Pattern Recognition 42 (2009) 167 – 182

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Off-line recognition of realistic Chinese handwriting using segmentation-free strategy

Tong-Hua Su*, Tian-Wen Zhang, De-Jun Guan, Hu-Jie Huang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, PR China

ARTICLE INFO

Article history:
Received 26 August 2007
Received in revised form 7 May 2008
Accepted 13 May 2008

Keywords:
Optical character recognition
Chinese handwriting recognition
Sliding window
Hidden Markov Model
Segmentation-free strategy
Classifier combination

ABSTRACT

Great challenges are faced in the off-line recognition of realistic Chinese handwriting. This paper presents a segmentation-free strategy based on Hidden Markov Model (HMM) to handle this problem, where character segmentation stage is avoided prior to recognition. Handwritten textlines are first converted to observation sequence by sliding windows. Then embedded Baum–Welch algorithm is adopted to train character HMMs. Finally, best character string maximizing the a posteriori is located through Viterbi algorithm. Experiments are conducted on the HIT-MW database written by more than 780 writers. The results show the feasibility of such systems and reveal apparent complementary capacities between the segmentation-free systems and the segmentation-based ones.

Published by Elsevier Ltd.

1. Introduction

Many advances have been achieved in Chinese handprinted character recognition since the 1980s [1–3]. In general, the recognition task is decomposed into four stages in literature (see Fig. 1). As for handwriting normalization, not only dozens of nonlinear methods are presented to remedy the variability of strokes [4,5], but also alternative schemes, such as elastic cell [6] and global transformation [7], are developed to avoid the zigzag effect. Quantity of effective feature extraction methods are contributed, for example, four plane features (FPFs) [8], directional element features [9], Gabor features [10,11], gradient histogram features [12], to capture the nature of Chinese character from different perspectives. As regards to character classifiers, support vector machine (SVM) [13], Hidden Markov Model (HMM) [14], quadratic discriminant function (QDF) [2,15], and structure matching [16] are explored and pleasing advances are observed. Recently, character-based and word-based language models [17,18] are available and show promising results.

These recognition algorithms are all trained on isolated-character databases, such as ETL-8/ETL-9 [19,20], IAAS-4M [21], HCL2000 [22]. However, strict restrictions are posed on the writers during collecting their character samples. Each participant is requested to write a large set of Chinese characters (e.g. at least 3755 characters concerning IAAS-4M and HCL2000), and write each character carefully in a preprinted character box. As a result, characters present little writing variability, and the same quantity of and enough instances per character (at least 100 samples per character) are available no matter whether that character is frequently used (refer to Ref. [23] for more thorough discussion). The restricted character style in isolated-character databases affects the experimental setup.

Current experiments on Chinese handprinted character recognition fall into two categories. Mostly, the evaluation is performed on the same database as what has been used to train character models [8–16]. The performance under this setup is unavoidably overestimated because it only considers the ideal and simplified situation, and pays no attention to handwriting complexities, such as the character touching, textline skewness.

In the other case, the recognizer is tested on high quality handwriting, though trained on isolated-character databases [17,18,24]. To identify the textline image, a character segmentation stage is employed first, and then feature selection and pattern classification are executed. Here, an ill-posed request should be fulfilled: the characters are written discretely, with no touching or overlapping, etc. However once character touching or conjunction are encountered, the character segmentation stage becomes the obvious bottleneck to the whole system.

Since the former is just to verify the matching algorithm, it can be seen as a special case of the latter. According to the definition in Ref. [25], they can be classified as segmentation-based strategy. In other words, the state-of-the-art recognizers for Chinese handwriting usually employ a character segmentation stage or the characters should be separated before the handwriting is sent into the recognition

* Corresponding author. Tel.: +86 451 86419692.
E-mail addresses: tonghuasu@hit.edu.cn (T.-H. Su), twzhang@hit.edu.cn (T.-W. Zhang), guandejun@hl.chinamobile.com (D.-J. Guan), hjh@hit.edu.cn (H.-J. Huang).

0031-3203/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.patcog.2008.05.012

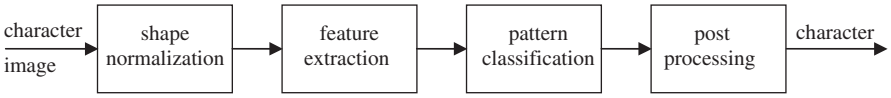

Fig. 1. The flowchart of Chinese handprinted character recognition. Most works fall into at least one of those four categories.

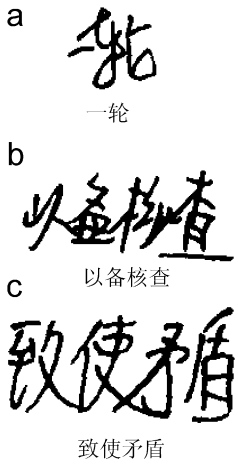

Fig. 2. Complex relationships between adjacent characters: (a) severe overlapping, (b) touching and (c) crossing. These handwritten phrases are extracted from HIT-MW database [23] and the ground truth of each phrase is placed under it.

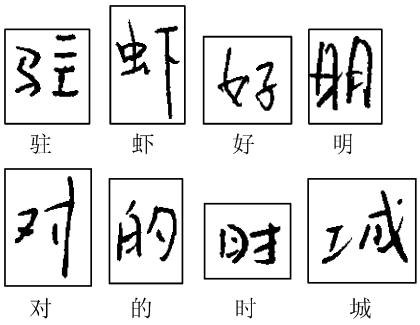

Fig. 3. Chinese characters with left–right structure. Each radical is also a valid character in itself. The ground truth is placed under each character box.

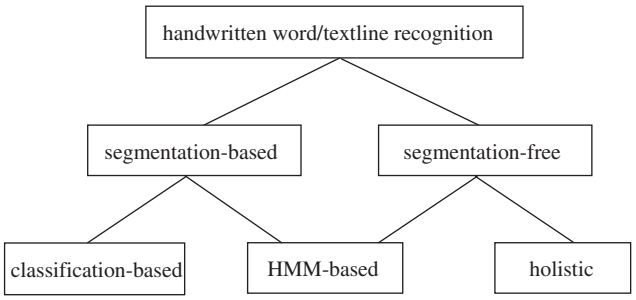

Fig. 4. The method hierarchy of handwritten word/textline recognition. As for Chinese handwritten character recognition, HMM is merely used in segmentation-based systems till now.

algorithms. Such recognizers may work smoothly on high quality Chinese handwriting. When the realistic one is fed, however, the results may deteriorate dramatically [26]. Great challenges, therefore, remain in the reliable recognition of the real-life Chinese handwriting.

1.1. Difficulties in the recognition of realistic Chinese handwriting

In fact, real-life Chinese handwritten documents may present complex textlines, instead of easily separated characters. Generally, there exist three hierarchies of complexities. At the document level, there are overlapping, touching, and crossing between adjacent textlines. Moreover, at the textline level, besides undulation and skewness of textline, overlapping, touching and crossing among character neighbors may also present. In addition, at the character level, the variable size of characters, deformation of strokes and even erasure of characters may exist. Present paper only covers the last two kinds of complexities.

These complexities pose great obstacles to recognizers. The substantial problems they face can be coarsely dichotomized as follows:

Ambiguity of segmentation. Commonly, realistic handwriting is written in cursive style and there is no extra gap between adjacent characters than that between radicals of the same character. If overlapping, touching, crossing or noise are presented, the determination of the right segmentation path is nontrivial (see Fig. 2). Even when it runs discretely, new problem–how to merge the over-segmented parts of Chinese character—arises to left–right structure character (see Fig. 3).

Intricacy of modeling. The writing style may vary from person to person. Even worse, character may differ much when produced by the same writer in different times, places or environments. In addition, there are many characters with negligible distinction (see Ref. [3, Fig. 5]). Two kinds of efforts have been made to enhance the modeling precision of Chinese character in the state-of-the-art strategy: reduce the variability at image level, for example, using shape normalization, and absorbing the distortion at abstract level resorting to robust feature representation, powerful classifier or contextual information. However, novel modeling techniques and perspectives are still in great need to the realistic handwriting recognition.

1.2. Related work

Parallel to segmentation-based systems, there are segmentation-free ones for the recognition of English text. Their relationships are depicted in Fig. 4. Just as Sayre's paradox goes [27], character segmentation is prone to error and difficult to make correction afterwards. The segmentation-free systems do not explicitly segment the textline into characters or graphemes in order to tackle the huge difficulties in the character segmentation. One kind of them is holistic word recognition [28]. Features are extracted to capture the word image as a whole and the underlying word is identified by exhaustively search a prior lexicon. This kind of systems is mainly used in a small lexicon or constrained domain, such as in check amount reading and address phrase interpretation.

The other is word/textline recognition where an HMM is adopted as the engine. A good review on word recognition is available in Ref. [29] and some combination strategies are developed, for instance, integration of recognition and verification [30], combination of segmentation-based system and segmentation-free system [31], ensemble of classifiers [32]. As for textline recognition, two typical systems for English handwriting with a large vocabulary are described in Refs. [33,34], respectively. The method employs the similar framework to continuous speech recognition [35]. In the training

stage, the initial character models are linked together and optimized by embedded Baum–Welch algorithm. During recognition, the recognizer outputs a character string of maximum probability through Viterbi algorithm [36]. Such strategy possesses desirable characteristics. For instance, it avoids the crisp character segmentation stage and extra language information can be easily incorporated. In Ref. [37], this kind of system is claimed as implicit segmentation strategy. In our study, the term "segmentation-free strategy" will be used consistently to stress on the primary aim of the maximizing the string probability criteria.

Extending this framework from speech recognition to cursive English handwriting recognition may be straightforward, due to their one-dimensional nature [38]. However, as for Chinese handwriting, no evidence has been provided yet since Chinese character has typical two-dimensional structure, and even worse, there are a large set of characters to be distinguished from each other.

As shown in Fig. 4, HMM method can be used in segmentation-based systems too. Herein, we merely focus our interest on the Chinese character recognition field. To our knowledge, HMM method is first applied to the Chinese printed character recognition in 1990 [39]. To each predefined character class, seven states with left-to-right transition were given. The underlying distribution is a discrete type. The codebook in this case composes of 64 entries, whose index encodes the projection values in horizontal and vertical directions. Similar system is adapted to recognize the handprinted Chinese character [14]. Its feature is extracted with a column-wise or row-wise manner, which is to simulate the time variable in speech recognition. The codebook is generated on feature vectors of training data by a K-means clustering [40]. Instead of codebook, a mixture of multivariate Gaussian probability density functions is used to represent the output probability in Ref. [10]. In their implementation, the character image is partitioned into horizontal (vertical) slices, and within each slice, Gabor features are extracted and their deviations are integrated. As for the HMM structure, eight states and four Gaussian components are given. The above HMM-based Chinese character recognizers are all trained and later tested on isolated character database, and the results are pleasing.

1.3. Our motivation

We attempt to recognize Chinese handwriting with a large vocabulary from segmentation-free strategy in present paper. This investigation is motivated by the following observations:

Low expenditure in preparation of training data. Labeling each character in training data, which is a tedious and time-consuming process, is passed by in segmentation-free systems. In the segmentation-free strategy, we only input the textlines and their underlying character string. The system automatically aligns each character to its image counterpart and then estimates the model of that character. It simplifies the expansion of training data as well as quickens the experimental process.

Desirable features of different origins. Segmentation-free systems usually adopt sliding windows moving along the textline without regard to the boundary of underlying characters, while segmentation-based ones extract features within each character segment. Their different origins result in different abilities in characterizing the character pattern, which plays important role in the system combination. For example, segmentation-based systems can express the certain class more elaborately, especially in vertical direction; segmentation-free ones can utilize the conjunction relationship between adjacent characters in a simple way.

Intrinsic advantages of the strategy. Segmentation-free strategy performs character segmentation and recognition in one step, and outputs a best transcription in probabilistic criteria. Compared with the segmentation-based strategy, it can utilize both the boundary information of sliding windows and the knowledge of character models, and avoid the blind character segmentation. Moreover, due to its solid mathematic foundation and powerful dynamic capacity, HMM can properly absorb the writing variability [30]. In addition, language models can be incorporated conveniently [33,34], though that is out of the scope of this paper.

Extra options to multiple classifier combination research. Given the huge challenges in the recognition of Chinese handwriting, as in the English word recognition task, multiple classification methods or properly combining a variety of classifier outputs may be the future trends. Segmentation-free strategy is devised from overall different perspective to the segmentation-based one and may serve as a desirable complementary strategy.

Unlike previous Chinese character recognition setup, an unconstrained Chinese handwriting database, which is written by more than 780 writers, is used as experimental dataset [23,41]. The experiments show the promising results of the segmentation-free strategy as well as the apparent evidence on the complementary capacities between the segmentation-free strategy and the segmentation-based one. We hope that this paper will encourage more researchers to re-examine the task and eventually advance the realistic Chinese handwriting recognition.

The paper is organized as follows. The next section briefly describes the HIT-MW database. Section 3 details components of our system including sliding window, feature extraction method and HMM modeling. Experiments are conducted and results are summarized in Section 4. Finally, discussions and conclusions are drawn from this work in Sections 5 and 6, respectively.

2. Database

The benchmark data used in this paper come from the HIT-MW database [41]. It is collected from more than 780 participants with an unconstrained manner. The writers are mainly college students and the department distribution and gender distribution of them are near to those of real distributions of college students. The underlying texts of the HIT-MW database are randomly sampled from China Daily corpus with a systematic way. As a result, a high character coverage on China Daily corpus (with approximately 80 million characters) is obtained (the coverage rate is 99.33%). The HIT-MW database can be seen as a representative subset of real Chinese handwriting (Ref. [23] for more details).

Currently, the HIT-MW database provides 5667 textlines which can be used freely. To partition the data into train, validation and test sets, we consider following procedure (the HIT-MW database and the experimental data in this paper are available at: http://hitmwdb.googlepages.com).

Step 1: Randomly select 383 textlines as test set (8471 characters).

Step 2: Within the 5284 remainder textlines, discard the textlines which are written by the same writer of test set. By this way, only 3172 textlines are retained for the following processing.

Step 3: Similar to step 1, draw 189 textlines uniformly out of 3172 ones as validation set (4100 characters).

Step 4: Further discard textlines written by the same writer of validation set (2306 textlines are left).

Step 5: Among the 2306 textlines, randomly select 953 ones as train set (20,701 characters).

This process is to reproduce a realistic scenario where the handwritten textlines for recognition in test set have not been seen before (in train or validation set). The validation set here is used to tune the parameters, such as the number of Gaussian components in mixture density and training iterations, which will be used to evaluate the recognizer's performance on test set. Some textlines selected from test set are demonstrated in Fig. 5.

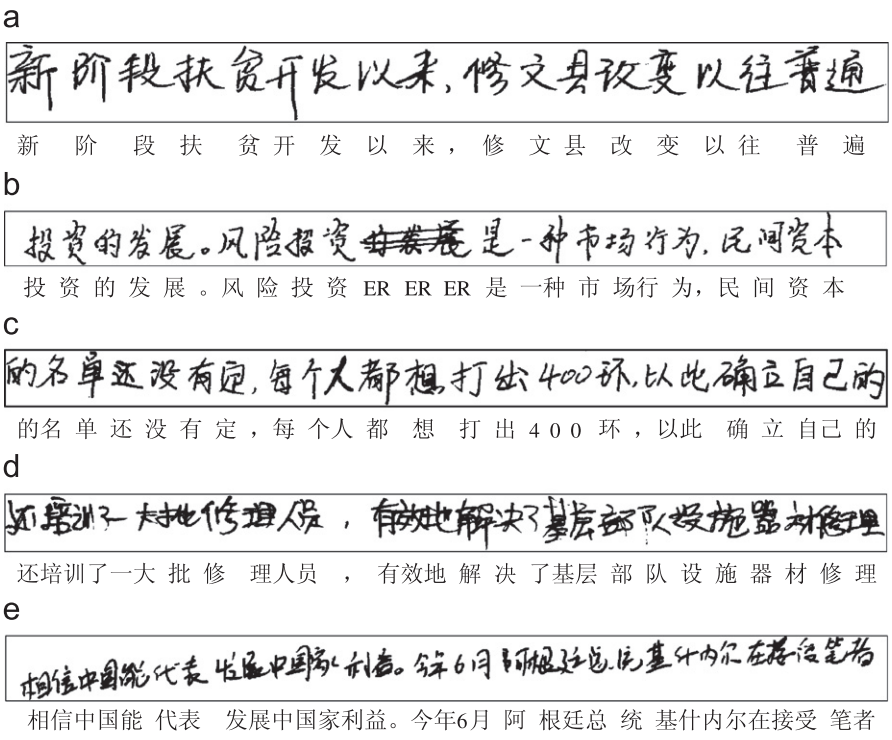

Fig. 5. Some complex textlines in test set which pose great challenges to the state-of-the-art segmentation-based systems. The ground truth of each textline is placed under it. (a) high-quality textline with long downwards strokes and a little stroke overlapping; (b) textline with three instances of erasures (represented with ER); (c) textline with digit, punctuation and Chinese characters; (d) textline with severe stroke touching, also with broken strokes; (e) textline with undulation and skewness.

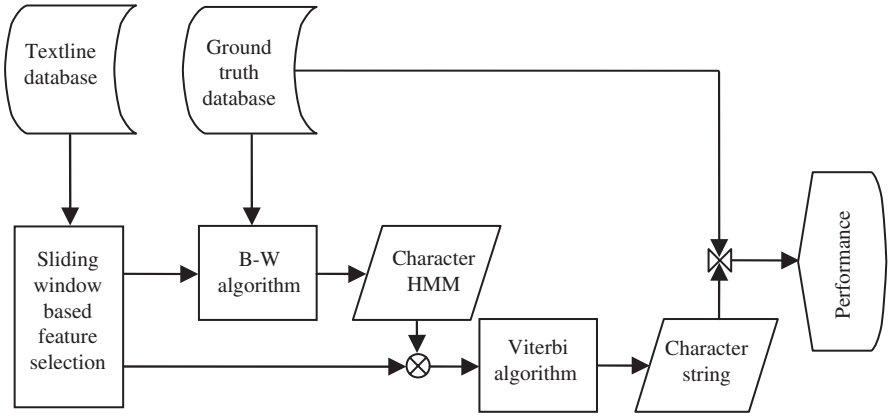

Fig. 6. System architecture. The ground truth database is the reference text of textline database, and it is used not only in the training stage but also in the performance evaluation stage; the character HMMs are generated by embedded Baum–Welch algorithm (B–W algorithm) on training and validation sets after the sliding window-based feature selection stage; using Viterbi algorithm, the test set is mapped into the symbol space.

3. System descriptions

When a textline image is fed into a recognizer, it is converted to a sequence of feature vectors or observations $\mathbf{O}=\mathbf{o}_1,\ldots,\mathbf{o}_m$. The recognition task is to identify a string of characters $\hat{S}=s_1,\ldots,s_n$ maximizing the a posteriori (MAP) $P(S|\mathbf{O})$. The recognizer described in this paper consists of three main components: sliding window, feature extraction and selection, HMM training and decoding. The system architecture is illustrated in Fig. 6.

3.1. Sliding window

Windowing is a common localization technique in signal processing domain. Since the character string runs in certain direction, a movable window called sliding window following the same order can be used to draw an interested zone and then extract features within it. Generally, the height of the sliding window is the same as that of textline. The other two parameters, the width W and the shift step S, should be assigned by researchers or determined through experiments. Supposing $f(p,q)$ is the digital image of a textline, the running mechanism of the sliding window is illustrated in Fig. 7. The window function can be expressed as

$$g(w,v)=\begin{cases}1, & w=1,\ldots,W,\\ 0 & \text{otherwise.}\end{cases} \tag{1}$$

Then, the window at step i can be denoted as

$$f_i=f((i-1)S+w,v)\cdot g(w,v), \tag{2}$$

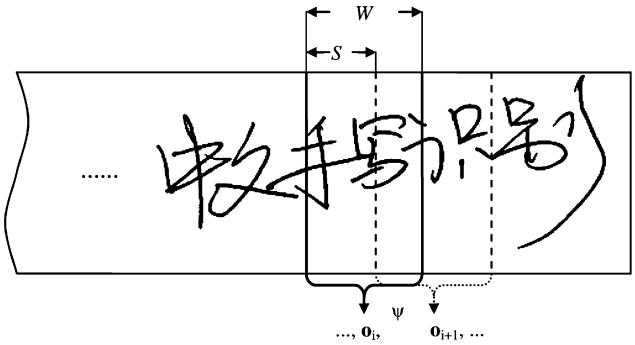

Fig. 7. Feature selection process using sliding window. $(W-S)$ columns are shared between adjacent windows in ith step and $(i+1)$th step (marked with solid and dashed lines, respectively).

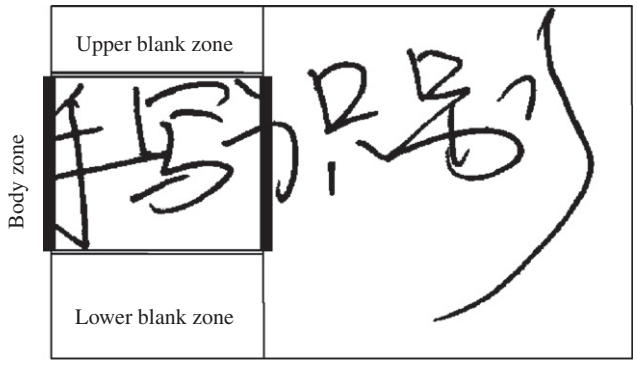

Fig. 8. The zone used to extract features. The blank parts in the upper and lower window are excluded.

where $\cdot$ is the multiplication operator. The feature vector at step i, $\mathbf{o}_i$, in observation sequence $\mathbf{O}=\mathbf{o}_1,\ldots,\mathbf{o}_m$ is calculated as follows:

$$\mathbf{o}_i=\psi(f_i), \tag{3}$$

where ψ is the feature mapping function from image space to feature space.

The sliding window used in Refs. [31,33] is of one pixel width and one pixel step. There is no overlap between adjacent windows. Another kind of sliding window with step of one pixel and width of 10 pixels is given in Ref. [34]. Obviously, nine-tenth contents of two consecutive windows are identical. In addition, [42] presents a bit adaptive sliding window relative to the height of textline: the width is of one-fifteenth its height and the step of one-third its width. This paper sets $W=12$ pixels (one-fifth the character's average width) and $S=2$ pixels (one-sixth its width). The window width W is selected to make a balance between the HMM structure and the smallest character width, and the tuning process of window step S is detailed in Ref. [43].

3.2. Feature representation

To resist the undulation of textlines, we partition the window into three zones, as shown in Fig. 8 and only the body zone is used to extract feature vector instead of the whole window. The body zone is separated by the topmost and bottommost foreground pixels in vertical direction.

Our baseline system is based on cell features. The window is divided into 8×2 cells, and then the foreground pixels are summed in each cell. We also adapt another feature representation method, FPF [8], which is originally used in Chinese isolated-character recognition. Each character image is scanned in four directions (horizontal, vertical, right diagonal and left diagonal) and only the strokes longer than a threshold are retained. Then each plane is divided into cells within which stroke crossing can be counted. Some modifications of the original algorithm are needed to fit the problem of handwriting recognition: (1) the feature vector is extracted from certain sliding window instead of character segment; (2) the average stroke width SW is estimated on whole textline by the analysis of stroke histogram (as in Ref. [44]) rather than on single character by contour following; (3) the four planes are formed by excluding the strokes smaller than $2\times SW$. We use this 64-dimensional feature vector in this paper to improve the performance of the baseline system. Fig. 9 shows the extraction process of FPFs. Further, we incorporate the previous two features as an extended 80-dimensional feature vector to enhance the recognizer.

In addition, principal component analysis (PCA) is adopted. The covariance matrix is calculated from train and validation sets and 36 out of 80 principal components are retained according to the criteria declared in Ref. [45] as a compact representation of previous mentioned 80-dimensional fused features. Further information in this aspect is available in Ref. [46].

3.3. HMM training and decoding

Supposing the sequence of feature vectors corresponding to handwritten textline is $\mathbf{O}=\mathbf{o}_1,\ldots,\mathbf{o}_m$, the task of the recognizer is to identify a character string, $\hat{S}=s_1,\ldots,s_n$, by MAP:

$$\hat{S}=\arg\max_S P(S|\mathbf{O}). \tag{4}$$

Using Bayes' theorem, we get

$$\hat{S}=\arg\max_S \frac{P(\mathbf{O}|S)P(S)}{P(\mathbf{O})}. \tag{5}$$

Since $P(\mathbf{O})$ is the a priori of feature vectors and independent of S, above equation is equivalent to

$$\hat{S}=\arg\max_S P(\mathbf{O}|S)P(S). \tag{6}$$

We call $P(\mathbf{O}|S)$ the string model and $P(S)$ the language model. The former encodes the probability of feature vectors $\mathbf{O}$ under character string S and the latter constrains the search space. Provided that $\mathbf{O}$ is of conditional independence, $P(\mathbf{O}|S)$ can be further approximated as

$$P(\mathbf{O}|S)\approx\prod_{i=1}^{n}P(\mathbf{O}_i|s_i), \tag{7}$$

where $\mathbf{O}_i$ is the observations of character s_i. The $P(\mathbf{O}_i|s_i)$ is the character HMM and estimated by embedded Baum–Welch algorithm in this paper. When $P(\mathbf{O}|S)$ and $P(S)$ are in order, the best character string in the MAP sense (as in Formula (6)) can be located by Viterbi algorithm [36]. Note that the meaning of $\mathbf{O}_i$ is different from $\mathbf{o}_i$ in that $\mathbf{o}_i$ is derived from certain sliding window and $\mathbf{O}_i$ is a series of $\mathbf{o}_i$ which is the counterpart of a character. Their relationship can be characterized as

$$\mathbf{O}=\mathbf{o}_1,\ldots,\mathbf{o}_m=\mathbf{O}_1,\ldots,\mathbf{O}_n,\quad m\geqslant n. \tag{8}$$

An HMM is an extended Markov chain. The transitions between states represent the shifts of character segments and each state of the chain associates a probability density spanned on a d-dimensional feature space. A continuous-density HMM (CDHMM) can be notated as $\lambda=(\mathbf{A},B,\Pi)$. $\mathbf{A}=(a_{ij})$ is the state transition probability. $B=\{b_j(\mathbf{o})\}$ is the observation probability. Further, the probability of observation $\mathbf{o}$ for state j, $b_j(\mathbf{o})$, can be approximated by a finite mixture of Gaussian density of the form:

$$b_j(\mathbf{o})=\sum c_{jk}\Omega(\mathbf{o},\mu_{jk},\Sigma_{jk}),\quad 1\leqslant j\leqslant N, \tag{9}$$

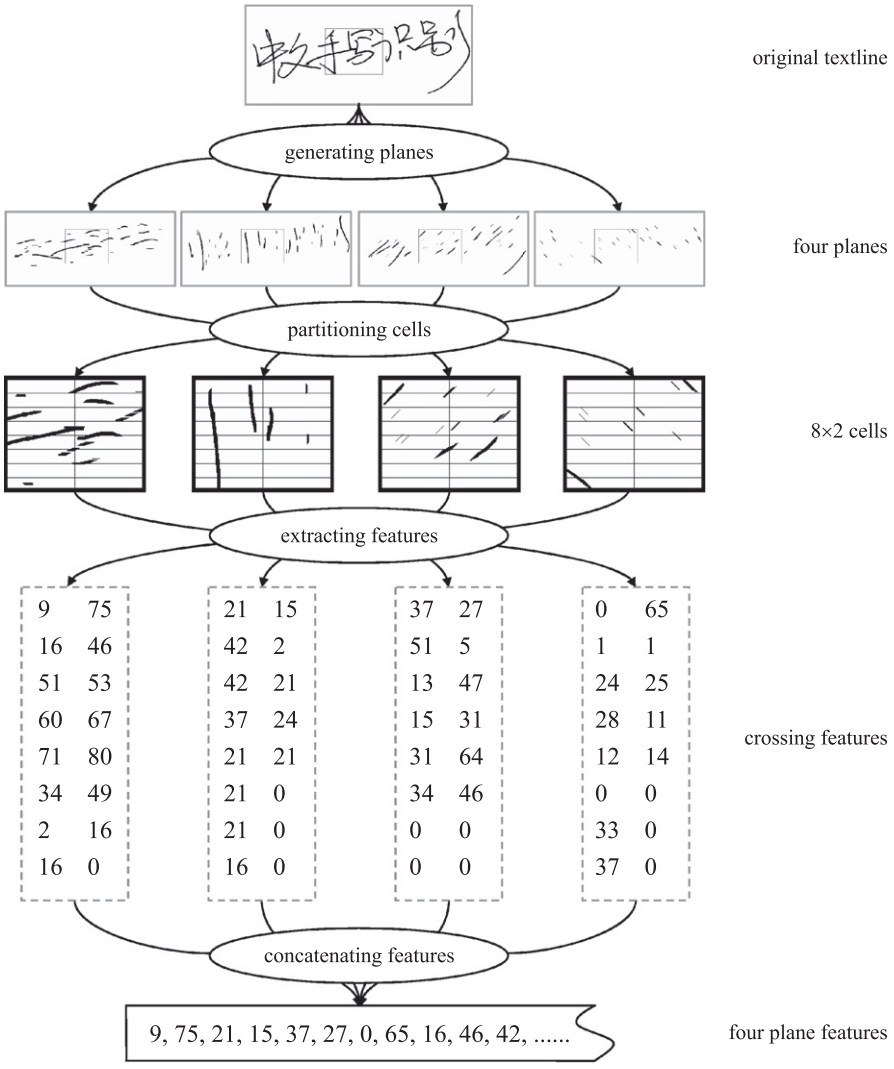

Fig. 9. The process of extracting 64-dimensional four plane features. The sliding window plotted here is larger than the actual size to give a clear illustration.

where N is the total states, $\mathbf{o}$ a d-dimensional vector, c_{jk} the mixture coefficient and μ_{jk}, Σ_{jk} the mean vector and covariance matrix for Gaussian distribution Ω, respectively. Π is the initial parameter set concerning the HMM topography, mean vector and covariance matrix in each state, mixture coefficients etc.

Our recognizer models each Chinese character class as a 11-state CDHMM, while a four-state CDHMM is given to the character class in digit or punctuation. The structure of the HMM is a Bakis form (left-to-right, with no skip). The initialization of parameters $\mathbf{A}$ and B, is done by a flat start. Before detail the technique aspect of embedded Baum–Welch algorithm, we visualize the key factors and explain their roles in the training process as shown in Fig. 10. To simplify the expression, one entry state and one exit state are added to each character HMM. However, output probability is not associated to them. From Fig. 10, we can see that character HMMs are embedded in sentence HMMs. The parameter set of sentence HMM is updated during training and the character HMMs in the sentence HMM will be renewed simultaneously.

Once the initialization of λ's is done, the embedded HMMs are re-estimated by embedded Baum–Welch algorithm. Suppose the training data (sequence of textline observations) is denoted as $\mathbf{O}^r(1 \leqslant r \leqslant R)$, N_q is the number of states of the qth HMM in a certain sentence HMM. As the estimation of output distribution can be easily derived from the estimation formula of Baum–Welch algorithm [36], we only provide the re-estimation of transition matrix as follows:

$$\hat{a}_{ij}^{(q)} = \frac{\sum_{r=1}^{R}(1/P_r)\sum_{t=1}^{T_r-1}\alpha_i^{(q)r}(t)a_{ij}^{(q)}b_j^{(q)}(\mathbf{o}_{t+1}^r)\beta_j^{(q)r}(t+1)}{\sum_{r=1}^{R}(1/P_r)\sum_{t=1}^{T_r}\alpha_i^{(q)r}(t)\beta_i^{(q)r}(t)} \quad (i \neq 1, j \neq N_q), \tag{10}$$

where P_r is the probability of the rth observation, and $\alpha_i^{(q)r}(t)$, $\beta_j^{(q)r}(t+1)$ are the forward and backward probability, respectively. The transition updating of entry state and exit state is trivial and omitted here.

The forward probability at time t, $\alpha_j^{(q)r}(t)$, can be calculated in a recursive way:

$$\alpha_j^{(q)r}(t) = \left[\alpha_1^{(q)r}(t)a_{1j}^{(q)} + \sum_{i=2}^{N_q-1}\alpha_i^{(q)r}(t-1)a_{ij}^{(q)}\right] b_j^{(q)}(\mathbf{o}_t^r) \quad (t > 1, j \neq 1, j \neq N_q). \tag{11}$$

Similarly, backward probability can be expressed.

In the recognition phase, the character models are concatenated to strings [34]. Currently there is no extra language information incorporated yet. Since any character can occur at any position, a string

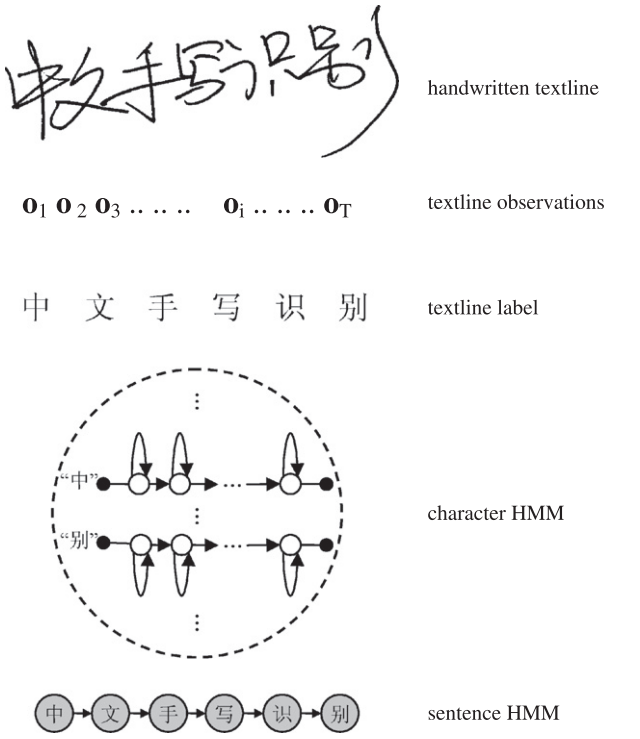

Fig. 10. The factors in the running process of embedded Baum–Welch algorithm. Handwritten textline is first mapped to textline observations by feature extraction function. Character HMMs are often coarsely initialized and concatenated to form the sentence HMM at the guidance of textline label. As the last step, embedded Baum–Welch algorithm will update the character HMMs using textline observations.

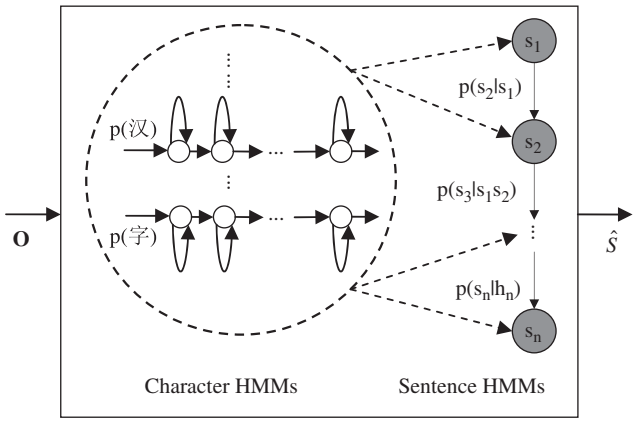

Fig. 11. Recognition process is to map features to symbols. Here, h_n is the history of s_n.

network can be formed. The best path is found out through Viterbi algorithm by a MAP criterion [36]. This phase actually maps the textline image to a character string (see Fig. 11). There is no explicit segmentation of the textline into characters, though the soft segmentation is delivered as a byproduct in recognition phase.

4. Experimental results

4.1. Experimental setup

In our recognizer, an HMM per character is given and all erasures (three erasures are presented in the second line of Fig. 5) are modeled as one HMM. In addition, to model the large space between characters, we add a "blank" HMM. Eventually, there are 2075 character models. The results are counted excluding English letters, due to their too limited occurrences in Chinese environment (13 letters with only 20 instances in all).

We use an incremental way to enhance the segmentation-free systems. As mentioned previously, the baseline system uses a 16-dimensional feature vector, which is derived from cell features (labeled as 16DCELL). Then we replace the 16-dimensional cell features with 64-dimensional FPFs, 80-dimensional fused features (the corresponding systems are denoted as 64DFPF, 80DFUS, respectively) to study the effect of different features. Moreover, a dimension reduction process by PCA is evaluated (the system is labeled as 36DPCA). Finally, a grand variance tying method (refereed as 36DGVS) which is originally used in speech recognition is taken into consideration. Grand variance system is a typical data sharing method to alleviate the data insufficiency and it means that all HMMs share the same Gaussian variance. When re-estimating the variance, training data which would have been used for each of original unshared variance are deposited. As a result, more reliable estimation of Gaussian density can be obtained.

Segmentation-based systems are also inspected. All of them are trained and tested on the same data as in segmentation-free systems, since our main concern is to investigate the complementary abilities between segmentation-free and segmentation-based systems. As stated in Section 1, most state-of-the-art recognizers can only deal with isolated characters. Therefore, we briefly describe the character extraction process before the specifications of segmentation-based systems are clarified.

To the characters in train and validation sets, we extract them manually. Here we only use linear paths, considering the sliding window adopted in segmentation-free systems uses a linear boundary. A character segmentation method similar to Ref. [24] is adopted to segment the characters in test set before classifier runs. However, to simplify the process, only the basic segmentation and the fine segmentation stages are used to generate the candidate paths. The path with minimal variance is selected as the final segmentation path. Parameters in the basic segmentation stage are 17, 13, 9, 5 and 1 pixel(s).

Modified quadratic discriminant function (MQDF) has become one of the most leading classifiers due to its advantages in digit and Chinese character recognition. Multiple-prototype template matching (MPTM) classifier is also evaluated besides the MQDF. It may be more probable to yield complementary abilities when distinct features are used in segmentation-based systems and segmentation-free ones. However, we pose more strict restrictions: the segmentation-based systems just employ the fused features similarly derived from FPFs and cell features in segmentation-free systems.

Previous empirical results on isolated-character recognition have shown that shape normalization can greatly reduce the within-class dispersions of character shape and improve the recognition rate. We study the effect of character size scaling, one-dimensional nonlinear normalization and elastic cell in MPTM approach and the right combination of them yielding best classification rate is adopted in the MQDF approach. Specially, to the Gaussian classifiers (MQDF in this paper), Box–Cox variable transformation is evaluated.

In the following, we will provide the technique details and experimental setup of above segmentation-based systems. As regards to MPTM approach, eight systems (MPTM-1 $\sim$ MPTM-8) are designed and they are put aside in Fig. 12 as vertices of a cube. In Fig. 12, three techniques are evaluated. To determine the aspect ratio, either fixed aspect ratio or sine of aspect ratio can be used [37]. To accomplish the coordinate transformation, nonlinear normalization with the line density initialization of Tsukumo and Tanaka [4] or linear normalization with bicubic interpolation are examined. And to partition the character images, 8×8 uniform cell or 8×8 elastic cell (the line density is also calculated by the method of Tsukumo and Tanaka) can be

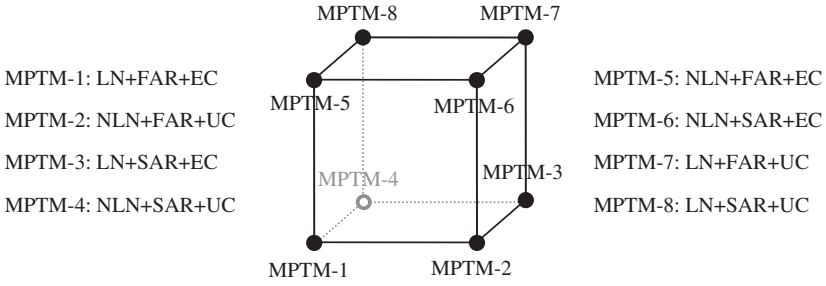

Fig. 12. Segmentation-based systems in MPTM approach. Eight systems are launched to evaluate the effect of three technologies. LN stands for linear normalization, NLN for nonlinear normalization, FAR for fixed aspect ratio, SAR for sine of aspect ratio, UC for uniform cell and EC for elastic cell.

applied. Since the FPFs are dependent on stroke width, four directional planes are generated before any shape normalization methods are applied. During the training stage, 12 models per character class are produced using K-means clustering method. In the classification phase, the feature vector of a character segment to be recognized is compared with all prototypes, and the character class associating the minimal city block distance is selected as the output class.

As for MQDF classifier, minor modification is applied to consider the unbalanced distributions of different classes:

$$g_3(\mathbf{x},\omega_i)=\sum_{j=1}^{k}\frac{1}{\lambda_{ij}}[\phi_{ij}^{\mathrm{T}}(\mathbf{x}-\mu_i)]^2+\sum_{j=k+1}^{d}\frac{1}{\delta_i}[\phi_{ij}^{\mathrm{T}}(\mathbf{x}-\mu_i)]^2 +\sum_{j=1}^{k}\log\lambda_{ij}+(d-k)\log\delta_i-2\log P(\omega_i). \tag{12}$$

where ω_i is the ith character class, μ_i denotes the mean vector of ω_i, λ_{ij} is the eigenvalue in nonincreasing order of the covariance matrix of ω_i and ϕ_{ij} is its eigenvector, and δ_i is optimized proportional to the global variance. For our modification is applied to MQDF2, we denote above classifier as MQDF3. In addition, special attention is given to the character class of small sample size. If the number of samples is below 10, λ_{ij} and ϕ_{ij} will be degenerated as the eigenvalue and eigenvector of global covariance matrix. The character segment will be classified into class ω_j, if

$$g_3(\mathbf{x},\omega_j)=\min_i g_3(\mathbf{x},\omega_i). \tag{13}$$

4.2. Analytical techniques

4.2.1. Correct rate and accurate rate

The output of certain recognizer is compared with the reference text and two metrics, the correct rate (CR) and accurate rate (AR), are calculated to evaluate the results. Supposing the number of substitution errors (S_e), deletion errors (D_e) and insertion errors (I_e) are known, CR and AR are defined, respectively, as

$$\begin{cases}\mathrm{CR}=(N_t-D_e-S_e)/N_t,\\ \mathrm{AR}=(N_t-D_e-S_e-I_e)/N_t,\end{cases} \tag{14}$$

where N_t is the total characters in the reference text. In general, CR is nonnegative while AR may be negative if there are overmany insertion errors.

4.2.2. Curve of character matching rate

Character matching rate (CMR) is used to reflect the complementary capacity of two systems at certain recognition rate. Suppose A_i, B_i are the sets of characters whose CR is bigger than i% given by two systems, respectively. More specifically, if we assume A_i = {"1", "2", "9", "7"} and B_i = {"2", "9", "7"}, we say $A_i \cap B_i$ (={"2", "9", "7"}) is the match set between A_i and B_i. The cardinality of the match set is given by $|A_i \cap B_i|$ (= 3, in this example). The CMR of them at the CR of i% is defined as the normalization of the above cardinality:

$$\mathrm{CMR}_i=\frac{|A_i\cap B_i|}{\min\{|A_i|,|B_i|\}}. \tag{15}$$

From the formula, we can see that less matches between two systems, the smaller the value of CMR_i. So, we can use CMR to characterize the possible complementary capacities between two systems when CR is bigger than i%. The curve of CMR manifests the possible complementary capacity dynamically by visualizing the CMR_i vs different i's.

Table 1
The digit recognition rates of different systems evaluated on test set (%)

	16DCELL	64DFPF	80DFUS	36DPCA	36DGVS
CR	42.61 (▽)	44.35	57.39 (△)	56.96	52.17
AR	36.09	36.52	39.13	46.09 (△)	35.22 (▽)

Table 2
The punctuation recognition rates of different systems evaluated on test set (%)

	16DCELL	64DFPF	80DFUS	36DPCA	36DGVS
CR	29.28 (△)	19.39	27.63	28.14	5.58 (▽)
AR	25.48 (△)	16.22	22.69	17.36	4.18 (▽)

Table 3
The Chinese character recognition rates of different systems evaluated on test set (%)

	16DCELL	64DFPF	80DFUS	36DPCA	36DGVS
CR	33.72 (▽)	35.2	36.57	39.54	48.09 (△)
AR	29.37 (▽)	32.47	32.43	34.93	42.91 (△)

4.3. Comparison within segmentation-free systems

4.3.1. The recognition rates

Due to their distinct differences in shape, the results of digit, punctuation and Chinese character by the segmentation-free systems are separately evaluated and are summarized in Tables 1–3, respectively. The maximal and minimal items in each row are highlighted with △ and ▽, respectively. Seen from Table 1, no system achieves the best or worst CR and AR simultaneously. As for CR, 16DCELL is the lowest one. The performance of 64DFPF in this respect outperforms 16DCELL. The 80DFUS reaches the best CR by the fusion of cell features and FPFs. The CRs of 36DPCA and 36DGVS drop down in turn. Similar trend can be observed in their AR performances. However, a mass of insertions are occurred in digit, and 80DFUS and 36DGVS are the two systems that were most suffered by this problem. As a

result, 80DFUS does not reach the maximum while 36DGVS reaches the minimum.

The best performance is demonstrated in 16DCELL when identifying the punctuation (see Table 2). The results of 64DFPF are inferior to 16DCELL. The feature fusion, dimension reduction and variance sharing techniques present no improvement at all in recognition rates, and a sharp decrease in CR and AR can be easily observed relating to 36DGVS.

Unlike the punctuation recognition, almost all enhancement operations increase the performance with regard to the Chinese character recognition, as shown in Table 3. The best performance is achieved by 36DGVS and the worst is by 16DCELL.

Moreover, we consider the average recognition rates, as shown in Table 4. It is clear that a noticeable improvement has been achieved after incremental enhancement. The CR and AR have increased from 33.54% and 29.18% (of baseline system) to 44.22% and 39.08% (of 36DGVS), respectively. Both promotions account for about 10%. Among such segmentation-free systems, the smallest two promotions are laid in the CR of 64DFPF and the AR of 80DFUS. Using Wilcoxon signed-rank test in Ref. [47], their statistical significance holds at 0.16 and 0.09 levels, respectively. Other improvements are all statistically significant with confidence more than 99%.

4.3.2. The analysis of errors

We proceed to analyse the error distribution of the segmentation-free systems. Three types of errors, delete error, insertion error and substitute error, are separately considered to each system and the error ratios are summarized in Table 5 as regards to digit, punctuation and Chinese characters, respectively. For example, the deletion errors of digit constitute 7.93% of whole deletion errors. Among 8471 characters in test set, 2.72%, 9.36% and 87.92% are comprised by digit, punctuation and Chinese character samples, respectively. Comparing 64DFPF with 16DCELL, obvious advantages are shown in characterizing the Chinese characters, seeing that all types of error ratios in Chinese characters are lower, while the opposite effect is observed in punctuation. On the fusion of FPFs and cell features (80DFUS), the insertion and the deletion error ratios in Chinese characters decrease. However, higher error ratios of insertion and deletion are occurred in digit and in punctuation, respectively. The small strokes of Chinese character are more likely to be misinterpreted as digit. On the contrary, punctuation tends to be viewed as a part of the neighboring Chinese character. The error ratios of Chinese character decrease from 80DFUS to 36DPCA. This finding verifies the positive role of PCA technique in the discriminative description of Chinese characters. As for 36DGVS, a remarkable reduction of deletion error ratio and substitution error ratio are observed in Chinese characters and as a result, a promising recognition rates are achieved in Table 3. Above robustness is achieved at some loss of modeling precision. For example, the GVS gives a larger insertion error ratio than 36DPCA.

The recognition results of the textlines in Fig. 5(c)–(e) are illustrated in Figs. 13–15, respectively. The correctly identified characters are underlined. As the character boundary can be obtained as a byproduct of the recognition process (see Section 3.3), we plot it in the first row of each subfigure whose intensity expressing the probability in that position and the segments enclosed by two adjacent boundary lines are labeled with numbers. The boundary line is of the same width of the sliding window.

We can see that substitution errors often occurs between similar characters. Two kinds of similar characters are clarified in Ref. [48] due to their inherent similarity or distortion during writing process. The former is the main source of substitution errors. Such instances are marked with “S1” in Figs. 13–15. In cursive characters, substitutions often fall into the latter. Such cases are marked with “S2” in Figs. 13 and 15. The other substitution errors are mainly resulted from the precision of the HMMs. Moreover, we can see that the misidentified segments contribute to the deletion and insertion errors.

In addition, the noise resistance can be verified. Some outliers incurred from small stroke can be solved. Examples cast in this category are marked with “H1” in Figs. 13–15. Even unacceptable segments may be correctly identified (see the examples marked with “H2” in Figs. 13–15).

To establish a reliable inference in statistics, we only consider those characters whose sample size is over 20. Following points are observed. First, substitutions are often found between similar characters. Several characters with large substitution errors are given in Fig. 16. Second, deletion is mainly occurred in digit and punctuation, as digit or punctuation is often mis-explained as a component of its neighboring Chinese character. Such digit and punctuation, whose deletion error rates are bigger than 1%, include zero, comma, period, caesura sign and left-double quotation mark. Last, insertion seems more dependent on system setup. Only the insertion error rate of the digit, “1”, is consistently bigger than 1% in all segmentation-free systems in this paper.

To sum up, it is preferable and important to design discriminative features and represent them properly. The cell features are sensitive to the stroke width but more suitable than FPFs to represent the punctuation. On the contrary, the FPF seems a good descriptor in characterizing the directional structures most presented in digits or Chinese characters. The fusion of above two kinds of features yields acceptable tradeoff or improvement of performance. The dimension reduction by PCA and variance sharing through GVS show efficiencies in alleviating the insufficient training data. However, adverse effects are observed in the recognition of digits and punctuation. We will revisit this in Section 5.

4.4. Comparison between segmentation-based and segmentation-free systems

4.4.1. The recognition rates

The results of MPTM approach are summarized in Table 6. Herein we first compare MPTM-1 vs MPTM-5, MPTM-7 vs MPTM-2, MPTM-3

Table 4
The average recognition rates of different systems evaluated on test set (%)

	16DCELL	64DFPF	80DFUS	36DPCA	36DGVS
CR	33.54 (▽)	33.96	36.29	38.94	44.22 (△)
AR	29.18 (▽)	31.06	31.7	33.59	39.08 (△)

Table 5
The error ratios on test set between digit punctuation and Chinese character of segmentation-free systems

	16DCELL (%)			64DFPF (%)			80DFUS (%)			36DPCA (%)			36DGVS (%)		
	DI	PU	CH	DI	PU	CH	DI	PU	CH	DI	PU	CH	DI	PU	CH
Delete error	7.39	17.43	75.18	5.47	19.50	75.03	5.35	20.36	74.29	5.65	20.68	73.67	6.00	37.67	56.33
Insertion error	3.79	8.13	88.08	6.50	10.16	83.34	10.80	10.03	79.17	5.52	18.76	75.72	8.97	2.53	88.50
Substitute error	1.62	8.95	89.43	1.64	10.00	88.36	1.31	9.26	89.43	1.41	9.71	88.88	1.49	10.82	87.69

CH, DI and PU are the acronyms of Chinese characters, digit and punctuation, respectively.

Fig. 13. Recognition results of the textline in Fig. 5c by: (a) 16DCELL, (b) 64DFPF, (c) 80DFUS, (d) 36DPCA and (e) 36DGVS. Upper row of each subfigure shows the soft segmentation boundary imposed on the textline. Middle row of each subfigure is copied from the recognition result of the textline and the correctly recognized characters are underlined to give a clear view. Lower row of each subfigure highlights some segments and characters with S1 (substitute error is due to their inherent similarity), S2 (substitute error results from writing distortion), H1 (correctly identified segment with small outliers) and H2 (correctly identified block with large outliers).

vs MPTM-6 and MPTM-8 vs MPTM-4. Clear advantages of nonlinear normalization are observed in digit recognition. Also, we compare MPTM-1 vs MPTM-3, MPTM-2 vs MPTM-4, MPTM-5 vs MPTM-6 and MPTM-7 vs MPTM-8. We can see a positive effect of sine of aspect ratio in both digit and punctuation recognition. Finally, we compare results using elastic cell with those using uniform cell (MPTM-2 vs MPTM-5, MPTM-4 vs MPTM-6, MPTM-7 vs MPTM-1 and MPTM-8 vs MPTM-3). Elastic cell yields a consistent improvement than uniform cell in Chinese character recognition rate and the average recognition rate. However, the best average recognition rates are not achieved by the combination of nonlinear normalization, sine of aspect ratio and elastic cell (MPTM-6). Instead, MPTM-1 is the averagely best recognizer due to its great discriminative ability in Chinese character. We will directly employ this kind of combination in the evaluation of MQDF3 classifier.

Two systems with MQDF3 classifier are experimented. One of them (MQDF3-2) applies Box–Cox transformation in the feature extraction stage besides the combination of linear normalization, fixed aspect ratio and elastic cell methods. The power of the transformation is set to 0.5 intuitively. The other (MQDF3-1) is a baseline system which uses an uniform cell but no Box–Cox transformation is applied. The parameters, k and δ_i, are tuned on validation set with MQDF3-2, and it shows that the classification rate yields best when k and δ_i are set to 7 and 1.7 times of the global variance. Their results are presented in Table 7. Following two points can be concluded:

(1) The recognition rates of MQDF3-1 are much lower than 80DFUS. As previous description of segmentation-free systems, neither shape normalization (including size scaling, nonlinear normalization and elastic cell) nor variable transformation is exploited till now. Thus, it appears that HMM may properly model the stroke variabilities and the sliding window-based feature extraction method may encode appealing information between adjacent characters (refer to Section 1.3).

(2) MQDF3-2 yields a great improvement than MQDF3-1. Further compared with 80DFUS, obvious increments are observed in the recognition rates of Chinese character. Certainly, after three decades research on handwritten Chinese isolated-character recognition, there are many techniques available to improve the recognizers. However, 80DFUS manifests advantages in the recognition of digit and punctuation. This consideration pushes us to combine MQDF3-2 with 80DFUS.

4.4.2. The time and memory consumption

We also report computational time and memory both on a desktop PC computer and a laptop. The description of computers is listed in Tables 8. The average CPU time per character of segmentation-free and segmentation-based systems are separately given in

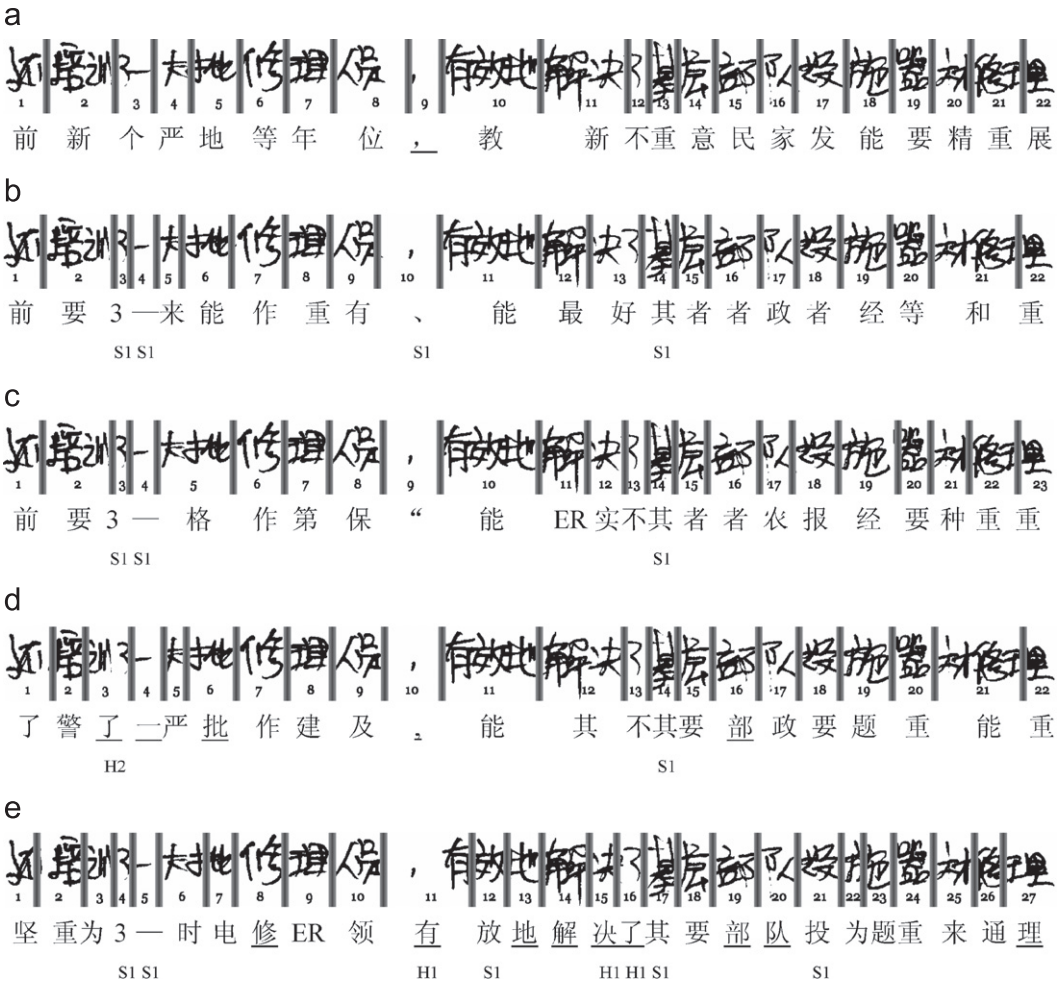

Fig. 14. Recognition results of the textline in Fig. 5d by: (a) 16DCELL, (b) 64DFPF, (c) 80DFUS, (d) 36DPCA, and (e) 36DGVS. Upper row of each subfigure shows the soft segmentation boundary imposed on the textline. Middle row of each subfigure is copied from the recognition result of the textline and the correctly recognized characters are underlined to give a clear view. Lower row of each subfigure highlights some segments and characters with S1 (substitute error is due to their inherent similarity), H1 (correctly identified segment with small outliers) and H2 (correctly identified block with large outliers).

Tables 9 and 10. The former include the CPU time of feature extraction and that of recognition. As for the latter, the time of character segmentation is provided, too. To estimate the memory consumption of each recognizer, the storage size of character models is summarized in Table 11.

From Tables 9–11, following points can be inferred:

(1) The CPU time and memory needed by segmentation-free systems in this paper are mainly dependent on their dimensionality of feature vector and the number of Gaussian mixtures; The 16DCELL (with five Gaussian mixtures) consumes smallest time and memory. With the increase of feature size, 64DFPF (also with five Gaussian mixtures) requires above double time and nearly quadruple storage. The 80DFUS (with six Gaussian mixtures) becomes the most intensive recognizer in time and storage consumptions. Once PCA is used, the overall time and storage are greatly saved in 36DPCA (with four Gaussian mixtures) though the time spent on feature extraction increases. More Gaussian mixtures are needed in 36DGVS (with eight Gaussian mixtures) when GVS is further adopted, which results in a bit more time consumption and a slight increase of storage. However, with an eye to the recognition rates, PCA and GVS are effective techniques to enhance the segmentation-free systems.

(2) MQDF3 runs faster than MPTM with an acceptable memory requirement, though MQDF is often claimed with intensive time and memory consuming in literature. Due to the shortage of samples, the parameter k in Formula 12 is set with 7. However, experiments on isolated-character databases choose k around 40 wherein a huge memory and an increase of time are required. Commonly, coarse classification stage is often performed to give a small set of candidate characters before the MQDF3 classifier and the CPU time to classify the candidates can be effectively reduced. And the huge memory requirement of MQDF3 is often alleviated by dimensionality reduction to some extent.

(3) Disadvantages in speed are observed in segmentation-free systems when compared with MQDF3-2. Among them, 80DFUS requires the most intensive CPU time and memory. In Section 4.5.2, the integration of 80DFUS and MQDF3-2 provides improvement in average recognition rates, while made a tradeoff between their CPU time.

4.5. Verification of the complementary capacities

Due to the huge complexities and difficulties in the recognition of realistic Chinese handwriting, the combination of segmentation-free and segmentation-based systems may be an important direction in the future. To pave the road to the fusion trends, we mainly explore the complementary capacities between them in Section 4.5.1 and then a simple integration mechanism is provided in Section 4.5.2.

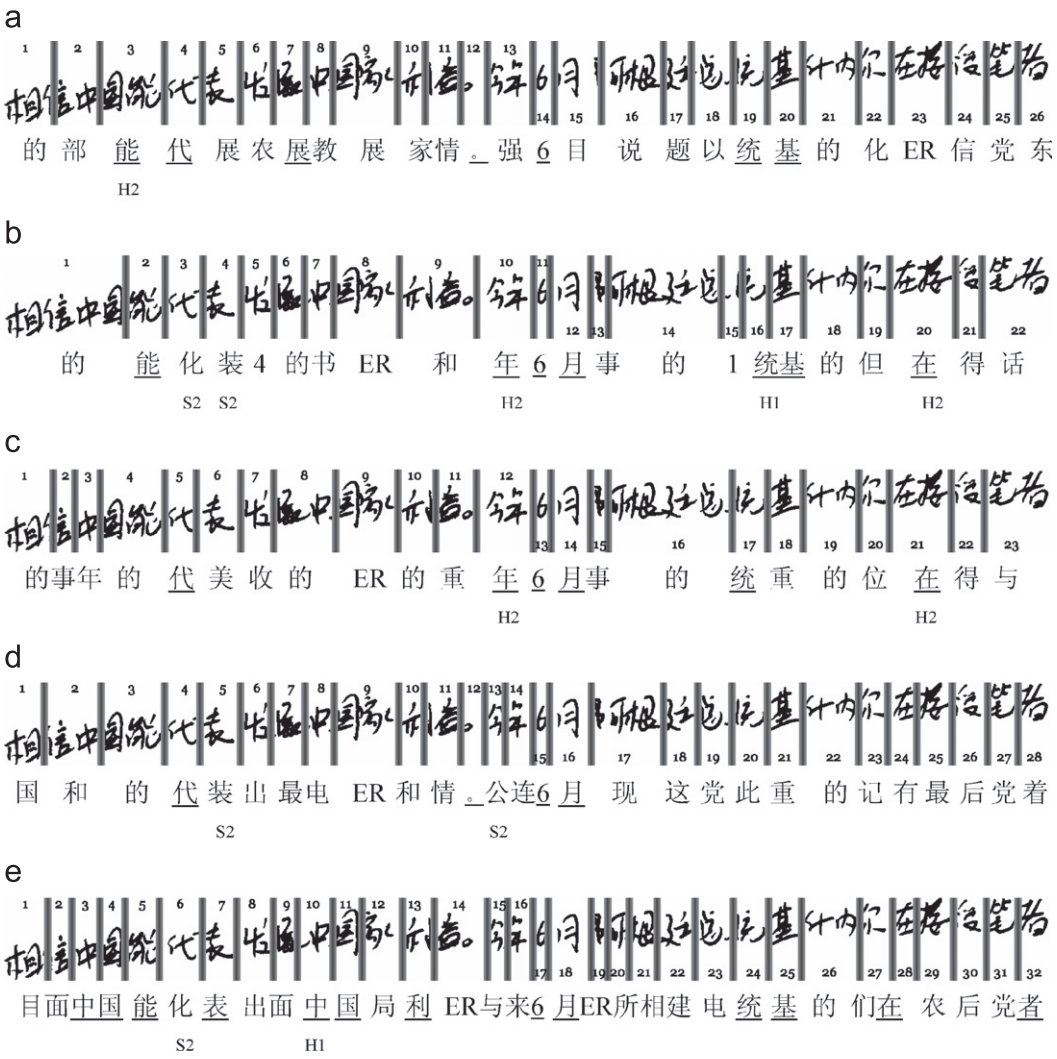

Fig. 15. Recognition results of the textline in Fig. 5e by: (a) 16DCELL, (b) 64DFPF, (c) 80DFUS, (d) 36DPCA and (e) 36DGVS. Upper row of each subfigure shows the soft segmentation boundary imposed on the textline. Middle row of each subfigure is copied from the recognition result of the textline and the correctly recognized characters are underlined to give a clear view. Lower row of each subfigure highlights some segments and characters with S1 (substitute error is due to their inherent similarity), S2 (substitute error results from writing distortion), H1 (correctly identified segment with small outliers) and H2 (correctly identified block with large outliers).

2→工 “→” ，→、 、→，
2→之 。→0 。→、 。→，

Fig. 16. Character pairs with high substitution errors in confusion matrix. Substitution errors are often found between similar characters.

4.5.1. CMR curves

We first plot the CMR curve in Fig. 17 between 80DFUS and MQDF3-2 (denoted as 80DFUS+MQDF3-2). It can be regarded as the measurement of complementary capacities between two different recognition strategies when they are trained on the same training data and their features are of the same type.

As references, we add another two CMR curves between two different segmentation-free systems: one is 80DFUS and 16DCELL (80DFUS+16DCELL), the other is 80DFUS and 64DFPF (80DFUS+64DFPF). Seen from the previous experiments, the fusion of cell features and FPFs demonstrates apparent improvement to each original set of features. Thus, we also illustrate their CMR curve in Fig. 17 (it is labeled as 64DFPF+16DCELL) to give a more concrete reference.

Table 6
The recognition rates of MPTM-based systems on test set

	Digit (%)		Punctuation (%)		Chinese character (%)		Average (%)	
	CR	AR	CR	AR	CR	AR	CR	AR
MPTM-1	11.30	8.26	12.12	9.22	28.81	26.33	26.74	24.22
MPTM-2	13.04	10.43	10.35	6.82	23.73	21.30	22.16	19.63
MPTM-3	13.91	12.61	13.76	9.22	27.66	25.41	25.96	23.52
MPTM-4	14.35	10.43	12.75	8.96	23.15	20.92	21.92	19.50
MPTM-5	13.04	10.87	11.36	9.34	24.05	21.39	22.54	19.96
MPTM-6	16.96	13.91	14.77	10.61	24.09	21.70	23.00	20.43
MPTM-7	12.61	9.57	12.37	9.85	23.85	21.47	22.45	20.04
MPTM-8	13.04	10.87	14.02	10.86	24.04	21.67	22.78	20.35

Table 7
The recognition rates of MQDF3-based systems on test set

	Digit (%)		Punctuation (%)		Chinese character (%)		Average (%)	
	CR	AR	CR	AR	CR	AR	CR	AR
MQDF3-1	13.91	10.44	13.01	10.99	29.05	26.43	27.11	24.53
MQDF3-2	17.83	14.35	14.02	11.11	38.36	35.98	35.49	33.03

Table 8
Computers used to measure performance

	CPU			Memory (MB)
	Name	Alias	Clocks (GHZ)	
Desktop PC	Pentium 4	P68, Willamette	1.37	256
Laptop	Mobile Duo T5600	Merom-2M	1.79	1024

Table 9
Average CPU time consuming of segmentation-free systems (unit: ms/character)

	16DCELL	64DFPF	80DFUS	36DPCA	36DGVS
Feature extraction	24.52 (8.23)	155.02 (41.29)	169.72 (45.30)	171.71 (46.36)	171.17 (46.36)
Recognition	1950.21 (704.13)	3599.86 (1514.28)	5788.50 (1923.47)	2272.52 (1089.20)	4039.63 (1769.87)

The CPU time measured on laptop is given in parenthesis.

Table 10
Average CPU time consuming of segmentation-based systems (unit: ms/character)

	MPTM-1	MQDF3-1	MQDF3-2
Character segmentation	1.14 (0.37)	1.14 (0.37)	1.14
Feature extraction	336.47 (101.53)	209.89 (75.80)	346.53 (106.53)
Classification	702.10 (322.08)	401.86 (153.35)	574.33 (164.48)

The CPU time measured on laptop is given in parenthesis.

Table 11
Memory consuming of segmentation-based and segmentation-free systems (unit: MB)

16DCELL	64DFPF	80DFUS	36DPCA	36DGVS	MPTM	MQDF3
17.57	58.63	86.48	28.05	30.96	5.92	33.09

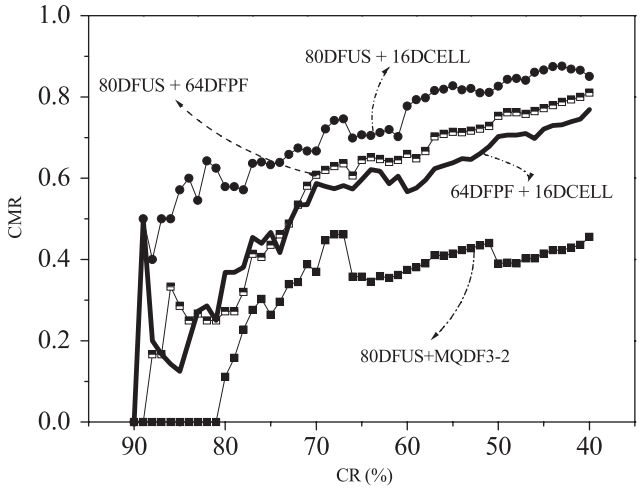

Fig. 17. Curves of CMR. Clear complementary capacities can be inferred between the segmentation-free strategy and the segmentation-based one.

Seen from the figure, the curve of 80DFUS+MQDF3-2 is completely lower than other three curves derived from segmentation-free systems. The smallest differential between them is more than 9% when CR ranges from 40% to 88%. Moreover, when more characters are considered (with CR decreasing), the differentials may largen. Therefore we can safely conclude that the two strategies under same training data may greatly complement each other, even when they employ the same type of features.

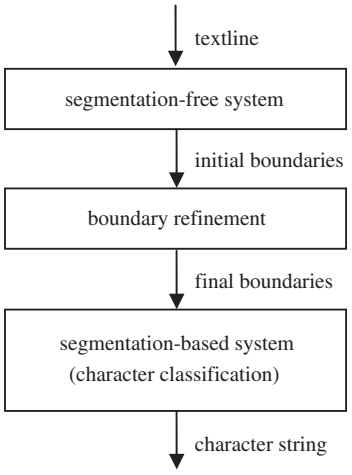

Fig. 18. The connection diagram of segmentation-free and segmentation-based systems. The segmentation-free system is launched to locate the initial character boundaries.

4.5.2. Integration of segmentation-free and segmentation-based systems

In Ref. [49], we insert a segmentation-free system into a segmentation-based system. The block diagram is illustrated in Fig. 18, wherein the segmentation-free system is used to locate the initial character boundaries. After tuning the boundaries, a considerable improvement in recognition rates is observed. However, the CPU time is intensively consumed, since two systems should be performed one by one.

This paper presents a distinct mechanism to provide a more concrete evidence for the complementariness between the segmentation-free system (80DFUS) and segmentation-based system (MQDF3-2). A similar strategy for English handwritten word has been presented in Ref. [31]. If the output of the fed textline is more “confident” than a threshold *TH*, 80DFUS will stand away. Otherwise, we start 80DFUS to give the final output. The confidence

Table 12
The recognition rates of integrated system on test set

	Digit (%)		Punctuation (%)		Chinese character (%)		Average (%)	
	CR	AR	CR	AR	CR	AR	CR	AR
Combination	37.83	26.09	22.73	16.79	41.23	36.84	39.37	34.64

is evaluated on the rth textline by LL^r as follows:

$$LL^r = M^r_{cs} - M^r_{df}, \tag{16}$$

where M^r_{cs} measures the quality of character segmentation and M^r_{df} measures the closeness of discriminant function.

Initially, M^r_{cs} and M^r_{df} are zeros. If the width of any character segments are larger than 1.5 times average character width (58 pixels in our experiment), M^r_{cs} will be updated with $M^r_{cs} \leftarrow M^r_{cs} + 1$. To diminish the effect of textline length, M^r_{cs} will be divided by the number of average characters in the rth textline. Similarly, M^r_{df} increases with 10, if any of the differentials of minimum two $g_3(\mathbf{x}, \omega_i)$ is smaller than 0.34. Also, the M^r_{df} should be divided by the number of character segments in the rth textline.

The integrated system works well, and the results when $TH = 95$ are shown in Table 12. We can see an obvious improvement in average recognition rates. It consumes 3078.01 ms on the desktop PC and 1002.38 ms on the laptop.

5. Discussions

The recognition rates of both segmentation-based systems and segmentation-free ones are no more than 50%. The low rates are partially attributed to the complexities of the realistic handwriting. However, the data sparseness also has remarkable impact.

Data sparseness is a common problem in natural language processing. As soon as Chinese handwriting is concerned, a great deal of Chinese characters may occur few times ever never. In other words, many character models have insufficient training samples. As a result, robust parameter estimation is impossible and on the other hand the recognizer easily falls into "overfitting". This problem in alphabetic languages like English may be not as severe as in Chinese. In recognition systems for English text, they can model the letters instead of words and then arrange letter models to word model, since plenty of letters are available. Unfortunately, Chinese character is the basic writing unit, thus no straightforward way to condense the large number of models. We plot the distribution of the samples in Fig. 19. The coordinate (10, 56) means that there are 56 character class, which possess 10 samples in train and validation sets. Among them, 13.83% have no training samples at all and 52.10% possess less than five training samples.

Alleviating the data sparseness reasonably may improve the recognition rate remarkably. We further inspect the correlation-ship between the number of samples and CR. All previously mentioned systems manifest significant correlations at 0.05 level [45] as regards the recognition of Chinese characters. The correlation coefficients of them are listed in Table 13. However, their significance is not confirmed as regards digit and punctuation. This is the main reason that the alleviation of insufficient data by PCA and GVS displays no positive effect (please refer to Section 4.3). If we expand the available samples of Chinese characters, the performance will be improved further, which can be inferred from Fig. 20. As expected, we can see that MQDF3-2 climbs up slower than segmentation-free systems.

Data sparseness is a hard problem indeed. However, two principles can alleviate it. The most obvious one is increasing the copies of samples, for example, collecting the same database several times; generating artificial samples by computer (refer to Ref. [50] for more details); incorporating existing character database (we can utilize the isolated-character database to initialize the character models or to produce textlines for a more reliable training); data sharing between similar models. Another principle is to reduce the size of parameter set to be estimated, for example, using dimension reduction method (as in present paper) or dispensing the number of Gaussians proportional to available samples (an example is given in Ref. [51]).

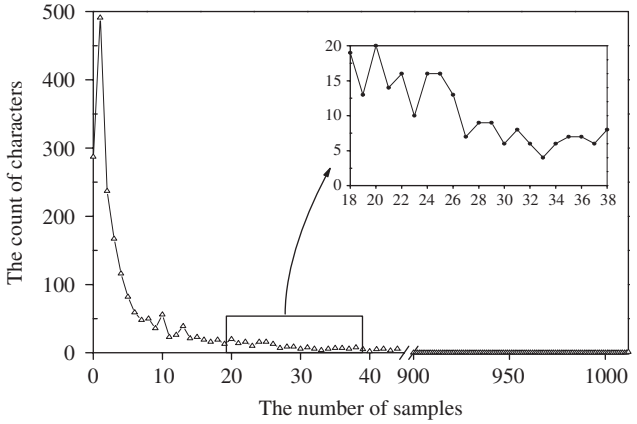

Fig. 19. The histogram of the number of samples. The data are derived from train and validation sets.

Table 13
The correlation coefficients of different systems between the number of samples and CR

16DCELL	64DFPF	80DFUS	36DPCA	36DGVS	MQDF3-2
0.66	0.65	0.69	0.65	0.57	0.48

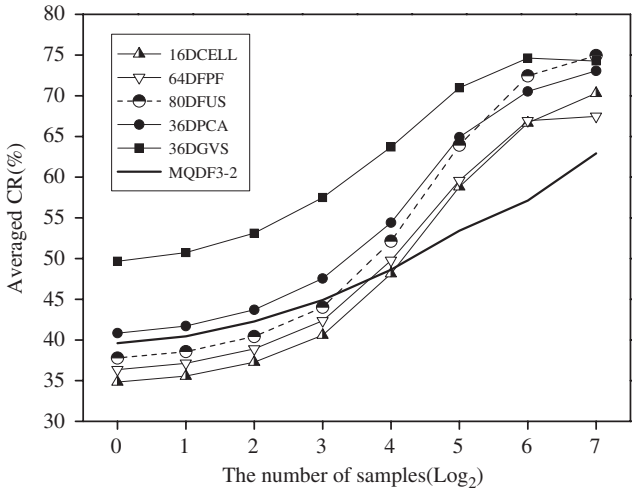

Fig. 20. Relationship between the number of samples and CR. With the exponential increase in the number of samples, it shows proportional improvements in CR.

6. Conclusions

Realistic Chinese handwriting poses great challenges to the state-of-the-art recognizers. On the one hand, the character separation is still far from solved. The segmentation algorithm falls into under-segmentation, once it encounters touching, overlapping, and crossing phenomena. On the contrary, over-segmentation cases are often raised to left–right structure characters. On the other, due to the great variability in character shape, the character modeling techniques

should be updated. Currently, the features are extracted within each character segment. However, the conjunction relationship between adjacent characters also encodes valuable information. Thus, it is in great need to handle these problems.

This paper describes HMM-based recognizers for realistic Chinese handwriting under a segmentation-free framework. The textlines fed into recognizer have no need to provide the position of the characters, and the output from recognizer is a best string of characters in MAP sense. Experiments are conducted on HIT-MW database. Promising results are achieved, which not only show the feasibility of the segmentation-free strategy but also provide the apparent evidence on the complementary capacities between the segmentation-free strategy and the segmentation-based one. Based on the observations found in this paper, we will investigate following points in the future: the discriminative feature extraction method, which can make preferable tradeoff between punctuation and Chinese characters; the synthetic handwriting algorithm to alleviate the insufficient training data; the approach to properly combine segmentation-free and segmentation-based systems.

Acknowledgments

We would especially like to thank the anonymous reviewers for their valuable suggestions. Great help has also been provided by Cheng-Lin Liu (Institute of Automation, Chinese Academy of Sciences, PR China) and his group in preparing this revision. This work is supported in part by the National Natural Science Foundation of China (NSFC, Grant no. 60475011) and the Heilongjiang Natural Science Foundation of China (Grant no. F0322).

References

[1] T.H. Hildebrandt, W. Liu, Recognition of handwritten Chinese characters: advances since 1980, Pattern Recognition 26 (1993) 205–225.
[2] S.N. Srihari, X. Yang, G.R. Ball, Offline Chinese handwriting recognition: an assessment of current technology, Front. Comput. Sci. China 1 (2007) 137–155.
[3] R. Dai, C. Liu, B. Xiao, Chinese character recognition: history, status and prospects, Front. Comput. Sci. China 1 (2007) 126–136.
[4] J. Tsukumo, H. Tanaka, Classification of handprinted Chinese characters using non-linear normalization and correlation methods, in: Proceedings of the 9th International Conference on Pattern Recognition, 1988.
[5] C.-L. Liu, K. Marukawa, Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition, Pattern Recognition 38 (2005) 2242–2255.
[6] L. Jin, G. Wei, Handwritten Chinese character recognition with directional decomposition cellular features, J. Circuit Syst. Comput. 8 (1999) 517–524.
[7] C.-L. Liu, H. Sako, H. Fujisawa, Handwritten Chinese character recognition: alternatives to nonlinear normalization, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 2003.
[8] Y. Chen, X. Ding, Y. Wu, Off-line handwritten Chinese character recognition based on crossing line feature, in: Proceedings of the Fourth International Conference on Document Analysis and Recognition, Ulm, Germany, 1997.
[9] N. Kato, M. Suzuki, S.i. Omachi, H. Aso, Y. Nemoto, A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance, IEEE Trans. Pattern Anal. Mach. Intell. 21 (1999) 258–262.
[10] Y. Ge, Q. Huo, A comparative study of several modeling approaches for large vocabulary offline recognition of handwritten Chinese characters, in: Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, 2002.
[11] X. Wang, X. Ding, C. Liu, Gabor filters-based feature extraction for character recognition, Pattern Recognition 38 (2005) 369–379.
[12] C.-L. Liu, Normalization-cooperated gradient feature extraction for handwritten character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1465–1469.
[13] J.-X. Dong, A. Krzyzak, C.Y. Suen, An improved handwritten Chinese character recognition system using support vector machine, Pattern Recognition Lett. 26 (2005) 1849–1856.
[14] B. Feng, X. Ding, Y. Wu, Chinese handwriting recognition using Hidden Markov Models, in: Proceedings of the 16th International Conference on Pattern Recognition, Barcelona, Spain, 2002.
[15] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, IEEE Trans. Pattern Analy. Mach. Intell. 9 (1987) 149–153.
[16] C.-L. Liu, I.-J. Kim, J.H. Kim, Model-based stroke extraction and matching for handwritten Chinese character recognition, Pattern Recognition 34 (2001) 2339–2352.
[17] Y. Li, X. Ding, C.L. Tan, Combining character-based bigrams with word-based bigrams in contextual postprocessing for Chinese script recognition, ACM Trans. Asian Lang. Inf. Process. 1 (2002) 297–309.
[18] Y. Li, C.L. Tan, X. Ding, A hybrid post-processing system for offline handwritten Chinese script recognition, Pattern Anal. Appl. 8 (2005) 272–286.
[19] S. Mori, K. Yamamoto, H. Yamada, T. Saito, On a handprinted Kyoiku–Kanji character data base, Bull. Electrotech. Lab. 43 (1979) 752–773.
[20] T. Saito, H. Yamada, K. Yamamoto, On the data base ETL9 of handprinted characters in JIS Chinese characters and its analysis, IEICE Trans. J68-D (1985) 757–764.
[21] Y. Liu, J. Tai, J. Liu, An introduction to the 4 million handwriting Chinese character samples library, in: Proceedings of the International Conference on Chinese Computing and Orient Language Processing, Changsha, China, 1989.
[22] H. Zhang, J. Guo, Introduction to HCL2000 database, in: Proceedings of Sino-Japan Symposium on Intelligent Information Networks, Beijing, China, 2000.
[23] T.-H. Su, T.-W. Zhang, D.-J. Guan, Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text, Int. J. Doc. Anal. Recognition 10 (2007) 27–38.
[24] C. Hong, G. Loudon, Y. Wu, R. Zitserman, Segmentation and recognition of continuous handwriting Chinese text, Int. J. Pattern Recognition Artif. Intell. 12 (1998) 223–232.
[25] R.G. Casey, E. Lecolinet, A survey of methods and strategies in character segmentation, IEEE Trans. on Pattern Anal. Mach. Intell. 18 (1996) 690–706.
[26] C.Y. Suen, S. Mori, S.H. Kim, C.H. Leung, Analysis and recognition of Asian scripts—the state of the art, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 2003.
[27] K. Sayre, Machine recognition of handwritten words: a project report, Pattern Recognition 5 (1973) 213–228.
[28] S. Madhvanath, V. Govindaraju, The role of holistic paradigms in handwritten word recognition, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 149–164.
[29] A. El-Yacoubi, R. Sabourin, C. Suen, M. Gilloux, An HMM-based approach for off-line unconstrained handwritten word modeling and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 21 (1999) 752–760.
[30] A.L. Koerich, R. Sabourin, C.Y. Suen, Recognition and verification of unconstrained handwritten words, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1509–1521.
[31] M. Mohamed, P. Gader, Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques, IEEE Trans. Pattern Anal. Mach. Intell. 18 (1996) 548–554.
[32] S. Gunter, H. Bunke, Ensembles of classifiers for handwritten word recognition, Int. J. Doc. Anal. Recognition 5 (2003) 224–232.
[33] U.V. Marti, H. Bunke, Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system, Int. J. Pattern Recognition Artif. Intell. 15 (2001) 65–90.
[34] A. Vinciarelli, S. Bengio, H. Bunke, Offline recognition of unconstrained handwritten texts using HMMs and statistical language models, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 709–720.
[35] J. Picone, Continuous speech recognition using Hidden Markov Models, IEEE ASSP Mag. 7 (1990) 26–41.
[36] L.R. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, Proc. IEEE 77 (1989) 257–285.
[37] M. Cheriet, N. Kharma, C.-L. Liu, C.Y. Suen, Character Recognition Systems: A Guide for Students and Practioners, Wiley, NJ, 2007.
[38] N. Lindgren, Machine recognition of human language: part III—cursive script recognition, IEEE Spectrum 2 (1965) 104–112.
[39] B.-S. Jeng, M.-W. Chang, S.-W. Sun, C.-H. Shih, T.-M. Wu, Optical Chinese character recognition with a Hidden Markov Model classifier—a novel approach, Electron. Lett. 26 (1990) 1530–1531.
[40] B. Feng, X. Ding, Y. Wu, Off-line handwritten Chinese character recognition, Pattern Recognition Artif. Intell. 15 (2002) 84–88 (in Chinese).
[41] T.-H. Su, T.-W. Zhang, D.-J. Guan, HIT-MW dataset for offline Chinese handwritten text recognition, in: Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, 2006.
[42] P. Natarajan, Z. Lu, R.M. Schwartz, I. Bazzi, J. Makhoul, Multilingual machine printed OCR, Int. J. Pattern Recognition Artif. Intell. 15 (2001) 43–63.
[43] T.-H. Su, T.-W. Zhang, Z.-W. Qiu, D.-J. Guan, Gabor-based recognizer for Chinese handwriting from segmentation-free strategy, in: Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns, 2007.
[44] T.-H. Su, T.-W. Zhang, H.-J. Huang, Y. Zhou, Skew detection for Chinese handwriting by horizontal stroke histogram, in: Proceedings of the 9th International Conference on Document Analysis and Recognition, 2007.
[45] A.C. Rencher, Methods of Multivariate Analysis, Wiley, New York, 2002.
[46] T.-H. Su, T.-W. Zhang, H.-J. Huang, Y. Zhou, HMM-based recognizer with segmentation-free strategy for unconstrained Chinese handwritten text, in: Proceedings of the 9th International Conference on Document Analysis and Recognition, 2007.
[47] J.E. Freund, Modern Elementary Statistics, Prentice-Hall, New Jersey, 1984.
[48] C.-L. Liu, Handwritten Chinese character recognition: effects of shape normalization and feature extraction, in: Arabic and Chinese Handwriting Recognition, 2008.
[49] T.-H. Su, T.-W. Zhang, H.-J Huang, Character segmentation of handwritten Chinese text based on HMM recognizer, in: Proceedings of the 1st Chinese Conference on Pattern Recognition, Beijing, 2007 (in Chinese).

[50] T. Varga, H. Bunke, Offline handwriting recognition using synthetic training data produced by means of a geometrical distortion model, Int. J. Pattern Recognition Artif. Intell. 18 (2004) 1285–1302.

[51] T.-H. Su, T.-W. Zhang, Z.-W. Qiu, HMM-based system for transcribing Chinese handwriting, in: Proceedings of the 6th International Conference of Machine Learning and Cybernetics, Hong Kong, China, 2007.

About the Author—TONG-HUA SU received the B.S. degree in Computer Science from Harbin Engineering University, Harbin, China and the M.S. degree in Computer Science from Harbin Institute of Technology, Harbin, China in 2001 and 2003, respectively. He is currently a Ph.D. student at the Department of Computer Science, Harbin Institute of Technology. During 2003–2005, he had collected the first Chinese handwriting database, HIT-MW database, to establish the fundamental data for Chinese handwriting recognition. His research interests include Pattern Recognition, Signal and Image Processing, Machine Learning, Evolutionary Algorithm and especially the applications to Handwriting Recognition.

About the Author—TIAN-WEN ZHANG received the B.S. degree in Computer Science from Harbin Institute of Technology, Harbin, China in 1964. Since then he has been teaching in the Department of Computer Science at that University, where he is currently a professor. During 1994–1995, he has been a Visiting Professor at the Massachusetts Institute of Technology (MIT). His current interests include Active Vision, Pattern Recognition, Artificial Intelligence, Wavelet Image Compression, Virtual Reality and Artificial Life.