A Component-based On-line Handwritten Tibetan Character Recognition Method Using Conditional Random Field

Long-Long Ma, Jian Wu
National Engineering Research Center of Fundamental Software
Institute of Software, Chinese Academy of Sciences
Beijing, P. R. China
{longlong, wujian}@iscas.ac.cn

Abstract

This paper presents a new component-based recognition method using conditional random field (CRF) for on-line handwritten Tibetan characters. The character pattern is over-segmented into a sequence of sub-structure blocks. Integrated segmentation and recognition method based on the CRF model is used to determine the component segmentation points from these block sequences. The CRF model combines component shape likelihood with geometrical likelihood. The parameters are learned using an energy minimization method. We build a component-based spelling rule model to ensure the correct component appearing at a specific structural position. A character-component generation model is presented to reduce component recognition error rate and accelerate the recognition process. Experimental results on MRG-OHTC database show that the proposed method gives promising performance comparing with the holistic method and the component-based conventional path evaluation method.

1. Introduction

Tibetan is a set of minority characters with a long history over 1,300 years. Tibetan language is still used by more than six million people in China, especially in Xizang, Yunnan and Qinghai provinces. Research on Tibetan character, which will enable easier the modernization of Tibetan culture and digitization of Tibetan document, is very important in the theoretical value as well as in extensive application perspective.

With the emergence of digitizing tablets, tablet PCs, digital pens, pen-based PDAs and mobile phones, online handwritten character recognition is gaining renewed interest. However, compared to the existing research work on CJK (Chinese, Japanese and Korean) and Arabic, Online handwritten Tibetan character recognition (OHTCR) is a relatively unexplored field.

More research works focus on printed Tibetan character. Ding designed a novel and effective recognition method for multi-font printed Tibetan OCR [1]. Ngodrup proposed local self-adaptive binary algorithm and grid-based fuzzy stroke feature extraction to improve the recognition accuracy [2]. Masami used Euclidean distance with deferential weights to discriminate similar characters [3]. A feature extraction method based on fractal moment is effective to enhance recognition accuracy [4]. There is far little reported work on OHTCR. Wang [5][6] combined HMM based on stroke type with HMM based on the position relation between strokes to improve the recognition performance. IMLDA (image matrix linear discriminate analysis) feature extraction and MQDF classifier are used to online Tibetan recognition [7]. We published an online handwritten Tibetan character database named MRG-OHTC [8] and got preliminary experimental results [8][9]. However, the recognition accuracy is far from that of human. From the characteristic of structure, Tibetan characters are combinations of structure elements located at specified positions. We select components as structure elements. The shape of components and the relative positions between components are preserved to some extent. The structure information is helpful to improve the recognition accuracy.

We propose a component-based recognition method using CRF. CRF is an undirected graphical model, which can effectively capture the dependencies between components, and provide principled tools for parameter learning and

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.153

contextual recognition. Tibetan character is generated based on the structure relations between components where components as structure elements. Based on character over-segmentation results, the CRF models are used to find the optimal component segmentation point sequences. We get the component labels and the character recognition result from these component segmentation points. Experimental results on MRG-OHTC database show the recognition accuracy is improved and the store space of model dictionary is saved.

2. Characteristic of Tibetan characters

Tibetan script consists of 4 vowels and 30 consonants, which are called basic elements. There are two kinds of characters used in handwritten Tibetan characters, that is, single characters (SC) and combined characters (CC).

Syllables are basic spelling units [18], whose structure is shown in Fig.1. Each syllable consists of at most 4 characters (those parts surrounded by red dash line bounding boxes in Fig.1. Some characters are called as CC, which are made up of EC (essential consonant), TV (the top vowel), CaEc (the consonant above the EC), CbEc (the consonant below the EC) and BV (the bottom vowel). Some consonants (in total 20) can be as a valid character and they locate at the left (CbCC, the character before CC) or right (1-CaCC and 2-CaCC, the first and second characters after CC) of the CC. Fig.2 gives an example of 4-character syllable.

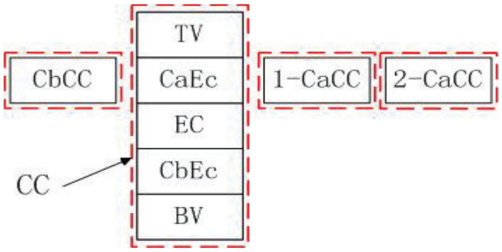

Figure 1 The syllable structure

From linguistic viewpoint, Tibetan character is composed of 34 basic elements (vowels and consonants). In this paper, we call basic elements as components. However, these components are hard to separate from characters by computer algorithm. We add some component models referring to selection criterion in [10]. Combing the component segmentation algorithm, we obtain 120 components from 562 Tibetan character classes. Table 1 gives all these component models. The columns *Idx* and *C* correspond to the index and shape of every component.

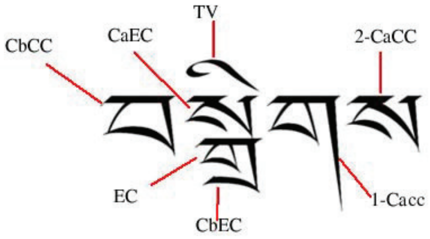

Figure 2 A example of a 4-character syllable

Table 1 Component models

Idx	*C*	*Idx*	*C*	*Idx*	*C*	*Idx*	*C*	*Idx*	*C*
1	ཀ	25	ར	49	ཁྱ	73	ཅུ	97	རྩུ
2	ཁ	26	ལ	50	ཀྲ	74	ཅྭ	98	རྭ
3	ག	27	ཤ	51	ཀྭ	75	ྂ	99	◌ྤ
4	ང	28	ས	52	ཀྱ	76	སྦ	100	ཤྭ
5	ཅ	29	ཧ	53	ཏུ	77	སྦྱ	101	ཀུ
6	ཙ	30	◌ཿ	54	སྡུ	78	∆	102	ཐྲ
7	ཆ	31	◌ེ	55	དུ	79	སྤ	103	ཏྲུ
8	ཉ	32	◌ི	56	ཛེ	80	སྣ	104	སྒུ
9	ཏ	33	◌ུ	57	རྗི	81	སྤྱ	105	སྒྱ
10	ཐ	34	◌ོ	58	རྗེ	82	ཐ	106	སྒ
11	ཇ	35	ཏྲ	59	སྦྲ	83	ཏ	107	ཎ
12	ཛ	36	བྲ	60	རྐུ	84	རྫ	108	ཞ
13	ད	37	བྱ	61	ཁྲྭ	85	རྐ	109	རྦ
14	ན	38	ཊ	62	ཁྲུ	86	རྐུ	110	ཏ
15	པ	39	དྲ	63	ཁུ	87	སྐྱ	111	ཅེ
16	ཕ	40	གྲ	64	ཁྱུ	88	རྗ	112	ཇེ
17	བ	41	གྲྭ	65	གུ	89	སྐ	113	ཇི
18	མ	42	གྭ	66	ཙུ	90	སྐྲ	114	ཇོ
19	ཚ	43	གྱ	67	ལྭ	91	སྨ	115	ཅི
20	ཞ	44	ཧྲ	68	སྭ	92	སྨྱ	116	ཅོ
21	ཟ	45	ཧུ	69	སུ	93	རྣ	117	ཟྭ
22	འ	46	ཧྭ	70	སྱ	94	ཛ	118	དྭ
23	ཡ	47	ཁྲ	71	ཌ	95	ཏྭ	119	ཞུ
24	ཨ	48	ཁྭ	72	ཉུ	96	རྩ	120	ཞྭ

3. System overview

The flowchart of the proposed component-based recognition method using CRF is shown in Fig 3. The input character is composed of a sequence of strokes.

Considering the characteristic that Tibetan character is presented as a vertical combination of consonant and vowel, we over-segment the input character into the sequence of sub-structure blocks using vertical overlapping degree as in [11] and rule-based small component combination. These sub-structure blocks are re-ordered according to the upper boundary of their bounding boxes. Fig 4 shows several over-segmentation examples. The part surrounded by a red bounding box represents a sub-structure block.

The module named CRF-based integrated segmentation and recognition is to find the optimal component segmentation points from sub-structure block sequences. The CRF model fuses four different models into a principled framework. The parameters are learned using an energy minimization method. According to the component segmentation points we can get the component sequences and character recognition result.

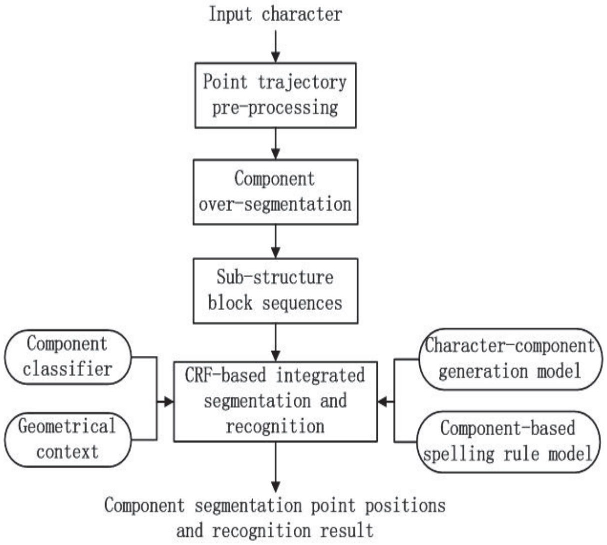
Input character
Point trajectory pre-processing
Component over-segmentation
Sub-structure block sequences
Component classifier
Geometrical context
CRF-based integrated segmentation and recognition
Character-component generation model
Component-based spelling rule model
Component segmentation point positions and recognition result

Figure 3 Flowchart of the proposed method

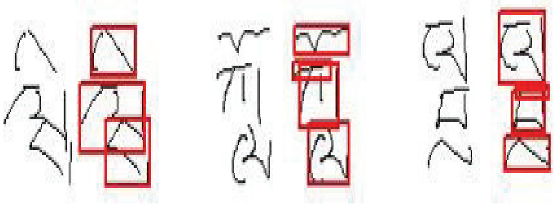

Figure 4 Component over-segmentation examples

4. CRF for component-based recognition

CRF is a discriminative graph model which is designed for labeling sequence data [12] and has been proposed for word and character string recognition [13][14]. Considering Tibetan character composed of ordered component sequences, we use CRF to label component segmentation points based on component over-segmentation results.

4.1. Integrated CRF model

After the component over-segmentation module, input Tibetan character is divided into sub-structure block sequence $C=(c_1,c_2,\cdots,c_n)$ ordered by the upper boundary of their bounding boxes, as shown in Fig 4. Our objection is to find the optimal segmentation S^* from C, that is, to maximize the posteriori probability $P(S|C)$:

$$S^* = \arg\max_{S} P(S\mid C) \qquad (1)$$

where $S=(s_1,s_2,\cdots,s_m)$ denotes a candidate segmentation sequence and m is the number of candidate components.

Although various path evaluation criteria have been proposed, most of them still lack a principled framework. We refer to a principled maximum a posteriori (MAP) framework based on CRF [13], which demonstrates the superiority for Japanese character string recognition.

A CRF may be viewed as an undirected graphical model [12]. Combining the definition of CRF, let (V, E) be a graph where each sub-structure block c_i as a node and the edges between nodes denote candidate components. (C, S) is a CRF if the probability of S conditioned on C obeys the Markov property. Combing our component-based recognition within a character, we use unary and binary cliques on the graph to construct the CRF model.

The graphical structure of a CRF may be used to factorize $P(S|C)$ with parameter λ into a normalized product of strictly positive, real-valued potential functions

$$P(S\mid C,\lambda)=\frac{1}{Z(C)}\exp(\sum_{j}\lambda_j F_j(S,C)) \qquad (2)$$

where $Z(C)$ is a normalized factor. λ_j is weighting parameter. $F_j(S,C)$ is a potential function and can be written as

$$F_j(S,C)=\sum_{i=1}^{m} f_j(s_{i-1},s_i,C,i) \qquad (3)$$

where each $f_j(s_{i-1},s_i,C,i)$ is either a unary function (for single candidate component) or a binary function (for two consecutive candidate components).

4.2. Potential functions

In our CRF-based integrated segmentation and recognition, we adopt two types of potential functions, that is, integrated function and auxiliary function. The integrated function is fused into formula (2) and used to capture the node (candidate component) attributes and local dependencies between nodes. The auxiliary function is use to remove some impossible candidate components and accelerate the recognition speed.

4.2.1. Integrated function. Integrated functions include unary and binary functions. Unary functions are built for single candidate component, which is described from shape and geometry features. Modified quadratic discriminant function (MQDF) is used to model component shape features, which are extracted using character feature extraction method. The unary geometric features extracted from single candidate component pattern are: width, height, (x,y) coordinates of the center of bounding box, normalized with respect to the character size. We use the Gaussian PDF to model these unary geometric features.

To derive a binary geometric feature function, geometric features are extracted from pairs of consecutive candidate components and include the signed differences of width and height, and of x- and y-coordinates of the centers of bounding boxes. Another Gaussian PDF is used to model these binary features.

4.2.2. Auxiliary function. Compared to integrated function, auxiliary function isn't fused into formula (2). Auxiliary function can enhance the recognition accuracy of candidate component and accelerate the search speed during the process of finding the optimal segmentation path. The type of function includes the function of component-based spelling rule model and character-component generation model.

Component-based spelling rule demands some component classes only can locate at valid structure position. Table 2 gives part of valid components at specific position. If candidate components aren't content with the validity, they will be refused. Candidate component is classified as one of several component classes instead of all component classes. The number of component class at specific position is far smaller than that of all component classes. So this spelling rule model can reduce the classification error rate of component.

Table 2 Part of valid components at specific position

Position	Valid component classes
EC	ཀ ཁ ཅ ཙ ཆ ཉ ◌ྲ ཏ ཇ ཛ པ ཕ ཝ ཞ ཟ ཡ ཨ ཤ ད ཐ
CaEC	ལ ར ས
CbEC	ལ ཡ ར ཝ
TV	◌ེ ◌ོ ◌ི
BV	◌ུ

Tibetan character is represented as component sequences by the upper boundary of component bounding box. Character-component generation model describes the component classes and the component context within a character. The model is measured using a tree structure where each character is a string (path) of components. The strings with a common prefix have a common path of parent nodes for the prefix. This can save both the storage of dictionary and the computation because the common prefix is stored only once and matched only once in finding the optimal segmentation *S**.

4.3. Parameter learning

Assuming the component-based character training samples $\{(c^k,s^k)\}$ are independently and identically distributed, the likelihood of formula (2) over all training samples is denoted as

$$\begin{aligned} P(\{s^k\}\mid\{c^k\},\lambda) &= \sum_k\left[\frac{1}{Z(c^k)}\exp(\sum_j \lambda_j F_j(s^k,c^k))\right] \\ &= \sum_k\left[\frac{1}{Z(c^k)}\exp(\sum_j \lambda_j \sum_{i=1}^{m} f_j(s_{i-1}^k, s_i^k, c^k, i))\right] \end{aligned} \tag{4}$$

To find the optimal parameters, we adopt the negative log-likelihood loss (NLL) [15] as the loss function. This function is convex, guaranteeing convergence to the global optimization.

$$\begin{aligned} L(\lambda) &= -\log P(\{s^k\}\mid\{c^k\},\lambda) \\ &= \sum_k\left[\log Z(c^k) - \sum_j \lambda_j \sum_{i=1}^{m} f_j(s_{i-1}^k, s_i^k, c^k, i)\right] \end{aligned} \tag{5}$$

The parameters are iteratively optimized by stochastic gradient descent algorithm as

$$\lambda(t+1) = \lambda(t) - \varepsilon(t)\frac{\partial L(\lambda)}{\partial(\lambda)}\Big|\lambda = \lambda_t \tag{6}$$

where $\varepsilon(t)$ is the learning step that controls how far the parameters move in the direction of the gradient.

5. Experimental results

We evaluated the performance of our method on MRG-OHTC database of online handwritten Tibetan character database of 910 classes, each class with 130 samples produced by 130 writers. We add the samples (recently collected by us) of 20 writers. So there are the samples from 150 writers. The samples of 562 classes were used in our experiments. We used 120 samples per class for training classifiers for Tibetan component and character recognition, and the remaining 30 samples per class for evaluating the performance. By removing the characters with component over-segmentation error, actually 59,807 samples were used for training CRF parameters and 14,914 samples were used for testing.

Component samples are extracted from the character samples using self-learning algorithms [11]. Character samples are re-labeled with correct component segmentation points. Accordingly, component-labeled database with component segmentation index are built based on character-based database. These two databases are stored sample by sample with fixed format.

Each component pattern undergoes the same procedures of normalization, feature extraction and classification as done in a holistic character recognition method [16]. Specifically, a moment normalization method is used to normalize the coordinates of pen trajectory points, and direction histogram features are extracted directly from pen trajectory using a normalization-cooperated feature extraction (NCFE) method [17]. The resulting 512-dimensional feature vector is reduced to 160 by Fisher linear discriminant analysis (LDA), and a MQDF classifier [18] with 20 principal eigenvectors per class is used to assign the component pattern to 10 top-rank component classes.

Our method was implemented with Visual Studio 2005 and all experiments evaluated on a PC with Intel Dual core 2.66 GHz CPU.

To evaluate the proposed integrated CRF model, we compared our method with our previous normalized path evaluation method [19] and holistic character recognition method. Table 3 lists the experimental results of three methods. As shown in Table 3, the CRF-based method outperforms the normalized path evaluation method, and enhances 4.32% recognition accuracy to holistic character recognition method. The main advantage of component-based recognition is the smaller number of component classes (120) than the number of character classes (562). Accordingly, the store space of component model dictionary is smaller than that of character model dictionary.

Table 3 Comparative results of three methods

Method		#Class	Accuracy
Component-based	CRF	120	93.51%
	Normalized path evaluation		90.76%
Holistic character		562	89.19%

For the CRF-based recognition method, samples with component over-segmentation error in our experiments are removed. Component-based recognition errors are mainly caused by candidate component classification error and path evaluation. Component classification error is attributed to component confusion. Table 4 gives some similar component pairs. While for path evaluation we need fuse other contexts and design other parameter learning method to improve path evaluation.

Table 4 Similar component pairs

ཅ ཙ	ཏ ཏ ཊ	ང ད ཌ
པ ཕ བ	འ ལ	གྲ གྱ གྲྱ
ཁྲ ཁྱ ཁྲུ	ཀུ ཀྱ	◌ེ ◌ི

6. Conclusion

In this paper, we proposed a novel component-based recognition method using the CRF model for online handwritten Tibetan characters. The CRF model integrates four different models (potential functions) into a principled recognition framework. Integrated functions are fused into potential functions while auxiliary functions are used to enhance the performance of path evaluation. Experimental results demonstrate the proposed method is superior over the normalized path evaluation method and holistic character recognition method. However, we remove the samples caused by component over-segmentation error during building component-labeled database. In actual demand, we also should consider these samples. So we need implement a better version to handle these samples and eliminate those errors led by path evaluation. In the future we will extend the method to online handwritten Tibetan syllable and character string recognition.

Acknowledgements

This work is supported by the CAS Action Plan for the Development of Western China (No.KGCX2-YW-512) and National Science & Technology Major Project (No.2010ZX01036-001-002, 2010ZX01037-001-002).

References

[1] X.Q. Ding, H. Wang, Multi-font printed Tibetan OCR, *Advance in Pattern Recognition*, pp.73-98, 2007.

[2] Ngodrup, D.C. Zhao, Study on printed Tibetan character recognition, *Proc. AICI*, pp. 280-285, 2010 .

[3] K. Masami, K. Yoshiyuki, K. Masayuki, Character recognition of wooden blocked Tibetan similar

manuscripts by using Eucliden distance with deferential weight. *IPSJ SIGNotes Computer and Humanities*, pp.13-18, 1996.

[4] Y.Z. Li, G. Fe, Y.L. Wang, Z. Z. Liu, Research on printed Tibetan character recogniton technology based on fractal moments, *Proc 3rd ICCSIT*, pp.57-60, 2010.

[5] B. Liang, W.L. Wang, J. J. Qian, Application of Hidden Markov Model in on-line Recognition of handwritten Tibetan characters (in chinese), *Journal of Microelectronics& Computer,* 26(4): 98-101, 2009.

[6] W.L. Wang, X.Q. Ding, K.Y. Qi, Study on simlitude characters in Tibetan character recognition (in chinese), *Journal of Chinese Information Processing*, 16(4): 60-65, 2002.

[7] J.J. Qian, W.L. Wang, D.H. Wang, A novel approach for online handwriting recognition of Tibetan characters, IMECS, pp. 333-337, 2010.

[8] L.-L. Ma, H.-D. Liu, J. Wu, MRG-OHTC Database for Online Handwritten Tibetan Character Recognition , *Proc. 11th ICDAR*, Beijing, China, 2011, pp.207-211.

[9] L.-L. Ma, J. Wu, A Recognition System for On-line Hand-written Tibetan Characters, *Proc. 9th GREC*, Seoul, Korea, 2011, pp.89-92.

[10] C.-L Liu, L.-L Ma, Radical-based hybrid Statistical-structural approach for Online Handwritten Chinese Character Recognition, Pattern Recognition, Machine Intelligence and Biometrics , P.S.P.Wang （Ed.), Springer, pp.633-655, 2011.

[11] L.-L. Ma, C.-L. Liu, On-line handwritten Chinese character recognition based on nested segmentation of radicals, *Proc of 2009 CCPR & First CJKPR*, Nanjing, China, 2009, pp.929-933.

[12] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, *Proc 18th ICML*, 2001, pp.282-289.

[13] X.D. Zhou, C.L. Liu, M. Nakagawa, Online handwritten Japanese character string recognition using conditional random fields, *Proc. 10th ICDAR*, Barcelona, Spain, 2009, pp.521-525.

[14] S. Shetty, H. Srinivasan, S. Srihari, Handwritten word recognition using conditional random fields, *Proc. 9th ICDAR*, Brazil, 2007, pp.1098-1102.

[15] Y. LeCun, S. Chopra, R. Hadsell, R. Marc'Aurelio, F. Huang, A tutorial on energy-based learning. In: G. Bakir, et al. (Eds.), *Predicting Structured Data*, MIT Press, 2007.

[16] Y. LeCun, S. Chopra, R. Hadsell, R. Marc'Aurelio, F. Huang, A tutorial on energy-based learning. In: G. Bakir, et al. (Eds.), *Predicting Structured Data*, MIT Press, 2007.

[17] C.-L. Liu, X.-D. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, *Proc. 10th IWFHR*, La Baule, France, 2006, pp.217-222.

[18] M. Hamanaka, K. Yamada, J. Tsukumo, On-line Japanese character recognition experiments by an off-line method based on normalized-cooperated feature extraction, *Proc. 2nd ICDAR*, Tsukuba, Japan, 1993, pp.204-207.

[19] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(1): 149-153, 1987.