

3D Space Handwriting Recognition with Ligature Model

Dae Hwan Kim, Hyun Il Choi, and Jin Hyung Kim

Korea Advanced Institute of Science and Technology,
373-1 Guseong-dong, Yuseong-gu, Daejeon, Korea
{kimdh, hichoi, jkim}@ai.kaist.ac.kr

Abstract. 3D space handwriting is a gesture-like character written in the air, and it is a promising input method for its portability. In this work, we propose a practical development of 3D space handwriting recognition system by combining 2D handwriting and the ligature of 3D handwriting based on that the most different part between these handwritings is the shape of ligature. We designed a ligature model not to depend on each character shape but to depend only on incoming and outgoing vectors. Therefore with a few ligature models, various ligature shapes are modeled. By using 2D space handwriting models and data, we can use existing models for various writing styles, and overcome the problem of the lack of data.

Keywords: ligature model, 3D space handwriting, online handwriting recognition, Bayesian network.

1 Introduction

A hand gesture is often used to communicate with remotely located people. Since the gesture is intuitive and familiar with human being, it can be a good interface between human and computer. Previously, a hand gesture was tracked by cameras [1]. In this system, they used skin color to segment a hand from background. Therefore, when there are skin-colored objects such as furniture, or when the user wears gloves, the tracking system does not work properly.

Recently, with development of MEMS (micro-electromechanical systems) sensors, inertial sensor based hand tracking began [2] [3] [4] [5]. In this system, we need a sensor-equipped device in hand while we need nothing to hold in the camera based system. However, the sensor based system gives more detail information to the extent that complex shapes such as handwriting characters can be described properly. If a character recognition system is applied to this tracking system, a new interface - handwriting in the air - is made.

From now on, for convenience, we will call 3D space handwriting for handwriting in the air, and 2D space handwriting for existing handwriting inputs with pen and tablet. The advantages of 3D space handwriting are as following. First, compared to the 2D handwriting input system, users can write characters freely without the limitation that writing should be on a specific plane such as tablet. Second, the 3D

input device can be designed in small size since it needs only small inertial sensors. Third, it is a one-hand input interface.

There are already some inertial sensor based applications. A mobile game phone is developed, which uses the orientation of the phone as an additional input of the game [6]. A motion sensing phone is introduced, which recognizes gestures for digits, and simple commands of the phone [7].

If the 3D handwriting character input system can recognize not only digits, but also letters, such as alphabets, Hangul, and so on, it can take the place of the keyboard input system. For example, an Internet TV, which has many channels, searches the Internet, and finds movies, is a promising home appliance in the near future. The remote controller of the TV has to understand complex commands such as web addresses and movie titles. If we use buttons for inputs, the remote controller should have as many buttons as a keyboard has. However, if we use 3D handwriting input system, what we have to do is drawing commands with the controller. The size of the controller will be even smaller than that of a current TV controller. Likewise, 3D space handwriting can be a promising interface between human and computer for its portability and ability of understanding complex input, in the age of Ubiquitous computing. Figure 1 shows the examples of current 3D input device.

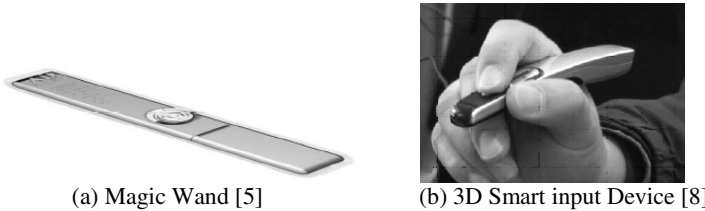


Fig. 1. Examples of 3D input devices that track 3D hand movements. The devices are small and portable.

3D handwriting is, however, hard to recognize. First, the shapes of 3D handwriting have large variance since users have difficulty in getting visual feedback. Second, 3D handwriting characters have no distinction between pen-downs and pen-ups. A pen-down means a pen state when the pen is touched to the writing surface such as a paper, and draws some parts of the character, and a pen-up means a case when the pen is up and moving to another position to be written. The natural shape of 3D space handwriting is a connected stroke. You can see the shape differences between 2D space handwriting and 3D space handwriting in Figure 2. Third, currently there are only small amount of 3D handwriting data.

In order to recognize 3D space handwriting, Oh et al. and Cho et al. defined recognition target trajectory [4] [5]. The trajectory was, so called, graffiti which is in the form of uni-stroke, and has a similar shape to the character. By restricting the number of the shapes of target traces, the recognition problem was simplified. This policy, however, forces user to use only pre-defined shapes of characters, and moreover, when the user writes a multiple stroke character such as '4', the user have to learn a new uni-stroke shape which he/she has never seen.



Fig. 2. 2D space handwriting and 3D space handwriting. 3D space handwriting is drawn with one connected stroke.

In this paper, we propose a connected stroke character recognition system in order to recognize 3D space handwriting. We do not place a limitation on writing styles. We use character models in Bayesian network framework to recognize characters that have large variance. Bayesian network based character models dynamically predict the whole shapes of characters given various inputs [9]. In order to solve the problem from no distinction between pen-downs and pen-ups, we construct a 3D handwriting ligature model which describes the shapes of pen-up movements. For the problem of lack of data, we utilize existing 2D handwriting models and data.

The rest of this paper is organized as following. In chapter 2, we introduce character models in Bayesian network framework. In chapter 3, we describe a ligature model. In chapter 4, we describe the implementation of our system. In chapter 5, we evaluate the performance of our system. Finally in chapter 6, we conclude the work and discuss future work.

2 Character Models in Bayesian Network Framework

The Bayesian network based recognizer shows high recognition performance since it models character components and their relationships explicitly. In this framework, a character model is composed of stroke models with their relationships, and a stroke model is composed of point models with their relationships [9]. We focus on the stroke model since it is applied to the ligature model to be explained in the next chapter.

A stroke is assumed to be set of points, and the stroke model is represented by point distributions approximated by Gaussian distributions and their relationship which is represented in Bayesian network. Figure 3 (a) shows various stroke examples of a stroke and their end points are displayed with points. Figure 3 (b) shows the scatter plot of positions of end points, and Figure 3 (c) shows the Gaussian distributions approximating the scatter plot. The point distribution is displayed with one node as random variable in Bayesian network structure in Figure 3 (d).

The middle point of a stroke can be represented in the same way. The distribution of the middle point, however, depends on the positions of two end points. This dependency can be represented by an arc in Bayesian network structure as you see in Figure 4.

For representing dependency, it is impossible to find all distributions corresponding to possible values since the position of the point has continuous values. However by adopting conditional Gaussian distribution [10], we can settle this problem. Conditional Gaussian distribution assumes that the mean of the distribution is the linear combination of values of dependent variables and a constant, but covariance does not depend on other values. For example, the matching probability of the middle point in Figure 4 given that two end points are observed is as following.

Let O be the instance of IP , O_1 instance of EP_1 , O_2 instance of EP_2 , \mathbf{W} be coefficient matrix

$$\begin{aligned}
 &P(IP = O \mid EP_1 = O_1, EP_2 = O_2) \\
 &= (2\pi)^{-1} |\Sigma|^{-1/2} \exp\left[-\frac{1}{2} (O - \mu)^T \Sigma^{-1} (O - \mu)\right] \\
 &\mu = \mathbf{W}[x_1 \ y_1 \ \dots \ x_n \ y_n \ 1]^T
 \end{aligned}
 \tag{1}$$

As above, we can represent point distribution by finding coefficient matrix \mathbf{W} and covariance matrix Σ .

A stroke model is defined as a joint distribution of point distributions. In Figure 4, matching probability of stroke is as follows.

$$\begin{aligned}
 &P(S = s(O_1, O, O_2)) \\
 &= P(EP_1 = O_1, IP = O, EP_2 = O_2) \\
 &= P(EP_1 = O_1)P(EP_2 = O_2) \times \\
 &\quad P(IP = O \mid EP_1 = O_1, EP_2 = O_2)
 \end{aligned}
 \tag{2}$$

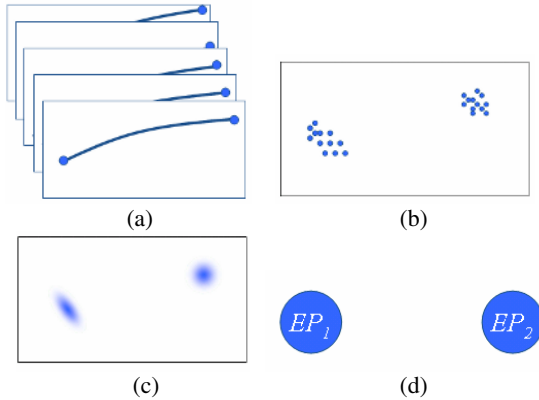


Fig. 3. Modeling of end points. (a) end point examples of strokes. (b) scatter plot of end points. (c) Gaussian distribution approximating end points distribution. (d) Bayesian network representation.

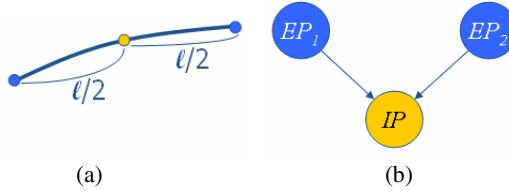


Fig. 4. Modeling of a middle point. (a) An example of a middle point. (b) Bayesian network representation of the middle point distribution.

3 Ligature Model

3.1 Need of Ligature Model

A ligature is a trajectory drawn in moving from the previous stroke to the next stroke, and it is the most different part between 2D space handwriting and 3D space handwriting. As we see in Figure 2, in 2D space handwriting, ligatures are usually drawn with pen-up, and therefore it is relatively easy to find the ligature parts from the entire trajectory of a character. Since the device cannot track the movement of the pen when the pen is up, a ligature is assumed to be a straight line connecting the end point of the previous stroke and the start point of the next stroke, and this kind of assumption ignores the detail shape of the ligature. On the other hand, in 3D space handwriting, it is not easy to find the ligature because there is no pen-up/pen-down concept. The shape of the ligature is cursive since the device gives full trajectory information including strokes and ligatures.

3D space handwriting trajectory except ligature part is similar to 2D space handwriting. Therefore by modeling the ligature of 3D space handwriting, we can efficiently adapt 2D space handwriting model to 3D space handwriting.

3.2 Characteristics of Ligature Model

The shape of a ligature has two characteristics.

First, the shape of a ligature does not depend on each character but depends on the incoming vector and the outgoing vector of the ligature. As you see in Figure 5, the ligature shape of the first consonant ‘ㄷ’ of Hangeul – the Korean alphabet – is similar with that of the vowel ‘ㅏ’. Although characters are different, only if the incoming vectors and the outgoing vectors of the ligatures are almost same, the ligature shapes are similar.

Second, as you see in Figure 6, a ligature is composed of two strokes which are nearly straight. We can assume that there is boundary which divides a ligature into two strokes, and the boundary is displayed in Figure 6 as one point. Let it called as a segmentation point. The shape of the first stroke of the ligature, that is, the stroke from the start point of ligature to the segmentation point, depends more on the incoming vector than the outgoing vector, and the shape of the second stroke depends more on the outgoing vector.

From this observation, we can model the ligature as the shape which is composed of two segments and which is affected by the shape of the previous stroke and the

next stroke. In this assumption, the ligature does not depend upon a character shape itself. If the ligature is much affected by the shape of each character, the number of ligature models is bigger than the number of the characters, and it means that we need extremely much data. However, in this model, the shape of the ligature is dynamically determined by the neighboring strokes, so the number of ligature models is one. Therefore, the model can be trained with small amount of data.

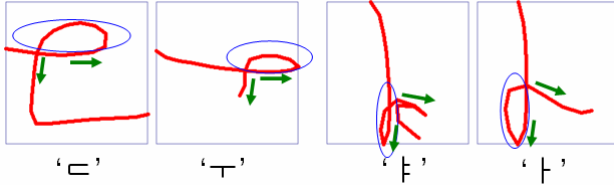


Fig. 5. Ligature shapes according to incoming vector and outgoing vector. The ligature shape is much dependent on the shape of input/outgoing vectors.

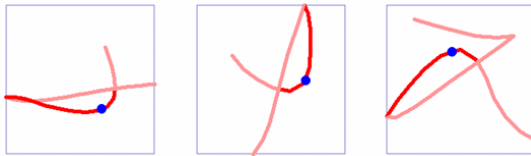
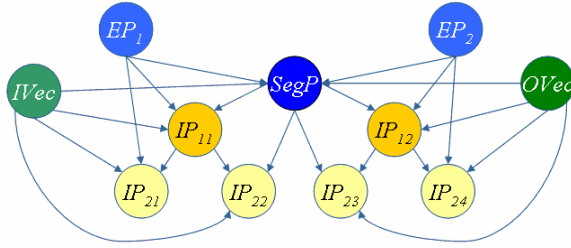


Fig 6. A ligature is composed of two nearly linear strokes. Segmentation point displayed as one big dot divides one stroke to two nearly linear segments.

3.3 Structure of Ligature Model

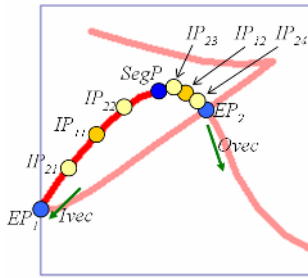
Based on characteristics of ligature shape, we form the Bayesian network structure of ligature as Figure 7 (a). Figure 7 (b) shows the points in a consonant ‘ㅌ’ that corresponds to the nodes in the structure. The proposed ligature model is composed of 9 point distributions, one incoming vector distribution, and one outgoing vector distribution. First stroke which is from the start point EP_1 to the segmentation point $SegP$, and second stroke from $SegP$ to EP_2 , have symmetric structure. The distributions of the middle points in each stroke depend on their end points. The distributions of all middle points in first stroke and the segmentation point depend on starting point EP_1 and incoming vector $IVec$, and this represents the fact that the shape of first stroke depends on incoming vector. The distributions of the middle points in second stroke and the segmentation point depend on end point EP_2 and outgoing vector $OVec$, and this reflects that the shape of second stroke depends on outgoing vector.

The segmentation point is different from the middle point of a stroke in that segmentation point is not positioned in the middle of strokes, but positioned where it can divide ligature into two two nearly straight strokes. The distribution of segmentation point is determined by two end points of ligature and incoming/outgoing vectors.



EP: end point
 IP: middle point
 IVec: incoming vector
 OVec: outgoing vector

(a) Bayesian network representation



(b) Corresponding points in the consonant 'ス'

Fig 7. Structure of a ligature model

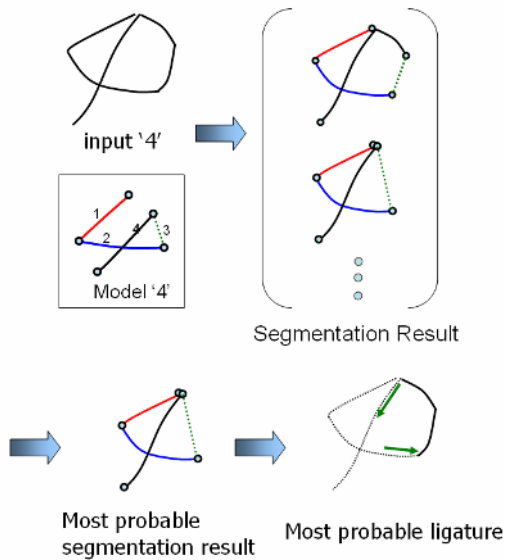


Fig. 8. Ligature extraction procedure

3.4 Ligature Extraction

Ligature extraction is a step where ligatures are extracted from an input trajectory, and this is a data collection step in ligature modeling. Ligature extraction is performed with algorithm, not with hand, by finding the part of trajectory that is most likely to be ligature by using Bayesian network based 2D space handwriting recognizer.

When a connected trajectory is inputted, the ligature extractor finds all possible segmentation results. The system segments input trajectory into N strokes. N is the number of strokes the 2D handwriting character model has. For each segmentation result, the ligature extractor finds the stroke that corresponds to ligature in the model, and makes the stroke straight. Making straight line ignores the detail shape of ligature and makes the input trajectory adapted to the 2D space handwriting model. The ligature extractor finds the segmentation result that has the maximum matching probability with the model. From the resulting trajectories, the extractor finds the linearized part, and the original shape of the part is the most probable ligature (See Figure 8).

4 System Implementation

Recognition step is as following. For a given handwriting input, evaluate all the matching probabilities between the input and all the character models that the recognizer has. Return the label of the model that shows the maximum probability as recognition result.

Therefore, in order to recognize a character, we need a corresponding character model. As the number of characters to recognize and the number of writing styles increase, the number of character models increases. We need many data to train the models, but we have only small amount of data.

In this chapter, we propose two methods that solve the problem of lack of data.

4.1 Combination of Models

3D space handwriting trajectory except ligature has similar shape to 2D handwriting trajectory. Since there are already robust 2D handwriting models for large variation

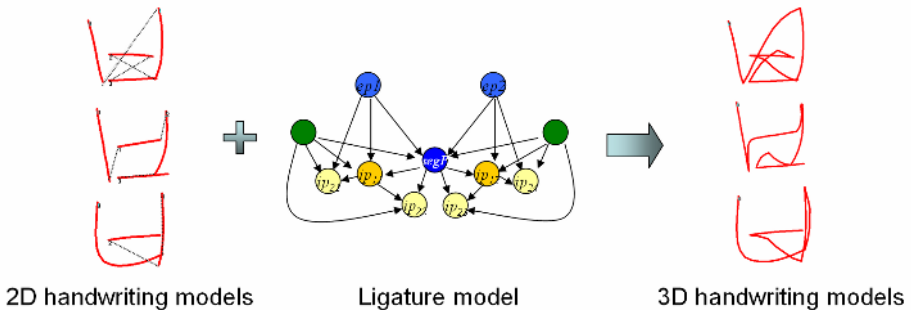


Fig. 9. Combination of models

and various writing styles, we can reduce the time and the efforts for constructing 3D space handwriting models by adding 3D space ligature models to existing 2D space handwriting models.

In Figure 9, we see that 2D models for various writing styles of first consonant ‘ H ’ in Hangul is combined with a ligature model to make 3D handwriting models. As you see, training data that we need to make various 3D models are only the 3D ligature data for the ligature model. However, there is one weak point that the 3D model except ligature part only reflects the shapes of 2D handwriting.

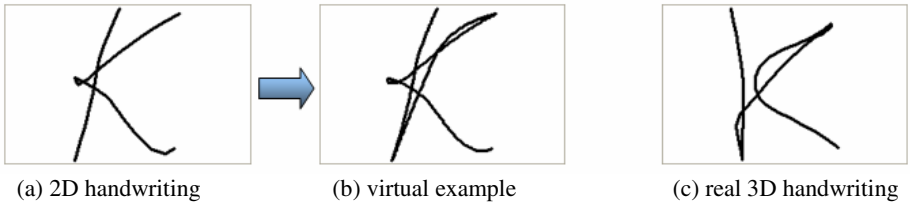


Fig. 10. Virtual example and real 3D handwriting

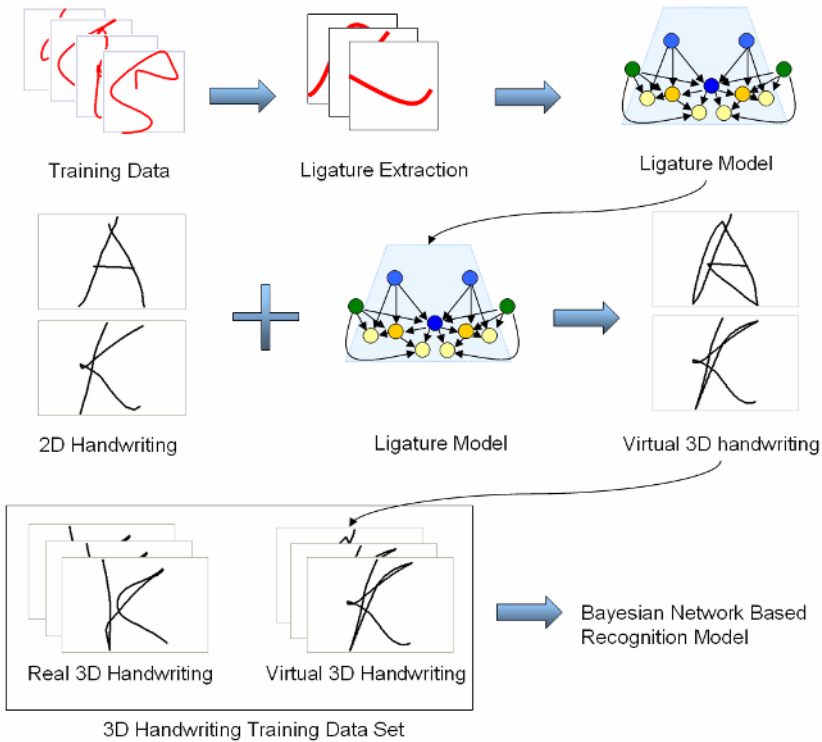


Fig. 11. Flow of the system using virtual examples

4.2 Use of Virtual Examples

Since the research on 2D space handwriting recognition has been done widely, we have much amount of 2D space handwriting data, and a large part of the data is available on the Internet. Since we know the 2D space handwriting and 3D space handwriting are quite similar, we can make enough data for training by making virtual 3D handwriting data from 2D handwriting.

It is possible to generate virtual 3D space handwriting by adding ligatures generated by the ligature model to existing 2D space handwriting examples. As you see in Figure 10, generated virtual 3D handwriting looks very similar to real 3D handwriting.

By using these virtual examples and real 3D space handwritings for training the recognizer, we can overcome the lack of data. The procedure is shown in Figure 11. In this method, 3D models reflect shapes of 3D handwriting as well as 2D handwriting.

5 Experiments

Experiments are performed with digits, alphabets, and Hangul. Digit and Hangul recognition system used combination of models, and alphabet recognition system used the virtual examples.

5.1 Digit Recognition

Digit data has 3,100 examples from 14 users. The data were divided into train data and test data with 2:1.

In the result, the total recognition rate was 95.4% (See Figure 12). As you see in Figure 13, the proposed system correctly recognize characters even if they have large variance. Errors in digit recognition were caused mainly two problems (See Figure 14). The first one is the ambiguity between ‘0’ and ‘6’, and between ‘1’ and ‘9’.

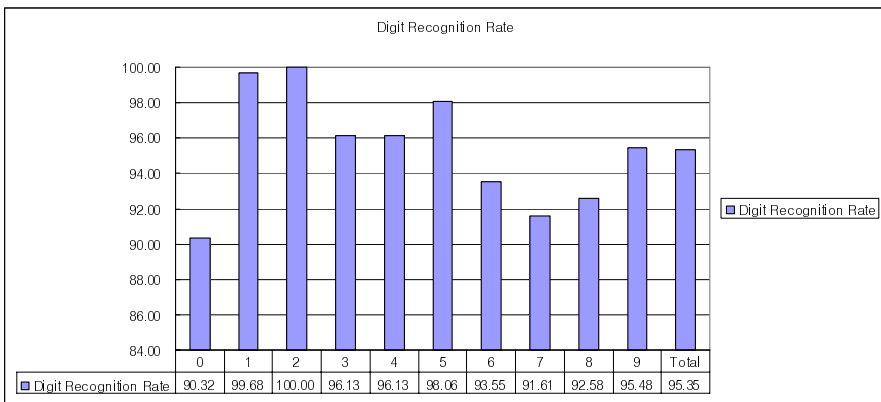


Fig. 12. Digit recognition result

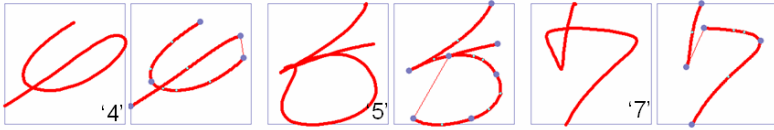


Fig. 13. Examples of correct recognition. In each pair, left one shows an input trajectory and its label. Right one is the shape after ligatures are detected and removed.

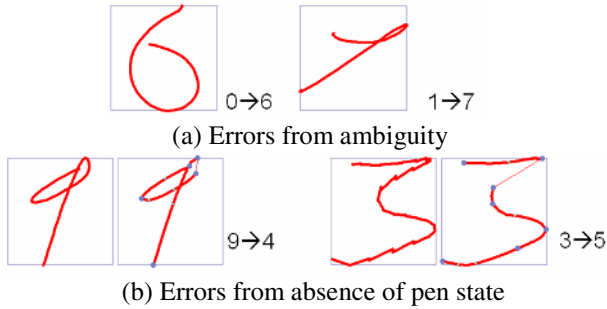


Fig. 14. Examples of errors in digit recognition

kind of problem is basic, and often observed even in 2D handwriting recognition. This ambiguous trajectory is hard to recognize even with human eyes. Second one is the ambiguity from the absence of pen-up/pen-down. According to where the ligature is assigned, the recognition result can be different.

5.2 Alphabet Recognition

In Alphabet recognition, we used real 3D handwriting and virtual 3D handwriting for training the recognizer. The data were composed of 2,600 capital letters and 2,600 small letters. The half of the data was used in modeling the ligature, and the rest was used in test. Though Alphabet is composed of 26 capital letters, and 26 small letters, we used 46 handwriting models for capital letter and 42 models for small letters in order to recognize the handwriting with various styles.

As a result, the recognizer showed 97.8% recognition rate for capital letters, and 93.8% for small letters. By increasing the number of the recognition candidates, the recognition rate reached over 99% (See Figure 15). We see that this system can be also used in word recognition by adopting a language model or word set.

Figure 16 shows the examples correctly recognized. Although the input traces have large variance, the system can recognize them correctly. Most of the errors are caused by the ambiguity from the absence of pen state. In Figure 17, 'K' and 'R' have a same writing order, and their shapes are so similar that even human eyes cannot determine whether the trace is 'K' or 'R'. Small letter 'i' shows same trace as that of 'l'. This ambiguity cannot be dissolved by shape models, but by using the context or using writing speed, this kind of ambiguity can be reduced.

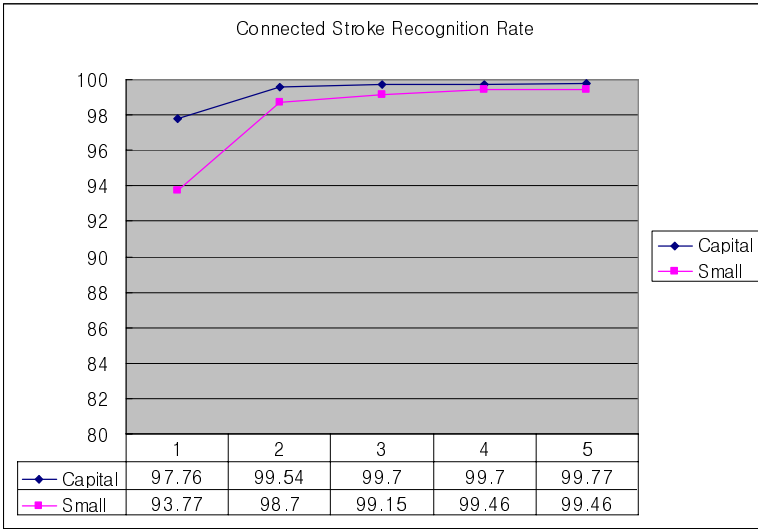


Fig. 15. Recognition rate of capital letters and small letters

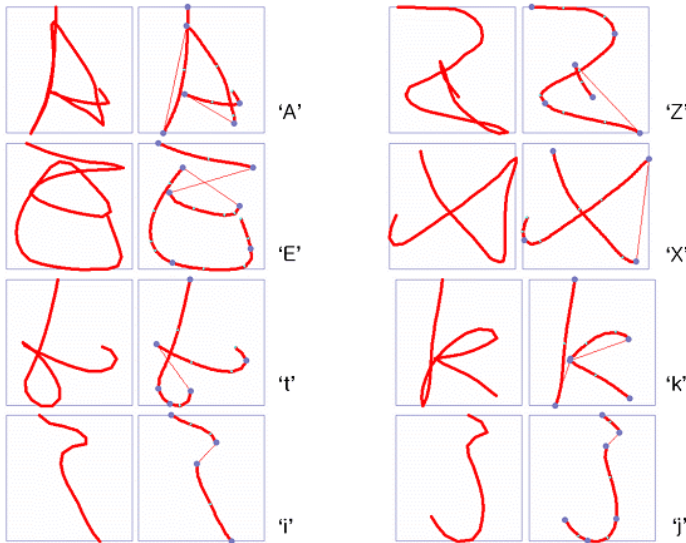


Fig. 16. Examples of correct recognition. The left one of each pair is an input trajectory, and the right one is the shape after ligatures are detected and removed.

5.3 Hangul Recognition

Since the number of Hangul characters is over ten thousand, making a handwriting model for each character needs many data to collect and many models to evaluate. Hangul is, however, composed of graphemes – first consonant, vowel, and last

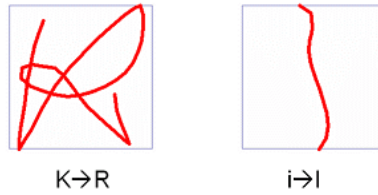


Fig. 17. Errors from the absence of the pen state

consonant – from its forming rule. By recognizing graphemes and by combining the recognition result, we can reduce the number of parameters of models. The state of the art of 2D space online Hangul recognizer adopts this grapheme recognizing policy [9] [11]. We implemented a Hangul character segmentator which segment a Hangul character into first consonant, a vowel, last consonant, and between-grapheme ligatures. We constructed a Bayesian network based model for each component using the model combination method explained in chapter 4.1 (See Figure 18).

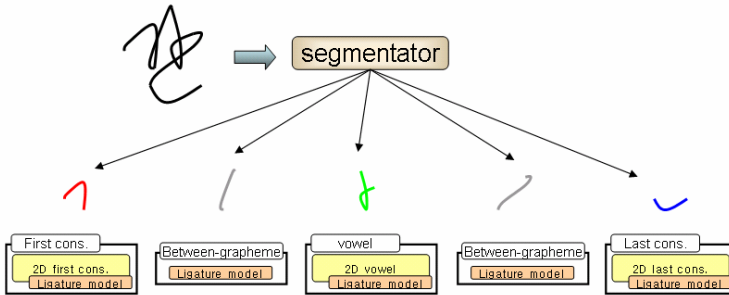


Fig. 18. Components of a Hangul character model

Hangul data has 3,600 examples from 9 users, and the data were divided into train data and test data with 2:1. Experiments are performed in two categories according to recognition targets. Recognition targets of first experiment were 900 best frequently used characters, and those of second experiment were 2,350 characters in ‘Wansung-hyung’ set.

Recognition rate was 78.6% and 64.0% for 900 characters and 2,350 characters respectively (See Figure 19). In Figure 20, we see that even for complex characters, the proposed system correctly finds grapheme boundaries and ligatures. Errors are mainly from two problems. First one is the error from segmentation. If character segmentator does not give correct grapheme boundary, Bayesian handwriting model cannot recognize it, and these errors are about a half of total errors. Second one is the error from ambiguity by absence of pen-up/pen-down. As you see in Figure 21, the first trajectory can be any one of the second trajectory or the third trajectory by assigning ligature. This kind of error will not be fixed by modeling since the trajectory is hard to recognize even with human eyes. In order to reduce this ambiguity, context information such as language model is required.

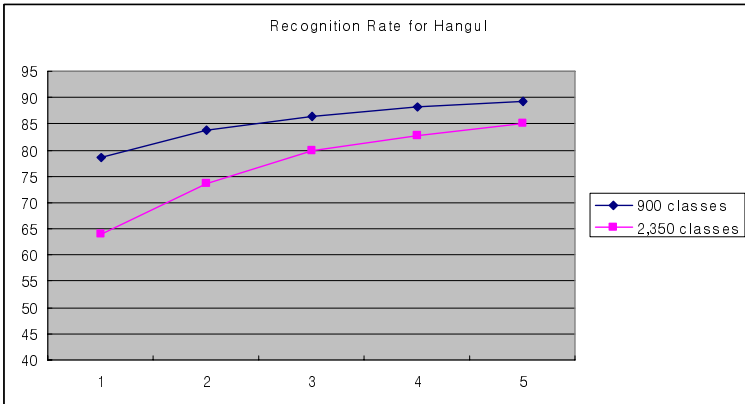


Fig. 19. 5 best recognition rate of Hangul recognition

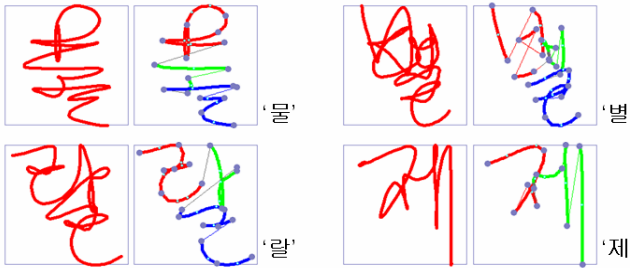


Fig. 20. Examples of correct recognition

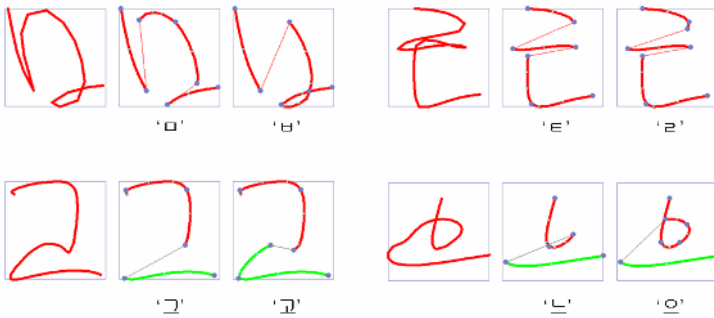


Fig. 21. Errors from absence of pen state. In each group, left most one is input trajectory. It can be either of middle and right most one.

6 Conclusion

3D space handwriting is a gesture-like character that a user writes in the air, and it is promising input method for its portability and ability of understanding complex and various inputs.

In this work, we proposed a 3D space handwriting recognition system by adapting 2D space handwriting models and data to 3D handwriting, based on that the most different part between 2D and 3D handwriting is the shape of ligature.

We designed ligature model not to depend on each character but to depend only on incoming and outgoing vectors. Therefore, with a few ligature models, various ligature shapes are modeled.

With the ligature model, we proposed two kinds of method of constructing 3D space handwriting character models. First one is combining 2D handwriting models and a 3D ligature model. It is useful when we have a robust 2D handwriting character recognizer. We can use existing models for various writing styles with only ligature modeling. Second one is using virtual 3D handwriting data in training. Virtual 3D data are generated by the ligature model and 2D handwriting data. This method is useful when we have much amount of 2D handwriting data. In either method, we can construct 3D handwriting character models with small amount of 3D data.

Recognition experiments are performed for digits, alphabets, and Hangul characters. The proposed system showed high recognition rates for digits, and alphabets. In Hangul recognition, many errors are from Hangul segmentation and various case of ambiguity. It is needed to use additional information such as writing speed and context information in order to solve ambiguity problem.

Acknowledgement

This research is partially supported by the Ubiquitous Computing and Network (UCN), one of the 21st century frontier R&D programs supported by the Ministry of Information and Communication (MIC) in Korea. The sub-project we participate in is "Agent Situation Awareness for Mobile Community Computing".

References

1. Hyeon-Kye Lee and Jun H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, October 1999
2. D. Reynaerts and H. Van Brussel, "Design of an Advanced Computer Writing Tool," *Proc. of the 6th Int. Symposium on Micro Machine and Human Science*, pp. 229-234, 1995
3. Won-Chul Bang, Wook Chang, Kyeon-Ho Kang, Eun-Seok Choi, Alexey Potanin and Dong-Yoon Kim, "Self-contained Spatial Input Device for Wearable Computers," *Proc. of the 7th IEEE Int. Symposium on Wearable Computers*, pp. 26-34, 2003
4. Jong K. Oh, Sung-jung Cho, Won-Chul Bang, Wook Chang, Euseok Choi, Jing Yang, Joonkee Cho, Dong Yoon Kim, "Inertial Sensor Based Recognition of 3-D Character Gesture with an Ensemble of Classifiers", *Proceedings of the 9th international Workshop on Frontiers in Handwriting Recognition*, 2004
5. Sung-jung Cho, Jong Koo Oh, Won-Chul Bang, Wook Chang, Eunseok Choi, Yang Jing, Joonkee Cho, Dong Yoon Kim, "Magic Wand: A Hand-Drawn Gesture Input Device in 3-D Space with Inertial Sensors", *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9)*, 2004
6. www.cyon.co.kr, "LG-SV360 / LG-KV3600", 2005

7. www.sec.co.kr, "SCH-S310", 2005
8. www.minfinity.co.kr, "3D Smart Input Device", 2005
9. Sung-Jung Cho and Jin H. Kim, "Bayesian Network Modeling of Strokes and Their Relationships for On-line Handwriting Recognition," *Pattern Recognition*, vol. 37, no.2, pp. 253-264, 2004
10. Kevin P. Murphy, "Inference and Learning in Hybrid Bayesian Networks", Report No. UCB/CSD-98-990, 1998
11. Bong-Kee Sin, Jin H. Kim, "Ligature Modeling for Online Cursive Script Recognition", *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 19, no. 6, pp.623-633, 1997