An Online Hand-Drawn Electric Circuit Diagram Recognition System Using Hidden Markov Models

Yingmin ZHANG [1,2] Christian VIARD-GAUDIN [1] Liming WU[2]
1. IRCCyN/URM CNRS 6597 Ecole Polytechnique de l'Université de Nantes, France
2. College of Information Engineering, Guangdong University of Technology, Guangzhou, China
ymzhang09@gmail.com

***Abstract*—In this paper we experiment the capabilities of Hidden Markov Models (HMM) to model the time-variant signal produced by the movement of a pen when drawing a sketch such as an electrical circuit diagram. We consider that the sketches have been generated by a two-level stochastic process. The underlying process governs the stroke production from a neuro-motor control point of view: go straight, change direction, produce a curve. A second stochastic process delivers the observed signal, which is a sequence of sampled points. Three different architectures of HMM are proposed and compared. On a dataset of 100 hand-drawn sketches, the proposed method allows to classify correctly more than 83% of the points with respect to the connector and symbol classes.**

Keyword: Pen-based interaction, stroke segmentation, Hidden Markov Model, Electrical circuit diagram, hand-drawn sketch.

I. INTRODUCTION

With the development of Tablet PCs, digital pen and paper technologies, electronic whiteboards and other electric digitizers, pen interfaces are an actual convenient modality to input sketches in a computer. While handwritten text recognition is already widely addressed as a research topic [1], much less effort has been devoted to hand-drawn sketch understanding. Support Vector Machine (SVM), Hidden Markov Model (HMM), Neural Networks (NN), Genetic Algorithms (GA), and many other pattern matching algorithms have been fostered by this challenging application and very promising results are obtained by combining these techniques [2].

Hand-drawn sketch is a natural and direct way to express people's thought and meaning and is of common use in many different fields. Diagram sketches are widely used in engineering and architecture fields. This is mainly due to the fact that a sketch is a convenient tool to catch rough idea, so that the designers can focus more on the critical issues rather than on the intricate details [3]. In this work, we propose to investigate the problem of automatic understanding of online hand-drawn sketches and as an important processing step, we aim at modeling the pen trajectory when switching from connectors to symbols drawing with the use of Hidden Markov Models.

Hidden Markov Models (HMM) were first introduced in the late 1960s, and they are especially good at modeling sequential and temporal phenomenon. It has received many achievements in applications, such as speech recognition [4], gesture recognition [5], and handwritten text recognition [6]. However, fewer people attempt to spread HMM into sketching recognition research because it's not as regular as characters. Sezgin and Davis [7] proposed converting the original sketches strokes into basic geometric objects, such as line and arcs. Then they viewed the sketch recognition as an interactive and incremental process allowing users to sketch the objects in multiple ways [8]. Recently, they attempt to group the symbols in object-level because one object would comprise multiple strokes or have shared strokes, learning users' drawing style [9]. Artieres [10] introduced basic geometric shapes to deal with two-dimensional graphical shapes such as Latin and Asian characters. More generally, these 2-D graphical shapes can be used to encode patterns like gestures, symbols, small drawings. Herry and Wardhani [11] proposes to use chain-code line features to recognize isolated symbols. Corners here are detected simply by direction change, and it can only deal with line primitives, while our approach can recognize strokes mixing lines and arcs.

Although different people may scribble the same symbol in different strokes numbers and orders, the drawing of sketches itself is temporally related. We believed the drawing context can help to do segmentation. In this paper, we present HMM models that can present strokes in lines and arcs. The Stroke segmentation is done by finding out the optimal path of the input with respect to the model. The organization of the paper is as follows. Section Ⅱ describes the pre-processing techniques from the raw input data. Section Ⅲ presents the architecture and topology of the HMM models. Experiments and results are discussed in section Ⅳ, followed by the concluding remarks in the final section.

II. PRE-PROCESSING STAGE

The velocity of the sketching tool can be very different in different places of the sketch or in sketches from different users. In order to be independent with respect to

978-0-7695-3494-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ISISE.2008.222

the speed of the tool, but also of the size of the symbols, and the resolution of the sampling device, a pre-processing stage is carried out. It consists in estimating a scaling parameter, which will be used to resample the sketch at a fixed spatial interval. However, straight lines and irregular segments are not processed in the same way. For long line segments, it is done with an idiographic equal distance based on the number of original points and on the line length. For arcs or lines with many turning-points in small distances, we will detect all the possible turning-point based on the symbol height variant and set the same number of points with an idiographic equal distance between two extremity turning-points.

Consequently, after pre-processing, we expect to obtain approximately the same number of points for symbol features such as resistor height whatever their original size was. The scaling parameter is automatically computed from the histogram of the length of the small line-segments.

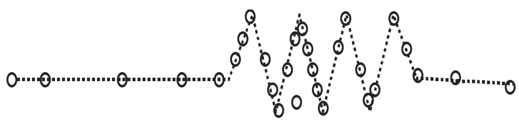

Figure 1. Raw input data

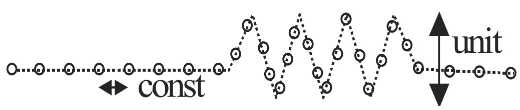

Figure 2. Normalized data

In Fig.2, the sketch normalized is classified into long line and short line (or arc) resample, instead of normalizing with the same fix distance everywhere, which can farthest avoid neglecting the inflexion or corner information. For the long line segment, it is done with an idiographic equal distance based on the number of original points and the line distance. For arcs or a line with many turning-points in short distance, we will detect all the possible turning-point based on the symbol height variant and set the same number of point with an idiographic equal distance between two neighboring turning-points.

III. OBSERVATION VECTOR FRAME

From the normalized re-sampled sequence of points, we derive a vector stream which will define the observation sequence of the HMM. At this stage only two main features have been considered, namely local direction and local curvature. Later, to achieve a specific symbol recognizer, additional geometric features will be included, as in [6].

The local direction θ_t is computed by the mean of the first two components f_1 and f_2 of the observation vector. Consider P_t for example, its local direction is that of line $P_{t-1}P_{t+1}$, which are defined by the cosine and sine value of the slant angle of line $P_{t-1}P_{t+1}$, (1) and (2).

The last two components f_3 and f_4 f are linked to the local curvature, computed as the direction bias from line $P_{t-1}P_{t+1}$ to line P_tP_{t+2}. When these two segments are aligned, the bias is very small, it can even reach zero. On the contrary, it increases up to the value of 2π. Equation (3) and (4) can derive from (1) and (2).

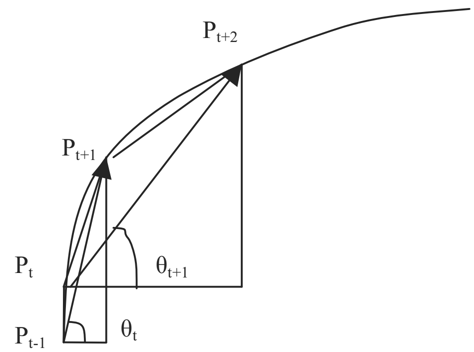

Figure 3. Direction and curvature

$$f_1(t) = \cos\theta_t = \frac{x_{t+1} - x_{t-1}}{\sqrt{(x_{t+1} - x_{t-1})^2 + (y_{t+1} - y_{t-1})^2}} \quad (1)$$

$$f_2(t) = \sin\theta_t = \frac{y_{t+1} - y_{t-1}}{\sqrt{(x_{t+1} - x_{t-1})^2 + (y_{t+1} - y_{t-1})^2}} \quad (2)$$

$$\begin{aligned} f_3(t) &= \cos(\Delta\theta) = \cos(\theta_{t+1} - \theta_t) \\ &= \cos\theta_{t+1}\cos\theta_t + \sin\theta_{t+1}\sin\theta_t \\ &= f_1(t+1).f_1(t) + f_2(t+1).f_2(t) \end{aligned} \quad (3)$$

$$\begin{aligned} f_4(t) &= \sin(\Delta\theta) = \sin(\theta_{t+1} - \theta_t) \\ &= \sin\theta_{t+1}\cos\theta_t - \cos\theta_{t+1}\sin\theta_t \\ &= f_2(t+1).f_1(t) - f_1(t+1).f_2(t) \end{aligned} \quad (4)$$

To adapt the HMM model, we define the Probability Density Function (PDF) of the four features as the observation emission probability, which will be described in section 4.

We will suppose that these 4 components are independent so that the overall Probability Density Function (pdf) of the observation vector is the product of the respective four pdf components.

IV. HMM ARCHITECTURES

An HMM is a statistical model that describes a dynamic process, and consists of *states* with *transitions* between the states [3]. There are N *emitting* states $\{q^0, q^1, \ldots, q^{N-1}\}$ that have observation PDFs associated with them. The two states q^s and q^f, without associated PDFs, are called *non-emitting* states. These two additional non-emitting states serve as initial and terminating states, thus eliminating the need for separate initial and terminating probabilities (see [4] for more detail).

Specifically, each emitting state qi is associated with a PDF $pdfi(\mathbf{x}(t))$, where $t \in \{1, \ldots, T\}$ and $i \in \{0, \ldots, N\text{-}1\}$. where $\boldsymbol{x}(t)$ denotes a D-dimensional feature vector at discrete-time instant t, and T is the length of the observed sequence.

In general, one can express an HMM as:

$$\lambda = \{\boldsymbol{A}, \{ pdfi(\mathbf{x}(t)), i = 1, \ldots, N\}\}, \tag{5}$$

where $\boldsymbol{A}$ is a matrix representation of the transition links and $pdfi(\mathbf{x}(t))$ is the observation PDF of qi evaluated at $\mathbf{x}(t)$, for $i \in \{1, \ldots, N\}$ and $t \in \{1, 2, \ldots, T\}$, where $\mathbf{x}(t)$ is a D-dimensional vector.

Now, let us describe our segmental HMM architecture. The role of HMM model is to decode pen-based signals into a sequence of elementary stroke states (see Fig.4 and 5) using the Viterbi algorithm, which is a stroke level representation (SLR) process. In SLR, the models implement time-varying functions describing the time evolution of the tangent angle of the drawing.

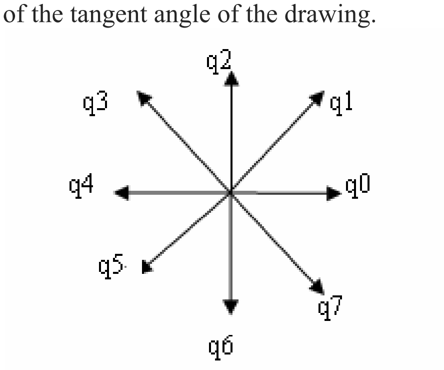

Figure 4. Definition of straight line direction

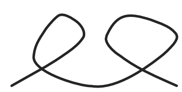

Figure 5. Example of non-straight segments

As in Artieres's character basic shape dictionary [9], we use a fixed stroke alphabet of nine elementary strokes as primitives, which correspond to nine states for the HMM: eight straight lines states and one single non-straight line state. The 8 straight line elements (q0-q7) express the 8 directions of a straight line (see Fig.4). The single non-straight line element models all other situations (see Fig.5), such as angle, round, arc, and curve. This alphabet is rich enough to represent a large collection of diagram symbols.

A basic topology HMM model is shown in Fig.6. In this model, all the states are treated with equal transition probabilities because we assume the same probability to go to any direction from the current one. Since different diagram sketches have different kinds of symbol, each symbol has its own stroke connect relationship. To make a comparable evaluation, this is the best suitable model for any arbitrary sketches.

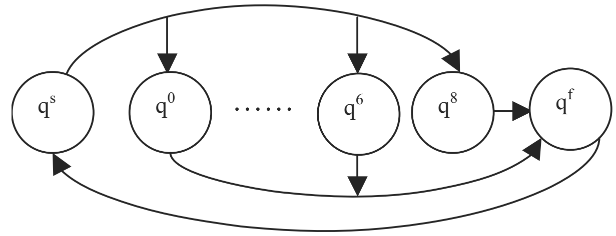

Figure 6. Basic equal probability HMM model

This equal transition probability model is a very generic one, but does not favor any specific geometric structure. When sketching for an instance an electric diagram, it is much more likely to follow the same direction most of the time, and to switch from one direction to a new one only occasionally. Thus to handle such a situation, it is possible to increase the self state transition when compared to other ones. Fig. 7 corresponds to the implementation of such a model.

However, for a special kind of diagram, the equal probability model is an optimal one. Online hand writing research and recognition is a process of studying human's writing habits. Different users have different stroke order on the same diagram, or different diagrams have different special stroke connect relationships. Hence, the equal probability model will neglect the details, leading to a decrease of the recognition rate. Fig.7 is an improved model to solve the problem, with raising the probability keeping in the same state. Continuing in the same state is a frequent occurring phenomenon. The sampling duration of the hand-writing devices often goes to the millisecond or microsecond level and it's impossible for user to finish one stroke even one short segment drawing in such a short time. For example, in an electric circuit diagram, a connector will be finished with keeping drawing in the same direction line. Thus, this model suits one special kind of diagram application.

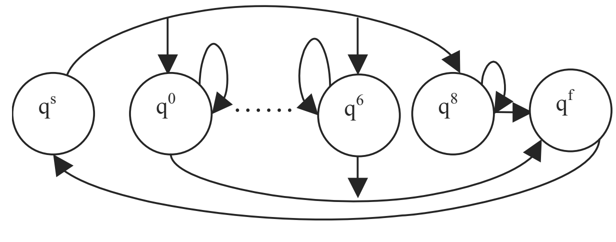

Figure 7. Temporal improved HMM model

A yet more precise model is proposed in Fig.9. In our research, we take the electric circuit diagram as the study object. The new model raises some transition probability based on the practical circuit situations, like turning to right angle. Consider q0 of Fig.4 for example, it is a left-right horizontal direction line state. Compared to other states, it has a higher probability keeping in the same state or turning to right angle (see Fig.8). Other states have similar condition. With this special model we can encode user stroke order and habit in different kinds of diagram and increase the specificity of the modeling system.

These different HMM models represent different diagram study targets. With these three models, various hand-drawn diagrams or diagram of a kind are able to expand multi-angle research.

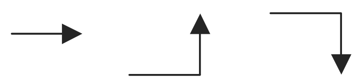

Figure 8. **Turning directions for state q0**

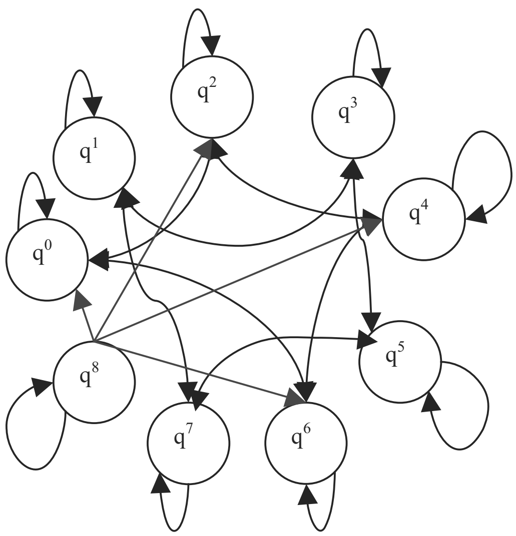

Figure 9. Specialized HMM model

Since the PDF components are independent, the joint observation PDF of q^i, evaluated at feature vector $\mathbf{o}(t)$, is given by:

$$pdf_i(t) = \prod_{j=1}^{4} pdf_{i,j}(o_j(t)) \quad (6)$$

Here, the feature pdf is assumed to have Gaussian distribution with a standard deviation corresponding to be on the frontiers between two consecutive directions.

An observation sequence of points $\boldsymbol{O}$ (derived from the pen trajectory) is matched to λ using the Viterbi algorithm [4]. The result is the optimal hidden state sequence $\mathbf{q} = [q_1, q_2, \ldots, q_m]$. The globally optimized likelihood of $\mathbf{q}$, based on λ and $\boldsymbol{O}$, is then given by:

$$\delta = a_{q_0 q_1} \prod_{t=1}^{T} a_{q_t q_{t+1}} pdf_{s_t}(\mathbf{o}(t)) \quad (7)$$

where q_0 = q^s corresponds to the non-emitting initial state of λ, q_{T+1} = q^f corresponds to the non-emitting terminating state of λ, and $f_{st}(\mathbf{o}(t))$ is described by (4). As alluded to above, $\mathbf{q}$ is useful in our application to identify the label corresponding to straight lines (q^0 to q^7) and non-straight lines (q^8) in $\boldsymbol{O}$.

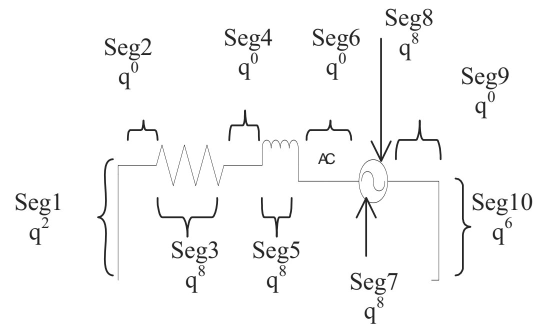

Figure 10. Example of stroke segmentation

In this figure, the Viterbi algorithm first finds a best state sequence according to the time-varying feature value, then groups the data with the same state value together and makes a fragmentation for different state segments at last.

V. SYSTEM EVALUATION

In our research, electric circuits have been considered as typical diagrams to test the proposed segmentation system. Based on the segmentation carried out by the Viterbi algorithm, we aim to extract all the connectors, which are supposed to be only straight line segments, and consequently to increase the level of understanding of the sketch by providing a classification of the segments within connector or component classes. Sketches will be made up 9 possible components as defined in Fig.11.

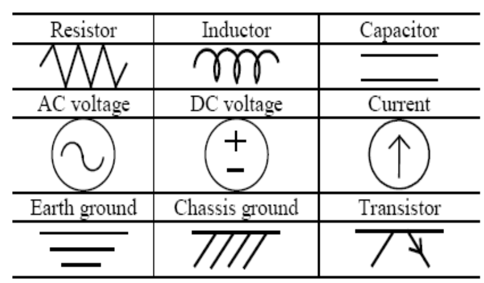

Figure 11. Electric circuit components used in our experiment [12]

We asked 10 subjects to draw each a series of electric circuits. Samples are collected using the Anoto digital pen and paper technology, so that drawing is done very freely, as on ordinary document. Each of them is asked to copy 10 diagrams and the information is stored as a series of 2D coordinates. As the samples are drawn on the paper, modification is not allowed. However, there are no constraint on the direction and the size. Figure 12 is an example of a collected electrical sketch.

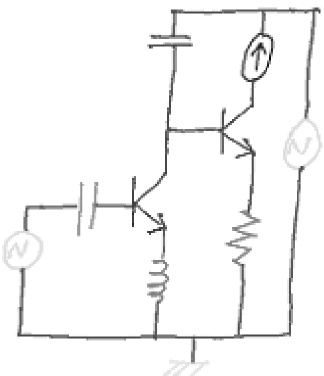

Figure 12. Hand-drawn electric circuit diagram

Table 1 shows the confusion matrix corresponding to the classification results obtained with the three HMM models. Table (a) is the basic equal probability model of Fig.6. Table (b) is the model of Fig.7 with self-loops. Table (c) is the specialized model of Fig.8.

Table 1. Recognition results of the 3 HMM models *

(a)

Reco \ Ground truth	Connector	Component
Connector	68. 75% (A)	50. 04% (C)
Component	31. 25% (B)	49. 96% (D)

(b)

Reco \ Ground truth	Connector	Component
Connector	75.62% (A)	21.08% (C)
Component	24.38% (B)	78.92% (D)

(c)

Reco \ Ground truth	Connector	Component
Connector	82. 94% (A)	16. 10% (C)
Component	17. 06% (B)	83.82% (D)

*zone A = P (reco- Connector | label-Connector)

$$= \frac{num-recognized-connector}{num-labeled-connector} \times 100\%$$

(following zones are the same)

zone B = P(reco- Component | label-Connector)

zone C = P(reco-Connector | label-Component)

zone D = P(reco -Component | label-Component)

From the tables it can be seen that, different HMM models have different recognition result because of the different targets. Concerning the connector extraction rate, it is 68.75% in table (a), 75.62% in table (b), and 82.94% in table (c). Adding some knowledge based constraint for state transition in electric circuit sketch allows to increase the performances. The specialized HMM models have higher efficiency than the basic equal probability model. This can be observed from the model of table (c). It means our system can locate most of the real segment points. The component recognition rate has a similar result.

Points belonging to zone B and C are miss-recognized points. Some connector points are recognized as components (17% of the overall set of connector points), this could be the case when some retracing appear in the drawing of the connector, which destroy the expected regular straight lines and conversely some points actually belonging to component are recognized as connector points (16%). Those situations are typical of some symbols composed of only straight lines as a capacitor when large segments are used. It also concerns the extremities of the symbols, where the exact segmentation points are not localized precisely, hence some points are missing because they are assigned to the connectors which are attached to this component.

VI. Conclusion

In this paper, we proposed a partial sketch understanding based on HMMs. The sketch is considered as a continuous time-varying stroke stream. The HMMs allow first to segment an arbitrary stroke into regular straight lines from eight different directions or in irregular arcs for the complementary regions. Then, the analysis of the best state sequence produced by the Viterbi algorithm has been used to classify points into connectors or components. With the most accurate model, an average recognition rate of 83% has been achieved. This demonstrates that the extension of the use of HMM from handwritten character recognition to sketch understanding is a promising area of interest. In this domain, the more constraints can be integrated in the HMM topology, the more accurate the results would be.

References

[1] R. Plamondon, S.N. Srihari, "On-Line and Off-line Handwriting Recognition: A Comprehensive Survey", *IEEE Transactions on PAMI*, 2000, Vol.22, No. 1, pp.63-84.

[2] E. Caillault, C. Viard-Gaudin, "Mixed Discriminant Training of Hybrid ANN/HMM Systems for Online Handwritten Word Recognition", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 21, No. 1 (2007), pp. 117–134.

[3] C. Calhoun, T.F. Stahovich, T. Kurtoglu and L.B.Kara. "Recognizing multi-stroke symbols". Proceedings of the

AAAI Spring Symposium on Sketch Understanding, Palo Alto, USA, 2002, pp. 15-23.

[4] L.R. Rabiner. “A tutorial on HMM and selected applications in speech recognition”. *Proceedings of the IEEE* 77,2 (1989), pp.257-286.

[5] S. Marcel, O. Bernier, J.E. Viallet, and D. Collobert. Hand gesture recognition using input-output hidden markov models. In Proc. of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, *IEEE Press*(2000), pp. 456-461.

[6] Y.H. Tay, P.M. Lallican,. Khalid Marzuki, C. Viard-Gaudin. “An Analytical Handwritten Word Recognition System with Word-level Discriminant Training”. ICDAR’2001 (IAPR), Seattle, September, 10-13, 2001.

[7] T.M. Sezgin, T. Stahovich and R. Davis. “Sketch Based Interfaces: Early Processing for Sketch Understanding[C]”. Workshop on Perceptive User Interfaces, Orlando FL , 2001.

[8] T.M. Sezgin and R. Davis. “HMM-Based Efficient Sketch Recognition[C]”. In Proceedings of the International Conference on Intelligent User Interfaces (IUI'05). New York, January 9-12 2005.

[9] T.M. Sezgin and R. Davis. “Sketch Interpretation Using Multiscale Models of Temporal Patterns”. *Computer Graphics*, 2007 (Vol. 27, No. 1) pp. 28-37.

[10] T. Artières, S. Marukatat and P. Gallinari. “On-line handwritten shape recognition using segmental Hidden Markov Models[J]”. *In IEEE. Trans. on Pattern Analysis and Machine Intelligence*. 29.2 (2007), pp. 205-217.

[11] D. Henry, A. Wardhani. “Diagram recognition using Hidden Markov Models”. *Technical Report, Faculty of Information Technology,* Queensland University of Technology, 2003.

[12] G. Feng, C. Viard-Gaudin and Z. Sun. “Hand-drawn Electric Circuit Diagram Understanding using 2D Dynamic Programming”. ICFHR2008 , Montreal, Aug. 2008, pp. 493-498.