키워드 인식을 위한 한글 Pseudo 2D HMM의 동적 합성 방법

정회원 조 범 준*

Dynamic Synthesis of Pseudo 2D HMMs for Korean Characters in Key Character Recognition Tasks

Beom-Joon Cho* *Regular Member*

요 약

한글은 둘 또는 세 개의 자모가 사각형 영역 안에 적절히 배치된 구조로 되어있다. 이와 같은 구성 방법에 따라 글자의 영상을 합성하고 이를 실시간에 Pseudo 2D HMM으로 변환하는 방법을 제안한다. 본 방법에 따라 실시간 합성된 모델과 추가의 필러(filler) 모델, 여백 모델을 문서 영상의 글자 영역에서 핵심어 검출에 적용하였다. 실험 결과 최소한의 설계 변수 조정으로도 오검출, 미검출률이 낮고 언어 모델 없이 숫자 89%, 한글 80%의 검출 성능을 보였으며, 따라서 제안된 방법이 인쇄 문자 패턴의 실시간 모델링 및 키워드 검출에 효과가 있음을 보였다. 본 연구 결과는 내용 기반의 광학 문서 색인 등에 활용할 수 있다.

ABSTRACT

A Korean Hangul character is composed of two or three component letters with a spatial arrangement of fitting them into a box. Based on the similar composition of letter image templates, we propose a new method of synthesizing a character image template and then converting it into a P2DHMM in real time. The realtime models synthesized in this way have been applied to a key character spotting task with additional filler models and a space model. In the experiment with minimal tuning of design parameters within and across models, we observed the performance of 89% and 80% for digit and Hangul spotting without a language model with low rate of false positives and negatives, which confirms that the proposed method is effective for the real time modeling and spotting of machine printed character patterns. The result of the research can be applied to content-based retrieval of optical documents.

I. Introduction

It is rarely argued against that neural network is a good model for the recognition of machine printed characters. However, one problem with the neural network is that the sequential nature of texts running left to right is not well captured without sophisticated network architectures like that of TDNN [1]. As a result, most of the neural network systems with ordinary architectures assume external segmentation of character blocks prior to recognition. In this case the overall system performance is usually limited by the segmenter's performance and quality of the

* School of computer Engineering, chosun university(bjcho@mail.chosun.ac.kr)

논문번호 : K01089-0228, 접수일자 : 2001년 2월 28일

※ This paper was accomplished with Research Fund provided by Korea Research Foundation, Support for faculty Research Abroad

segments.

Another good tool for modeling sequential signal is hidden Markov model or HMM. HMM is an equally famous tool employed in diverse areas including speech recognition and on-line handwriting [2]. The application of HMMs benefit from the wide range of experience accumulated in speech recognition and many other fields. Since characters in texts run sequentially and, mostly, left to right, the HMM can effectively be used for modeling the variability of shape and size of text images. This research is focused on the application of the HMM methodology in the text analysis in 2D optical documents, and deals with real time generation of Korean Hangul character models for spotting key characters in the content analysis of optical documents.

Printed characters in documents are two-dimensional. Therefore it is natural to believe that 2D HMMs will be helpful for analyzing 2D character patterns. Since characters are of two dimension by nature, it is natural to believe that 2D HMMs will offer great potential for character recognition problems. But a fully connected 2D HMM leads to an algorithm of exponential complexity [3]. To avoid the problem, the connectivity of the network has been reduced in several ways, two among which are Markov random field and its variants [4] and pseudo 2D HMM [5,6,7]. The latter model is a very simple and efficient 2D model that retains all of the useful HMM features. The basic idea of this paper is about the real time construction of the pseudo 2D HMM for composite Korean characters.

The rest of the paper consists as follows. In Section 2 we will briefly review the HMM. In Section 3 the pseudo 2D HMM and its algorithm, and then a procedure for developing character models are discussed. Section 4 describes the additional models for spotting task. Section 5 presents results from preliminary experiments. Section 6 concludes the paper.

Ⅱ. Hidden Markov Model

The hidden Markov model is a doubly stochastic process that can be described by three sets of probabilistic parameters as λ = (A, B, π). Given a set of N states and a set V of observable symbols, the parameters are formally defined by [2]:

- Transition probability distribution: $A = \{ a_{ij}, = p(q_t = j \mid q_{t-1} = i), 1 \leq i, j \leq N \}$.
  $\sum_j a_{ij} = 1$.
- Output probability distribution: $B = \{ b_i(v) = p(x_t = v \mid q_t = i), 1 \leq i \leq N, v \in V \}$.
  $\sum_v b_i(v) = 1$.
- Initial transition probability distribution: $\pi = \{\pi_i, = p(q_1 = i), 1 \leq i \leq N \}$.
  $\sum_i \pi_i = 1$.

The most frequent task with an HMM is the evaluation of the model matching score given an input sequence $X = x_1 \; x_2 \; ... \; x_T$. It is given by the likelihood function of the sequence generated from the model

$$P(X \mid \lambda) = \sum_{Q} \pi_{q_1} b_{q_1}(x_q) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(x_t) \tag{1}$$

Although simple in form, the time requirement is exponential. With the use of the DP technique, this can be computed in linear time in T. However when it comes to 2D HMM formulation, even the DP technique alone is not enough. One research direction is the structural simplification of the model, and the pseudo 2D HMM is one solution.

Ⅲ. Pseudo-2D HMM Construction

3.1 Description

Pseudo 2D HMM in this paper is realized as a horizontal connection of vertical sub-HMMs (λ_k). But it is not the only one. The alternative realization is the vertical connection of horizontal sub-HMMs as in the work of Xu and Nagy [8]. In order to implement a continuous forward search method, the former type has been used in this research.

Let X_t = x_{1t} x_{2t} ... x_{St}. This is a one-dimension sequence like that of X in equation (1). This is modeled by a sub-HMM $_k$ with the likelihood $P(X_t|_k)$. You may regard each sub-HMM r_t as a super-state whose observation is a vertical frame of pixels.

$$P_{r_t}(X_t \mid \lambda_{r_t}) = \sum_Q \pi_{q_1} b_{q_1}(x_{1t}) \prod_{s=2}^{S} a_{q_{t-1}q_t} b_{q_t}(x_{st}) \tag{2}$$

Now consider that we are given a bitmap image which we define as a sequence of such vertical frames as X = X_1 X_2 ... X_T. Each frame will be modeled by a super-state or a sub-HMM. Let be a sequential concatenation of sub-HMMs. Then the evaluation of given the sample image X is

$$P(X \mid \Lambda) = \sum_R P_1(X_1) \prod_{t=2}^{T} a_{r_{t-1}r_t} P_{r_t}(X_t) \tag{3}$$

where it is assumed that super-state process starts only from the first state. The P_{r_t} function is the super-state likelihood. Note that both of the above equations can be calculated efficiently using the Viterbi algorithm.

One of the immediate goal of the Viterbi search is the calculation of the matching likelihood score between X and an HMM. The objective function for an HMM λ_k is defined by the maximum likelihood as

$$\Delta(X_t, \lambda_k) = \max_Q \prod_{s=1}^{S} a_{q_{t-1}q_t} b_{q_t}(x_{st}) \tag{4}$$

where $Q = q_1 q_2 \Lambda\ q_S$ is a sequence of states of λ_k, and $a_{q_0 q_1} = \pi_{q_1}$. $\Delta(X_t, \lambda_k)$ is the similarity score between two sequences of different length. The basic idea behind the efficiency of DP computation lies in formulating the expression into a recursive form

$$\delta_s^k(j) = \max_i \delta_{s-1}^k(i) a_{ij}^k b_j^k(x_{st}),$$

$$j = 1, ..., M_k,\ s = 1, ..., S,\quad k = 1, ...\ K \tag{5}$$

where $\delta_s^k(j)$ denotes the probability of observing the partial sequence x_{1t} ... x_{st} in model k along the best state sequence reaching the state j at time/step s. Note that

$$\Delta(X_t, \lambda_k) = \delta_S^k(N_k) \tag{6}$$

where N_k is the final state of the state sequence. The above recursion constitutes the DP in the lower level structure of the P2DHMM. The remaining DP in the upper level of the network is similarly defined by

$$D(X, \Lambda) = \max_k \prod_{t=1}^{T} \vec{a}_{r_{t-1}r_t} \Delta(X_t, \lambda_{r_t}) \tag{7}$$

that can similarly be reformulated into a recursive form. Here $\vec{a}_{r_1 r_2}$ denotes the probability of transition from super-states r_1 to r_2. According to the formulation described thus far, a P2DHMM adds only one parameter set for super-state transitions to the conventional HMM parameter sets.

3.2 Design of Character Models

One of the most important tasks in hidden Markov modeling is estimating the probabilistic parameters. For this task it is assumed that we are given a set of typical samples of character images X = { $X^{(1)}$, ... , $X^{(D)}$ } of an equal dimension.

The focus of the section is the construction of the P2DHMM for a Korean Hangul character. Each character comprised in two or three letters of phonetic consonants and vowels. The combination follows a general rule to fit the letters into a rectangle. There are six types of combination according to the shape of the vowel letter (horizontal, vertical, or both) and the presence of consonant suffix. The proposed method of model creation is based on the given set of bitmap images. The overall procedure is shown in Figure 1, and explained as follows:

(1) Letter segmentation. This step involves extracting the individual letters of character

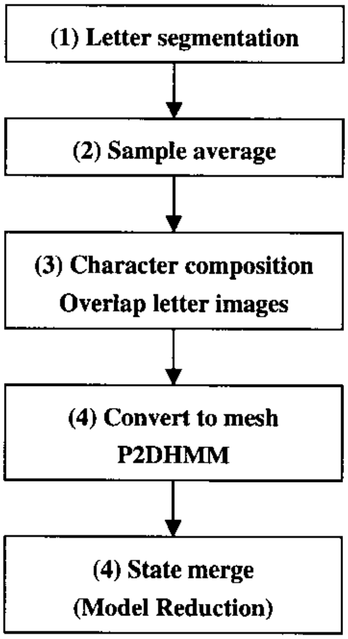

Fig. 1 Design of a Character Model

samples in the context of the box enclosing the character. As illustrated in Figure 1, the letters are separated while retaining the position with the box.

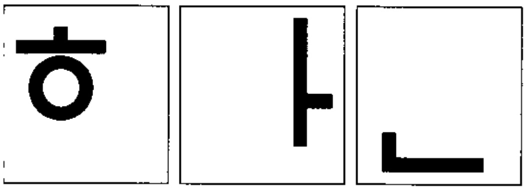

Fig. 2 Korean letters separated out from a syllabic character '한'. From left to right: the initial consonant, the vowel, and the suffix consonant. Note that the letter position information is retained.

In its simplest form this step is the most costly in the proposed method. But the problem can be avoided by using a bootstrapping strategy or a little more sophisticated prototyping idea [8].

(2) Average the sample extracts. Now there is a set of letter samples. First we classify the samples according to the type of the letter arrangement pattern of the original character. For the initial consonant letter there are six types, and two for each vowel letter. Then, take the sample average of the set of categorized images pixel by pixel so that a smooth grayscale-like image is obtained. Assuming binary samples, the average intensity of the (i, j)-th pixel is

$$\bar{x}_{ij} = \frac{N_{ij}}{N}$$

where N_{ij} is the number of samples whose (i, j) pixel is black (or white) and N is the total number of samples. Essentially the training phase is finished.

(3) Overlap the letter images. From this step on the process belongs to the decoding or recognition phase, and is performed in real time. Here the given task is to spot or recognize a character. The image template of the character is synthesized in the image domain by overlapping the component letter images generated in the previous step.

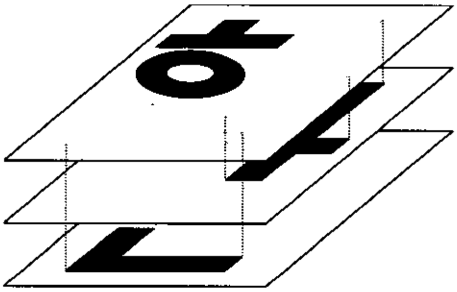

Fig. 3 Overlapping letter image templates

(4) Conversion to P2DHMM. Given a character image, it is straightforward to construct a P2DHMM. First assign a state to every pixel with the output probability being the intensity value. Then the states are linked according to the topological constraint of P2DHMM: vertical sub-state transition, and then horizontal transition between super-states. Note that all the transition probabilities are one without self-transitions. There is no space-warping in the current model.

(5) State merge. When two or more successive states are similar in the output probability (gray

scale), they are replaced with a new node with a modified output probability

$$\bar{x}_{i'j} = w_i x_{ij} + w_{i+1} n_{i+1,j}$$

where w_i is the weight as a function of the duration in the state i. The state similarity is measured by the output probability of the states. In the case of super-states, the distance measure is

$$d(x, n_k) = |x - n_k|^{\alpha}$$

where $\alpha \in R$. If $\alpha = 2$, then this measures the dissimilarity in the least square sense. Then we estimate the transition probability similarly, or the whole transition parameter set may be replaced by state duration probabilities.

Ⅲ. Keyword Spotting Model

For keyword spotting task, we developed two more types of P2DHMMs in addition to key character models, and then combined then into a network model continuous decoding of input streams.

4.1 Filler model

In keyword spotting task, a filler corresponds to any non-key characters. In this paper the filler is defined as the model for all characters including the key characters. For better thresholding capability in Korean Hangul characters, we defined six fillers, one for each of the six character composition types. Figure 3 shows the filler images before conversion to P2DHMMs.

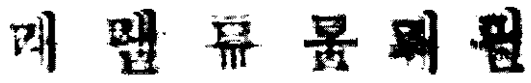

Fig. 4 Bitmap images for filler models.

4.2 Space model

The region excluding the text is white space. The white space will be limited to the white frames between characters. It is modeled with a small number of nodes. Actually the state merge step reduced the nodes to one or two most of the time in practice.

4.3 Spotting network

For character spotting task we have designed a network-based transcription model. It is a circular digraph with a backward link via the space model so that it can model arbitrary long sequence of non-key as well as key patterns. Given such a network, an input sequence of will be aligned to every possible path circulating the network. One circulation is called a level. An l level path hypothesis represents a string of l characters [9]. The result is retrieved from the best hypothesis.

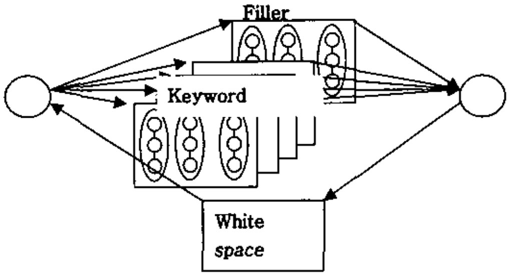

Fig. 5 Circular network of P2DHMMs for spotting multiple keywords.

4.4 Search method

The spotter network models a small set of key patterns and used to locate them while ignoring the rest of the words of no interest. One efficient search is the one stage DP. For the continuous spotting with forward scanning, however, we applied a modified form of two-level DP; this performs a single forward pass consisting of alternating partial forward search and output. The time requirement for the two-level DP is $O(NT^2)$ where N is the total number of states or nodes, and T is the number of vertical frames in a line [10]. In the proposed method, this is reduced to $O(N|w|T)$ where $|w|$ is the horizontal length or the number of vertical frames per character.

Ⅴ. Experiments

5.1 Digit recognition

The proposed method of character spotting has been tested first on digit string recognition; recognition corresponds to spotting all characters. Training sets include 20 character samples per class. The average images derived from the sets were further smoothed to accommodate unknown variants including noise. Test sets were prepared from another set of text images allowing a limited degree of skewing in the data scanning stage. A digit test image is a string of 10 digits in arbitrary order. In this test as well as the Hangul character experiment to follow, we introduced a new set of features that capture the vertical derivatives of the image (i.e., $\Delta = I(x,\ y) - I(x,\ y1) \in \{0,\ 1,\ -1\}$) as the second codebook of P2DHMM, and employed the relative entropy measure for deciding state merge [11].

The result of the test is shown in Table 1. The figure is essentially a confusion matrix that shows the rates of data-conditional misclassification, i.e., the ratios of which data is recognized to which class. Most of the confusion errors are located in the foreground triangular sections of the matrix. According to the figure 36.8% of the errors came from samples of nine. The overall recognition rate is 88.9% with 180 samples.

Table 1. Confusion matrix from digit recognition test

		Model									
		0	1	2	3	4	5	6	7	8	9
input data	0	0	18	0	0	0	0	0	0	0	00
	1	1	0	18	0	0	0	0	0	0	00
	2	2	0	0	14	3	0	0	0	0	01
	3	3	0	0	2	16	0	0	0	0	00
	4	4	0	0	0	0	18	0	0	0	00
	5	5	0	0	2	0	0	16	0	0	00
	6	6	0	0	0	0	0	1	16	0	01
	7	7	0	0	1	0	0	0	0	17	00
	8	8	0	0	1	0	0	0	0	0	017
	9	9	0	0	0	1	0	1	3	0	112

Figure 6 shows sample images used for the experiment. We took an average for each character class and created a P2DHMM. In the case of filler modeling, we do not need to prepare an additional set of samples; rather we computed the average of the entire set for a filler model.

Fig. 6 Digit sample images used in training and testing.

5.2 Hangul character spotting

One significant characteristic of Korean text is that there are no natural italic fonts. And there are not so many fonts as in Latin alphabets. These observations justify the use of simple image-based models as proposed in the paper. A limited test has been performed in the context of 10 point (*Myongjo* font) character images scanned in 200dpi resolution. The letter models were created from the hand-segmented letter images. Instead of manually preparing a large set of segments, a single image was chosen and blurred by Gaussian filter with the radius of two pixels. Refer to individual letters in Figure 8. The most frequently used 97 character classes were used in character (not word) spotting task. The character set constitutes approximately the half of the test text. The search process uses a two-level DP with a modification to the level processing so that a continuous forward output stream can be produced.

The test result has been analyzed in terms of correct spotting(H), false positives(P) and false negatives(N). The overall spotting performance was 79.7 percent as shown in Table 1. But the rather high rate of failures remains to be explored further. For a detailed investigation, we provided a character type hits and failures in the table. The character type Type I ~ VI corresponds to the six different arrangement (also called 'clusters') of Hangul vowels and consonants. Here the type hit means that the type of the character is correct regardless of the correctness of the label.

Table 2. Korean Hangul character spotting results. (H = the number of hits, P = the number of false positives, and N = the number of false negatives)

	Correct hits $\left(\frac{H}{H+P+N}\right)$	False positives $\left(\frac{H}{H+P+N}\right)$	False negatives $\left(\frac{H}{H+P+N}\right)$	# Classes	Remarks
Overall	79.7%	10.9%	9.4%	97	Character spotting
Type I	90.9%	0.0%	9.1%	20	Type spotting
Type II	91.7%	8.3%	0.0%	22	
Type III	81.3%	12.5%	6.3%	17	
Type IV	88.9%	11.1%	0.0%	18	
Type V	80.0%	0.0%	20.0%	19	
Type VI	87.5%	0.0%	12.5%	11	

Figure 7 shows a sample result containing spotted characters with enclosing boxes and labels. Although not marked in the figure, all the characters are correctly transcribed into either key labels or appropriate filler types.

Although not necessary for the consistent description of the paper, some examples of synthetic character images are provided in Figure 8 to help readers' understanding. They were prepared by composing individual letter images which had been averaged over a collection of type (of Hangul character composition) dependent samples and then smoothed using a Gaussian filter. For the vowel images in the middle image, we divided 'ㅗ' and 'ㅜ' into two classes according to shape of the first consonants and the final consonants respectively. Compare the length of the vertical strokes of the two basic vowel letter 'ㅗ' in the second and the third characters. Note also that the final consonant in the last image is not correctly aligned vertically with the vowel.

According to the test results, we have found one interesting case of failures that cannot be easily resolved under the current P2DHMM architecture. The case is illustrated in Figure 9 where the two images are potentially very similar when analyzed in vertical frame basis. This may be resolved simply by introducing state or model duration parameters.

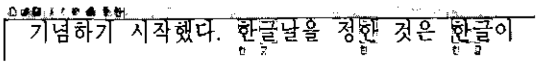
기념하기 시작했다. 한글날을 정한 것은 한글이

Fig. 7 A sample result of spotting two characters.

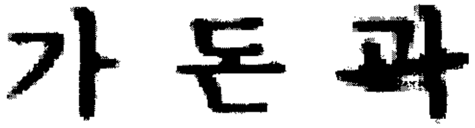

Fig. 8 Compound characters composed of several consonants

Fig. 9 Two characters similar in the context of P2DHMM and vertical frames.

Ⅵ. Conclusion

Using a set of letter image templates, a very effective method is presented for real time synthesis of key character P2DHMMs. The method is based on the principle of composing Hangul syllable characters. The composition itself is very efficient and its conversion to a P2DHMM is highly intuitive considering that we are dealing with machine printed character images. With experimental results form the application to key character spotting tasks, we consider that the feasibility is confirmed and the future refined system would work to meet our demand for the application to content-based document image indexing and retrieval. Currently the only point of simplification has been the equal dimension of the sample character bitmaps. Alleviation of this constraint is one of the immediate future works on our research goal.

References

[1] K. Lang, A. Waibel and G. Hinton, "A time delay neural network architecture for isolated word recognition," *Neural Networks*, Vol. 3, pp. 23-44, 1990.

[2] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. IEEE*, Feb. 1989.

[3] E. Levin and R. Pieraccini, "Dynamic planar warping for optical character recognition," in *Proc. 1992 ICASSP*, San Fransisco, California, Vol. 3, pp. 149-152, 23-26 March 1992.

[4] R. Chellapa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEE E Trans. on ASSP*, Vol. 33, No. 4, pp. 959-963, Aug. 1985.

[5] O. E. Agazzi and S. Kuo, "Hidden Markov model based optical character recognition in the presence of deterministic transformations," *Pattern Recognition*, Vol. 26, No. 12, pp. 1813-1826, 1993.

[6] O. E. Agazzi, S. Kuo, E. Levin and R. Pieraccini, "Connected and degraded text recognition using planar hidden Markov models," in *Proc. 1993 ICASSP*, Minneapolis, 27-30 April 1993.

[7] S. Kuo and O. E. Agazzi, "Keyword spotting in poorly printed texts using pseudo 2D hidden Markov models," in *Proc. 1993 IEEE Conf. on CVPR*, New York, 15-17 June 1993.

[8] Y. Xu and G. Nagy, "Prototype extraction and adaptive OCR," *IEEE Trans. PAMI*, Vol. 21, No. 12, pp. 1280-1296, December 1999.

[9] C. S. Meyers and L. R. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.*, Vol. ASSP-29, No. 2, pp. 284-297, April 1981.

[10] H. Sakoe. Two-level DP-matching - a dynamic programming-based pattern matching algorithm for connected word recognition, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol.ASSP-27, No.6, pp.588-595, December 1979.

[11] S. Kullback, *Information Theory and Statistics*, New York: Dover Publications, 1968.

조 범 준(Beom-Joon Cho)

Beom-Joon Cho has been a professor in the School of Computer Engineering at Chosun University since 1980. He did post-doctoral work at the Dept. of Computer Science and Engineering at the University of Connecticut from Aug. 1989 to Aug. 1990. He also did post-doctoral work at the Dept. of Computer and Information Science at the University of Massachusetts from Sep. 1990 to Aug. 1991. He was a visiting professor at the Center for Automation Research at the University of Maryland from Dec. 1997 to Feb. 1999. He was the director of the Computer Center at Chosun University from Apr. 1993 to Apr. 1997. He has been Dean of College of Electronics and Information since Nov. 2000. He received the PhD in Electrical Engineering from Hanyang University in the Republic of Korea in 1988. His research interests include pattern recognition, neural networks, and artificial intelligence.