Dynamic Character Model Generation for Document Keyword Spotting

Beom-Joon Cho[1] and Bong-Kee Sin[2]

[1] Department of Computer Engineering, Chosun University,
Dong-ku, Gwangju 501-759 Korea
bjcho@chosun.ac.kr
[2] Department of Computer Multimedia, Pukyong National University,
Dayon-dong 599-1, Nam-ku, Busan 608-737 Korea
bkshin@pknu.ac.kr

Abstract. This paper proposes a novel method of generating statistical Korean Hangul character models in real time. From a set of grapheme average images we compose any character images, and then convert them to P2DHMMs. The nonlinear, 2D composition of letter models in Hangul is not straightforward and has not been tried for machine-print character recognition. It is obvious that the proposed method of character modeling is more advantageous than whole character or word HMMs in regard to the memory requirement as well as the training difficulty. In the proposed method individual character models are synthesized in real-time using the trained grapheme image templates. The proposed method has been applied to key character/word spotting in document images. In a series of preliminary experiments, we observed the performance of 86% and 84% in single and multiple word spotting respectively without language models. This performance, we believe, is adequate and the proposed method is effective for the real time keyword spotting applications

1 Introduction

In the field of OCR the neural network is a highly successful model for recognizing 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 performance of the segmenter and the quality of the resulting segments. Another problem with the neural network model is that it is a purely wholistic model that cannot be decomposed, analyzed nor synthesized; therefore training thousands of character models is extremely difficult, if not impossible.

Since the early nineties one model has come into the arena of document analysis; it is the hidden Markov model or HMM. Stimulated by the success in speech recognition, the modeling capability of the variability and sequential flow of a pattern, HMM

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 1114–1125, 2004.

has been used in diverse areas successfully [2]. The application of HMMs benefits from the wide range of experience accumulated in speech recognition and many other fields.

Since document texts run sequentially and, mostly, left to right, it is natural that the idea of using HMM occurs to researchers. To date, the HMM application to English has been reported in several places in the literature. But, although texts run linearly, individual character patterns are not linear but two-dimensional. This fact has not been a barrier to the modeling of Latin alphabet-based texts that run strictly left-to-right down at the letter level. In fact the idea is simply straightforward. But in the case of Korean Hangul characters the problem is not so simple. At the character level or above, texts run linearly. One problem is that there are thousands of characters used in Hangul texts, and we may need the corresponding number of models. An observation below the character level is that a Korean Hangul character is composed of either two or three graphemes arranged two-dimensionally in a way to fit into a rectangle. The two-dimensional composition of grapheme models in Hangul is not straightforward and thus the HMM has not been tried for machine-print character recognition. Without doubt, however, the composition method of character modeling is more advantageous than that of designing thousands of whole character models in regard to the memory requirement as well as the training difficulty.

This research is focused on the application of the HMM method to the analysis of document text images. The basic idea lies in the real time generation of Korean Hangul character models for spotting key characters in the content analysis of optical documents. In the proposed method individual character models are synthesized in real-time using the trained grapheme image templates.

Since characters are two-dimensional, it is natural to believe that a 2D HMM, an extension to the standard HMM, will be helpful and offer a great potential for analyzing and recognizing character patterns. 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]. The latter model, called P2DHMM, 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 Hangul character pseudo 2D HMM using trained grapheme image templates. We believe the proposed method is feasible and particularly appropriate thanks to the absence of natural italic fonts corresponding to the English italics, a rationale for using P2DHMM.

In the proposed method, we prepared a set of grapheme patterns for each grapheme and obtained their average, the grapheme template. By superposing appropriate grapheme templates, we can compose a character image template. This character template is converted to a P2DHMM in a systematic way. In this method the new idea of location-preserving 2D superposition is very simple but highly elegant and efficient for real-time processing. The idea of character composition is not new, but the application to strictly 2D model design is. It is especially true in 2D HMM framework. Another feature of the proposed method is the conversion of the gray-scale template into P2DHMM, which is theoretically correct in the sense of maximum

likelihood estimation. Additional noteworthy feature is model reduction by noting the information redundancy in the templates; successive HMM states are merged based on the similarity between their output PDs. The resulting models are often much smaller than the original and thus speed up the spotting task, and sometimes, improves the performance.

The rest of the paper consists as follows. In Section 2 we will briefly review the HMM and P2DHMM. In Section 3 the pseudo 2D HMM and its algorithm are described; and then a procedure for developing character models is discussed in detail. Section 4 describes auxiliary models needed for the proposed method of key character spotting. Section 5 presents results from preliminary experiments. Section 6 concludes the paper.

2 HMM Theory

This section reviews briefly the theories of HMM and pseudo 2D HMM.

2.1 HMM

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

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

The most frequent task with an HMM is the evaluation of the model matching score for an input sequence $X = x_1\, x_2 \ldots x_T$. 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. Thanks to 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.

2.2 P2DHMM

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 [6]. In order to implement a continuous forward search method and sequential composition of word models, the former type has been used in this research.

Let us consider a t-th vertical frame $X_t = x_{1t}\, x_{2t} \ldots x_{St}$, $1 \le t \le T$, in a text line image. This is a one-dimension sequence like that of X in Equation (1). This is modeled by a sub-HMM λ_k with the likelihood $P(X_t|\lambda_k)$. You may regard each sub-HMM λ_k 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 let us consider a bitmap image which we define as a sequence of such vertical frames as $X = X_1\, X_2 \ldots X_T$. Each frame will be modeled by a super-state or a sub-HMM. Let Λ be a sequential concatenation of sub-HMMs. Then the evaluation of Λ given the sample image X is

$$P(X \mid \Lambda) = \sum_{R} P_1(X_1) \prod_{t=2}^{T} \vec{a}_{r_{t-1}r_t} P_{r_t}(X_t) \tag{3}$$

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 Equations (2) and (3) can be effectively approximated by the Viterbi score.

One 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

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

where $Q = q_1 q_2 \cdots q_S$ 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

$$\delta_s^k(j) = \max_i \delta_{s-1}^k(i) a_{ij}^k b_j^k(x_{st}) \quad j = 1, \ldots, M_k,\ s = 1, \ldots, S,\ k = 1, \ldots K \tag{5}$$

where $\delta_s^k(j)$ denotes the probability of observing the partial sequence $x_{1t} \ldots x_{st}$ 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, the super-state transitions, to the conventional HMM parameter sets. Therefore it is a simple extension to conventional HMM.

3 Character Modeling

One of the most important tasks in hidden Markov modeling is estimating the probabilistic parameters. For this task we assume a set of typical samples of character images $X = \{ X^{(1)}, \ldots, X^{(D)} \}$ of an equal dimension. Different size raises no problem if we scale the images bilinearly. Moreover, the scale difference in test images is naturally resolved with HMM method.

The focus of the section lies in the construction of the P2DHMM for a Korean Hangul character. A Hangul character consists of either two or three graphemes of phonetic consonant and vowel letters. The composition follows a general rule to fit the graphemes into a rectangle. There are six types of combination (Fig. 1) according to the shape of the vowel (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 Fig. 2, and explained as follows:

(1) Grapheme segmentation. This step involves extracting the individual graphemes from character samples while retaining the location inside the box enclosing the character. As illustrated in Fig. 3, the graphemes are separated while retaining the position with the box. 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 [6].

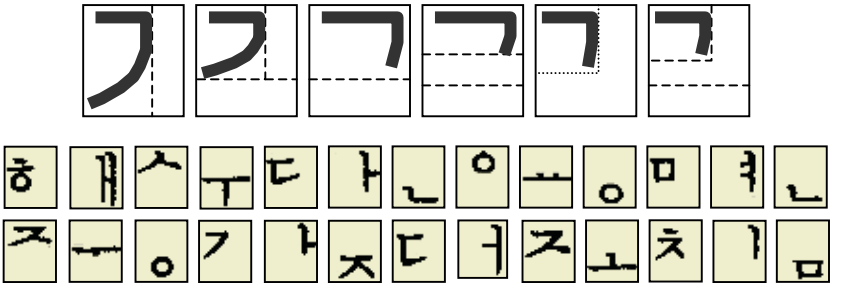

Fig. 1. (Top) Six types of grapheme arrangement inside a Korean syllable character box. A grapheme changes its shape according to the type. (Bottom) Grapheme segment samples.

(2) Average the extracted samples. Now there is a set of grapheme samples. First we classify the samples according to the type of the grapheme arrangement pattern of the original character. For the initial consonant grapheme there are six types (see Fig. 1), and two for each vowel grapheme. Then, take the sample average of the set of categorized images pixel by pixel so that a smooth grayscale-like image is obtained (see Fig. 3). Assuming binary samples, the average intensity of the pixel at (i, j) 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 at this stage.

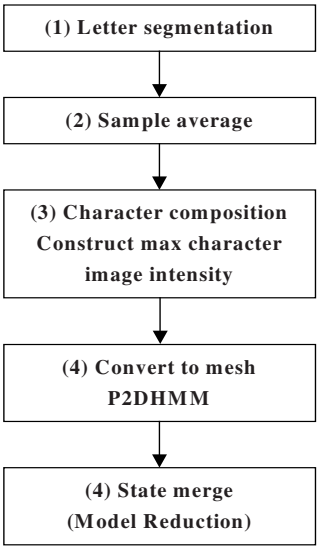

Fig. 2. Model design procedure.

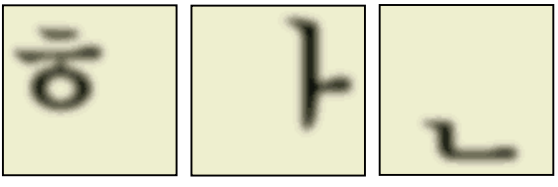

Fig. 3. Korean graphemes separated out from a syllabic character for /*han*/. From left to right: the initial consonant, the vowel, and the suffix consonant. Note that the grapheme position in the original character block is retained.

(3) Character image construction. 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 from the component grapheme images generated in the previous step(Fig. 4). The value of the (i, j)-th pixel of the character template takes the maximum of the two or three pixels (i, j) from each grapheme plane.

(4) Conversion into 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

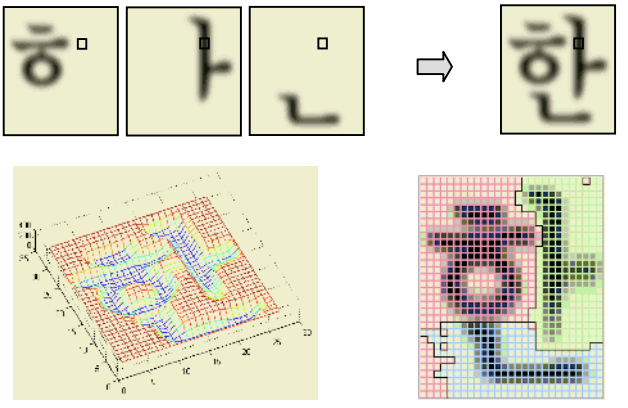

Fig. 4. Character image construction and the result.

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

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.

The proposed procedure of creating statistical model is theoretically correct in the sense of maximum likelihood sense. One problem with the method is found in the final stage of merging states. But it is justified because, although the method of state merging itself is coarse yet, the idea of merging is correct information-theoretically.

4 Keyword Spotting

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

4.1 Filler Model

In keyword spotting task, a filler corresponds to something between interesting things or keywords. It is also called a non-key character. Then a filler model is defined as the model for all non-key characters. For convenience sake, however, it does not

discriminate keys and non-keys but models all kinds of character patterns statistically. The desired characteristic of the filler model λ_F is

$$p(x^K \mid \lambda_F) < p(x^K \mid \lambda_K)$$
$$p(x^F \mid \lambda_F) > p(x^F \mid \lambda_K)$$

where λ_K is a key model for the key pattern x^K, and x^F is a non-key pattern. In general, however, the character patterns are not completely random and there is a certain degree of similarity between some characters. In addition it is not easy to design a single good model for numerous patterns of all characters. According to the work of Lee and Kim [7], the filler model can behave as a threshold. For better thresholding capability in Korean Hangul characters, we defined six fillers, one for each of the six character composition types as of Fig. 1. Fig. 5 shows the filler images before conversion to P2DHMMs. They are simple arithmetic averages over a large set of character samples. Unlike the key character models, the filler models are not synthesized in real-time. Rather they are prepared once and for all from the image templates. Compared to the key model construction, filler model creation is very simple.

Fig. 5. Bitmap images for filler models.

4.2 White 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(Fig. 6). 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 [8]. The result is retrieved from the best hypothesis.

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, we applied a modified form of

two-level one-stage DP; this performs a single forward pass consisting of alternating partial forward search and output. The time requirement for the DP is $O(N^2ST)$ where N is the total number of states or nodes, and S and T are the frame length and the number of vertical frames in a line [9] respectively. In the proposed method, this is reduced to $O(NST)$.

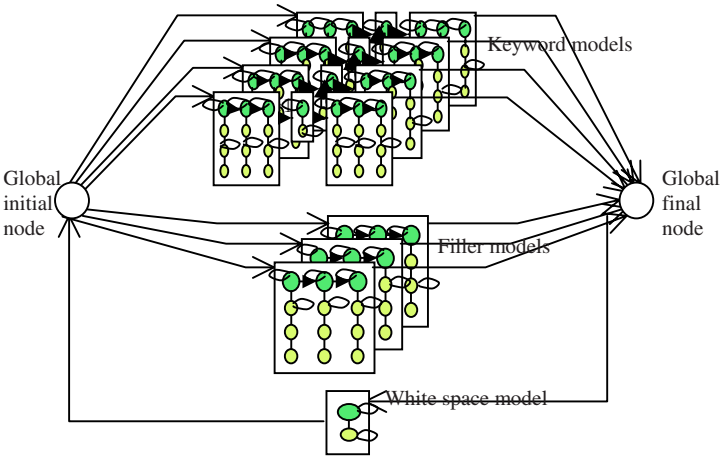

Fig. 6. The circular network of P2DHMMs for spotting keywords.

5 Experiments

5.1 Hangul Character Spotting

One significant characteristic of Korean text is that there are no natural italic fonts. This observation justifies the use of simple image-based modeling scheme. In the initial experiment a limited test has been performed using 10 point (*Myongjo* font) character images scanned in 200dpi resolution. The letter models were created from the hand-segmented letter images. In this test we prepared only a single image for each letter and blurred it by a Gaussian filter to the effect of averaging multiple images. 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 test result has been analyzed in terms of correct spottings(H), false positives(P) and false negatives(N). The overall spotting performance was 79.7 percent as shown in Table 1. In order to better understand the performance and weakness, we detailed the result into character type hits and failures in the same table. The character Types I to VI correspond to the six different arrangements (see Fig. 1) of Hangul vowels and consonants. Here the type hit means that the type of the character is correct regardless of the correctness of the character label. According to the table, the hit ratios of Type III and V are relatively low, for which false acceptance and rejection are high. We noted this fact to refine the models and tune merge parameters for the next set of experiments.

Table 1. Korean Hangul character spotting result. (h = the number of hits, p = the number of false positives, and n = the number of false negatives; H = h/(h+p+n), P = p/(h+p+n), and N = n/(h+p+n)) All figures are in %.

	H	*FA*	*FR*	#Classes
Character	79.7%	10.9%	9.4%	107
Type average	87.0%	5.6%	7.4%	6
Type I	90.9%	0.0%	9.1%	(20)
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)

5.2 Word Spotting

A word is a linear left-to-right concatenation of characters in Hangul system. For word spotting task we tested a mixture of fourteen keyword models on a set of one hundred journal papers' abstract images. In this test we fixed the filler models optimized previously since they need not be created dynamically at run time.

Fig. 7 shows the performance change by varying the state merging thresholds. In the graph the highest hit(H) reaches 66.7% at the threshold of 0.03. In this case the recall is very high but the precision is sacrificed a lot; the best precision score is obtained at threshold 0.01.

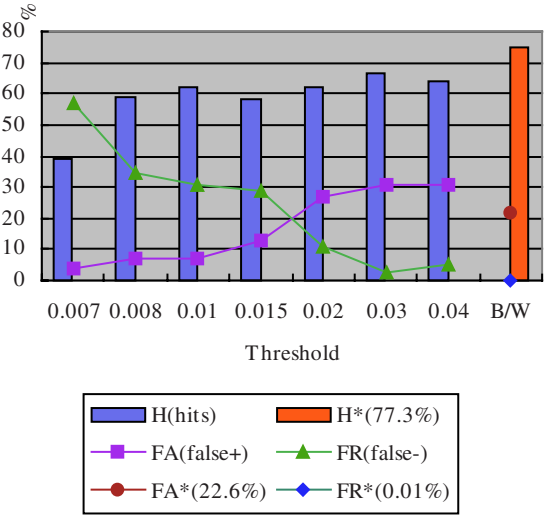

Fig. 7. Word spotting result.

Let us compare the performance figures of the proposed method with those of spotting with Baum-Welch-trained P2DHMMs which are assumedly optimal. The

latter models record 77.3%(H* for the hit rate), 22.6%(P* for the false positives) and 0.1%(N* for the false negatives), as are separately marked at threshold 1.0. The Baum-Welch modeling method for the P2DHMM, although superior, cannot be used for large vocabulary keyword spotting tasks that require training tens of thousand P2DHMMs and preparing a huge number of character samples. This implies that the proposed method of dynamic synthesis of key character P2DHMMs has a definite advantage over the traditional Baum-Welch modeling. Furthermore, if a higher precision is desired, we can pass the spotted word images to a high-performance recognizer for a more accurate spotting. This method will be far faster than the full recognition of the whole documents.

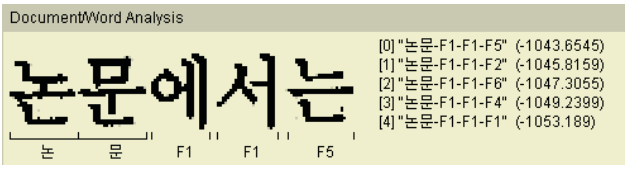

Fig. 8. A sample result, part of screen shot, showing correct spotting and filler type classification.

Fig. 8 gives a sample result, a part of screen shot, showing correct spotting and filler type labeling. Note the small gaps between fillers. They denote white spaces between characters.

5.3 Keyword Set Spotting

In the final experiment we compared the hit ratio by varying the number of keywords sought at a time. Table 2 summarizes the result. When the number of keywords $N = 1$, the hit ratio reached peak, above from character spotting performance. When N increases, the confusion among words also increases thus degrading the performance gradually. The last column corresponds to the highest hit ratio in the preceding experiment. But when N is moderately large, the word spotting task is more successful than the individual character spotting task where we used about four character models at a time, about two Korean Hangul syllable characters in a word.

Table 2. Character and word spotting performance with increasing number of keyword models spotted at a time.

	Character spotting ($N = 2$)	word spotting (N = # keywords)		
		$N = 1$	$N = 2.5$	$N = 14$
Hit ratio(H)	75.0%	86.3%	83.8%	66.7%

6 Conclusion

Using a set of letter image templates, we proposed a very effective method for real time synthesis of key word 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 word spotting tasks, we consider that the method is highly feasible and meets our ultimate demand for the application to content-based document image indexing and retrieval.

References

1. Lang, K., Waibel, A., Hinton, G.: A time delay neural network architecture for isolated word recognition, Neural Networks, 3(1990), 23–44
2. Rabiner, L. R.: A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE, *77*(1989), 257–286
3. Levin, E., Pieraccini, R.: Dynamic planar warping for optical character recognition, Proc. ICASSP, 3(1992), San Fransisco, CA, 149–152
4. Chellapa, R., Chatterjee, S.: Classification of textures using Gaussian Markov random fields, IEEE Trans. ASSP*,* 33(1985), 959–963
5. Agazzi, O. E., Kuo, S.: Hidden Markov model based optical character recognition in the presence of deterministic transformations, Pattern Recognition, 26(1993), 1813–1826
6. Xu, Y., Nagy, G.: Prototype extraction and adaptive OCR, IEEE Trans. PAMI, 21(1999), 1280–1296
7. Lee, H.-K., Kim, J.H.: An HMM-based threshold model approach for gesture recognition, IEEE Trans. PAMI, 21(1999), 961–973
8. Meyers, C. S., Rabiner, L. R.: A level building dynamic time warping algorithm for connected word recognition, IEEE Trans. ASSP, 29(1981), 284–297
9. Sakoe, H.: Two-level DP-matching - a dynamic programming-based pattern matching algorithm for connected word recognition, IEEE Trans. on ASSP, 27(1979), 588–595