# Ligature Modeling for Recognition of Characters Written in 3D Space

Dae Hwan Kim, Jin Hyung Kim

# Ligature Modeling for Recognition of Characters Written in 3D Space

*Dae Hwan Kim*
Artificial Intelligence and Pattern Recognition Lab.
KAIST, Daejeon,
South Korea
kimdh@ai.kaist.ac.kr

*Jin Hyung Kim*
Artificial Intelligence and Pattern Recognition Lab.
KAIST, Daejeon,
South Korea
jkim@ai.kaist.ac.kr

## Abstract

*In this work, we propose a 3D space handwriting recognition system by combining 2D space handwriting models and 3D space ligature models based on that the most different parts between 2D space handwriting characters and 3D space handwriting characters are the shapes of ligatures. We design the ligature model not dependent on each character but dependent on the incoming and the outgoing vector. Therefore with a few ligature models, various ligature shapes are modeled. Using 2D space handwriting models, we could use many models for various writing styles without training.*

**Keywords**: ligature model, 3D space handwriting, online handwriting recognition

## 1. Introduction

Recognition of characters written in 3D space is recognizing gesture-like characters written in the air with the device that gives a trajectory of hand movement. The input device has accelerators and gyroscopes which are used in estimating the trajectory.

Compared to 2D space handwriting characters i.e. handwriting characters with pen and tablet, 3D space handwriting characters need no writing surface. Therefore users can write more freely in the air, and the device can be designed in small size and with little weight since it needs only small pen-type sensing device. Therefore, a 3D handwriting input can be a promising input method for the portable devices.

A 3D space handwriting character, however, shows different feature in the shape of the trajectory – it has no pen-up/pen-down concept. The natural shape of 3D space handwriting character is the trajectory in the form of a connected stroke. You can see the shape difference between the 2D space handwriting character and the 3D space handwriting character in Figure 1.

In order to recognize 3D space handwriting characters, Oh et al. and Cho et al. defined thirteen recognition targets[1][2]. The target trajectory was, so called, graffiti which is in the form of a uni-stroke, and has a similar shape to a corresponding character. By restricting the targets, the recognition problem was simplified. This policy, however, forced users to use pre-defined shape of character while it showed high recognition performance. Moreover when a user writes multiple stroke character such as '4', the user has to write a new shape which is predefined in a uni-stroke and which he/she has never seen.

In this paper, we propose a connected stroke character recognition system in order to recognize 3D space handwriting characters. Based on that most of the shape differences between 2D space handwriting characters and 3D space handwriting characters are in the shapes of ligatures which are the connecting trajectories between strokes, we efficiently develop the recognition system by combining existing 2D space handwriting models and the ligature models of 3D space handwriting characters.

The rest of this paper is organized as following. We describe a ligature model in chapter 2, and the combination of 2D space handwriting models and 3D space ligature models in chapter 3. In chapter 4, we show the entire map of the proposed system. In chapter 5, we show the extension of the system for Hangul recognition. In chapter 6, we evaluate the performance of our system with experiments on digit and Hangul. Finally in chapter 7, we conclude the work, and discuss the future work.

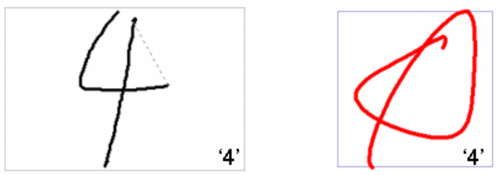



**Figure 1.** 2D space handwriting and 3D space handwriting

## 2. Ligature Model

### 2.1. Need of Ligature Model

A ligature is a trajectory drawn in moving from a previous stroke to a next stroke. A ligature is the most different part between the 2D space handwriting and the 3D space handwriting. As we see in Figure 1, in the 2D space handwriting, the ligature is usually drawn with a pen-up, and therefore it is relatively easy to find the ligature part from the entire trajectory of the character. Since the device cannot track the movement of pen when the pen is up, a ligature is assumed to be straight line connecting the end point of the previous stroke and the start point of the next stroke. This assumption ignores the detail shape of ligature. On the other hand, in the 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 the trajectory information including strokes and ligatures.

The shapes of 3D space handwriting trajectories except for ligatures are similar to those of 2D space handwriting trajectories. Therefore by modeling the ligatures of 3D space handwriting, we can efficiently adapt 2D space handwriting models to 3D space handwriting characters.

## 2.2. Characteristics of Ligature Shape

The shape of ligature has two characteristics.

First, the shape of ligature depends not upon each character shape but upon the incoming vector and the outgoing vector of the ligature. As you see in Figure 2, the ligature of the first consonant ‘ㄷ’ of Hangul – the Korean alphabet – has the similar ligature shape of vowel ‘ㅜ’. Although characters are different, the incoming vector and the outgoing vector of ligature are almost same, and this makes similar ligature shapes.

Second, as you see in Figure 3, a ligature is composed of two strokes which are nearly straight. We can assume that there is a boundary which divides a ligature into two strokes, and the boundary is displayed in Figure 3 as one point. From now on, we will call this point as a segmentation point. The shape of first stroke of ligature, that is, stroke from the start point of ligature to the segmentation point, depends more on the incoming vector than outgoing vector, and the shape of the second stroke depends more on the outgoing vector.

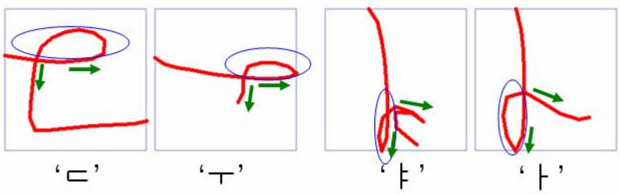



**Figure 2.** Ligature shape according to incoming vector and outgoing vector

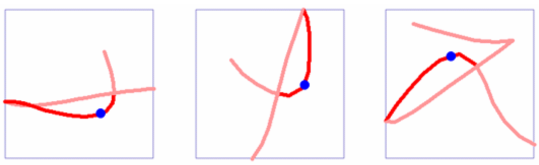


**Figure 3.** Ligature is composed of two nearly linear strokes

## 2.3. Baseline Model – Bayesian Network Based Stroke Model

The basic idea of ligature model is based on the stroke model in Bayesian network framework [3]. In this model, a stroke is assumed to be a set of points, and the stroke model is represented by point distributions approximated by Gaussian distribution, and their relationship which is represented in Bayesian networks. Since it explicitly models points, strokes, and their relationship, it shows higher performance than other handwriting recognition algorithms such as Hidden Markov Models and Neural Networks [3].

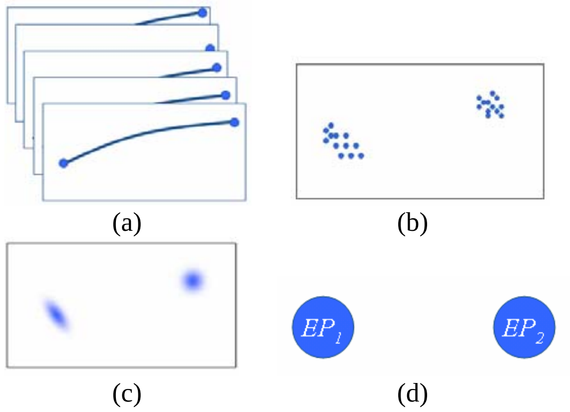



**Figure 4.** Modeling of stroke end points

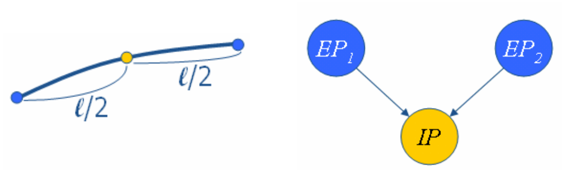



**Figure 5.** A model of a middle point. $\ell$ is the stroke length.

Figure 4 (a) shows various stroke samples of a stroke, and their end points are displayed with points. Figure 4 (b) shows the scatter plot of positions of end points, and Figure 4 (c) shows Gaussian distributions approximating the scatter plot. The point distribution is displayed with one node as random variable in the Bayesian network structure.

The middle point of a stroke can be represented in the same way. The distribution of middle point, however, depends on positions of two end points. This relationship can be represented with an arc in Bayesian network structure as you see in Figure 5.

For representing dependency, it is impossible to find all distribution tables corresponding to each value of the position since the position of point has continuous values. However by adopting conditional Gaussian distribution [4], we can solve this problem. In conditional Gaussian distribution, we assume that the mean of the distribution is linear combination of a constant and the values of dependent variable, but the covariance does not depend on other values. For example, the matching probability of the middle point in Figure 5 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 $EP_2$, and $\mathbf{W}$ coefficient matrix

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

As above, we get the point distribution by finding a coefficient matrix **W** and a covariance matrix $\Sigma$.

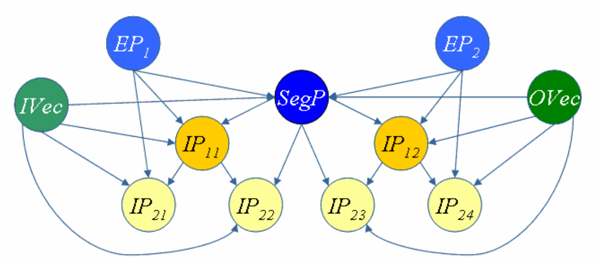
A stroke model is defined as joint distribution of point distributions. In Figure 5, matching probability of the stroke is as follows.

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

### 2.4. Structure of Ligature Model

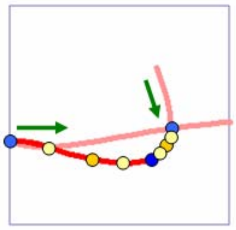
Based on the characteristics of ligature shape, we form the Bayesian network structure of the ligature as Figure 6 (a). Figure 6 (b) shows points corresponding nodes in the structure in vowel 'ㅗ'. A ligature model composed of 9 point distributions and the incoming/outgoing vector distributions. The first stroke which is from start point $EP_1$ to segmentation point *SegP* and the second stroke from *SegP* to $EP_2$ have symmetric structure. The distributions of the middle points *IP*s in each stroke depend on their end points. The distributions of all middle points in the first stroke and the segmentation point depend on the starting point $EP_1$ and the incoming vector *IVec*, and this represents the fact that the shape of first stroke depends on the incoming vector. The distributions of middle points in the second stroke and the segmentation point depend on the end point $EP_2$ and the outgoing vector *OVec*, and this reflects that the shape of the second stroke depends on the outgoing vector.

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




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

(a) Bayesian network representation



(b) corresponding points in vowel 'ㅗ'

**Figure 6.** Structure of ligature model

## 3. Combination of 2D Space Handwriting Model and 3D Space Ligature Model

As you see in Figure 1, the shape of the 3D space handwriting trajectory except for the ligature part has the similar shape to the 2D handwriting trajectory. Since there is already much data collected for 2D space handwriting characters and there exist robust handwriting models for a large variation and various writing styles, we can reduce time and efforts for constructing 3D space handwriting models by adding 3D space ligature models to existing 2D space handwriting models.

We use a Bayesian network based character model as a 2D space handwriting model. This model is robust against writing variation since it explicitly models the relationship between strokes. Since it represents likelihood as a probability and our ligature model also gives a probability, it is easy to combine two models.

In order to add a ligature model to a 2D space handwriting model, it is necessary to define an incoming vector and an outgoing vector since there are many candidates for these vectors. By evaluating the correlation coefficient of each candidate and points in the ligature, we choose vectors which are most related to ligature shape. From experiments, we chose the vector from the middle point of previous stroke to the starting point of the ligature as the incoming vector and the vector from the end point of the ligature to the middle point of the next stroke as the outgoing vector.

In order to find the best position where the 3D ligature model is added, the system finds all possible segmentation results. However, by restricting the number of segmentation boundaries, we can overcome the time complexity.

## 4. System Implementation

The flow of the proposed system can be summarized by Figure 7. First, ligatures are extracted from connected input data, and with this data, ligature models are trained. 3D space handwriting models are formed by adding ligature models to 2D space handwriting models. In the recognition step, the matching probabilities between input trace and all the models the recognizer has are evaluated, and return the label of model which shows the maximum probability as the recognition result.

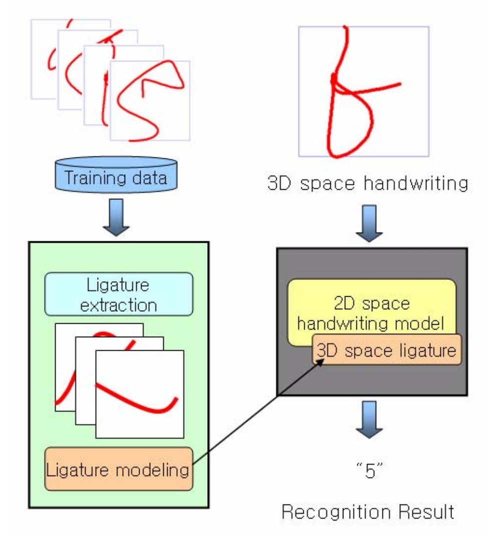



**Figure 7.** System flow

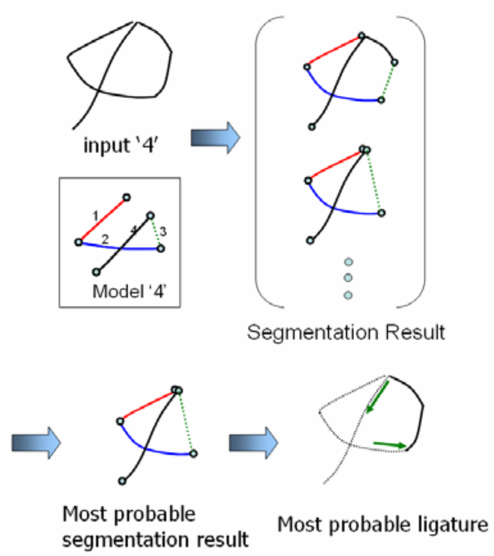



**Figure 8.** Ligature extraction

### 4.1. Ligature Extraction

Ligatures are extracted from input trajectories. This step is data collection for ligature models. Ligature extraction is performed with algorithm, not with hand, by finding the parts which are most likely to be ligatures by using Bayesian network based 2D space handwriting recognizer.

We can see the procedure in Figure 8. When a connected trajectory is inputted, the ligature extractor finds all possible segmentation results. The number of strokes in each segmentation result is the number of strokes of the 2D handwriting model that corresponds to the label of the input trajectory. For each segmentation result, the ligature extractor finds the stroke that corresponds to the ligature in the model, and makes the stroke straight. Making straight line is ignoring the detail shape of the ligature and making the input trajectory adapted to the 2D space handwriting model. For all trajectories which are adapted to the 2D handwriting, the ligature extractor finds the most probable segmentation result which has the maximum matching probability with the corresponding model. From the resulting trajectory, the original shape of the linearized part is the most probable ligature.

### 4.2. Ligature Model Evaluation

A matching probability of ligature model is a joint probability of point matching probabilities. Among all the points of the ligature, the coordinate value of the segmentation point is not unique whereas the other points such as middle points and end points are fixed when the end points of ligature and the segmentation point are determined. Therefore, we assign the position of the segmentation point to the point which makes a maximum stroke matching probability. Given that the incoming/outgoing vectors and the end points of the ligature are observed, a matching probability between ligature model and ligature instance is as follows.

$$\begin{aligned} & P(S = O(1,\dots,T)) \qquad (3) \\ & = P(EP_1 = O(1),\dots,IP_i = O(ip_i),\dots,EP_2 = O(T)) \\ & = \max_{\text{all possible } segP\text{'s}} \{P(EP_1 = O(1))P(EP_2 = O(T)) \\ & \quad P(SegP = O(segp) \mid Pa(SegP)) \prod_{\text{all } i\text{'s}} P(IP_i = O(ip_i) \mid Pa(IP_i))\} \end{aligned}$$

Here, $Pa(X)$ means parents of random variable $X$.

### 4.3. Training Ligature

In order to find the parameters of a ligature model, we should find those of point model. As we mentioned earlier, the parameters of point model is the coefficient matrix $\mathbf{W}$ and the covariance matrix $\Sigma$. These parameters are calculated by maximum likelihood estimation. Although the positions of all the points should be given in order to use maximum likelihood estimation, the position of segmentation point is not determined, and it is difficult to find the segmentation point of all input data by an algorithm. By estimating parameters similar to EM (Expectation and Maximization) method, this kind of problem can be solved.

## 5. System Extension for Hangul Recognition

### 5.1. Need of Character Segmentator

Since the number of Hangul characters (Korean characters) is over ten thousand, making handwriting models for each character needs much data to collect and needs many model to evaluate. Hangul is, however, composed of graphemes – first consonant, a vowel, and last consonant – from the forming rule. By recognizing graphemes and combining the recognition result, we can reduce the number of parameters of models. The state of the art of the 2D space online Hangul recognizer adopts this grapheme recognizing policy [3] [5].

A Hangul character in the form of connected stroke is segmented into first consonant, a vowel, last consonant, and between-grapheme ligatures as you see in Figure 9. Each component recognizer is formed with 3D space handwriting models which are explained in chapter 4. The main problem of Hangul recognition is how to segment input connected stroke into graphemes and between-grapheme ligature. Finding all possible segmentation results costs extremely high. Therefore we need a character segmentator which gives relatively small number of segmentation results, and guarantees a correct segmentation should exist in the candidates.

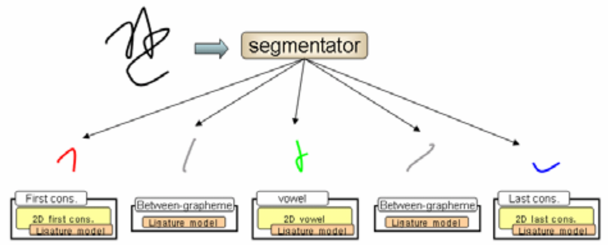



**Figure 9.** Hangul recognition system

## 5.2. HMM-Based Character Segmentator

We use the recognition result from HMM-based recognizer for character-to-grapheme segmentation. A HMM-based recognizer uses direction code as feature. In order to improve Hangul recognition performance we adopted BongNet [5] which explains how graphemes can be combined to one Hangul character. For finding the segmentation result, the recognition label is not used, but only grapheme boundaries of recognition result were used.

The model structure of a HMM is automatically formed. Since one stroke in handwriting character is nearly straight, it can be one state in a HMM. We assigned one stroke in the Bayesian network model to one state in the HMM, and one ligature to two states in the HMM as we assumed that ligature is composed of two strokes. Figure 10 shows correspondence between consonant 'ㅅ' and its HMM structure.

The HMMs of Hangul components are trained separately. The data of each component are collected by manual segmentation from the Hangul characters

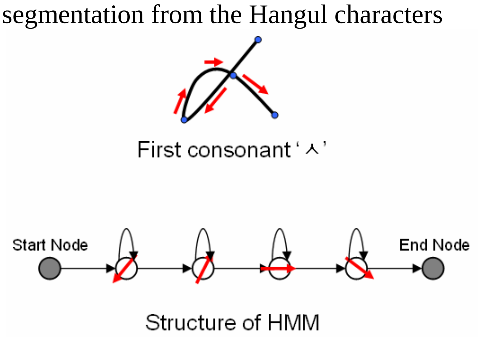



**Figure 10.** A HMM structure according to strokes of character

# 6. Experiments

Experiments are performed with digit and Hangul.

## 6.1. Digit Recognition

Digit data have 2,100 examples from 14 users. The recognition rate is calculated from the 3-fold cross test. For comparison, the recognition rate of the proposed system and the 2D space handwriting recognizer which does not have any 3D space ligature model are evaluated.

The recognition result is summarized in Figure 11 and Table 1. The proposed system showed 24% of error reduction rate, and total recognition was 93.27%. As our expectation, digits that has ligature such as '4', '5', and '6' showed high error reduction rate. However, there was small reduction of recognition rate in '3' and '9'. As you see in Figure 12, the proposed system correctly recognize characters even if they have large variance. Errors in digit recognition were caused mainly by two problems. The first one is the ambiguity between '0' and '6', and between '1' and '9'. This is an avoidable problem and we can also find this problem in the 2D space handwriting recognition. This ambiguous trajectory is even hard for human to recognize. Second one is the ambiguity from the absence of pen-up/pen-down (Figure 13). According to where the ligature is assigned, the recognition result can be different.

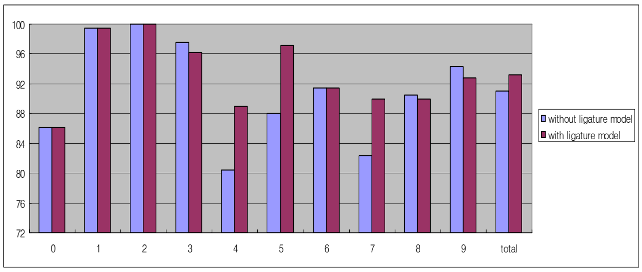



**Figure 11.** Digit recognition result.

**Table 1.** Error reduction rate by ligature modeling

| | Error reduction rate (%) |
|---|---|
| Total | 24.5% |
| Digit '4' | 43.9% |
| Digit '5' | 76.0% |
| Digit '7' | 43.3% |


**Figure 12.** Examples of correct recognition

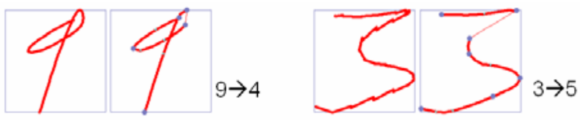



**Figure 13.** Errors from absence of pen state

## 6.2. Hangul Recognition

Hangul data have 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 target size. The Recognition target of the first experiment is 900 best frequently used characters, and that of the second experiment is 2,350 characters in Hangul 'Wansung-hyung' set which covers most of Hangul characters in daily use.

The result showed that the proposed system reduces 45% of errors compared to the 2D space handwriting recognition. Recognition rate was 79% and 64% for 900 characters and 2,350 characters respectively. In Figure 14, we see that even for a complex character, proposed system correctly finds grapheme boundaries and ligatures. Errors are mainly from two problems. First one is error in the segmentation. If the character segmentator does not give correct grapheme boundaries, Bayesian handwriting model cannot recognize the character. These errors are about a half of total errors. Second one is the error from the ambiguity by the absence of pen-up/pen-down. As you see in Figure 15, the left-most trajectory can be any one of the middle trajectory or the right-most trajectory, corresponding to where the ligature is assigned. This kind of errors will not be fixed only by modeling since the trajectory is hard for human to recognize. In order to reduce this ambiguity, a context such as a language model is required. Figure 16, shows 10 best recognition rate when the targets are 900 characters, and we see that the number of errors from ambiguity decreases as the number of candidates increase, but still 8% is not correctly recognized – most of this errors are from segmentation errors.

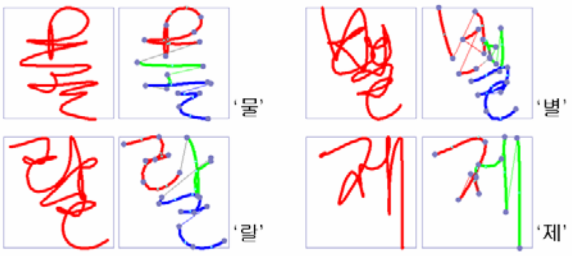



**Figure 14.** Examples of correct recognition

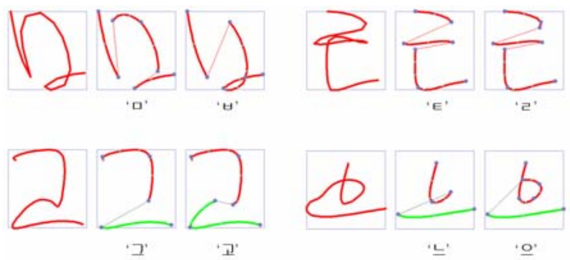


**Figure 15.** Errors from absence of pen state

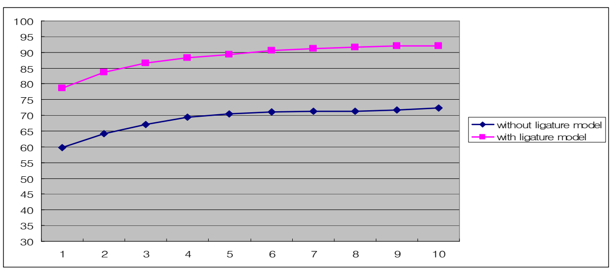



**Figure 16.** 10 best recognition rate. The number of recognition targets is 900.

## 7. Conclusion

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

We designed ligature models not dependent on each character shape but dependent on the incoming and the outgoing vector. Therefore with a few ligature models, various ligature shapes are modeled. Using 2D space handwriting models, we could use many models for various writing styles without training.

The Results showed that the proposed system highly reduces error compared to the 2D space handwriting recognizer. However, it has limitation for practical uses, since there are many hard examples to the extent that even human cannot recognize it. However, the area where 3D space handwriting is applied is not likely to be a word processor, but to be a short command input. In this situation, since the number of targets is small, a strong language model can be applied. Therefore we can achieve high recognition rate.

For Hangul recognition, a half of errors are from the segmentation. Therefore, by reducing this kind of errors, we will see a large improvement. Since HMM use only local information, we can expect improvement by using duration modeling.

## References

[1] 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

[2] 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

[3] 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

[4] Kevin P. Murphy, "Inference and Learning in Hybrid Bayesian Networks", Report No. UCB/CSD-98-990, 1998

[5] 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

[6] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. Of the IEEE, vol. 77, no. 2, pp.257-286, 1989