

# Offline Handwritten Chinese Character Recognition by Radical Decomposition

DAMING SHI

School of Computer Engineering, Nanyang Technological University, Singapore  
and

ROBERT I. DAMPER and STEVE R. GUNN

School of Electronics and Computer Science, University of Southampton

---

Offline handwritten Chinese character recognition is a very hard pattern-recognition problem of considerable practical importance. Two popular approaches are to extract features holistically from the character image or to decompose characters structurally into component parts—usually strokes. Here we take a novel approach, that of decomposing into radicals on the basis of image information (i.e., without first decomposing into strokes). During training, 60 examples of each radical were represented by “landmark” points, labeled semiautomatically, with radicals in different characteristic positions treated as distinctly different radicals. Kernel principal-component analysis then captured the main (nonlinear) variations around the mean radical. During the recognition, the dynamic tunneling algorithm was used to search for optimal shape parameters in terms of chamfer distance minimization. Considering character composition as a Markov process in which up to four radicals are combined in some assumed sequential order, we can recognize complete, hierarchically-composed characters by using the Viterbi algorithm. This gave a character recognition rate of 93.5% characters correct (writer-independent) on a test set of 430,800 characters from 2,154 character classes composed of 200 radical categories, which is comparable to the best reported results in the literature. Although the initial semiautomatic landmark labeling is time consuming, the decomposition approach is theoretically well-motivated and allows the different sources of variability in Chinese handwriting to be handled separately and by the most appropriate means—either learned from example data or incorporated as prior knowledge. Hence, high generalizability is obtained from small amounts of training data, and only simple prior knowledge needs to be incorporated, thus promising robust recognition performance. As such, there is very considerable potential for further development and improvement in the direction of larger character sets and less constrained writing conditions.

Categories and Subject Descriptors: I.5.4 [**Pattern Recognition**]: Applications—*Text processing*

General Terms: Algorithms

Additional Key Words and Phrases: Chinese computing; offline character recognition; active shape modeling, Viterbi decoding

---

Authors' addresses: D. Shi, School of Computer Engineering, Nanyang Technological University, Nanyang Ave., Singapore 639798; R. I. Damper and S. R. Gunn, School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2003 ACM 1530-0226/03/0300-0027 \$5.00

## 1. INTRODUCTION

According to Sampson [1985, p. 145], it is estimated that until the end of the 18th century, more than half of all books ever published were written in Chinese or Chinese-derived scripts. Nowadays, one quarter of the world's population reads and writes in Chinese scripts. So the advancement of Chinese information processing will undoubtedly bring great commercial and social benefits. However, Tang et al. [1998] describe offline handwritten Chinese character recognition as one of the most challenging topics in pattern recognition, since it involves a large number of characters with complex structure, serious interconnection among the components, and considerable pattern variation. Generally, online recognition is considerably easier, as it can exploit dynamic pen-movement information to help distinguish between characters or features that would otherwise be difficult to discriminate in a static image (Kim et al. [1996, 1997]; Chen and Lee [1997]). Given these difficulties, we believe a theoretically well-founded approach is necessary, one that is based on learning from data and avoiding *ad hoc* rules and heuristics, but one that nonetheless allows a modicum of prior knowledge to constrain the learning and pattern-matching processes.

The Chinese writing system is intensely hierarchical: words are composed of characters, which in turn are composed of *radicals* [Chang 1973; Suen and Huang 1984], which in turn are composed of a sequence of strokes (readers unfamiliar with Chinese script, should see Shan [1995] for a simple introduction). So the two extreme possibilities are to recognize characters holistically, or to base recognition on extraction of low-level features such as strokes. If we are to recognize holistically, then we need to determine the size of the set of Chinese characters. But as Sampson [1985, p. 161] writes: "to ask how many graphs there are in Chinese script is like asking how many words there are in the English language, and this is not a question with a well-defined answer." However, according to Shan [1995], more than 400,000 characters are used in Chinese newspapers, of which about 4,000 are used daily by the average person. This scale obviously challenges holistic recognition approaches. Hence, most attempts to solve the problem use stroke extraction as a precursor to subsequent recognition. By contrast, according to Ip et al. [1997, p. 185], "very few studies have focused on... recognizing a character by first extracting and recognizing its radicals." This is probably because earlier work concentrated on the easier task of online recognition, in which stroke information is fairly directly encoded in the input—but this is not the situation in offline recognition. In this work we pursue a radical-decomposition approach to the offline problem based on the intuition that it effects a simplifying tradeoff between holistic recognition of large numbers of complete characters and extraction and recognition of large numbers of highly variable strokes. Again, asking how many radicals there are in Chinese script is not a question with a well-defined answer, so that estimates of the number vary according to how different authorities count different radicals. However, the number is often said to be about 350, which is manageably small [Liao and Huang 1990; Chung and Ip 2001]. Hence, radical-based

recognition is likely to avoid the worst difficulties of each of the more extreme strategies.

The remainder of this paper is organized as follows. The composition (and corresponding decomposition) of Chinese characters in terms of radicals is described in Section 2; we discuss the different sources of variability in Chinese character recognition, and argue that the radical-decomposition approach offers the potential to model each source in the most appropriate way in Section 3; Section 4 then describes the proposed technique of active radical modeling, including the database used, landmark-point labeling, eigenvector computation, and extensions to handle nonlinear variation; in Section 5, we detail how radical matching within a character is achieved by adjusting the shape parameters during a search for the model parameters that best explain the observed data, and give results of some experiments on radical matching; subsequently, the recognition of complete, multiradical characters by Viterbi decoding is described in Section 6; finally, conclusions are presented in Section 7, along with a discussion of some related work.

## 2. CHINESE CHARACTERS AND RADICALS

Unlike the Semitic-derived family of scripts (e.g., alphabetic writing system of English and Arabic), which are basically phonographic, Chinese writing is logographic. According to Sampson [1985], a graph (i.e., character) of the Chinese writing system stands not for a unit of pronunciation but for a morpheme—a minimal meaningful unit of the Chinese language. Graphs are composed of simple graphs in Sampson's terminology (called radicals in this work) which appear in a specific position, such as left-hand radical, upper radical and so on. Only a small number of such radicals can compose many different Chinese characters [Suen and Huang 1984].

Radical approaches decompose large numbers of Chinese characters into a smaller set of radicals. Thus, the complex character recognition problem is converted to a simpler one of radical extraction and optimization of combination of the radical sequences. Figure 1 illustrates how four complex Chinese characters can be represented by hierarchical composition of radicals from the set {+, 日, 大, 土, 巾, 力}. If the redundant information is ignored, the four characters can be classified by recognizing elements of the peripheral radical set {土, 日, 巾, 力}. Intuitively, this should be easier than recognizing whole characters.

To make a legitimate character, radicals are placed in specific positions with respect to each other. Since the size of a handwritten Chinese character is approximately constant, independent of the number of radicals, it follows that what is nominally the same radical varies in size depending upon its position within a character. Quite apart from such size effects, the "same" radical is often written differently, depending on its position. Hence for recognition purposes, it is sensible to treat what is nominally the same radical as different (position-dependent) radicals. (This fact contributes to the difficulty, mentioned above,

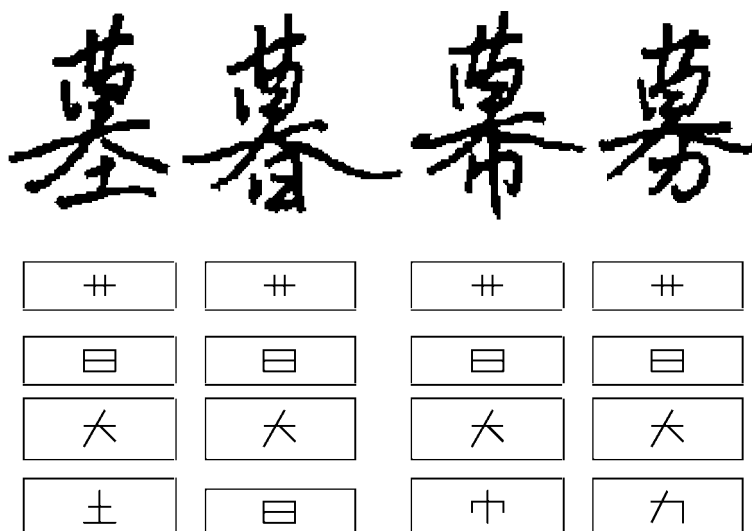


Fig. 1. Examples of complex Chinese characters decomposed into idealized radicals. They look similar, but they are absolutely different (one-word) characters which mean *tomb*, *dusk*, *curtain* and *subscription*, respectively.

of deciding the precise number of radicals in Chinese script.) So, in our active radical models, any mean radical has its own initial location. This treatment simplifies the matching phase such that we do not need to search over the whole character image (only around the expected position for that radical), and so avoid having to segment the image—a potentially error-prone process. However, it increases somewhat the cardinality of the set of radicals that we have to deal with.

In general, in a typical Chinese character there is less interconnection within the peripheral structures than within the inner structures. Also, the peripheral radicals carry more classification information (e.g., Hildebrandt and Liu [1993]). Hence, for the present we restrict attention to peripheral radicals only. This obviously places a limit on the number of characters that can be recognized, but at this stage our purpose is to demonstrate the capabilities and potential of the radical approach rather than build a complete and final recognition system. In this research, all the peripheral radicals can be sorted into 9 types: left-hand side, right-hand side, upper side, lower side, surrounding, and 4 diagonal corner radicals.

Let  $L$ ,  $R$ ,  $U$ ,  $D$ , and  $SU$  denote left, right, up, down, and surrounding radicals, respectively, and  $TL$ ,  $TR$ ,  $BL$ , and  $BR$  denote top-left, top-right, bottom-left, bottom-right radicals, respectively. Figure 2 shows the 12 basic compositions for producing characters from these 9 radicals. This description in terms of 12 types simplifies both the radical matching and combination procedures. Note that the radical structure of the four characters depicted in Figure 1 is obtained as a hierarchical composition of  $U$  and  $D$  substructures.

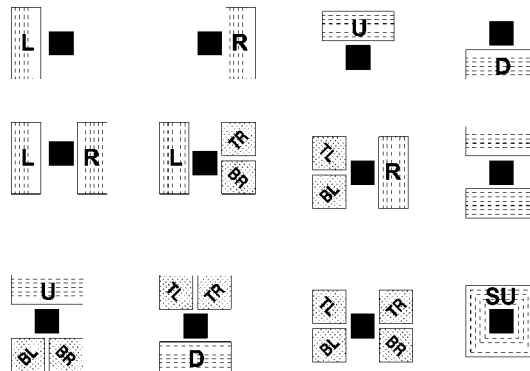


Fig. 2. Twelve basic structures of Chinese characters composed from nine peripheral radicals. The filled rectangle indicates that other structures can appear in the indicated position, so building characters up hierarchically.

### 3. SOURCES OF VARIABILITY IN CHINESE CHARACTER RECOGNITION

There are three main sources of variability in Chinese character recognition:

- (1) writer variability *within* radicals, according to which the same radical is written very differently by different writers, and by the same writer at different times;
- (2) the variability of spatial relationships *between* radicals. Again, this depends on the particular writer and different times of writing;
- (3) the statistics on usage of radicals to compose characters in the Chinese writing system.

In our opinion, these very different sources of variability need to be handled very differently. It makes little sense to try to model such widely different kinds of variation within a single framework. Crucially, the radical approach allows us to do this, as we now show.

Figure 3 depicts our Chinese character recognition system as a two-stage process in which we separately model sources of variability 1 and 3, respectively. The first stage of radical recognition is naturally model-based. Here, we capture the within-radical variability by using active shape modeling, as described in detail in Section 4. At the second stage, we capture the statistics on the use of radicals by frequency counts in a lexicon, and use the Viterbi algorithm to effect overall character recognition, see Section 6.

Note that we do not need to model variability 2 at all. It is implicit in our use of a position-dependent representation of radicals. For example, we only need to know that a *U* radical is above a *D* radical: The exact spatial relationship and its variation is unimportant in our scheme. This kind of high-level understanding of how radicals relate spatially is known to any Chinese writer, and can be much more simply encoded in a knowledge-based way, as described in the previous section, than learned from example data. Yet this source of variability causes major problems in other approaches that use holistic recognition, or recognition based on strokes or other low-level features. In particular, where

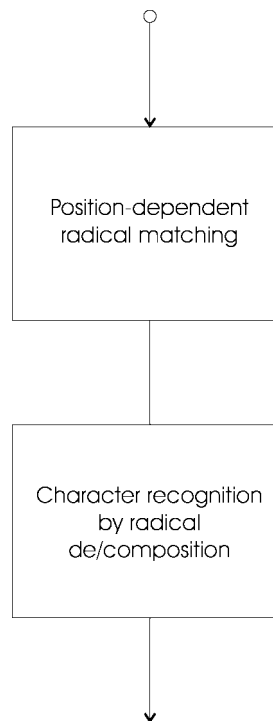


Fig. 3. Chinese character recognition as a two-stage process: In the first stage, radicals are matched in a position-dependent way using image information alone (i.e., without segmentation). In the second stage, characters are recognised by finding the best composition of radicals. The advantage of this two-stage approach is that the important sources of variability in Chinese handwriting can be handled separately and in the most appropriate manner (see text).

different radicals overlap, it can be extremely difficult to separate them. Yet our method, by matching radicals independently of one another at the first stage of recognition, handles this situation directly and without difficulty. It is as simple to recognize overlapping radicals as it is to recognize well-separated ones.

#### 4. ACTIVE RADICAL MODELING (ARM)

Our research aims to avoid stroke extraction by building up a statistical representation of each radical's shape and its variability with active shape modeling (Cootes et al. [1995]), in which the principal variations of the radical are encoded in a small number of parameters. Active shape models have similarities to snakes, in which a contour is fit to the image by minimizing an energy function. But a snake has only very generic prior knowledge, such as smoothness, about the shapes that it fits. Yet a great deal of prior information obviously exists about radicals, otherwise readers would not be able to read them! This prior knowledge can be recovered from training data, encoded within an active shape model, and exploited in recognition.

In recent years, active shape modeling has become established as a powerful and popular technique in computer vision. In the present work, we exploit this

technique for offline Chinese character recognition in the form of active radical modeling (ARM). The first step is to extract character skeletons by an image-thinning algorithm (i.e., Jang and Chin [1992]), so our approach is skeleton-rather than stroke-based. Landmark points are then labeled semi-automatically to represent a radical; principal component analysis is applied to obtain the main shape parameters that capture the radical variations.

Although semiautomatic, landmark labeling requires manual intervention and is time consuming. This is obviously not ideal; but at this stage of our research, while we are exploring the potential of ARM, the expense is justifiable.

#### 4.1 HITPU Database

Ideally, we would like to use a large, publicly-accessible database of Chinese characters. Public accessibility makes it possible for authors to compare their different techniques on the same data. Unfortunately, such a database did not exist at the commencement of this work.

We use data collected at the Harbin Institute of Technology and Hong Kong Polytechnic University [Shi et al. 2000], which we call the HITPU database. The complete database consists of each of the 3,755 characters taken from the GB2312 (simplified Chinese) set, which was written by 200 writers under loosely-constrained conditions. Hence, there are 751,000 characters in total. We stress that there is only one version of each character from each writer (i.e., no repetitions). Ideally, we would have liked multiple versions of each character so that intrawriter variation could be represented in the database. However, it was felt to be enough of an imposition to have each writer produce 3,755 characters and, in any event, interwriter variation is likely to be considerably greater than intrawriter variation. So there is probably more than enough variability in the database for research and evaluation purposes. Figure 4 shows representative examples of character *ying* (in *pinyin*, a scheme in widespread international use for the romanization of the Chinese language), meaning *elite*, written by 25 of the writers. This figure gives a good idea of the degree of interwriter variability in the HITPU database and, therefore, of the difficulty of the recognition tests described.

In the present work we have simplified the recognition problem by restricting attention to a subset of 200 of the most common and productive radicals in the database (as selected by author DS). This simplification is appropriate because at this stage our concern is to demonstrate the power of the radical decomposition approach: the time and effort required for semiautomatic landmark labeling of the whole database is not justified. Consequently, only 2,154 of the 3,755 character classes are used. These characters contain 4 radicals or fewer. They all include one or more of the selected 200 radicals, although others may be included also (and are ignored when assessing recognition accuracy in this work). The  $2,154 \times 200 = 430,800$  characters are each represented as  $64 \times 64$  pixel binary images, thinned to one pixel-wide. Note that there are some 2.6 radicals per character on average, although this figure includes the “irrelevant” radicals (i.e., those not from the 200 classes mentioned above). To fill the need for a large publicly-accessible dataset of Chinese



Fig. 4. Example Chinese characters from the database: the character *ying* (meaning *elite*) written by 25 different writers, illustrating the degree of inter-writer variability in the HITPU database.

characters on which different approaches can be compared by different authors, the HITPU database is made freely available for research purposes at <http://www.ntu.edu.sg/asdmshi/hitpu.html>.

Since the commencement of this work, we have become aware of a large database of Chinese characters produced under the China National 863 Project. It contains freely-written versions of the 3,755 character classes in GB2312 from 1,000 writers. As in the HITPU database, and presumably for the same reasons, there is only a single example of each character from each writer. Although the 863 database is five times larger than HITPU (i.e., there are 1,000 writers instead of 200), we have been unable to obtain a copy in spite of strenuous and sustained efforts. Also, no character recognition results that made use of this database have yet appeared in the open literature.

#### 4.2 Landmark Point Labeling

Active shape modeling requires training data from which models may be built. For the work reported here, we have used 60 examples of each of the 200 radical classes, randomly sampled from among the 200 writers. Each is represented by



a vector of landmark points, to capture shape constraints in training and to construct plausible new examples of the shape for use in matching.

The labeling of landmark points (by author DS) was semiautomatic. We did not wish to be distracted at this stage by tackling the difficult task of fully-automatic landmark labeling. First, the terminating points of a stroke were marked manually and then the computer attempted to interpolate automatically to give a total of 10 equally-spaced points along each such stroke. This process used a snake-like technique to attract the interpolated points onto the image (external energy), while preventing adjacent points from moving in divergent directions (internal energy). It was occasionally necessary to adjust some of the interpolated points manually—especially when strokes intersected. Figure 5 illustrates landmark labeling for the simple radical *mù*, meaning *tree*.

Although it is semiautomatic, landmark point labeling is still labor intensive. Note too that only a small proportion (about 1%) of the total data is used in training. In particular, not all writers are represented in the training data. Hence we will not maintain the traditional distinction between seen and unseen data when reporting results later. To a first approximation, all the data are “unseen.”

#### 4.3 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a powerful technique for extracting structure from possibly high-dimensional datasets. It aims to find a low-dimensional representation by an orthogonal transformation of the coordinate system, which explains most of the variance [Duda and Hart 1973; Jolliffe 1986; Cherkassky and Mulier 1998]. It is readily done by solving an eigenvalue problem. Usually, only a small number of principal components (or “modes of variation”) is sufficient to account for most of the structure variance.

This section describes how to obtain the eigenvectors in order to capture the main directions of variance within the radical, which is a crucial point in active radical modeling. We choose to model variation about the mean radical on the assumption that this will be much less than the variance in the image space. Given a set of examples  $\{\Gamma_1, \Gamma_2, \dots, \Gamma_M\}$  for a radical, which are represented by  $N$  landmark points, i.e.,  $\Gamma_k = (x_{k0}, y_{k0}, \dots, x_{k(N-1)}, y_{k(N-1)})^T$ , the mean radical of the set is given by  $\Psi = \frac{1}{M} \sum_{k=1}^M \Gamma_k$ . In this work,  $M = 60$  and  $N = 10S$  where  $S$  is the number of (idealized) strokes in a given radical. The centralized radical difference from the mean radical is given by  $\Phi_k = \Gamma_k - \Psi$ . The centralized radical matrix  $\mathbf{A}$  is then formed as  $[\Phi_1 \ \Phi_2 \ \dots \ \Phi_M]$ , having  $2N$  rows and  $M$  columns. The eigenvectors  $\mathbf{u}_i$  (each of dimension  $2N \times 1$ ) and eigenvalues  $\lambda_i$  ( $i = 1, \dots, M$ ) of the covariance matrix

$$\mathbf{A}\mathbf{A}^T = \frac{1}{M} \sum_{k=1}^M \Phi_k \Phi_k^T$$

are solutions to

$$(\mathbf{A}\mathbf{A}^T)\mathbf{u}_i = \lambda_i \mathbf{u}_i.$$

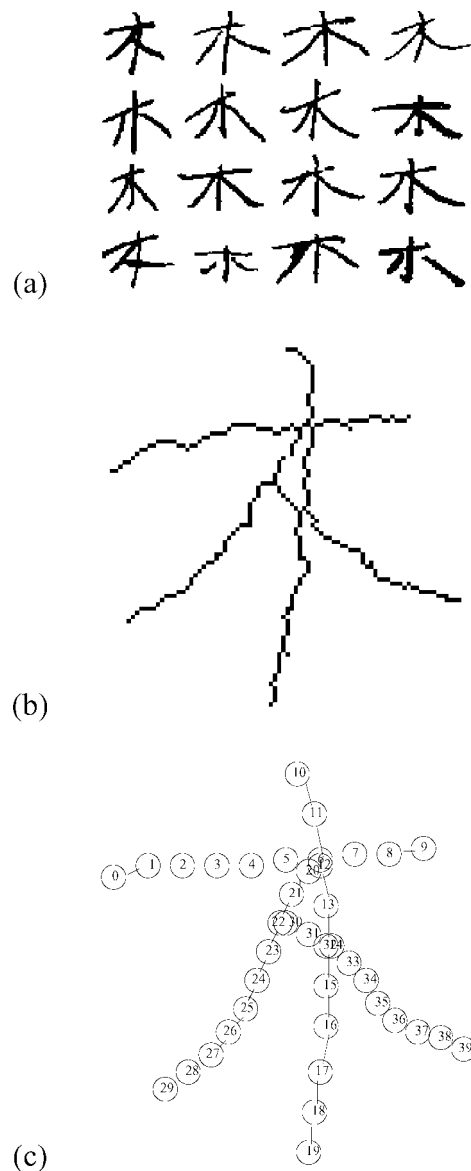


Fig. 5. Illustration of landmark labeling of the radical *mù* (meaning *tree*). (a) 16 different original image examples. (b) The skeleton for the first example in (a). (c) Landmark points for the skeleton in (b).

The variance explained by each eigenvector is equal to the corresponding eigenvalue. However, most of the variation can usually be explained by a small number of modes,  $M'$ . In practice, a small  $M'$  is sufficient for pattern recognition, since accurate reconstruction of the image is not a requirement. Any radical in the training set can be approximated using the mean radical and a weighted

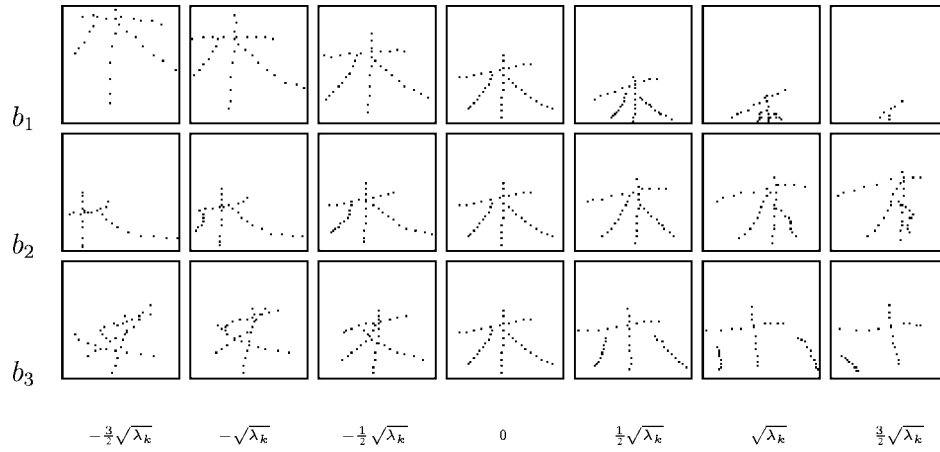


Fig. 6. The effects of varying the first three shape parameters for the radical  $mù$ . Note that the allowable bounds of variation for parameter  $b_k$  in matching an active radical model to a test image are  $\pm 3\sqrt{\lambda_k}$  (i.e., wider than shown here.)

sum of these shape variations obtained from the first  $M'$  modes:

$$\Gamma_{(2N \times 1)} = \Psi_{(2N \times 1)} + \mathbf{U}_{(2N \times M')} \mathbf{b}_{(M' \times 1)}. \quad (1)$$

Here  $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_k, \dots, \mathbf{u}_{M'})$  is the matrix of the first  $M'$  eigenvectors, and the shape parameters  $\mathbf{b} = (b_1, \dots, b_{M'})^T$  control the variation in shape around the mean radical. In this work,  $b_k$  is only allowed to vary in the range  $-3\sqrt{\lambda_k} \leq b_k \leq 3\sqrt{\lambda_k}$  when generating models for the matching phase, and  $M'$  is set such that  $(\sum_{i=1}^{M'} \lambda_i) / \lambda_T > 90\%$ , which means that  $M'$  will be different from radical to radical; it is typically 5 or 6.

Figure 6 shows the radical skeletons generated by varying the shape parameters independently for the  $mù$  radical. It is apparent that, for this radical, the first parameter captures the variation of image scale and, to some extent, mean position. The main effect of varying the second shape parameter is to alter the angle between the two oblique strokes, while the third parameter mainly controls the spatial separation of these oblique strokes. The matching phase will be carried out by varying the  $b_k$ 's actively to fit the radical model to the image object.

#### 4.4 Extending ARM with Kernel PCA

The approach described until now uses *linear* active radical models, and so it implicitly assumes that nonlinear handwriting effects are negligible. Judging from the degree of interwriter variability illustrated in Figure 4, this simplifying assumption is unlikely to hold. We therefore extended linear active radical modeling to nonlinear ARM by using kernel PCA [Schölkopf et al. 1998]. The basic idea of kernel PCA is that linear PCA is performed not in the original input space but in a high-dimensional feature space related to it by a nonlinear

map. We do not go into details of the method here, but instead refer the reader to Shi et al. [2002] and Shi et al. [2003].

## 5. MATCHING RADICALS WITHIN A CHARACTER

Given a test image in the form of a (thinned) skeleton and our set of active radical models, we need a means to determine the distance between the  $N$  landmark points of each candidate radical and the test image. This can be done by one of several methods, such as computing the Hausdorff distance [Huttenlocher et al. 1993], but in the present work we use the chamfer distance transform (see below), primarily because it converts binary to continuous, gray-scale images in a way that gives gradient information that can be used in subsequent gradient descent search for optimal shape parameters. In the transformed continuous-valued image, each point represents a global, weighted distance to the original image points. Distance calculation is then a simple matter of “picking off” points in the chamfer-transformed test images corresponding to the  $N$  landmark points of the active radical model, summing these  $N$  chamfer distance values and normalizing by  $N$ .

Furthermore, the models are deformable; by varying the  $\mathbf{b}$  parameters within some bounds, according to Eq. (1), we can vary the match to the test image. Hence, we can employ gradient descent search to adjust the shape parameters to fit the character image. Due to the orthogonality of the principal components, we can search for each  $b_k$  separately, which reduces computational complexity enormously. However, a good basin of attraction is required to find the shape parameters efficiently. Again, the chamfer distance transform can help to achieve this by providing gradient information.

In matching radical models to a test image, no penalty is applied for varying a shape parameter  $b_k$  from its mean value. We rely on the penalty (i.e., distance metric) implicit in the chamfer transform itself. However, to prevent implausible shape deformations, the variations are constrained to lie within  $\pm 3\sqrt{\lambda_k}$ , as mentioned above. We now give a more detailed treatment of the matching process.

### 5.1 Chamfer Distance Transform

The chamfer distance transform was first proposed by Barrow et al. [1977], and then systematically developed by Borgefors [1988]. The basic idea is to replace each pixel in the image by some global weighted measure of distance to image points. Borgefors improved the original technique in two aspects. First, she used the root-mean-square average as a distance measure instead of the arithmetic mean. Second, she applied a hierarchical structure in matching the image to the model.

One significant property of the chamfer distance transform is its ability to handle noisy and distorted data, as the edge points of images are transformed by a set of parametric transformation equations, which describe how the images can be geometrically distorted in relation to one another. The transform approximates global distances by propagating local distances at image pixels. In the simple case of a binary edge image, each edge pixel is first set to zero and

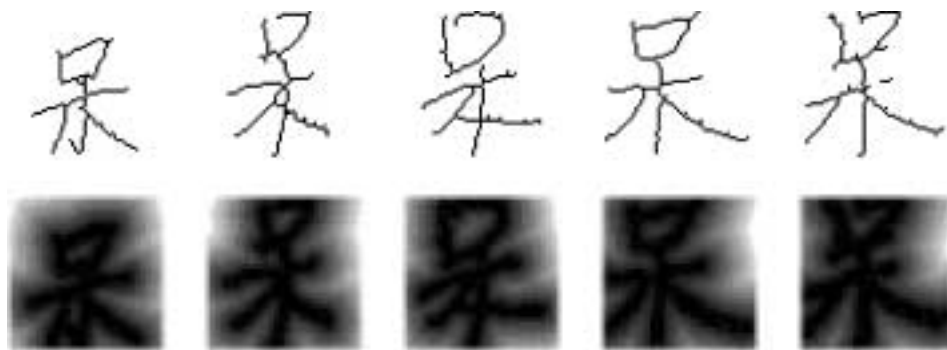


Fig. 7. Character skeletons and their chamfer distance transformed images for some different versions of *dai* (meaning *stupid*). The effect of the transform is to blur the original image so as to tolerate noise and distorted data. Hence, a basin of attraction is created for gradient search for appropriate shape parameters.

each nonedge pixel is set to infinity. A chamfer distance is a sequential distance transform. A  $(3 \times 3)$  window is used to scan the image forwards (from top-left to bottom-right) and backwards (from bottom-right to top-left). For each pixel, a new value is calculated iteratively, as follows:

**Forwards:**

$$I_{\text{new}}(x, y) = \min \begin{bmatrix} I_{\text{old}}(x-1, y-1) + C_2 \\ I_{\text{old}}(x, y-1) + C_1 \\ I_{\text{old}}(x+1, y-1) + C_2 \\ I_{\text{old}}(x-1, y) + C_1 \\ I_{\text{old}}(x, y) \end{bmatrix}$$

**Backwards:**

$$I_{\text{new}}(x, y) = \min \begin{bmatrix} I_{\text{old}}(x, y) \\ I_{\text{old}}(x+1, y) + C_1 \\ I_{\text{old}}(x-1, y+1) + C_2 \\ I_{\text{old}}(x, y+1) + C_1 \\ I_{\text{old}}(x+1, y+1) + C_2 \end{bmatrix}$$

where  $C_1$  and  $C_2$  are positive constants, equal to 3 and 4, respectively, in the work of Borgefors [1988] and here. The above calculation is carried out iteratively until there is no further change to any pixel, to produce the chamfer transformed image  $I(x, y)$ .

From Figure 7, we see that the effect of the transform is to blur the original image so as to tolerate noise and distorted data, hence a basin of attraction is created. As stated above, the distance between a model and test image is calculated by superimposing the model's landmark points onto the chamfer distance transformed image, summing the corresponding chamfer image point values and normalizing by the number of landmark points. In the following, we

treat this distance as an “energy,” as the latter terminology is more usual in active shape modeling.

### 5.2 Chamfer Distance Minimization

The chamfer distance transformed image is a functional, as follows:

$$I(x, y) = D_{\text{chamfer}}[I(x, y)].$$

Hence, the energy of a radical model  $\Gamma$  is given by

$$E_{\Gamma}(\mathbf{b}) = \frac{1}{N} \sum_{j=1}^N I(x_j, y_j) + \frac{\|\mathbf{b}\|^2}{M'}$$

where  $(x_j, y_j)$  are the coordinate points of the  $j$ th radical model deformed according to  $\mathbf{b}$ . Here, as previously stated, we ignore the second term to leave the distance contribution only (see above):

$$E_{\Gamma}(\mathbf{b}) = \frac{1}{N} \sum_{j=1}^N I(x_j, y_j). \quad (2)$$

We want to find the minimum energy according to

$$\begin{aligned} \frac{\partial E_{\Gamma}}{\partial \mathbf{b}} &= \frac{1}{N} \sum_{j=1}^N \frac{\partial}{\partial \mathbf{b}} I(x_j, y_j) \\ &= \frac{1}{N} \sum_{j=1}^N \left( \frac{\partial I}{\partial x_j} \frac{\partial x_j}{\partial \mathbf{b}} + \frac{\partial I}{\partial y_j} \frac{\partial y_j}{\partial \mathbf{b}} \right) \\ &= \frac{1}{N} \sum_{j=1}^N \left( \frac{\partial I}{\partial x_j} \mathbf{u}_{2j-1} + \frac{\partial I}{\partial y_j} \mathbf{u}_{2j} \right) = \mathbf{0} \end{aligned} \quad (3)$$

where  $\mathbf{u}_{2j-1}$  and  $\mathbf{u}_{2j}$  are the column vectors forming the relevant columns of the matrix of eigenvectors  $\mathbf{U}$ .

One solution to this problem is to use gradient descent, with the partial derivatives estimated from finite differences. Given an initial position of a model  $\Gamma$ , the solution is obtained by adjusting the model coefficients  $\mathbf{b}$ , using the gradient information so as to satisfy Eq. (3). A given test image is matched against each active radical model and the corresponding minimum energy  $E(\mathbf{b})$  from Eq. (2). These values can then be ranked to allow a recognition decision to be made.

### 5.3 Gradient Descent with Dynamic Tunneling Algorithm

In previous work, standard gradient descent was applied to search for shape parameters [Shi et al. 2001a, 2001b]. Early results were obtained from a small number of radicals (66 only), which means there are only a few possible radicals in some positions, yet the correct matching rate with standard gradient descent was just 75.4%, indicating the need for a better search procedure. The problem

is caused by local minima in the basin of attraction nearest to the starting point. In other words, as commonly found in gradient descent search, the effectiveness of the algorithm depends heavily on the initial point from which the search starts and the topology of the surface associated with the objective function.

To address the problem of local minima in optimization problems, Yao [1989] proposed the dynamic tunneling algorithm (DTA), which is based on a physical analogy to the quantum-mechanical tunneling of a particle through a potential barrier to a point of lower energy. In the context of search, this means that we can escape from a local minimum by “tunneling” to a point of even lower energy without visiting large numbers of intermediate points of relatively higher energy. The reader is referred to the original paper of Yao [1989] for details of the algorithm.

RoyChodhury et al. [2000] utilized gradient descent with DTA. That is, a gradient descent phase finds a minimum local to the search point, after which a dynamic tunneling phase is entered to search for a point of even lower energy. They point out (on p. 386 of RoyChodhury et al. [2000]) that the concept of quantum-mechanical tunneling is probabilistic. To avoid using a probabilistic framework, they allow a search point in the dynamic tunneling phase “to penetrate freely by invoking a small perturbation around the [potential] well...”. For searching along a unidimensional ( $x$ ) axis with energy function  $f(x)$ , the dynamics of this perturbation are conveniently governed by the following equation:

$$\frac{dx}{dt} = \rho(x - x^*)^{1/3} \cdot \text{sgn}(f(x) - f(x^*))$$

where  $\rho$  is the strength of the “repeller,”  $x^*$  is the last local minimum computed, and  $\text{sgn}()$  is the signum or Heaviside function that is equal to 1 for positive values of its argument and equal to 0 otherwise.

RoyChodhury et al. [2000] show that, from any initial condition, a search point will be attracted to an attractor or repelled from a repeller (according to the sign of the RHS) in finite time, given by

$$t = \frac{3}{2}\rho(x - x^*)^{2/3}.$$

Hence, in the dynamic tunneling phase, we jump from local minimum  $x^*$  to a new search point:

$$x = x^* + \sqrt{\left(\frac{2t}{3\rho}\right)^3} \quad (4)$$

where the energy  $f(x)$  may or may not be lower. Treating  $t$  as an index of iteration, we can increment  $t$  until such time as  $f(x) < f(x^*)$ . Hence, the dynamic tunneling procedure can jump to another basin of attraction where the new, initial search point is even lower in energy. From this new starting point, gradient descent can again be used to find a lower minimum. If no such point is found, the search terminates. Here  $\rho$  acts as a parameter adjusted to suit the particular search problem; increasing  $\rho$  will decrease “step size” for the search. For kernel PCA, we used  $\sqrt{\left(\frac{2}{3\rho}\right)^3} = 0.001$ .

Since all the eigenvectors are orthogonal in our ARM work, the shape parameters can be specified one by one. On the grounds that the global optima are close to  $b_k = 0$  in most cases, it is more efficient to start the search procedure at this point (rather than from extremal points as in RoyChodhury et al. [2000]). An optimum  $b_k^+$  is obtained in the positive direction first, and then the global optimum  $b_k^*$  is obtained after the search is completed in the negative direction. The algorithm for searching the  $k$ th shape parameter  $b_k$  by gradient descent with DTA is then as follows:

<b>Step 1:</b>	$b_k^* = 0, b_k^+ = 0.$
<b>Step 2:</b>	Search for local optimum with gradient descent from point $b_k^+$ with equation (3), to obtain new current global minimum $b_k^*$ .
<b>Step 3:</b>	Dynamic tunneling phase begins, setting the tunneling time $t = 0.$
<b>Step 4:</b>	$t = t + 1.$
<b>Step 5:</b>	In the case of positive direction, select the point $b_k = b_k^* + 0.001\sqrt{t^3}$ according to equation (4); In the case of negative direction, select the point $b_k = b_k^* - (b_k^+ + 0.001\sqrt{t^3})$ (i.e., treating $b_k^+$ as an offset)
<b>Step 6:</b>	In the case of positive direction, if $b_k > 3\sqrt{\lambda_k}$ , then $b_k^+ = b_k^*$ , and go to <b>Step 3</b> ; In the case of negative direction, if $b_k < -3\sqrt{\lambda_k}$ , go to <b>Step 10</b> .
<b>Step 7:</b>	Calculate the energy function in $b_k.$
<b>Step 8:</b>	If $E(b_k) > E(b_k^*)$ go to <b>Step 3</b> to continue dynamic tunneling phase.
<b>Step 9:</b>	Otherwise, go to <b>Step 2</b> to start a gradient descent phase.
<b>Step 10:</b>	End.

This is repeated for all  $k, 1 \leq k \leq M'.$

After gradient descent with DTA, the optimal shape parameters for each radical class specify the corresponding energy minimum according to Eq. (2). The decision rule is then simply to select the radical class with the overall minimum.

#### 5.4 Experiments and Results on Radical Recognition

Experiments were conducted on the HITPU database (Section 4.1) to evaluate the radical recognition rate. This was done on a PC (Pentium III 450 MHz, 128M RAM) using C++, and Matlab to perform the eigenvector computations. The 200 radicals in our experiments appear in 2,154 commonly used Chinese characters in our database. A lexicon of these 2,154 characters has been built up, in which each character is a 9-dimensional feature vector corresponding to the 9 types of radicals ( $L, R, U, D, SU, TL, TR, BL,$  and  $BR$ ), as in Figure 2. This lexicon, which implicitly defines the allowable compositions of characters from radicals, is necessary for subsequent whole-character recognition (Section 6).

The number of correct radicals recognized is then determined as follows: Consider a given character with  $1 \leq \Omega \leq 4$  radicals from the set of 200 under consideration in its lexical entry. Each of these  $\Omega$  radicals has its own characteristic position, defined by its position in the feature vector, which it shares with (on average) something like 10 other possible radicals. For  $\omega = 1, \dots, \Omega,$  we count one correct recognition if and only if the  $\omega$ th radical is the top-ranked radical



of those sharing the same characteristic position. Here, “top-ranked” refers to the ranked list of chamfer distances computed as described in Section 5.2. This is repeated for all characters to give an overall figure for the 430,800 characters.

On this basis, the matching rate is 96.5% radicals correct, relative to a chance level of about 10% (since there are approximately 10 candidate radicals at each position). This is actually a rather strict measure of correctness, since each of the  $\Omega$  radicals has to out-rank all of its competitors at the same position. That is, we disregard knowledge encoded in the lexicon about which radicals can legally appear in combination with other radicals. Of course, Viterbi decoding (described in Section 6) takes precisely this information into account.

Since kernel PCA is more computationally expensive than linear PCA, its use requires some justification. Using linear PCA earlier, our radical recognition rate was 94.2%, compared to 96.5% obtained here under the same conditions. If we assume that the distribution of errors in radical recognition is binomial (the result is either right or wrong), then the standard deviation is  $\sqrt{n(1-p)p}$ , where here  $n$  is the number of radicals (approximately 1.1 million) and  $p$  is the error probability. For kernel PCA, this equates to  $\sqrt{1.1 \times 10^6 \times 0.035 \times 0.965} \sim 193$ . For  $n$  large, as here, the binomial distribution is well represented as a normal distribution. Hence, we can have 99% confidence that the true number of errors is in the range  $np \pm 3\sqrt{n(1-p)p}$ , or  $1061500 \pm 579$ , corresponding to a range of recognition rates from 96.4% to 96.6%. It follows that kernel PCA provides a small but highly significant improvement on linear PCA.

## 6. CHARACTER RECOGNITION WITH THE VITERBI ALGORITHM

The output of the radical extraction stage is a set of radicals ranked by their chamfer distance to the given character. Treating Chinese character composition as a discrete Markov process corresponding to a sequence of radicals, the optimal radical combination is equivalent to the “best” path in a graph made up of all possible radical combinations. The best path can be determined according to estimated probabilities of initial state, transitions between states, and symbol probabilities at each state (see below and Xiong et al. [2001], who have recently presented a somewhat similar model).

### 6.1 Markov Process of Character Composition

As stated in Section 4.1, any character considered here consists of up to four radicals. Each radical corresponds to a state,  $x_k$ ,  $k \in \{1, \dots, K\}$ , with  $K = 4$ . The composition process is assumed to be Markov, in the sense that the probability of being in state  $x_{k+1}$  at index  $k+1$ , given all states up to index  $k$ , depends only on the state  $x_k$  at index  $k$ :  $P(x_{k+1}|x_0, x_1, \dots, x_k) = P(x_{k+1}|x_k)$ . With such a model, Chinese characters can be viewed as the outputs of an  $m^K$ -state Markov process, where  $m$  is the number of distinguishable radicals ( $m = 98$  for the results reported in this section). This, however, over-counts the number of states, because radicals are position-dependent and cannot appear anywhere else other than in their characteristic position. In practice, the total number of states will be considerably less than  $98^4$ , i.e., there will be many zero-probability transitions.

Transition probabilities are defined according to the allowable sequences of radicals, or decompositions of the Chinese characters. Hence, to allow the powerful Markov formalism to be used, we make the assumption that such sequences actually exist, whereas, in fact, sequential information is entirely absent from an offline character image. The assumed sequence in this work is  $L, U, R, D, TL, TR, BR, BL, SU$ . This is the order in which the composition of characters in terms of radicals is entered into our lexicon of allowable characters. We then find transition probabilities by frequency counts in the lexicon.

It is important to emphasize that the assumed sequence is arbitrary. In particular, a different choice might have led to different results, since the transitions would have differed. Our intuition is that the variation in performance is likely to be small but this issue remains to be investigated.

## 6.2 Decoding Algorithm

In this research, Chinese character recognition is associated with a graph where the nodes contain radical recognition scores (i.e., chamfer distances). A one-to-one correspondence exists whereby every path through the graph corresponds to a particular legal segmentation of the input character into radicals, and conversely, every possible legal segmentation of the input character corresponds to a particular path through the graph.

In the lexicon, each character consists of nine codes:  $1xx, 2xx, \dots, 9xx$ , representing the nine types of radical:  $L, U, R, D$ , etc. The Viterbi algorithm [Viterbi 1967; Forney 1973; Neuhoff 1975] provides a convenient method for rapidly determining (“decoding”) the best-scoring path corresponding to an interpretation for a character. Given a character, the symbol probability of the  $j$ th radical is estimated by

$$P(j) = 1 - \frac{\text{chamfer distance of radical } j}{\text{sum of chamfer distance of all radicals in same position as } j}$$

Transition probabilities are estimated as

$$P((i, j)|(a, b)) = \frac{\text{number of transitions from } (a, b) \text{ to } (i, j)}{\text{total number of transitions from } (a, b)}$$

The initial state probability is estimated as

$$\pi(j) = \frac{\text{number of characters beginning with radical } j}{\text{total number of characters}}$$

A *survivor* is defined as the shortest path leading to a node. According to the dynamic programming principle [Bellman 1957], only survivors need be considered in determining optimal paths. Let us define the following symbols:

$x(i, j)$  is the node at the  $i$ th row and  $j$ th column;

$\hat{x}(i, j)$  is the survivor path ending at  $x(i, j)$ ;

$L(i, j)$  is the survivor path value;

$K (= 4)$  is the total number of rows in the graph;

$J (= 98)$  is the total number of columns in the graph.

A formal statement of the decoding algorithm, modified from Jung and Kim [2000], is as follows:

**STEP 1 Initialization.**  $L(i, j) = 0, \quad \forall i, j \neq 0; \quad j = 1.$   
**STEP 2**  $L(1, j) = \pi(j) \times P(j); \quad \hat{\chi}(1, j) = (1, j).$   
**STEP 3**  $i = 2.$   
**STEP 4** Calculate:  

$$L(i, j) = \max_{1 \leq m \leq J} [L(i-1, j) \times P((i, m) | \hat{\chi}(i-1, j))] \times P(j);$$

$$\hat{\chi}(i, j) = (i, m), \quad \text{s.t.} \quad \max_{1 \leq m \leq J} [L(i-1, j) \times P((i, m) | \hat{\chi}(i-1, j))];$$
  
**STEP 5**  $i++$ ; Repeat Step 3 while  $i \leq K.$   
**STEP 6**  $j++$ ; Go to Step 2 while  $j \leq J.$   
**STEP 7 Termination and backtracking.** The best path is:  

$$\hat{\chi}(1, v), \hat{\chi}(2, v), \dots, \hat{\chi}(K, v) \quad \text{s.t.} \quad L(K, v) = \max_{1 \leq j \leq J} L(K, j)$$

### 6.3 Results on Character Recognition

Using Viterbi decoding, the performance for whole character recognition was evaluated under the same conditions as for the assessment of the radical recognition rate (see Section 5.4). The obtained recognition rate was 93.5% characters correct.

## 7. CONCLUSIONS AND DISCUSSION

An approach to radical recognition for offline handwritten Chinese characters has been proposed, based on nonlinear active shape modeling. We call this active radical modeling (ARM). Only a small number of principal components (modes of variation) can capture the main variations of radicals. Chamfer distance minimization is used to match radicals within a character using the dynamic tunneling algorithm (DTA) to search for the best shape parameters to describe the deformation of an active model to fit a test image.

Experiments for (writer-independent) radical recognition were conducted on 200 radical categories covering 2,154 loosely-constrained character classes from 200 writers, using kernel PCA to model variations around mean landmark values. Because the active models for a radical are position-dependent, recognition also corresponds to implicit extraction of radicals without any need for image segmentation. The matching rate obtained is 96.5% radicals correct. Kernel PCA was found to give a small but highly significant improvement in radical recognition performance relative to linear PCA, justifying the additional computational expense. The recognition of radicals within Chinese characters, although an important subproblem, is only part of the overall task of character recognition. Accordingly, we propose a subsequent stage of character recognition based on Viterbi decoding with respect to a lexicon, resulting in a character recognition rate of 93.5%.

We know of only one other study of Chinese character recognition that has produced comparable results—that of Ge et al. [2002], whose approach is considerably different from ours. They use four Gabor features (see Lades et al. [1993]) and their spatial derivatives to parameterize the whole character image in both horizontal and vertical directions. They then use linear discriminant analysis for data reduction and two continuous-density hidden Markov models (CDHMMs) to model the characters in the two (horizontal and vertical) directions. Gabor filters are spaced every 8 pixels, and the CDHMMs are trained using a minimum classification error criterion on 1,384,800 character examples. For a set of 4,516 character classes, Ge et al. [2002] obtain a recognition rate of 96.34% on a test set of 1,025,535 characters. Superficially, they obtain a better recognition rate on a larger test set with a larger number of character classes. Great care must be taken with this comparison, however, since we know nothing about the relative difficulty of recognition using the two datasets. It is not clear that Ge et al.'s results are necessarily any better than ours; it is safest to say that they are comparable. Also, as Ge et al. [2002] say, “offline recognition of . . . handwritten Chinese characters remains largely an unsolved problem,” a fact which encourages the search for “new ways to attack this difficult problem” (p. 1053 Ge et al. [2002]).

In this paper we have presented just such a “new way,” which we believe has many potential advantages over the work of Ge et al. [2002]. Their approach is effectively holistic, encoding the complete character image, which implicitly carries the disadvantage of trying to model simultaneously the three main sources of variability in Chinese handwriting, which we outlined in Section 3. As argued in that section, we believe that this is likely to be inferior to our problem decomposition, which allows each source to be handled in the most appropriate way. We can either learn the variation from example data as in the case of radical recognition with the ARM or estimation of the various probabilities used in Viterbi decoding, or use a simple knowledge-based approach as in the construction of the character lexicon in terms of radicals. In particular, as with most other work on the problem, Ge et al. [2002] implicitly attempted to model the spatial variability between radicals, which is entirely unnecessary in our method. We only need to know the relative positions of radicals in terms of very broad classes (*L*, *R*, *U* etc.). Like us, they avoided problematic stroke extraction, which is probably the main reason they achieved their high accuracy. But since they used no structural information whatsoever, they were forced to use a very computationally expensive procedure based on Gabor filtering for (nonstructural) feature extraction. Thus, unlike us, they required large amounts of training data (over one million characters compared to our 12,000 radicals). On the other hand, our method requires semiautomatic landmark labeling, while theirs is entirely automatic. It is clear nonetheless from our much smaller training set that our method has superior generalizability, and the possibility exists of developing techniques for automatic landmark labeling. Hence we believe there is very considerable potential for further development and improvement of the ARM method in the direction of larger character sets and less constrained writing conditions.

## REFERENCES

- BARROW, H. G., TENENBAUM, J. M., BOLLES, R. C., AND WOLF, H. C. 1977. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence* (Cambridge, MA). 659–663.
- BELLMAN, R. 1957. *Dynamic Programming*. Princeton University Press, Princeton, NJ.
- BORGEFORS, G. 1988. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* 10, 6, 849–865.
- CHANG, S. K. 1973. An interactive system for Chinese character generation and retrieval. *IEEE Trans. Syst. Man Cybern.* 3, 3, 257–265.
- CHEN, J.-W. AND LEE, S.-Y. 1997. On-line Chinese character recognition via a representation of spatial relationships between strokes. *Int. J. Pattern Recog. Artif. Intell.* 11, 3, 329–357.
- CHEKASSKY, V. AND MULIER, F. 1998. *Learning from Data*. Wiley, New York.
- CHUNG, F. AND IP, W. W. S. 2001. Complex character decomposition using deformable model. *IEEE Trans. Syst. Man, Cybern.—Part C: Applications and reviews* 31, 1, 126–132.
- COOTES, T. F., TAYLOR, C. J., COOPER, D. H., AND GARAHAM, J. 1995. Active shape models—their training and application. *Comput. Vision Image Understanding* 61, 1, 38–59.
- DUDA, R. O. AND HART, P. E. 1973. *Pattern Classification and Scene Analysis*. Wiley, New York.
- FORNEY, G. D. 1973. The Viterbi algorithm. *Proc. IEEE* 61, 3, 268–278.
- GE, Y., HUO, Q., AND FENG, Z.-D. 2002. Offline recognition of handwritten Chinese characters using Gabor features, CDHMM modeling and MCE training. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'02*, vol. 1 (Orlando, FL). 1053–1056.
- HILDEBRANDT, T. H. AND LIU, W. 1993. Optical recognition of handwritten Chinese characters: Advances since 1980. *Pattern Recogn.* 26, 2, 205–225.
- HUTTENLOCHER, D. P., KLANDERMAN, G. A., AND RUCKLIDGE, W. J. 1993. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.* 15, 9, 850–863.
- IP, W. W. S., CHUNG, K. F. L., AND YEUNG, D. S. 1997. Offline handwritten Chinese character recognition via radical extraction and recognition. In *Proceedings of the Third International Conference on Document Analysis and Recognition* (Ulm, Germany). 185–189.
- JANG, B. K. AND CHIN, R. T. 1992. One-pass parallel thinning: Analysis, properties and quantitative evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 11, 1129–1140.
- JOLLIFFE, I. T. 1986. *Principal Component Analysis*. Springer-Verlag, New York.
- JUNG, K. AND KIM, H. J. 2000. On-line recognition of cursive Korean characters using graph representation. *Pattern Recogn.* 33, 3, 399–412.
- KIM, H. J., JUNG, J. W., AND KIM, S. K. 1996. On-line Chinese character recognition using ART-based stroke classification. *Pattern Recogn. Lett.* 17, 12, 1311–1322.
- KIM, H. J., JUNG, J. W., AND KIM, S. K. 1997. On-line recognition of Chinese characters based on hidden Markov models. *Pattern Recogn.* 30, 9, 1489–1500.
- LADES, M., VORBRÜGGEN, J. C., BUHMAN, J., LANGE, J., VAN DER MALSBERG, C., WÜRTZ, R. P., AND KONEN, W. 1993. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers* 42, 3, 300–311.
- LIAO, C. W. AND HUANG, J. S. 1990. A transformation invariant matching algorithm for handwritten Chinese character recognition. *Pattern Recogn.* 23, 11, 1167–1188.
- NEUHOF, D. L. 1975. The Viterbi algorithm as an aid in text recognition. *IEEE Trans. Inf. Theory IT-21*, 222–226.
- ROYCHODHURY, P., SINGH, Y. P., AND CHANSARKAR, R. A. 2000. Hybridization of gradient descent algorithms with dynamic tunneling methods for global optimization. *IEEE Trans. Syst., Man, Cybern.—Part A: Systems and Humans* 30, 3, 384–390.
- SAMPSON, G. 1985. *Writing Systems*. Hutchinson, London.
- SCHÖLKOPF, B., SMOLA, A. J., AND MÜLLER, K. 1998. Kernel principal component analysis. In *Advances in Kernel Methods*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., Cambridge, MA: MIT Press, 327–352.
- SHAN, L. 1995. *Passport to Chinese: 100 Most Commonly Used Chinese Characters, Book 1*. EPB Publishers, Singapore.

- SHI, D., GUNN, S. R., AND DAMPER, R. I. 2001a. Active radical modeling for handwritten Chinese characters. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition, ICDAR'01* (Seattle, WA). 236–240.
- SHI, D., GUNN, S. R., AND DAMPER, R. I. 2001b. A radical approach to handwritten Chinese character recognition using active handwriting models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1 (Kauai, Hawaii). 670–675.
- SHI, D., GUNN, S. R., AND DAMPER, R. I. 2002. Handwritten Chinese character recognition using nonlinear active shape models and the Viterbi algorithm. *Pattern Recogn. Lett.* 23, 14, 1853–1862.
- SHI, D., GUNN, S. R., AND DAMPER, R. I. 2003. Handwritten Chinese radical recognition using nonlinear active shape models. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 2, 277–280.
- SHI, D. M., GUNN, S. R., DAMPER, R. I., AND SHU, W. H. 2000. Recognition rule acquisition by an advanced extension matrix algorithm. *Eng. Intell. Syst. Electrical Eng. Commun.* 8, 2, 97–101.
- SUEN, Y. AND HUANG, E. M. 1984. Computational analysis of the structural compositions of frequently used Chinese characters. *Comput. Process. Chinese Oriental Lang.* 1, 3, 1–10.
- TANG, Y. Y., TU, L. T., LIU, J., LEE, S. W., LIN, W. W., AND SHYU, I. S. 1998. Offline recognition of Chinese handwriting by multifeature and multilevel classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 5, 556–561.
- VITERBI, A. J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory IT-13*, 2, 260–269.
- XIONG, Y., HUO, Q., AND CHAN, C. K. 2001. A discrete contextual stochastic model for the off line recognition of handwritten Chinese characters. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 7, 774–782.
- YAO, Y. 1989. Dynamic tunneling algorithm for global optimization. *IEEE Trans. Syst., Man, Cybern.* 19, 5, 1222–1230.

Received January 2002; revised July 2003; accepted August 2003