

Estimating the Pen Trajectories of Static Signatures Using Hidden Markov Models

Emlri-Mari Nel, Johan A. du Preez, *IEEE, Member*, and B.M. Herbst

Abstract—Static signatures originate as handwritten images on documents and by definition do not contain any dynamic information. This lack of information makes static signature verification systems significantly less reliable than their dynamic counterparts. This study involves extracting dynamic information from static images, specifically the pen trajectory while the signature was created. We assume that a dynamic version of the static image is available (typically obtained during an earlier registration process). We then derive a hidden Markov model from the static image and match it to the dynamic version of the image. This match results in the estimated pen trajectory of the static image.

Index Terms—Pattern recognition, document and text processing, document analysis, handwriting analysis.

1 INTRODUCTION

PRODUCING cursive writing or handwritten signatures on documents involves a dynamic process—the pen position, pressure, tilt, and angle are functions of time. The end result, however, is a static image with little, if any, dynamic information encoded in it. Dynamic information, on the other hand, is particularly useful for automatic handwritten character or word recognition or for the verification of signatures [1]. Thus, extracting dynamic information from static handwritten images can have useful applications. In this paper, we study the problem of extracting the pen trajectories that created a static signature, i.e., the paths that the pen followed over the document. Thus, the problem is to unravel the script and present it as a collection of parametric curves.

There are several difficulties to overcome when recovering the pen trajectory from a static handwritten script. The first one is to find the starting position. It is often hidden inside the image (especially where signatures are considered) and not visible at all. Due to this ambiguity, strict constraints are normally required to decide where the pen trajectory starts: Typically, it is assumed that the pen trajectory must start and terminate at distinct positions [2], [3], [4]. Thus, characters such as “0,” cannot be successfully unraveled.

A second and more serious problem arises from regions containing multiple self-intersections. Signatures often have complicated regions consisting of many intersections, making it hard to track a particular path through those regions. One possibility is to assume that the direction of a line is maintained when entering and leaving an intersection. A choice between the different possibilities at the intersection is then typically based on some local smoothness criteria, as in

[5], [6], [7], [8], [9], [10]. This approach is, however, insufficient to completely resolve ambiguities. If the script becomes indistinct due to a large number of intersections in a small area, local information is not sufficient to find the correct path, and additional assumptions may be necessary; for example, restricting the number of lines that can cross at an intersection [2], [3]. Local methods struggle to take context into account and several studies therefore include global information by modeling the pen trajectory estimation problem as a graph-theoretical problem [11], [12], [13], [14], [3], [15], [16], [17].

All the studies mentioned above use only the 2D image of the script. Another approach is to use dynamic exemplars of the static image captured with a digitizing tablet at the time of registration [18], [19]. The idea is to compare a given static image with prerecorded dynamic exemplars. It is important to note that the static image is compared with generic dynamic representatives, and not a dynamic copy of itself. The additional information from the exemplars is exploited to partially resolve ambiguities in regions of multiple crossings. Some heuristic measures might still be required to resolve the ambiguities completely.

A dynamic exemplar is also valuable in resolving a third difficulty, namely, identifying turning points, where the pen stops, and then reverses direction. It should be clear that static signatures retain no information about the return portion of a pen trajectory that stops and then reverses direction, returning along the same path. Some studies therefore restrict the number of times the pen can revisit a line [2], [3].

Another major difficulty one has to address is disconnected trajectories, produced when the pen is lifted. This particular problem falls outside the scope of this paper, but, in the final section, we briefly outline the approach we are pursuing at present.

In this study, we consider only singlepath handwritten scripts, i.e., scripts that consist of a single curve created with uninterrupted, nonzero pressure. We make use of prerecorded dynamic exemplars of the static image. As alluded to above, ours is not the only approach assuming the availability of dynamic exemplars. Guo et al. [18], for example, locally compare the static and dynamic pen positions and line directions. Since a local approach does not take the global context into account, their method relies heavily on heuristics

• E.-M. Nel and J.A. du Preez are with the Department of Electrical and Electronic Engineering, University of Stellenbosch, Private Bag X1, 7602 Matieland, South Africa. E-mail: {emnel, dupreez}@dsp.sun.ac.za.

• B.M. Herbst is with the Department of Applied Mathematics, University of Stellenbosch, Private Bag X1, 7602 Matieland, South Africa. E-mail: herbst@sun.ac.za.

Manuscript received 1 Apr. 2004; revised 4 Feb. 2005; accepted 22 Feb. 2005; published online 15 Sept. 2005.

Recommended for acceptance by K. Yamamoto.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0160-0404.

to resolve ambiguities; for example, the smoothest pen trajectory is chosen at an intersection. Lau et al. [19] use a statistical approach. They derive and train four probability density functions (PDFs) for each signatory from prerecorded dynamic scripts. The skeleton of a static image is matched to these PDFs using a dynamic programming algorithm. The path with maximum likelihood is then extracted from the static image.

Our approach constructs a Hidden Markov Model (HMM) from the static image. An HMM is a probabilistic model that models a time-dependent sequence of events with a sequence of *states* having *transitions* between them [20]. In our case, the HMM describes the pen trajectory that created the image. Each state has an associated PDF, embedding geometric shape information of the static image. The HMM topology specifies the interconnection of states. Transitions between states are weighted with transition probabilities to dictate possible pen movements between static image coordinates. Normally, both the state PDFs and the transition probabilities are obtained through a training process. Although training is possible for this application as well, data scarceness is a serious problem and we chose to specify the PDFs and transition probabilities in advance.

The next step is to compare the constructed HMM with prerecorded dynamic exemplars of the image. This is done using the Viterbi algorithm [20]. Since the Viterbi algorithm is a global optimization algorithm, it is particularly useful for resolving local ambiguities due to multiple intersections. It should also be noted that the initial transition probabilities allow the estimated pen trajectory to start at any position, resolving the problem of the starting point. Turning points are dealt with by specifying appropriate transition probabilities and no restrictive assumptions are needed.

A basic first-order HMM is constructed from a static image, as described in Section 3. However, it is not able to completely resolve ambiguities in regions with multiple intersections. The problem is due to a loss of context caused by the use of first-order HMMs: transition probabilities depend only on the current state. Higher-order HMMs, whose transition probabilities depend not only on the current state but also on the previous states, are much better equipped to take context into account. Usually, higher-order HMMs tend to be computationally expensive. In this study, however, we use a second-order HMM with sparse transition probability matrices, reducing the computational cost to a manageable level. The suitable second-order HMM that is derived from a basic first-order HMM is described in Section 3.3. Further, context is provided by comparing not only pen positions but also local line directions.

The rest of this paper is organized as follows: Section 2 describes the primary preprocessing steps required to apply our method successfully. Section 3 presents our pen trajectory estimation technique. Section 4 presents a procedure for evaluating our technique and reports experimental results. Section 5 draws some conclusions.

2 PREPROCESSING STEPS

A recorded dynamic exemplar is interpolated using a cubic spline and parameterized so that any two successive points are approximately one pixel apart. Subsequent preprocessing is simple, consisting of resizing and translation, thinning, and orientation normalization. The static image is skeletonized, as

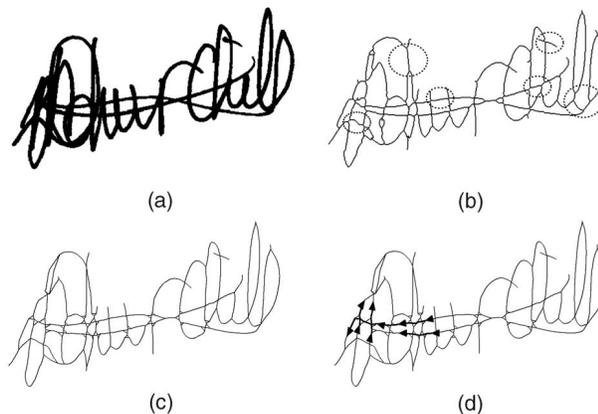


Fig. 1. (a) A binarized signature that is difficult to unravel. (b) Examples of artifacts that can occur in skeletons. (c) The final skeleton of (a), specific to our application (note the web-like structures). (d) Examples of trajectories that can be extracted from (c).

described in Section 2.1. The orientations of the static image and the dynamic exemplar are then aligned using the Radon transform, as discussed in Section 2.2.

2.1 Skeletonization

In order to extract a parametric curve from a static image, we first extract a *skeleton* from the image through a thinning process, where the skeleton coincides mostly with the centerline of the original image. Standard skeletonization techniques which do not remove noise or artifacts can be found in [21], [22], [23], [24].

Since we are attempting to extract the pen trajectory from the image, artifacts introduced by thinning algorithms can have a negative impact on our trajectory extraction algorithm. We have therefore implemented the rather sophisticated algorithm of Zou and Yan [25], as improved by Rocha [26], with a few important modifications specific to our application. The algorithm of Zou and Yan first determines the edges within the original image. By constructing Delaunay triangles [27] from control points representing these edges, one computes a skeleton that follows the centerline of the image. Additionally, one identifies the triangles that contribute to artifacts, resulting in a powerful technique to identify and remove skeletonization artifacts.

The most important modification for our application involves the skeletonization in complicated regions. The algorithms by Zou, Yan, and Rocha assume that lines do not change their orientation after entering an intersection. Due to the nature of human handwriting, especially signatures, this is not always true. When an image becomes indistinct due to multiple crossings in a small region, it is not clear which curves should be combined. If the skeletonization algorithm follows a dominant curve and strives to maintain its direction, the wrong curves may be connected, with the result that actual trajectories become irretrievably lost. In situations like these, we are careful to maintain all possible connections, while smoothing transitions at intersections as much as possible. This often results in a visually unappealing web of connected lines (see Fig. 1c). Although visually unappealing, these webs of connected lines are not a problem for our proposed method. The HMM is able to find the appropriate connections and thereby reconstructs the pen trajectory accurately.

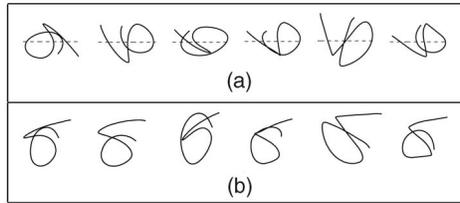


Fig. 2. Aligning signatures with (a) PCA and (b) the Radon transform.

Fig. 1a shows an example of a reasonably difficult signature to skeletonize: one that even the eye finds difficult to unravel. Fig. 1b is its skeleton as produced by the well-established thinning algorithm described in [24]. Some, but not all, artifacts are encircled with dotted lines. The modified Zou, Yan, and Rocha procedure is illustrated in Fig. 1c. Examples of trajectories that can be extracted from Fig. 1c are shown in Fig. 1d, illustrating how the web-like structures smooth transitions between intersections in complicated regions. The transitions are considerably smoother than those in Fig. 1b.

Finally, we note that our pen trajectory estimation algorithm is not particularly sensitive to a specific skeletonization scheme. The Zou, Yan, and Rocha scheme yields a 2.2 percent improvement in accuracy over the basic thinning algorithm described in [24]. A detailed discussion of these results can be found in Section 4.

2.2 Orientation Normalization

Any form of handwriting is generated in a specific overall or average direction relative to the horizontal axis, which we refer to as the *orientation* of the handwriting. Since our algorithm relies on local line directions, it is important that the static image and the dynamic exemplar have the same orientation.

Principal component analysis (PCA) is frequently used to align different shapes [22], [5], [28]. Unfortunately, this simple procedure is not reliable for signatures. Problems are encountered with shapes that do not display a clear “direction,” i.e., where the two principal values are approximately the same, as illustrated in Fig. 2a. The depicted dynamic signatures are rotated so that their principal axes (dashed lines) are aligned with the x-axis. It should be noted how the principal axes differ for the six different signatures of the same person.

A more robust approach is provided by shape matching algorithms in the Radon and Hough domains [29], [30]. Since a rotation of an image corresponds to a linear shift of the Radon and Hough transforms (see, for example, [31]), it is straightforward to calculate the optimal match. It should be noted that the Radon and Hough transforms can also be used to detect straight lines in an image. The estimated equations of the straight lines enable one to detect italic (*slanted*) handwritten characters or to determine the general orientation of a document (*document skew*); see, for example, [32], [33].

In order to compare a static image skeleton with a dynamic exemplar, the dynamic exemplar is translated until their centroids are aligned. The dynamic exemplar is then converted into a static image. Since the Radon transform is sensitive to the line width of the images, we thicken [22] the static image skeleton as well as the image derived from the dynamic exemplar to a line width of approximately five pixels. The two images are then matched, using a method

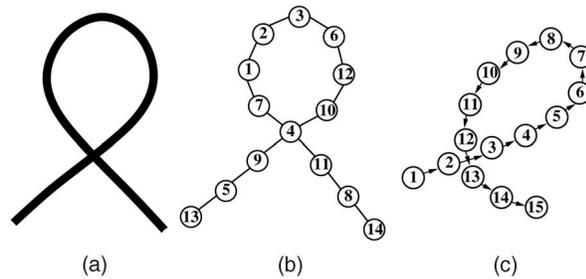


Fig. 3. (a) A straightforward singlepath static signature, with (b) its unordered skeleton samples and (c) a dynamic signature that can be used to extract the pen trajectory of (b).

very similar to the one described in [29], which is based on Euclidean distance measures in the Radon domain. The relative angle of the optimal match is calculated, and the dynamic exemplar is rotated through this angle to align it with the static image skeleton. This procedure is illustrated in Fig. 2b, where all the signatures are aligned with the orientation of the first signature.

3 ESTIMATING THE PEN TRAJECTORY

The technique we develop for extracting the pen trajectory from a static, normalized image is based on an HMM. An HMM is a probabilistic model describing a dynamic process that evolves from one state to the next. In our application, the sequence of states describes the sequence of pen positions as the image is produced. An HMM is constructed from the static image skeleton. Using the HMM, the dynamic exemplar is matched to the static image. The matching algorithm results in the most likely pen trajectory of the static skeleton, given the model. In addition to the pen trajectory, one also obtains a quantitative correspondence between the static image and dynamic exemplar.

We explain the main ideas by means of the simple example shown in Fig. 3a. The image is skeletonized, as described earlier. Since the order of skeleton samples is unknown, a typical numbering is shown in Fig. 3b. Fig. 3c shows a dynamic exemplar that must be matched to the static image. Note the shape differences between the two. Possible pen trajectories must be estimated from Fig. 3b and compared with the known sequence of Fig. 3c. Since we do not know the optimal sequence in Fig. 3b, or even the starting point, for that matter, a very large number of possible sequences need to be compared—far too many for an exhaustive search. The use of an HMM, however, makes the calculation of the optimal pen trajectory computationally feasible.

To estimate the pen trajectory of the static image, two basic issues are addressed. First, a probabilistic model of the static signature is created. More specifically, an HMM is created which describes the geometric shape of the signature and restricts the choice of possible pen movements. Second, the optimal pen trajectory is calculated by matching the known dynamic exemplar to the HMM.

3.1 The HMM

An HMM has N emitting states $\{q_1, q_2, \dots, q_N\}$ that have observation PDFs associated with them. The two states q_0 and q_{N+1} , without associated PDFs, are called *nonemitting* states. These two additional nonemitting states serve as

initial and terminating states, respectively, thus eliminating the need for separate initial and terminating probabilities (see [34] for more detail).

All state observation PDFs in the context of this paper are spherical Gaussians, described by

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{D}{2}}\sigma} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{x} - \boldsymbol{\mu})^T(\mathbf{x} - \boldsymbol{\mu})\right), \quad (1)$$

where \mathbf{x} is a D -dimensional vector that must be matched to the PDF and $\boldsymbol{\mu}$ is the D -dimensional mean of the Gaussian. The standard deviation σ is preset. For brevity, the PDF associated with state i having mean $\boldsymbol{\mu}_i$ and standard deviation σ will be referred to as $\mathcal{N}(\boldsymbol{\mu}_i, \sigma)$. Geometric shape information of the static image is embedded in the PDF parameters $\boldsymbol{\mu}_i$ and σ , as described in Section 3.2.

States are connected by transition links that dictate the possible pen movements. All transitions between states are weighted with transition probabilities. The *order* of the HMM specifies the number of previous states the HMM considers when transitioning to a next state. Sections 3.2-3.6 describe how the order of our HMM is increased to take context into account. In order to match a static image and a dynamic exemplar, the dynamic exemplar is presented as a sequence of quantifiable characteristics called *feature vectors*. The sequence is given by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T]$, where \mathbf{x}_t denotes a D -dimensional feature vector at discrete-time instant t and T is the number of feature vectors (number of samples in the dynamic exemplar). Using the Viterbi algorithm, \mathbf{X} is matched to our HMM to produce a hidden state sequence $\mathbf{s} = [s_1, s_2, \dots, s_T]$, which estimates the desired sequence of skeleton samples, as described in Section 3.7.

3.2 A First-Order HMM

The shorthand notation for an HMM λ is

$$\lambda = \{\mathbf{A}, \{\mathcal{N}(\boldsymbol{\mu}_i, \sigma), i = 1, \dots, N\}\}, \quad (2)$$

where \mathbf{A} is a matrix representing the transition links and $\mathcal{N}(\boldsymbol{\mu}_i, \sigma)$, as described by (1), is the observation PDF of state i for $i \in \{1, \dots, N\}$.

We begin by constructing a first-order HMM from the skeleton of the static image. The skeleton consists of M unordered samples $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M\}$, where \mathbf{p}_x is the 2D coordinate of sample x . Each emitting state i is associated with a skeleton sample via a mapping $r(i)$ and the sample coordinates are embedded in the observation PDF by setting $\boldsymbol{\mu}_i = \mathbf{p}_{r(i)}$. For a first-order HMM, we have $N = M$ and $r(i) = i$. Our first-order HMM matches only 2D feature vectors, in this case, the pen positions of the dynamic exemplar. We choose $\sigma = 0.7$ pixels in (2) for all states, in order to relate the match between the position coordinates of the dynamic exemplar and static image to Euclidean distance.

The HMM topology is crucial to our algorithm, as it constrains the range of possible pen movements that could generate the static image. For our first-order HMM, the probability of reaching the next state depends only on the current state, so that the transition probability matrix $\mathbf{A} = [a_{ij}]$, where $a_{ij} = P(s_{t+1} = q_j | s_t = q_i)$ is the probability of a transition from q_i to q_j at instance $t+1$, with $i, j \in \{0, 1, \dots, N+1\}$ and $t \in \{1, 2, \dots, T-1\}$. HMM states are called *neighbors* if their associated skeleton samples are adjacent. All emitting states are linked to their neighbors, to allow the pen to move to an adjacent skeleton point on a

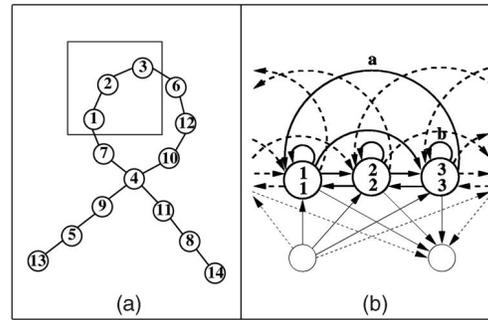


Fig. 4. (a) Isolated unordered skeleton samples in a simplified signature and (b) their corresponding first-order HMM.

transition. However, this only takes local information into account, and not context. Context is incorporated by using second-order HMMs, which allow us to include a directional feature, as described in Section 3.3.

Since we have no prior knowledge of where the pen trajectory of the static image may start or end, the nonemitting initial state can enter any emitting state. Also, each emitting state is directly connected to the nonemitting terminating state.

One also needs elasticity in the model, to allow the static image and dynamic exemplar to have different numbers of samples. This is accomplished by including skip-links and self-loops in the HMM. A *skip-link* is a transition between two states separated by a neighbor common to both. A *self-loop* connects a state back to itself. Self-loops are added to the emitting states. In this paper, we use skip-links to skip states with only two neighbors. Equal transition probabilities are assigned to all transition links leaving a state, normalized to sum to one.

These ideas are illustrated in Fig. 4. The first-order HMM for the isolated fragment in Fig. 4a is shown in Fig. 4b, where the three states indicated by the larger circles are emitting states. Each state is labeled with a pair of numbers: The top number is the state index i and the bottom number is the index $r(i)$ of its associated skeleton sample. The dashed lines indicate transition links to and from states outside the rectangular box in Fig. 4a. The smaller blank circles indicate the nonemitting initial and terminating states. All states are connected to these nonemitting states so that the pen trajectory can start and end at any skeleton sample. Skip-link 31 and self-loop 33 are also indicated. State 1 and its neighbors have two neighbors, two skip-links, one self-loop, and one to the nonemitting terminating state. The associated six transition probabilities are therefore all specified as $\frac{1}{6}$.

We emphasize an important feature of Fig. 4b. Any two neighboring emitting states and any two emitting states connected by a skip-link are connected both ways: If one enters the HMM at an emitting state with more than one neighbor it is not possible to determine locally in which direction one should move next and, therefore, both directions are allowed. Since all transition links are assigned the same probability, all skeleton samples are potential turning points. It is therefore entirely possible and it indeed happens, in practice, that the extracted pen trajectory may incorrectly reverse direction. One way to address this problem is to include more context. Bengio and Frasconi [35], supported by the experiments of Abou-Moustafa et al. [36], investigated the

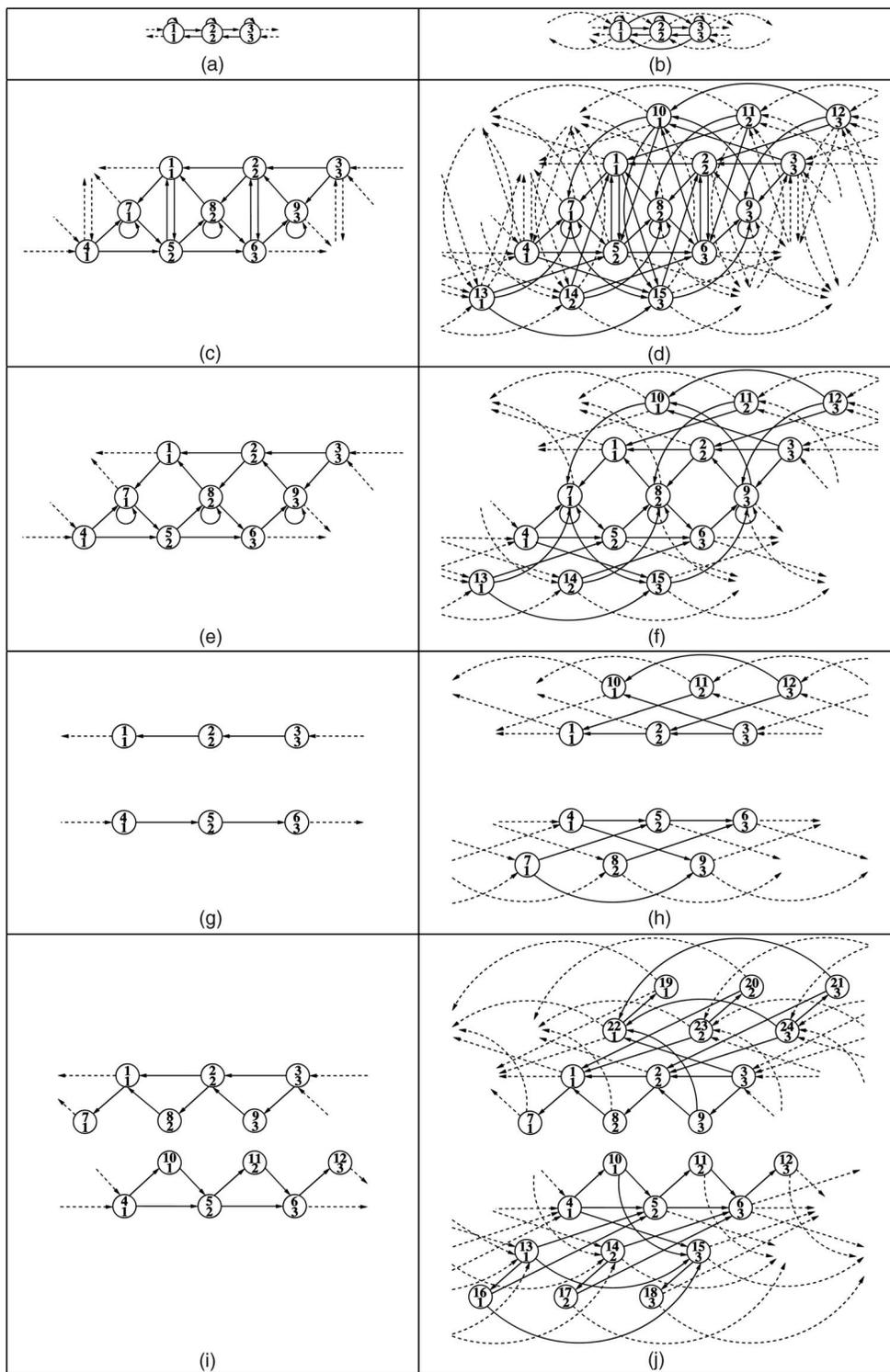


Fig. 6. The simplified HMM topology for a line segment in the left-hand column with a detailed version in the right-hand column when skip-links are added to our first-order HMM. (a) and (b) First-order HMMs. (c) and (d) Second-order HMMs. (e) and (f) Assigning the cost function. (g) and (h) Removal of self-loop states. (i) and (j) Inclusion of duration states.

(i.e., $N > M$), which represents different contexts in which the sample can be found. It is also worth noting the significant increase in model complexity in the right-hand column, due to the skip-links added in Fig. 6b.

The next step is to enforce pen movement in one direction. Let θ_{hi} be the angle between the two straight lines connecting points h to i and i to j , respectively. Then,

$$\cos(\theta_{hi}) = \frac{(\mathbf{p}_j - \mathbf{p}_i) \cdot (\mathbf{p}_i - \mathbf{p}_h)}{\|\mathbf{p}_j - \mathbf{p}_i\| \|\mathbf{p}_i - \mathbf{p}_h\|}, \quad (3)$$

where $\|\cdot\|$ is the Euclidean distance norm and $\mathbf{p}_h, \mathbf{p}_i$, and \mathbf{p}_j are the 2D coordinates of points h, i , and j , respectively. Let the predecessor q_h of q_i be any state for which $a_{hi} > 0$. Our second-order HMM guarantees that all predecessors of an emitting

state share the same skeleton sample, by virtue of its topology. For example, if there are two emitting states q_g and q_h that enter q_i so that $a_{hi} > 0$ and $a_{gi} > 0$, our second-order HMM topology guarantees that $\mathbf{p}_{r(g)} = \mathbf{p}_{r(h)}$. In order to encourage the system to follow the same direction along a line segment, the probability of a transition from state i , with predecessor state h , to emitting state j is chosen as the cost function

$$a_{ij} = \begin{cases} \cos(\theta_{r(h)r(i)r(j)}), & \text{for } |\theta_{r(h)r(i)r(j)}| \leq 90^\circ \\ 0, & \text{for } |\theta_{r(h)r(i)r(j)}| > 90^\circ \end{cases} \quad (4a)$$

where $\cos(\theta_{r(h)r(i)r(j)})$ is defined by (3). This does not apply to links entering or leaving self-loop states, where *self-loop states* are states with self-loops, e.g., State 7 in Fig. 6d. Figs. 6e and 6f show the HMMs in Figs. 6c and 6d after links with zero probability have been removed, based on the cost function (4).

Since the self-loop states are excluded from the cost function, it is still possible to turn around via them. An example from Fig. 6e is the state sequence (top numbers) [4, 5, 8, 1], which corresponds to the sample sequence (bottom numbers) [1, 2, 2, 1]. In order to prevent this, self-loop states and all their connections (both entering and leaving them) are removed. Figs. 6g and 6h show the HMMs in Figs. 6e and 6f after this step.

Introducing skip-links for more elasticity leads to the configuration of Fig. 6h. We use the term *skip-link states* when referring to states in the second-order HMM that result from skip-links in the corresponding first-order HMM, e.g., State 10 in Fig. 6d (top number), which results from the skip-link leaving State 3 and entering State 1 in Fig. 6b. Skip-link states can compensate for situations in which the static image has more samples than the dynamic exemplar. Self-loop states, on the other hand, can compensate for situations in which the dynamic exemplar has more samples than the static image. Since all self-loop states have been removed all emitting states are duplicated and each emitting state is allowed to enter its duplicated state. We refer to the duplicated states as *duration states*. These duration states have the same destinations as the states they duplicate as illustrated in Figs. 6i and 6j. States 7-12, for example, are the duplicated states of States 1-6 in Figs. 6g and 6h and, therefore, share the same skeleton samples (bottom numbers) and destinations of states they duplicate. Finally, it should be noted how the two directions that a pen can follow on a line segment are completely disjoint within the HMM, so that it is not possible to change direction in the middle of a line segment.

In this section, we discussed the topology of states associated with segment points. Specific transition weights will be provided at the end of Section 3.6. Next, we discuss the topology at states where the pen is allowed to change direction abruptly.

3.5 HMM Topology for Crosspoints and Endpoints

To enable the pen to immediately recross a line or suddenly change direction, we allow it to turn around or change direction abruptly at states associated with endpoints and crosspoints. We define *endpoints* as skeleton samples having one neighbor. *Crosspoints* are skeleton samples having more than two neighbors. The main difference between states associated with segment points and states associated with crosspoints and endpoints is that the direction constraint of (4) is not enforced for crosspoint and endpoint states. Instead, traversal to any immediate or skip-link state

neighbor is allowed from a state associated with a crosspoint or endpoint. This ensures that it is possible to change direction abruptly, or even to turn around, at these states.

Some situations, however, involve simple crossings, where it is easy to follow line directions that enter and leave the crosspoint. It is straightforward to unravel such crossings. If the line thickness is uniform and line directions are smooth near such a simple crosspoint and no more than two lines cross each other at a single point, it is unlikely that the pen has passed through that region multiple times. Let us, for the moment, assume that we identify such a simple crossing and label its associated sample as i . A simplified first-order model (excluding nonemitting states, skip-links, and self-loops) is shown in Fig. 7a, where $i = 4$. It should be noted that first-order HMMs are not able to model past context so that the transition probabilities for state i have to allow access to any of the four neighboring states.

The situation is different with second-order HMMs. Its first-order equivalent is shown in Fig. 7b. As the transition probabilities also depend on the previous state, which has a unique associated sample point, it is straightforward to follow lines through the crosspoint. At a simple crossing, one can then detach the two lines that cross, by setting the appropriate transition probabilities to zero, as shown in Fig. 7c. To do this, however, we need to be able to identify such simple crossings.

With the crosspoint labeled as i , we label the four neighboring coordinates clockwise, in order, as a , b , c , and d . The idea is to identify whether the sequenced samples $\{a, i, c\}$ and $\{b, i, d\}$ are intersecting lines. In Fig. 7b, for instance, the coordinates (bottom numbers) are labeled as $a = 7$, $b = 10$, $c = 11$, and $d = 9$. We consider only crossings where a , b , c , and d are all segment points, having only one other skeleton neighbor besides i . Let x be the other skeleton neighbor of a and y be the other skeleton neighbor of c . We now calculate three angles, θ_{xai} , θ_{aic} , and θ_{icy} , using (3). If $|\theta_{aic}| \leq 10^\circ$, $|\theta_{xai}| \leq 30^\circ$, and $|\theta_{icy}| \leq 30^\circ$, $\{a, i, c\}$ is considered a straight line. If, likewise $\{b, i, d\}$ also proves to be a straight line, the crossing is considered a simple crossing. The second-order HMM provides the necessary context to extract directions and decouple the two lines $\{a, i, c\}$ and $\{b, i, d\}$ so that the two intersecting lines can both be traversed in one direction or the other. Direction is now maintained through the crossing, and the inclusion of duration states provides the necessary flexibility, as shown in Fig. 7d.

A more detailed example is shown in Figs. 7e and 7f, indicating possible pen trajectories that can be extracted from Fig. 3b if first-order and second-order HMMs are used and detaching the simple crossing at State 4 in Fig. 7f. For the sake of clarity, Figs. 7e and 7f omit the fact that a pen trajectory can start and end at any sample (numbered circle). Self-loop symbols are used to indicate duration states in Fig. 7f. It is interesting to derive the number of first-order and second-order HMM states and nonzero transition probabilities for the signature in Fig. 3b, that allow the choices of possible pen motions shown in Figs. 7e and 7f:

1. The signature in Fig. 3b has 16 states in its first-order HMM and 107 states in its final second-order HMM (including nonemitting states).
2. The signature in Fig. 3b has $\frac{92}{256}$ (36 percent) nonzero transition probabilities in its first-order HMM. This is reduced to $\frac{428}{11449}$ (3.7 percent) in its final HMM. It should be noted that transition links leaving the

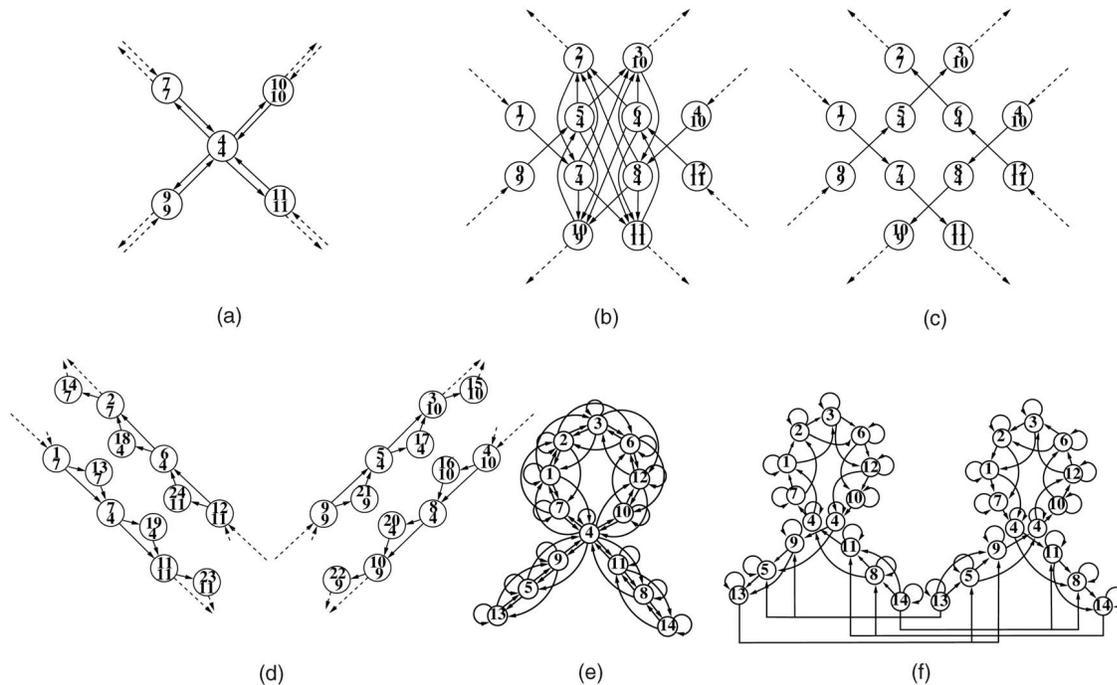


Fig. 7. (a) Simplified first-order HMM for a crosspoint (excluding nonemitting states, skip-links, and self-loops) with (b) the first-order equivalent of its second-order counterpart, (c) lines decoupled at the crossing by removing links from (b), and (d) duration states included in (c). (e) Possible pen trajectories for Fig. 3b using our first-order HMM, and (f) using the second-order HMM with detachment of the intersecting lines at the crosspoint. It should be noted that graphs (e) and (f) are not intended as HMMs, but as representations of allowed pen trajectories with skeleton samples as nodes and possible pen motions as arrows.

nonemitting initial state and entering the nonemitting terminating state are included in this computation.

We can conclude from the above statistics that the final HMM is notably sparser and has more states than its first-order counterpart, thereby improving the ability of our pen trajectory estimation algorithm to model long-term dependencies, as described in Section 3.2.

3.6 HMM PDFs

In the first-order HMM, each position coordinate is associated with a single state. For simplicity, the PDF associated with each state has so far only reflected information about the position variations of the static image. When unraveling a static image, the direction of pen motion at each coordinate is also important. Knowledge of pen direction allows us to match not only position coordinates, but also local directions in a dynamic exemplar, thus providing additional context.

The PDF of state i that reflects the pen position is given by $\mathcal{N}(\boldsymbol{\mu}_i^P, \sigma_P)$, where we defined $\boldsymbol{\mu}_i^P = \mathbf{p}_{r(i)}$. However, in our second-order HMM, each state i has a single predecessor skeleton sample h , enabling the use of a direction feature. A second Gaussian PDF is associated with each emitting state, which takes the form of (1) with

$$\boldsymbol{\mu}_i^V = \frac{\boldsymbol{\mu}_i^P - \mathbf{p}_h}{\|\boldsymbol{\mu}_i^P - \mathbf{p}_h\|}, \quad (5)$$

where $\boldsymbol{\mu}_i^V = (0, 0)$ if state i is preceded by the nonemitting initial state. The direction PDF is abbreviated as $\mathcal{N}(\boldsymbol{\mu}_i^V, \sigma_V)$. The two PDFs associated with state i are assumed to be statistically independent [39]. They reflect the typical correspondences between the coordinates (position and direction of pen motion) of the static image and dynamic exemplar. It should be noted that the directional feature described by (5) is

frequently used in first-order HMMs of online character recognition and signature verification applications, where each pen position has a unique previous position. In our application, each coordinate in the skeleton of a static image has one or more neighbors and we have no prior knowledge to choose appropriately. Second-order HMMs can model longer dependencies, effectively enforcing a single previous coordinate for each state. Thus, we are able to include an unambiguous directional feature in each state PDF.

All parameters of the HMM are designed specifically for our application and are independent of the test set. We now list the relevant empirically determined values used in our system (all transition probabilities leaving a state are normalized to sum to one, of course):

- $a_{ij}^E = \frac{1}{N_D}$: Probability of a transition from endpoint state i to state j , where N_D is the number of links leaving state i .
- $a_{ij}^I = \frac{1}{N_D}$: Probability of a transition from nonemitting initial state i to state j , where N_D is the number of links leaving state i .
- $a_{ij}^C = \frac{1}{N_D}$: Probability of a transition from crosspoint state i to state j , where N_D is the number of links leaving state i .
- $a_{ij}^{SS} = 0.05$: Probability of a transition from segment point state i to its duration state j .
- $a_{ij}^{ST} = 0.05$: Probability of a transition from segment point state i to the terminating nonemitting state j .
- $a_{ij}^S = \cos(\theta_{r(h)r(i)r(j)})$, $|\theta_{r(h)r(i)r(j)}| \leq 90^\circ$: Probability of any other transition from state i , associated with segment point $r(i)$ (see (4)).

- $\sigma_P = 17$: Standard deviation (in pixels) quantifying similarities between pen positions. This constrains the distance between points in the static image and dynamic exemplar.
- $\sigma_{VI} = 2$: Standard deviation that quantifies similarities between local line directions if a state is preceded by the nonemitting initial state.
- $\sigma_V = 0.2$: A tight standard deviation that quantifies similarities between local line directions if a state is preceded by an emitting state.

3.7 The Hidden State Sequence (Estimated Pen Trajectory)

We have now developed the full HMM from the static image. It consists of states associated with the position coordinates of the static image skeleton, and transition probabilities determining a transition from one state to the next. Each state has two statistically independent PDFs associated with it, describing the position and direction variations. The transition probabilities govern the possible pen motions, based on two basic assumptions:

1. The pen is not allowed to turn around suddenly within a line segment.
2. The pen is allowed to turn around at endpoints and crosspoints.

Most important, when the dynamic exemplar is matched to the HMM, one can determine the most likely state sequence. Since the states are associated with the position coordinates of the skeleton, this sequence yields the maximum likelihood pen trajectory as determined by the model. The dynamic exemplar $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T]$ is matched to the HMM λ of the static image using the Viterbi algorithm [40], [20]. This results in an optimum state sequence $\mathbf{s} = [s_1, \dots, s_T]$ as well as a likelihood.

The first two components of each dynamic feature vector \mathbf{x}_t form a subvector $\mathbf{x}_t^{1,2}$ describing the dynamic pen position. The last two components $\mathbf{x}_t^{3,4} = (\mathbf{x}_t^{1,2} - \mathbf{x}_{t-1}^{1,2}) / \|\mathbf{x}_t^{1,2} - \mathbf{x}_{t-1}^{1,2}\|$ are two direction components (normalized velocity), with $\mathbf{x}_1^{3,4} = (0, 0)$. Let $f_{s_t}^P(\mathbf{x}_t^{1,2})$ be the position PDF $\mathcal{N}(\boldsymbol{\mu}_{s_t}^P, \sigma_P)$ evaluated at $\mathbf{x}_t^{1,2}$. Likewise, let $f_{s_t}^V(\mathbf{x}_t^{3,4})$ be the direction PDF $\mathcal{N}(\boldsymbol{\mu}_{s_t}^V, \sigma_V)$ evaluated at $\mathbf{x}_t^{3,4}$. Since these PDFs are independent, the joint observation PDF of state s_t evaluated at feature vector \mathbf{x}_t is given by $f_{s_t}(\mathbf{x}_t) = f_{s_t}^P(\mathbf{x}_t^{1,2}) f_{s_t}^V(\mathbf{x}_t^{3,4})$, where $s_t \in \{1, \dots, N\}$.

The globally optimized likelihood of \mathbf{s} , based on the HMM and dynamic data \mathbf{X} , is then given by

$$\delta = a_{s_0 s_1}^I \prod_{t=1}^T a_{s_t s_{t+1}} f_{s_t}(\mathbf{x}_t), \quad (6)$$

where $s_0 = 0$ is the nonemitting initial state and $s_{T+1} = N + 1$ is the nonemitting terminating state. We now have a maximum likelihood state sequence for each available dynamic exemplar of a static image. This provides a pointwise correspondence between the static image and dynamic exemplar.

The likelihood δ is a useful similarity measure between a static image and a dynamic exemplar. It tends to decrease if a segment exists in the dynamic exemplar and not in the static image or in case of inconsistencies in size or orientation. However, it can happen that a dynamic exemplar matches only a portion of the static image very well. A dynamic

character “1,” for example, can produce a high likelihood on a static “7.” To prevent this, we weight the likelihood (6) in the following manner: First, the total path length T_L is computed, as the sum of distances between all the connected skeleton samples of the static image. Second, the path length R_L of the recovered pen trajectory is computed so that $R_L \leq T_L$. We now weight each of the maximum likelihood state sequences (one for each dynamic exemplar) as follows:

$$\delta_W = \frac{R_L}{T_L} \delta. \quad (7)$$

Finally, the dynamic exemplar’s state sequence that produces the maximum weighted likelihood δ_W is chosen as the estimated pen trajectory. It should be noted that the weighted likelihood in (7) can be used to identify forgeries in a signature verification application.

4 EXPERIMENTS

We showed in the previous section how to recover the pen trajectory of a static image, by comparing its HMM to a different dynamic exemplar. In this section, we demonstrate how the accuracy of this recovered trajectory is measured, by comparing it to its ground truth. The evaluation technique that we use is presented in Section 4.1. Experimental results are given and discussed in Section 4.2.

4.1 Evaluation Protocol

In order to evaluate the accuracy of our method, a ground truth is needed. We therefore developed a signature database, named US_SIGBASE, which contains the exact dynamic sequences of static signature images. Signatures were recorded from 55 persons (hereafter referred to as *signatories*) on paper placed on a WACOM digitizing tablet. The paper signatures were then scanned as gray-scale images at 600 dpi. Thus, a static image, with line thickness varying between five and 10 pixels, and its exact dynamic counterpart were obtained simultaneously. A second set was obtained from the Dolfing database [41], which consists of dynamic signatures of 50 signatories. These signatures were converted to their static counterparts, with line thickness varying between one and three pixels. Finally, we selected only singlepath signatures, as the described system is not capable of combining disconnected trajectories (see the discussion in the final section, regarding extensions to multipath signatures).

Due to the noise introduced while recording a dynamic signature and while scanning, binarizing, and skeletonizing its static counterpart, the image skeleton may differ from its exact dynamic counterpart. Furthermore, there is clearly not a one-to-one correspondence between the pen positions of the dynamic signature and the position coordinates of the skeletonized static image. The ground-truth pen trajectory is obtained by matching a slightly modified HMM of the static image with its exact dynamic counterpart. The slight modification tightens the standard deviation σ_P to 7 (measured in pixels), since the position coordinates of the static image match that of its exact dynamic counterpart much better than in situations where only a dynamic exemplar is available.

All that remains is to compare the two state sequences—the ground truth, as described above and the estimated sequence obtained from the dynamic exemplar. Both sequences are extracted from the same static image and it is

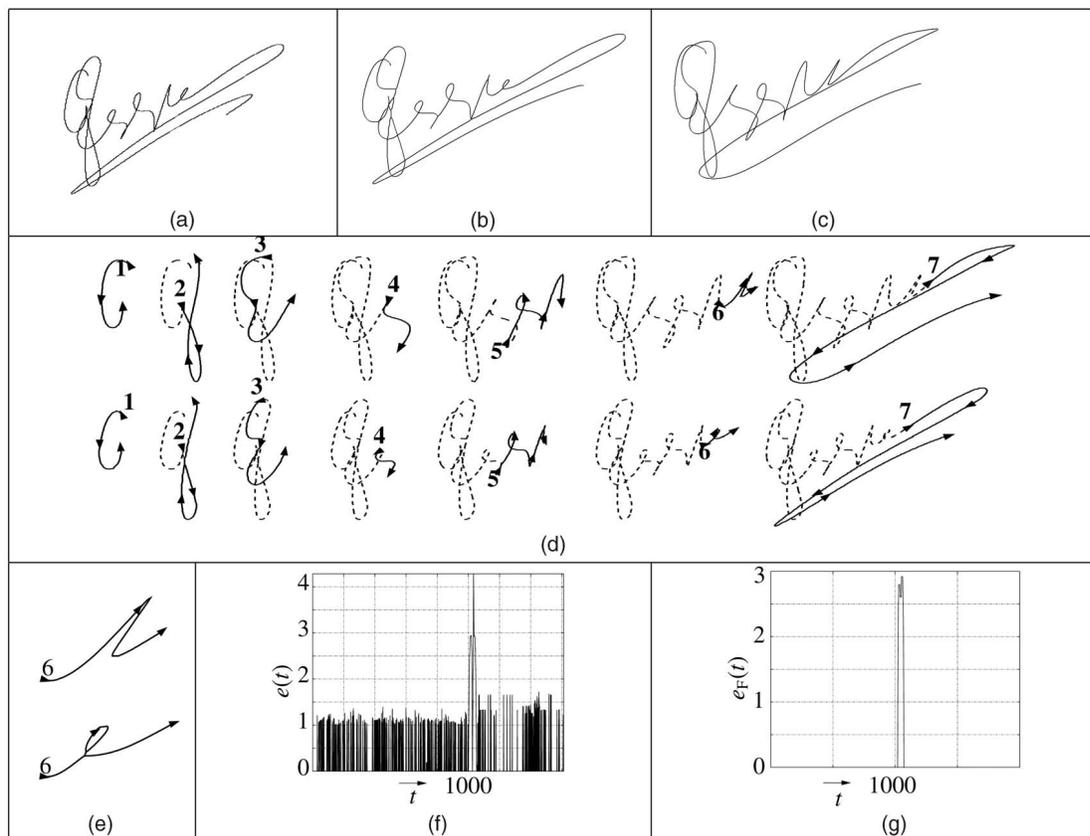


Fig. 8. Evaluating a recovered pen trajectory. (a) A static signature with (b) its skeleton and (c) the dynamic signature best corresponding to it. (d) Animation of the dynamic exemplar (top) and estimated (bottom) pen trajectories. (e) An error to quantify. (f) The errors $e(t)$ and (g) their filtered versions $e_F(t)$.

therefore possible to compare them. However, since the two sequences are obtained from different dynamic sequences, they do not necessarily have the same number of samples. A point-wise comparison is therefore not possible. Thus, we align the two sequences before comparison using a dynamic programming (DP) algorithm [40] to minimize the Euclidean distance between them. This allows for a direct comparison, from which errors can be identified.

Figs. 8a, 8b 8c, and 8d illustrate the pen trajectory estimation process. Fig. 8a shows the static image with its skeleton in Fig. 8b and a different dynamic exemplar in Fig. 8c. Fig. 8d shows an animation of the match between the static image (bottom) and a dynamic exemplar (top). The numbers indicate where curves start and the arrows indicate the positions of matched coordinates. Dashed lines are previous curves. The only significant error in the estimated pen trajectory occurs in the curve starting at 6 in Fig. 8d, which is traversed the wrong way around. The Euclidean error distance $e(t)$ as a function of time t , for the matched coordinates, is shown in Fig. 8f. The error, magnified in Fig. 8e, results in a prominent pulse, originating at approximately $t = 1,000$ in Fig. 8f. The smaller errors are due to the elasticity built into the model through skip-link and self-loop states, which accommodate different curve lengths. In order to isolate a true error, as indicated by a pulse, a morphological open-close filter [22] with width $w = 10$ is applied to $e(t)$, resulting in $e_F(t)$, as shown in Fig. 8g. Erroneous regions are now identified where $e_F(t) > 0$.

An error measure that is invariant to parameterization is given by the path length of the ground-truth trajectory in

erroneous regions expressed as a percentage of the total ground-truth path length. Accordingly, the accuracy of the pen trajectory shown in Fig. 8 is approximately 98 percent in terms of this measure.

4.2 Experimental Results

Our signature database, as described in the previous section, consists of a total of 710 singlepath signatures from 50 signatories. The static images of the first 35 signatories were obtained from the US_SIGBASE database, and the remaining 15 signatories were obtained from the Dolfing database. For each signatory, a static image was randomly selected. It should be noted that the dynamic counterpart that establishes a ground truth is available for each static image. The rest of the dynamic signatures are used as dynamic exemplars for estimating the pen trajectories of the static images. The pen trajectory of a static image is recovered, as described in Section 3.7. The estimated pen trajectory is then compared with the ground truth to obtain an accuracy score from $e_F(t)$, as described in Section 4.1. The *average accuracy* of a database is then calculated by averaging the accuracy scores over all the static images.

The experiments were run on an AMDXP1900+ at 1.6 GHz. The signatures are constrained to a bounding box of $50 \text{ mm} \times 20 \text{ mm}$, which is somewhat larger than what would normally be allowed on documents such as bank checks. It takes approximately 16 seconds to construct an HMM for a skeleton image consisting of 1,600 coordinates, and approximately 22 seconds to estimate its pen trajectory based on a

TABLE 1
Experimental Results, Showing the Average Accuracy
of Recovered Pen Trajectories

	Dolfing	US.SIGBASE	Combined
Number of static images	15	35	50
Number of dynamic exemplars	210	450	660
Accuracy for thinning in [24]	87.2%	90.3%	89.3%
Accuracy for our skeletonization	92.8%	91.0%	91.5%

different dynamic exemplar. We are using generic software with no optimization for this particular situation. It is expected that this computation time can be substantially reduced with code optimized for this application. Other optimization suggestions are made in the final section.

Two sets of experiments are conducted. The first set uses a standard thinning algorithm [24] without any removal of artifacts and yields an accuracy of 89.3 percent. The second set uses the more sophisticated skeletonization procedure described in Section 2.1 and yields an average accuracy of 91.5 percent. These results are summarized in Table 1. The simple thinning approach achieved a relatively high accuracy, which indicates that our approach is not highly sensitive to skeletonization artifacts. However, the approximately 20 percent error reduction between experiments 1 and 2 indicates that artifact removal and direction preservation does improve the procedure.

Fig. 9 shows examples of typical errors that were encountered. The dynamic exemplars and the skeletonized static images are shown in Figs. 9a and 9b, respectively. The errors as functions of t are shown in Fig. 9c, and the accuracy, as a percentage of the total ground-truth path length, is shown in Fig. 9d. The exact locations of the errors in the signatures are indicated by thickened lines in Fig. 9b, with the corresponding locations in the dynamic exemplars of Fig. 9a indicated by matching numbered arrows.

As expected, the main cause of errors is inconsistencies between a static image and a dynamic exemplar. More specifically, the system is prone to errors in regions where a line segment is present in either the dynamic exemplar or static image, but absent in the other. A line segment in the static image that is absent in the dynamic exemplar will not be extracted from the static image. The reason is that the dynamic exemplar does not provide the necessary information (see the arrows numbered 1 in Fig. 9, for signatures 1-3). In Fig. 9, arrows 2, 3, and 4 for signature 5 show errors caused by inconsistent pen movements on corresponding segments of the dynamic exemplar and the static image. This error type is illustrated in the two attachments. Attachment 1 (available at <http://computer.org/tpami/archives.htm>) is an animation of the extraction of the ground-truth pen trajectory of signature 5 in Fig. 9. The exact dynamic counterpart of the static image (top signature, rendered in blue) is used to estimate its ground truth (bottom signature, rendered in black). The animated red circles show corresponding dynamic pen positions. Attachment 2 (available at <http://computer.org/tpami/archives.htm>) animates the estimation process with a different dynamic exemplar, shown in Fig. 9a. The green segments in the bottom signatures (also shown by the thick solid lines in Fig. 9b) are erroneous, due to

compensation for dissimilarities. Note the inconsistent pen movements between the ground truth (bottom signature in Attachment 1) and dynamic exemplar of Attachment 2 (top), resulting in the relatively low accuracy score of signature 5. Extreme size differences between corresponding segments can also cause errors, e.g., segment 1 of signature 6. The erroneous segment originating at s is indicated by arrows. It should be noted, however, how easily the slight shape dissimilarities between signatures 7a and 7b can be accommodated. The estimated pen trajectory is shown with arrows in Fig. 9b, resulting in 100 percent accuracy.

Some images may have an excessive line thickness relative to the size of the signature. In this case, information loss due to multiple crossings in small areas becomes severe, making it difficult or impossible to unravel the image. Figs. 10a, 10b, and 10c show an example of such a signature (probably the most complex signature in our database). The original image, its skeleton, and the dynamic exemplar corresponding best to it are shown in Figs. 10a, 10b, and 10c, respectively. Not only is the shape of the image in Fig. 10a corrupted in the middle region, but the dynamic exemplars have inconsistent pen movements in corresponding regions. The accuracy of the pen trajectory, as calculated by our evaluation technique, is 68 percent. Thus, despite the obvious difficulties, a total path length of almost 70 percent is derived from its dynamic ground truth.

Finally, Figs. 11a, 11b, and 11c illustrate the effect on our system when the pen trajectory of a static image is estimated using an incorrect dynamic exemplar (i.e., one created by a different signatory). Fig. 11a shows the dynamic exemplar used to estimate the pen trajectory of the static signature in Fig. 8a. As expected, the pen trajectory is estimated incorrectly, and only a small part of the signature is extracted from its image, as shown in Fig. 11b. The error function $e_F(t)$ produced by our evaluation technique is erratic, as shown in Fig. 11c, and produces an accuracy of 3 percent.

5 CONCLUSION

Our system for estimating the pen trajectories of static images depends on the availability of dynamic exemplars. After skeletonization, an HMM is created from the static image. The necessary context to resolve ambiguities is provided by a second-order HMM. The Viterbi algorithm matches a dynamic exemplar to the HMM and determines the most likely state sequence, which can be translated into the most likely pen trajectory. Furthermore, the Viterbi likelihood provides a measure of similarity between a static image and a dynamic exemplar, which may form a useful basis for a static signature verification system.

Our experiments compare the ground truths with the estimated pen trajectories of static images. It is shown that more than 90 percent of the ground-truth path lengths are recovered (averaged over all the images). The experiments also show that the system is rather robust with respect to the type of skeletonization used. Moreover, we find that complex static images, even ones that are hard to unravel with the eye, do not pose serious problems. Of course, some images can be corrupted during skeletonization to such an extent that information loss becomes severe, making it very difficult to unravel the image. The main source of errors is inconsistencies between a dynamic exemplar and a static image, which is inevitable for handwritten documents. Although our system

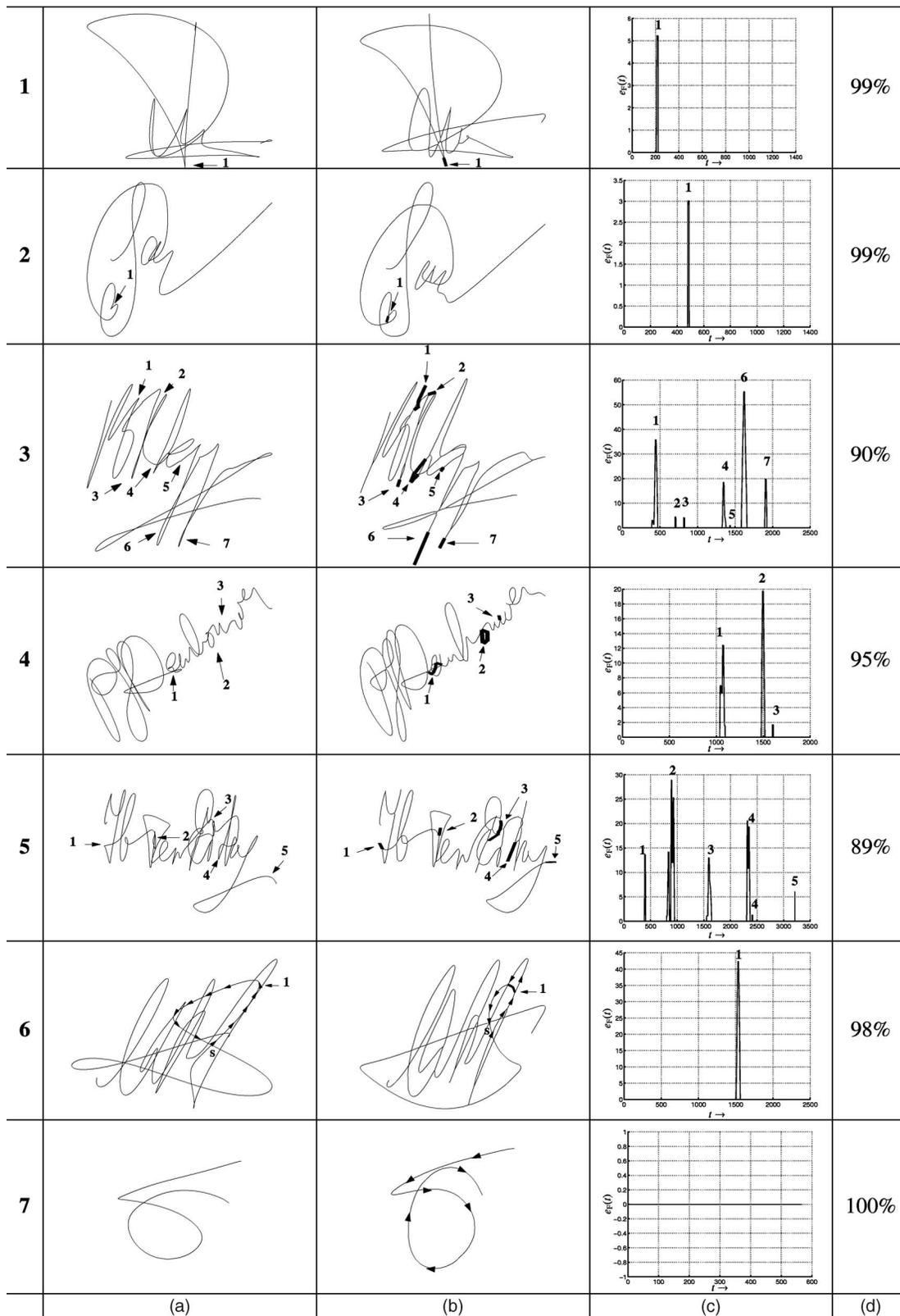


Fig. 9. (a) Dynamic exemplars that are used to unravel (b) skeletonized static images, with (c) resulting errors, and (d) the accuracy scores of the extracted pen trajectories.

is not immune to dissimilarities and ambiguities, it takes global context into account, making it more robust than algorithms that rely heavily on local correspondences. It is important to note, however, that in those cases where errors

do occur, one still has access to local correspondences. This can be useful in a signature verification application, as it allows the comparison of only those parts of the signature that were accurately recovered. At present, we are investigating

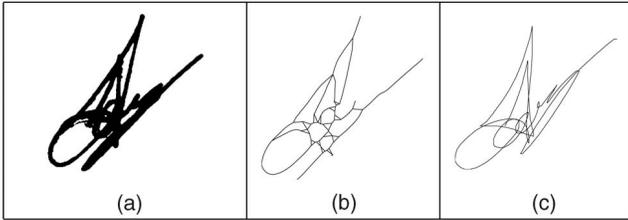


Fig. 10. (a) A complicated static image to unravel with (b) its skeleton and (c) the dynamic exemplar corresponding best to it.

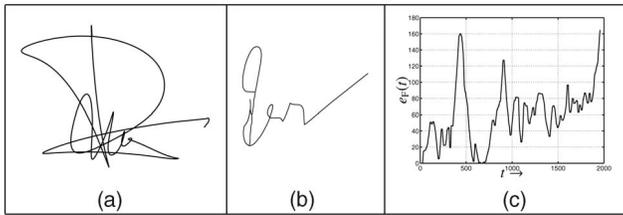


Fig. 11. (a) A dynamic exemplar of a different signatory to calculate the pen trajectory of Fig. 8a, with (b) its estimated pen trajectory, resulting in (c) a poor error function.

HMM training schemes to model inconsistencies between signatures.

At least two more issues still need to be addressed. The first relates to the speed of the system. As it stands, without any optimization, it is slow. A major improvement could presumably be effected by reducing the number of samples that constitute the static images and dynamic exemplars. The Viterbi algorithm is used to estimate the pen trajectory of a static image. The computational cost of the Viterbi algorithm is $O(Tm)$, where m is the number of nonzero transition probabilities at each time step [42] and T is the number of samples in the dynamic exemplar that is matched to the HMM. There are other approximate but faster algorithms that could be used instead (see [42]). The second issue concerns our assumption of a singlepath static image. This imposes a severe restriction on the application of the system. Preliminary experiments with hierarchical HMMs indicate that multipath static images can be accommodated with comparative ease and we hope to address the topic in more detail elsewhere. Unexpected disconnections, i.e., broken curves in the static image, can also be treated as part of the extension to multipath signatures.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers and to Ludwig Schwardt and Barry Sherlock for their numerous constructive comments.

REFERENCES

- [1] R. Plamondon and S.N. Srihari, "Online and Offline Handwriting Recognition: A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63-84, Jan. 2000.
- [2] T. Huang and M. Yasuhara, "A Total Stroke SLALOM Method for Searching for the Optimal Drawing Order of Off-Line Handwriting," *Proc. IEEE Systems, Man, and Cybernetics*, vol. 3, pp. 2789-2794, Oct. 1995.
- [3] Y. Kato and M. Yasuhara, "Recovery of Drawing Order from Single-Stroke Handwriting Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 938-949, Sept. 2000.

- [4] K.K. Lau, P.C. Yuen, and Y.Y. Tang, "Stroke Extraction and Stroke Sequence Estimation on Signatures," *Proc. Int'l Conf. Pattern Recognition*, pp. 119-122, 2002.
- [5] S. Lee and J.C. Pan, "Offline Tracing and Representation of Signatures," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 22, no. 4, pp. 755-771, July 1992.
- [6] G. Boccignone, A. Chianese, L.P. Cordella, and A. Marcelli, "Recovering Dynamic Information from Static Handwriting," *Pattern Recognition*, vol. 26, no. 3, pp. 409-418, 1993.
- [7] P.M. Lallican and C. Viard-Gaudin, "A Kalman Approach for Stroke Order Recovering from Off-Line Handwriting," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 519-523, 1997.
- [8] H. Chang and H. Yan, "Analysis of Stroke Structures of Handwritten Chinese Characters," *IEEE Trans. Systems, Man, and Cybernetics B*, vol. 29, no. 1, pp. 47-61, Feb. 1999.
- [9] W. Huang, G. Rong, and Z. Bian, "Strokes Recovering from Static Handwriting," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 861-864, 1995.
- [10] V. Govindaraju and R.K. Krishnamurthy, "Holistic Handwritten Word Recognition Using Temporal Features Derived from Off-Line Images," *Pattern Recognition Letters*, vol. 17, no. 5, pp. 537-540, 1996.
- [11] I.S.I. Abuhaiba, M.J.J. Holt, and S. Datta, "Processing of Binary Images of Handwritten Text Documents," *Pattern Recognition*, vol. 29, no. 7, pp. 1161-1177, 1996.
- [12] S. Jäger, "Recovering Dynamic Information from Static, Handwritten Word Images," PhD dissertation, Univ. of Freiburg, 1998.
- [13] S. Jäger, "A Psychomotor Method for Tracking Handwriting," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 528-531, 1997.
- [14] P.M. Lallican, C. Viard-Gaudin, and S. Knerr, "From Off-Line to On-Line Handwriting Recognition," *Proc. Seventh Int'l Workshop Frontiers in Handwriting Recognition*, pp. 303-312, 2000.
- [15] Y. Kato and M. Yasuhara, "Recovery of Drawing Order from Scanned Images of Multi-Stroke Handwriting," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 261-264, 1999.
- [16] Y. Al-Ohali, M. Cheriet, and C.Y. Suen, "Efficient Estimation of Pen Trajectory from Off-Line Handwritten Words," *Proc. Int'l Conf. Pattern Recognition*, pp. 323-326, 2002.
- [17] Y. Al-Ohali, M. Cheriet, and C.Y. Suen, "Dynamic Observations and Dynamic State Termination for Off-Line Handwritten Word Recognition Using HMM," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition*, 2002.
- [18] J.K. Guo, D. Doerman, and A. Rosenfeld, "Forgery Detection by Local Correspondence," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 15, no. 4, pp. 579-641, 2001.
- [19] K.K. Lau, P.C. Yuen, and Y.Y. Tang, "Recovery of Writing Sequence of Static Images of Handwriting Using UWM," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 1123-1128, 2003.
- [20] L.R. Rabiner and B.H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, pp. 4-16, Jan. 1986.
- [21] L. Lam, S. Lee, and C.Y. Suen, "Thinning Methodologies—A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869-885, Sept. 1992.
- [22] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*. Addison-Wesley, 1992.
- [23] M. Seul, L. O'Gorman, and M.S. Sammon, *Practical Algorithms for Image Analysis: Description, Examples, and Code*. Cambridge Univ. Press, 2000.
- [24] R.C. Gonzalez, R.E. Woods, and S.L. Eddins, *Digital Image Processing Using Matlab*. Pearson Prentice Hall, 2004.
- [25] J.J. Zou and H. Yan, "Skeletonization of Ribbon-Like Shapes Based on Regularity and Singularity Analysis," *IEEE Trans. Systems, Man, and Cybernetics B*, vol. 31, no. 3, pp. 401-407, June 2001.
- [26] J. Rocha, "Perceptually Stable Regions for Arbitrary Polygons," *IEEE Trans. Systems, Man, and Cybernetics B*, vol. 33, no. 1, pp. 165-171, Feb. 2003.
- [27] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry Algorithms and Applications*, second ed. Springer, 1997.
- [28] M.E. Munich and P. Perona, "Visual Identification by Signature Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 200-217, Feb. 2003.
- [29] P. Fränti, A. Mednongov, and H. Kälviäinen, "Hough Transform for Rotation Invariant Matching of Line-Drawing Images," *Proc. Int'l Conf. Pattern Recognition*, pp. 389-392, 2000.

- [30] O.R. Terrades and E. Valveny, "Radon Transform for Lineal Symbol Representation," *Proc. Int'l Conf. Document Analysis and Recognition*, pp. 195-199, 2003.
- [31] A.K. Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.
- [32] A.S. Rosenthal, J. Hu, and M.K. Brown, "Size and Orientation Normalization of On-Line Handwriting Using Hough Transform," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, p. 3077, 1997.
- [33] R. Kapoor, D. Bagai, and T.S. Kamal, "A New Algorithm for Skew Detection and Correction," *Pattern Recognition Letters*, vol. 25, pp. 1215-1229, Aug. 2004.
- [34] J.A. du Preez, "Efficient Training of High-Order Hidden Markov Models Using First-Order Representations," *Computer Speech and Language*, vol. 12, no. 1, pp. 23-39, Jan. 1998.
- [35] Y. Bengio and P. Frasconi, "Diffusion of Context and Credit Information in Markovian Models," *J. Artificial Intelligence Research*, vol. 3, pp. 249-270, 1995.
- [36] K.T. Abou-Moustafa, M. Cheriet, and C.Y. Suen, "On the Structure of Hidden Markov Models," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 923-931, June 2004.
- [37] J.A. du Preez and D.M. Weber, "The Integration and Training of Arbitrary-Order HMMs," *Australian J. Intelligent Information Processing Systems*, vol. 5, no. 4, pp. 261-268, 1998.
- [38] J.A. du Preez, "Efficient High-Order Hidden Markov Modelling," PhD dissertation, Stellenbosch Univ. Press, 1997.
- [39] P.Z. Peebles, *Probability, Random Variables, and Random Signal Principles*, third ed., McGraw-Hill, 1993.
- [40] J.R. Deller, J.H.L. Hansen, and J.G. Proakis, *Discrete-Time Processing of Speech Signals*. IEEE Press, 2000.
- [41] J.G.A. Dolfing, "Handwriting Recognition and Verification: A Hidden Markov Approach," PhD dissertation, Eindhoven, Netherlands, 1995.
- [42] Y. Bengio, "Markovian Models for Sequential Data," *Neural Computing Surveys*, vol. 2, pp. 129-162, 1999.



Johan A. du Preez joined the Department of Electrical and Electronic Engineering at the University of Stellenbosch in 1989 after four years in the telecommunications sector. He received the PhD degree in electronic engineering from the University of Stellenbosch in 1998. He is active in the broader fields of signal processing and pattern recognition. His research focuses on developing advanced statistical pattern recognition techniques such as higher-order hidden Markov modeling and applying them to speech, online handwriting, and image processing problems. He is a member of the IEEE and the IEEE Computer Society.



B.M. Herbst received the PhD degree in applied mathematics from the University of the Free State, Bloemfontein, South Africa, in 1982. He visited the University of Dundee as a postdoctoral student during 1983 and was a visiting associate professor at Clarkson University, Potsdam, New York (from 1988 to 1989), and the University of Colorado, Boulder (from 1989 to 1990 and again during 1994). He joined the Department of Applied Mathematics at the University of Stellenbosch in 1998. His current research interests include scientific computation, pattern recognition, and computer vision.

► **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**



Emli-Mari Nel received a degree in electrical and electronic engineering from the University of Stellenbosch, South Africa, in 2001. She is currently busy with her PhD in electronic engineering at the University of Stellenbosch. Her research interests include pattern recognition and computer vision.