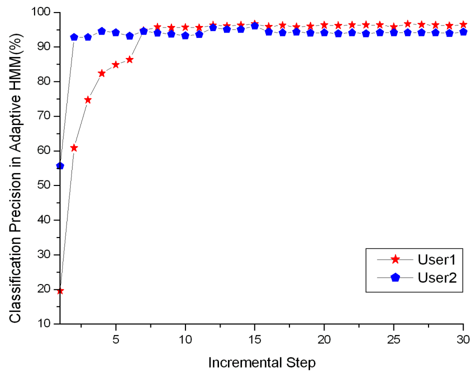
Figure 5. Recognition Precisions of Fixed State-number and Variable State-number HMM

As is shown in the figure, recognition precision of different classes depends on state-number when fixed state-number approach is been used. For one given fixed state-number, different classes of sketch cannot achieve the highest recognition precision at the same time. However, the recognition precisions of each class are nearly the same when using variable state-number approach. We make a conclusion that variable state-number approach can adapt different drawing styles, and it is better than fixed state-number approach in on-line multi-stroke sketch recognition.

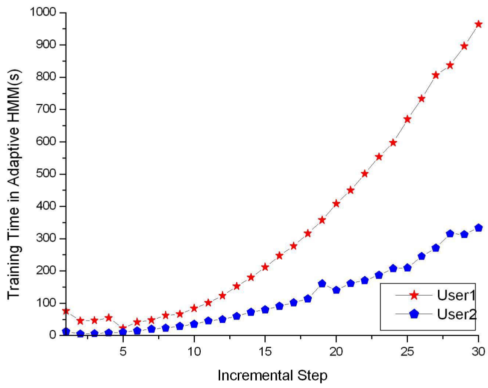
5.2. Recognition Performance of Adaptive HMM

In this experiment, 60% of first user's samples are used for training, while 75% of second user's are used for training, and the rest samples are used for testing. We divide the samples into 30 training sets for each user. The first five training sets contain 1%, 1%, 2%, 3% and 3% of the total samples of each user. Each of the rest 25 training sets contains 3.6% of the total samples of each user. The experiment results are shown in Figure 6.

The adaptive HMM obtain a high performance in multi-stroke sketch (sketches drawn by the second user) recognition, the recognition precision reaches to 95% after the second training, while in one-stroke sketch (sketches drawn by the first user), the recognition precision reaches to 95% after the seventh training. The results show that adaptive HMM has a good performance in adapting to user's drawing habits, especially under multi-stroke sketch and small train sets.



(a) Classification Precision in Adaptive HMM



(b) Training Time in Adaptive HMM

Figure 6. Experiment Results of Adaptive HMM

6. Conclusion

In this paper, we have exploited an HMM based method for multi-stroke sketch recognition. A multi-stroke sketch is modeled to one HMM chain and the strokes are mapped to different HMM states. First, we propose a new method to determine the HMM state-number which is variable to fit the on-line multi-stroke sketch recognition: a variable state-number determining method. Then we propose an adaptive HMM topology in sketch recognition. Finally, we review the features used in handwritten character and sketch recognition in some previous researches and propose a combined feature based on curvature, velocity and geometrical characteristics of stroke for sketch recognition. The experiment proves the proposed method both effective and efficient.

Acknowledgements

The work described in this paper is supported by the grants from the National Natural Science Foundation of China [Project No. 69903006 and 60373065] and the Program for New Century Excellent Talents in University of China.

References

[1] Landay J A and Myers B A, Sketching Interfaces: Toward more human interface design, IEEE Computer, 2001, 34(3): 56-64.

[2] Newman M W, James L, Hong J I, et al, DENIM: An informal web site design tool inspired by observations of practice, HCI, 2003(18): 259-324.

[3] Sun Z X, Zhang B, Qiu Q H and Zhang L S, A Freehand Sketchy Graphic Input System: SketchGIS, IEEE International Conference on Machine Learning and Cybernetics, Xian, Nov, 2003: 3232-3237.

[4] Takeo Igarashi, Freeform User Interfaces for Graphical Computing, Smart Graphics 2003, Lecture Notes in Computer Science, 2003, vol. 2733: 39-48.

[5] Levent Burak Kara, Thomas F, Stahovich, Sim-U-Sketch: A Sketch-Based Interface for Simulink, Proceedings of AVI-2004: 354-357

[6] Chris Calhoun, Thomas F Stahovich, Tolga Kurtoglu, et al, Recognizing multi-stroke symbols, In AAAI Spring Symposium on Sketch Understanding, AAAI Technical Report SS-02-08, Stanford University, AAAI Press, 2002: 15-23.

[7] Seong-Whan Lee, Recognizing hand-drawn electrical circuit symbols with attributed graph matching, In H S Baird, H Bunke, and K Yamamoto, editors, Structured Document Image Analysis, Springer-Verlag, 1992: 340–358.

[8] Xu X G, Sun Z X, Peng B B, et al, An online composite graphics recognition approach based on matching of spatial relation graphs, International Journal of Document Analysis and Recognition, 2004，7(1)：44～55。.

[9] Dean Rubine. Specifying gestures by example. Computer Graphics, 1991, 25:329–337.

[10] Huang Z and Cohen F, Affined-invariant b-spline moments for curve matching. IEEE Transactions on Image Processing, 1996,5(10): 1473–1480.

[11] Sun Z X, Liu W Y, Peng B.B, et al, User Adaptation for Online Sketchy Shape Recognition, Lecture Notes in Computer Science, 2004, Vol 3088: 303-314.

[12] Jen-Tzung Chien, On-line unsupervised learning of hidden Markov models for adaptive speech recognition, IEE Proceedings - Vision, Image and Signal Processing, 2001, 148(5):315-324.

[13] Jianying Hu, Michael K Brown, William Turin, HMM-based online handwriting recognition， IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, 18(10).

[14] Jay J.Lee, Jah Wan Kim and Jin H.Kim Data-driven Design of HMM Topology for On-line Handwriting Recognition, in the proceedings of 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, September 2000.

[15] Mitsuru Nakai, Naoto Akira, Hiroshi Shimodaira, et al, Substroke Approach to HMM-based On-line Kanji Handwriting Recognition, Sixth International Conference on Document Analysis and Recognition (ICDAR 2001), pp.491-495, 2001.

[16] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proc. Of the IEEEE, Vol.77, No.2, pp.257-286, Feb. 1989