Context Driven Chinese String Segmentation and Recognition

Yan Jiang[1], Xiaoqing Ding[1], Qiang Fu[1], and Zheng Ren[2]

[1] Department of Electronic Engineering, Tsinghua University, Beijing, China, 100084
{jyan, dxq, fuq}@ocrserv.ee.tsinghua.edu.cn
[2] Siemens AG, D-78467 Konstanz, Germany
zheng.ren@siemens.com

Abstract. This paper presents a context driven segmentation and recognition method for handwritten Chinese characters. We follow a split-merge technique in character segmentation. In this process, a Chinese text line is first pre-segmented into a sequence of radicals, which are then merged according to a cost function combining both recognition confidence and contextual cost. Two strategies are also proposed for implementation: bi-gram based merging and lexicon driven merging. In the former one, we generate a set of merging paths which are then evaluated by Viterbi algorithm. The radicals' best merging method is given by the path with the highest score. In the latter strategy, a lexicon is preset and compared with the radicals to determine both radicals' merging and candidate character selection. Experiments show that contextual information plays a crucial role in Chinese character segmentation and could obviously improve the segmentation and recognition results.

1 Introduction

Single character recognition has achieved impressive progress both in accuracy and speed in the past 40 years. However, it could not remarkably benefit a document reading system directly because some practical difficulties. For example, it is hard to extract text lines from a complex layout document containing both graphs and characters in different fonts and size. Though a text line is perfectly extracted, character segmentation is another ineluctable and decisive step since a general classifier could only recognized a single-character image at a time.

Recently, there many papers considering character segmentation of digits, western and eastern languages. According to Casey ([1]), these methods are categorized into three basic strategies: structural analysis, recognition based and holistic tactic. In this paper, we suggest these methods are concluded into two levels according to the information sources they are rested on (Fig.1). Low level methods make use of information directly from image and high level methods utilize contextual and grammatical constraints originated from prior knowledge.

Character segmentation is still an obstacle in Chinese OCR especially for off-hand case, it should deal with diverse writing styles, a large character set and complex character structures. Moreover, characters are written with touching

D.-Y. Yeung et al. (Eds.): SSPR&SPR 2006, LNCS 4109, pp. 127–135, 2006.

and overlapping in scripts. According to the recent work, low-level methods are effective to remove touching and overlapping by accurately locating the segmentation points for touching and overlapping strokes. But it may come with another problem that a character is wrongly separated into different parts.

In conventional methods, character segmentation contains two steps. The first step is pre-segmentation, which decomposes a text line into a series of radicals. Secondly, these radicals are reunited into characters. Previously, only low level information is considered in the second step, which is shown to be unreliable in practice. A Chinese character may be composed of some parts, each of them is indeed a character itself. Low level methods perform inadequately and always segment a character into more than one parts. Recent papers are considering to involve contextual relationship in this process. In western languages, these context methods are always based on a word dictionary ([5]). However, the methods for oriental languages are quite different from that for western languages. Liu ([3]) proposes a lexicon driven way for Japanese address reading. Takahiro([7]) introduces bi-gram in his likelihood function for on-line Japanese handwriting recognition. But related work has seldom been done for Chinese up to now.

In this paper, we introduce contextual restrictions by incorporating bi-gram and lexicon-dictionary in character segmentation. According to the experiments, we can see that both segmentation rate and recognition rate could get improved.

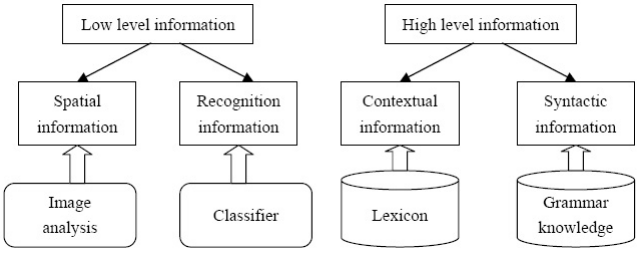
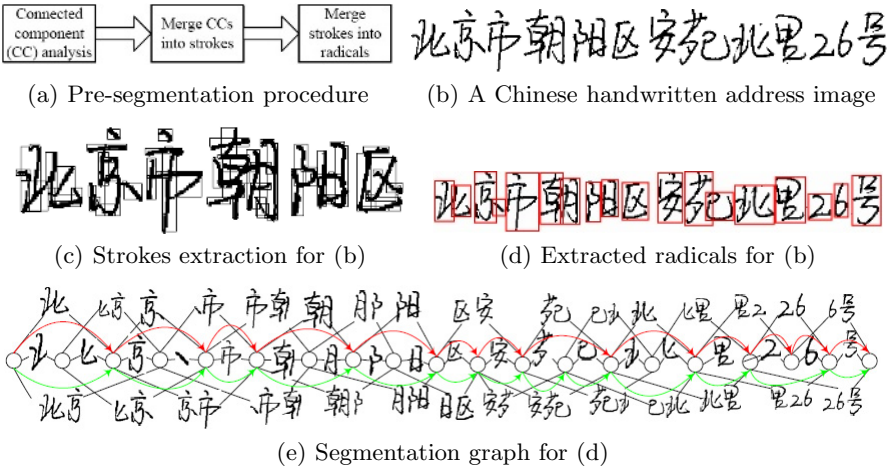


Fig. 1. Character segmentation strategies

2 Pre-segmentation

A Chinese character could be regarded as a composition of some primitive components. In pre-segmentation, we cite Xue's work([6]) to extract these components (Fig.2(a)). He extracts connected components from a text line, which are then merged into strokes (Fig.2(c)). These strokes are assembled to form radicals. Each radical should be warranted as just one part of a character (Fig.2(d)). A segmentation graph is accordingly established (Fig.2(e)), which is directed and acyclic. An arc corresponds to a certain radical combination and a path from the first node to the last represents a merging way for radicals. We assign each arc a cost to evaluate the likeness for a merged image of being a real character([6]).



(a) Pre-segmentation procedure (b) A Chinese handwritten address image

(c) Strokes extraction for (b) (d) Extracted radicals for (b)

(e) Segmentation graph for (d)

Fig. 2. Pre-segmentation for Chinese handwritten texts

3 Bi-gram Based Segmentation

3.1 Introduction of Bi-gram Model

Let $\mathrm{X} = \mathrm{x}_1\mathrm{x}_2 \ldots \mathrm{x}_T$ be a sequence of character images. A classifier commonly gives some hypotheses for each image, for example, $\mathrm{c}_{t,1}\mathrm{c}_{t,2} \ldots \mathrm{c}_{t,M}$ denote the candidates for x_t. Post-processing selects characters from each candidate set and composes the most likely string $\mathrm{c}_{1,k_1}\mathrm{c}_{2,k_2} \ldots \mathrm{c}_{T,k_T}$, where $1 \le k_t \le M, 1 \le t \le T$.

$$\mathrm{c}_{1,k_1^*}, \mathrm{c}_{2,k_2^*}, \ldots, \mathrm{c}_{T,k_T^*} = \underset{1 \le k_t \le M, 1 \le t \le T}{\arg\max} P(\mathrm{c}_{1,k_1}, \mathrm{c}_{2,k_2}, \ldots, \mathrm{c}_{T,k_T} | \mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_T) \quad (1)$$

By Bayesian formula, we have

$$\begin{aligned} &P(\mathrm{c}_{1,k_1}, \mathrm{c}_{2,k_2}, \ldots, \mathrm{c}_{T,k_T} | \mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_T) \\ &= \frac{P(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_T | \mathrm{c}_{1,k_1}, \mathrm{c}_{2,k_2}, \ldots, \mathrm{c}_{T,k_T}) P(\mathrm{c}_{1,k_1}, \mathrm{c}_{2,k_2}, \ldots, \mathrm{c}_{T,k_T})}{P(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_T)} \end{aligned} \quad (2)$$

Assuming that the classifier's decision for the current image is independent of the previous image ([9]), we have

$$P(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_T | \mathrm{c}_{1,k_1}, \mathrm{c}_{2,k_2}, \ldots, \mathrm{c}_{T,k_T}) = \prod_{i=1}^{T} P(\mathrm{x}_i) \times \prod_{i=1}^{T} \frac{P(\mathrm{c}_{i,k_i} | \mathrm{x}_i)}{P(\mathrm{c}_{i,k_i})} \quad (3)$$

In natural language processing (NLP), N-gram model assumes that only the n adjacent characters before the given character make sense. In bi-gram ($n = 2$), $P(\mathrm{c}_{1,k_1}\mathrm{c}_{2,k_2} \ldots \mathrm{c}_{T,k_T})$ is simplified to $P(\mathrm{c}_{1,k_1}) \prod_{i=2}^{T} P(\mathrm{c}_{i,k_i} | \mathrm{c}_{i-1,k_{i-1}})$. Instead of Eq.(1), we turn to maximize $P(\mathrm{c}_{1,k_1}) \prod_{i=2}^{T} P(\mathrm{c}_{i,k_i} | \mathrm{c}_{i-1,k_{i-1}}) \prod_{i=1}^{T} P(\mathrm{c}_{i,k_i} | \mathrm{x}_i)$ to form the most likely string. The maximum of the above criteria could be regarded

as a hybrid of context and recognition cost. In our method, given a merging path in the segmentation graph, the maximum of H (Eq.(4)) is applied in evaluation.

$$H = \frac{1}{T}[\log P(\mathrm{c}_{1,k_1}) + \sum_{i=2}^{T} \log P(\mathrm{c}_{i,k_i}|\mathrm{c}_{i-1,k_{i-1}}) + \sum_{i=1}^{T} \log P(\mathrm{c}_{i,k_i}|\mathrm{x}_i)] \tag{4}$$

3.2 Confidence Estimation

A general character classifier outputs a set of sorted characters with ascending distances. However, distance measure is not discriminating in judging whether an input image is a real character or not. On the other hand, classifiers based on different learning algorithm would output different types of distances, which makes it inconceivable for further discussion. As shown above, distance measure is required to be transformed into probability measure. We will briefly review some basic transformation techniques. Suppose we have M candidate hypotheses $\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_M$ for image x with corresponding ascending distances $d_1 \leq d_2 \leq \ldots \leq d_M$. (5) gives a set of experimental transformations for posterior probability estimation ([8]). In [4], Liu proposes his method based on Gauss distribution assumption (6), where variance parameter θ is estimated from training samples.

$$P(\mathrm{c}_i|\mathrm{x}) = \begin{cases} 1/d_i / \sum_{j=1}^{j=M}(1/d_j) & (5.1) \\ 1/d_i^2 / \sum_{j=1}^{j=M}(1/d_j^2) & (5.2) \\ 1/(d_i - d_1 + 1) / \sum_{j=1}^{j=M}[1/(d_j - d_1 + 1)] & (5.3) \end{cases} \tag{5}$$

$$P(\mathrm{c}_i|\mathrm{x}) = \frac{\exp((d_i - d_1)/\theta)}{\sum_{j=1}^{j=M} \exp((d_j - d_1)/\theta)} \tag{6}$$

3.3 Implementations

In post-processing, character images are fixed prior to candidate selection. However, in radical merging step, we don't know how radicals will be organized. The number of possible ways of merging increases exponentially with respect to the number of radicals. It is necessary to discuss some more applicable methods. In this section, two implementations are provided for bi-gram driven way.

Beam search is an optimization of the best first search algorithm where only a predetermined number of paths are kept as candidates. If more paths than a threshold are generated, the worst paths are discarded.

Bi-gram Driven Beam Search (BDBS)
$\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_N$—the pre-segmented radicals
$\mathrm{S}_{i,j}$—radical combination of radical $\mathrm{s}_i\mathrm{s}_{i+1}\ldots\mathrm{s}_j$
$\mathrm{c}_h(\mathrm{S}_{i,j})$—the h-th candidate character for radical combination $\mathrm{S}_{i,j}$, $1 \leq h \leq M$

Initialization step. For a predefined integer L, we test first L radical combinations $\mathrm{S}_{1,1}$, $\mathrm{S}_{1,2}$,... $\mathrm{S}_{1,L}$ and recognize them. If $\log P(\mathrm{c}_h(\mathrm{S}_{1,j})|\mathrm{S}_{1,j}) + \log P(\mathrm{c}_h(\mathrm{S}_{1,j}))$

is more than C, we add $< 1, j, \mathrm{c}_h(\mathrm{S}_{1,j}), \log P(\mathrm{c}_h(\mathrm{S}_{1,j})|\mathrm{x}_{1,j}) + \log P(\mathrm{c}_h(\mathrm{S}_{1,j})), 1 >$ to the node list as a valid expansion, in which, the fifth element records the number of characters up to current merging.

Expanding step. For each node $< i, j, \mathrm{c}_h(\mathrm{S}_{i,j}), Q, n >$ in the list, we expand the node list as follows. We recognize $\mathrm{S}_{j+1,j+1}$, $\mathrm{S}_{j+1,j+2}$,... $\mathrm{S}_{j+1,j+L}$, if there exist p, q that satisfy $j+1 \le p \le j+L$, $1 \le q \le M$ and $\log P(\mathrm{c}_q(\mathrm{S}_{j+1,p})|\mathrm{S}_{j+1,p}) + \log P(\mathrm{c}_q(\mathrm{S}_{j+1,p})|\mathrm{c}_h(\mathrm{S}_{i,j})) > C$, we update this node to $< j+1, p, c_q(\mathrm{S}_{j+1,p}), Q + \log P(\mathrm{c}_q(\mathrm{S}_{j+1,p})|\mathrm{S}_{j+1,p}) + \log P(\mathrm{c}_q(\mathrm{S}_{j+1,p})|\mathrm{c}_h(\mathrm{S}_{i,j})), n+1 >$. Meanwhile, if there are multiple choices, all the valid expansions must be added too.

Pruning step. In beam search, only B_0 nodes are allowed to be kept. If B, the number of nodes, exceeds B_0, we prune the redundant nodes for the sake of efficiency. All the nodes in the list are reordered according to the descending average scores. That is, node $< i, j, \mathrm{c}, Q, n >$ is ranked according to the average score $\frac{Q}{n}$. The nodes with the smallest $B - B_0$ average scores are removed.

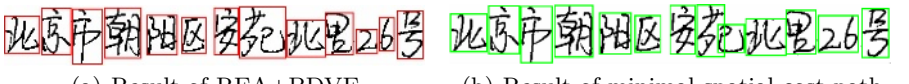
In another implementation, We first apply K-shortest algorithm ([2]) to the segmentation graph in order to generate a set of path hypotheses for evaluation.

If the character images, merged according to a certain path, are denoted by $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_T$ and the recognized characters of the t-th image are $\mathrm{c}_{t,1}, \mathrm{c}_{t,2}, \ldots, \mathrm{c}_{t,M}$, the maximum of H(see Eq.(4)) of this path is computed by Viterbi algorithm. In following procedure, $Q(p,q)$ denotes the accumulative total of the logarithm of the probability value for the most likely string from the start character to $\mathrm{c}_{p,q}$.

Bi-gram Driven Viterbi Evaluation (BDVE)
Step 1. For $1 \le q \le M$, we set $Q(1,q) = \log P(\mathrm{c}_{1,q}) + \log P(\mathrm{c}_{1,q}|\mathrm{x}_1)$.
Step 2. For $2 \le p \le n_k$ and $1 \le q \le M$, we calculate $Q(p,q)$ as follows: $Q(p,q) = \max_{1 \le j \le M} \{Q(p-1,j) + \log P(\mathrm{c}_{p,q}|\mathrm{c}_{p-1,j})\} + \log P(\mathrm{c}_{p,q}|\mathrm{x}_p)$.
Step 3. We output $\frac{\max_{1 \le q \le M} Q(T,q)}{T}$ as the maximum of H.

The maximum number of radicals in a character is usually less than 6. For the worst case, the complexity of REA is at most $O(6N + KN \log 6)$, the complexity of Viterbi algorithm is $O(M^2 T)$. Accordingly, generating K-shortest paths is very fast, however, the size of the candidate set M controls the total time.

In Fig.(3), we make a comparison between REA+BDVE method and the minimal spatial cost method. The proposed method achieves a segmentation rate of 100%, much better than the rate, 77%, given by the latter method. The merging paths recommended by the two methods are also illustrated in Fig.2(e).



(a) Result of REA+BDVE (b) Result of minimal spatial cost path

Fig. 3. Segmentation results comparison

4 Lexicon Based String Segmentation and Recognition

In the above, we try to promote the string recognition rate by improving segmentation and recognition of characters. However, perfect character segmentation may not be necessary, fractional characters in a string could also help in unique identification against a lexicon dictionary. The differences between western languages and oriental ones result in distinct strategies in holistic string recognition. In English, a string image is divided into word images first and then recognized by dictionary matching. However, there are no gaps between characters in Chinese, that means a Chinese string must be dealt with at the same time. Liu([3]) uses a trie structure to organize address lexicons for Japanese address reading. He also adopts a split-merge strategy, determining merging path for radicals, identifying characters for images and matching the text line with a certain lexicon. This process is implemented by beam search from the left-most radical to the end. More than 110,000 items are considered in his method and each lexicon is limited within ten characters. Unlike the sequential characteristic of Liu's method, we propose a novel algorithm that could start from all the substrings. By this adaption, we could hurdle the problem that one segmentation error or one character mis-classification may potentially break down the whole process.

Optimal Substring Alignment Algorithm (OSAA)
$\mathrm{w}_1\mathrm{w}_2 \ldots \mathrm{w}_m$—a given Chinese string
$R[p][q]$—the set of all the substrings that start with w_q and contain p characters

Step 1. For a predefined integer L, we recognize $\mathrm{S}_{i,i+j}$, where $0 \leq j \leq L-1$ and $1 \leq i \leq i+j \leq n$. If the candidate set of $\mathrm{S}_{i,i+j}$ contains w_k, we add $< i, j, k, k, c >$ to $R[1][k]$, where c is the confidence score.
Step 2. After initializing $R[1][k]$ for $1 \leq k \leq m$, for $1 \leq p \leq m, 1 \leq q \leq m-p+1$, we compute $R[p][q]$ as follows: if there exists $< i_1, j_1, k_{1,1}, k_{1,2}, c_1 >$ in $R[p-1][q]$ and $< i_2, j_2, k_{2,1}, k_{2,2}, c_2 >$ in $R[1][p+q-1]$ that satisfy $i_2 = j_1 + 1$, we then add a new element $< i_1, j_2, k_{1,1}, k_{2,2}, \frac{c_1 \times (p-1) + c_2}{p} >$ to $R[p][q]$.
Step 3.1. For a given threshold D, we select all the substrings $< i_l, j_l, k_{l,1}, k_{l,2}, c_l >, 1 \leq l \leq B$ from $R[p][q], p \geq D$ satisfying $1 \leq i_1 \leq j_1 < i_2 \leq j_2 < \ldots < i_B \leq j_B \leq n$ and $1 \leq k_{1,1} \leq k_{1,2} < k_{2,1} \leq k_{2,2} < \ldots < k_{B,1} \leq k_{B,2} \leq m$.
Step 3.2. For the selected substrings, we traverse all the characters and their possible corresponding radical combinations which are not covered by the substrings. By this means, each character in the given string could find its corresponding radicals. Then we calculate the value of a certain cost function which is designed to evaluate both dissimilarities between images and characters and differences between characters and the given lexicon. In step 3.1 and 3.2, we simply apply depth-first search to traverse all the possible cases.

Comparing with the bi-gram driven way, lexicon driven method is seen as a more strict rule to control both radical merging and candidate selection process.

5 Experiments and Discussions

We collect 1141 handwritten Chinese address lines including 14,970 characters written by different people. In bi-gram based case, we test both BDBS and REA+BDVE implementations. We found that BDBS may be attacked by a intervened error and thus result in poorer performance comparing with REA+BDVE (in Table 1, we adopt Eq.(6) for recognition confidence estimation, $K = 15N, M = 10$). In the following, we will mainly discuss REA+BDVE.

Table 2 compares confidence estimation methods (Eq.(5)&Eq.(6))and selection strategies of K. Different ways of estimation don't result in obvious distinctions. We are inclined to select K according to the number of the radicals. For a text line with fewer radicals, this adaption will save time, on the other hand, more paths will be examined if a text line has more radicals. In the following experiments, we use $K = 15N$ and Eq.(4) without specification.

Table 1. Comparison of BDBS and REA+BDVE

	BDBS ($B_0 = 40$)	REA+BDVE ($K = 15N, M = 10$)
Segmentation rate (%)	82.75	92.53
Time (s)	24.7	0.5

We then compare different factors in segmentation. In the columns of Table 3, we test the minimal spatial cost path, the maximal recognition confidence path, the minimal recognition distance path (i.e., we assign each arc the confidence/distance of the best candidate given for the image associated with the arc) and the best contextual ranked path (i.e., recognition confidence is omitted) respectively. Noticing the results from different criteria, contextual relation is most important.

Generally, we select 10 candidates for each character image in recognition (i.e. $M = 10$), since the size greatly affects the computation time of Viterbi. As shown in Table 4, extending the candidate set size will not improve the results obviously, however, brings about a rapid increase in time consumption.

Table 2. REA+BDVE segmentation results

	(5.1)	(5.2)	(5.3)	(6)
$K = 200, M = 10$	91.34	91.73	92.62	92.20
$K = 15N, M = 10$	91.50	91.99	92.89	92.53

Table 3. Analysis of different factors in segmentation

	Spatial cost segmentation	Recognition confidence segmentation	Recognition distance segmentation	Contextual relationship segmentation
Correct rate (%)	81.94	53.46	3.13	90.61

Table 4. Candidate set size for segmentation

	$M = 5$	$M = 10$	$M = 50$
Average time for Viterbi per text line (ms)	37	137	3368
Character segmentation rate (%)	92.02	92.53	92.91

Table 5. Right path distribution in the K-shortest paths

$K = 200$	$K = 400$	$K = 600$	$K = 800$	$K = 1000$
81.69	86.25	88.17	89.48	90.27

Table 6. Correct rate for different numbers of candidate paths

	$K = 200$	$K = 500$	$K = 1000$
Segmentation rate (%)	92.20	92.59	92.63
Total time (ms)	410	622	861

Table 7. Bi-gram driven segmentation results for a general document

Segmentation rate (%)	Recognition rate (%)
92.9	84.9
88.2	79.0

We cannot assure that the right answer must be in the candidate set, though K is very large. In Table 5, we give the rate of the correct merging path in the first K paths. However, sub-optimal paths are always included in the K-shortest paths, which are applicable to achieve a acceptable segmentation rate, so enlarging K will not benefit the segmentation rate remarkably (Table 6).

In the proposed method, we use an averaged score considering different number of characters merged, otherwise, if we don't take the difference into consideration as in [7], we may encounter a little drop in the segmentation rate. For example, if H is replaced by $\tilde{H} = \log P(\mathrm{c}_{1,k_1}) + \sum_{i=2}^{T} \log P(\mathrm{c}_{i,k_i}|\mathrm{c}_{i-1,k_{i-1}}) + \sum_{i=1}^{T} \log P(\mathrm{c}_{i,k_i}|\mathrm{x}_i)$, the segmentation rate drops from 92.5% to 90.0%.

The above experiments utilize the bi-gram training on the address lexicons of Beijing. If we turn to a more general bi-gram, trained on "People's Daily" of 2000 (covering politics, economics, science and etc.), we get a segmentation rate of 84.59%. Comparing with the rate of 92.53%, the general bi-gram weakens the contextual relationship of specific documents and degrades the performance, however, the result exceeds the performance of minimal spatial cost path.

Furthermore, we extend our idea to a more general case. We collect some text lines containing 238 characters from a technical document, by using the bi-gram of "People's daily", both segmentation and recognition rates get improved (Table 7).

OSAA is designed for lexicon driven holistic string recognition. We use 500 handwritten address lines and a database containing more than 370,000 lexicons in our experiments and achieve a string recognition rate of 86.8%.

6 Conclusions and Discussions

This paper presents a context driven way for unconstrained off-line handwritten Chinese characters segmentation. Unlike the previous techniques based on low level information, we pay more attention to the application of contextual knowledge in this process, which may be more useful as revealed by the experiments.

Contextual information could effectively determine how to merge the radicals, however, low level information is essential to get these radicals. Noticing that there have been many papers considering various techniques based on low level pre-segmentation to remove touching and overlapping for Chinese scripts, our method could be easily extended to the merging step of those methods.

Acknowledgements. This work has been funded by Siemens AG under contract number 20030829 - 24022SI202. The author would also thank to the anonymous reviewers for their kindly and helpful advice.

References

1. Richard G. Casey, Eric Lecolinet: A Survey of Methods and Strategies in Character Segmentation. IEEE Trans. PAMI **18**(7), (1996) 690–706
2. Víctor M. Jimenez and Andrés Marzal: Computing the K shortest paths: A new algorithm and an experimental comparison. *Proc. 3rd WAE*, 1999, 15–29. LNCS vol. 1668. Springer
3. Chenglin Liu, Masashi Koga, Hiromichi Fujisawa: Lexicon-driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading.IEEE Trans. PAMI **24**(11), (2002) 1425–1437
4. Chenglin Liu, Masaki Nakagawa: Precise Candidate Selection for Large Character Set Recognition by Confidence Evaluation.IEEE Trans. PAMI **22**(6),(2000) 636–642
5. Stefano Messelodi, Carla Maria Modena: Context Driven Text Segmentation and Recognition. Pattern Recognition Letters **17**(1), (1996) 47–56
6. Junliang Xue, Xiaoqing Ding, et al: Location and Interpretation of Destination Addresses on Handwritten Chinese Envelopes. Pattern Recognition Letters **22**(6),(2001) 639–656
7. Takahiro Fukushima, Masaki Nakagawa, On-line Writing-box-free Recognition of Handwritten Japanese Text Considering Character Size Variations, *Proc. 15th ICPR*, 359–363.
8. Xiaofan Lin, 1998. Theory and Application of Confidence Analysis and Multiple Classifier Combination in Character Recognition. Ph.d. dissertation, Tsinghua University.
9. Yuanxiang Li, 2001. The Research on Chinese Character Recognition Using Contextual Information. Ph.d. dissertation, Tsinghua University.