Combination of HMMs for the representation of printed characters in noisy document images

A J Elms and J Illingworth

Many methods of printed character recognition have been proposed to-date, but although performance figures are usually stated for a particular set of fonts or size of text, it is rarely clear under what conditions of noise the measurements were taken. Baird has suggested a model of Document Imaging Defects, which enables authors to compare results against an emerging standard where one figure can be quoted to quantify the level of noise present in the document image. In this paper, a novel method is proposed for the recognition of printed characters, and its extension to the segmentation and recognition of noisy printed words is outlined. The method is based on the representation of the shape of a character by two Hidden Markov Models. Recognition is achieved by scoring these models against the test pattern and combining the results. The method has been evaluated using Baird’s noise model, producing a peak performance of 99.5% on the test set in the presence of near-minimal noise. The method generalizes to recognize characters with noise levels greater than those included in the training set, and an investigation of the top-k performance suggests that much of the effect of noise on the recognition performance on images of natural language text could be overcome using a word recognizer employing shallow contextual knowledge.

Keywords: character recognition, Hidden Markov Models, shallow contextual knowledge

INTRODUCTION

Methods for the classification of printed character images largely fall into two groups: those which rely on structural analysis, and those which are based on template matching. Structural methods have the advantage of representing the pattern in an abstract form (a combination of strokes) which lends itself to polyfont recognition, whereas template matching methods are generally based on a more explicit representation of the pattern. Structural methods can fail when pixels introduced by a noise process either split or connect strokes.

Many authors claim performance figures for such methods, but without common databases of printed material they are difficult to compare. It is also often the case that results are reported for noise-free images which are generated on clean paper by a high-quality laser printer, and are immediately and carefully scanned in. Such images rarely occur in real applications due to the use of copiers and fax machines, as well as the effects of handling and ageing. However, it is unusual to see a quantitative evaluation of the performance of a character recognizer in such conditions.

Baird has defined a model representing defects typical of document images and the use of the model to generate artificial noisy images for the training and testing of a polyfont classifier has also been reported.

In this paper, a character recognizer is described which uses a novel combination of two Hidden Markov Models (HMM) to capture an abstract representation of a training set of character images. This method is applicable to polyfont recognition, but does not rely on segmenting each pattern into strokes and is therefore tolerant to noise. In a single-font application a peak recognition rate of 99.5% is achieved, and Baird’s Defect Model is used to quantify how the performance varies with the amount of noise present in the image.
It has been suggested that the limiting factor in printed word recognition is the segmentation of the word into characters, rather than the classification of the isolated characters. Following the evaluation of the HMM recognizer’s performance on isolated characters in the presence of noise, it is demonstrated how this scheme can be extended to produce a segmentation of a word image. The level building method of linking the Hidden Markov Models produces the segmentation as a by-product of the recognition result—this maximizes the joint probability of a correct segmentation and recognition, which is clearly preferable to a segment-then-recognition strategy.

Related work

Although much work on HMMs has been done in the field of speech recognition, only recently have they been applied to text recognition. Anighbugo and Belaid claim performance of between 96% and 98.65% on a range of fonts. A limitation of their method is that the characters have to be segmented prior to classification. A number of features such as number of black–white transitions and aspect ratio are extracted and ordered heuristically, creating a sequence of discrete observations from a character which can be modelled by an HMM. However, there is no obvious explanation as to why there should be any Markov dependency in the sequence. Vlontzos and Kung use a similar system of competing HMMs, with 95% accuracy, but use a parameterization of structural primitives such as strokes and arcs as the observation sequence. These are ordered according to a traversal of the skeleton of a segmented character which is guided by an analysis of junction points in the skeleton. This method also requires the prior segmentation of the characters, and the proposed method of structural analysis is unlikely to be reliable in the presence of noise.

An alternative approach is taken for the recognition of handwritten words by Kundu et al. A single model is used in which each state represents a letter and a word is recognized by using the Viterbi algorithm to trace out the maximum likelihood state sequence. Observations are derived from a prior segmentation of the word into its constituent letters. Features are extracted which are clustered to produce a codebook of observation symbols. An extension of this work is that of Chen et al., which is intended for recognition of words from a fixed lexicon of limited size. Rather than requiring an initial segmentation of a word into characters, it is split into small segments. Again, one model is used to represent the words in the lexicon, but each state may represent a fraction of a character, a whole character or touching characters. Another method which does not rely on a prior segmentation into characters is that of Bose and Kuo. In a method analogous to a recognition system for connected spoken words described by Rabiner and Levinson, a number of competing HMMs are used to recognize characters based on an observation sequence derived from a clustering of feature vectors from small segments of each word. A level building search procedure is used to produce the maximum likelihood character sequence at the same time as the segmentation, making it suitable for connected and degraded character images.

Chen et al. use HMMs to model a sequence of observations derived from the vertical columns of pixels in the image of a line of text. The observation features encode firstly the position of the upper and lower contour of the text at each column, and secondly the pixel structure between the contours (using an averaged autocorrelation function). The HMMs are used to represent characters, which can be recognized without prior segmentation. This method is applied to keyword spotting in images where characters may be split or merged. Agazzi and Kuo represent two-dimensional images of characters with pseudo-2D HMMs, in an attempt to avoid the computational complexity of fully-connected 2D models. The pseudo-2D models are not fully-connected—states are connected in horizontal chains to represent rows of pixels, and the chains of states, or ‘superstates’, are linked vertically. The P2DHMMs can be used to recognize degraded characters without prior segmentation, and can simultaneously normalize the characters with respect to scaling, slant angle and translation. The method is shown to be superior to a 1D model, where an observation feature is generated for each column of pixels in the text image. The feature is simply an encoding of the bit values in the column and the values in the adjacent columns. Park and Lee heuristically combine evidence from four 1D HMMs which model features derived from the Regional Projection Contour Transformation of a character image. It does not appear suitable for connected character recognition, however, because only one of the directional projections could be computed prior to segmentation, and the classification accuracy of the individual models is shown to be greatly inferior to that of the combination. Furthermore, methods which rely on contour tracing could prove fragile in the presence of noise which may corrupt the contours.

Our method is similar to that of Bose and Kuo in that a segmentation of a printed word into its constituent characters is not required prior to recognition. However, our method does not require a complex structural analysis of the input image, and the training and recognition require no manual intervention. The features used represent a complete column of pixels in a way that could be related to the method of Chen et al. use to represent the pixel structure between contours. We do not use contour tracing—instead adjacent columns are related by the deviation from a running average of the ‘centre of gravity’ of the pixel mass. Unlike the 1D model-based methods, we represent the shape of the 2D image using two 1D models, one representing the patterns of pixels found in columns of a character image, and one representing the row pixel patterns. However, our two models are separate, whereas Agazzi and Kuo link 1D models together to
form the P2DHMM. Our two 1D models show a good classification accuracy on their own, albeit not as good as when their results are combined, so that unlike Park and Lee's method, our 1D model which represents the columns of pixels has sufficient power to reliably segment a word image and hypothesize a label for the characters. The 1D row model can then be used to verify the hypotheses.

A full characterization of the performance of our algorithm on isolated characters has been undertaken to determine its peak performance, as well as its susceptibility to document image noise.

HMM RECOGNIZER

A discrete HMM is a representation of the statistics of a random process which produces discrete observations \( O(t) \) at sampling instants \( t = 1, \ldots, T \). If such a sequence is observed, a model \( C \) can be 'scored' in terms of a likelihood that the process which \( C \) represents could have produced the sequence. Given a number of sequences, each one derived from one of \( Q \) classes of pattern, and a set of models \( C_q, q = 1, \ldots, Q \) where each model \( C_q \) has been trained on pattern class \( q \), it is possible to classify the sequences by scoring all models against each sequence and choosing the model with the highest likelihood score. Thus, the HMM is a natural mechanism for recognizing a pattern which can be represented as a sequence of discrete observation symbols.

Consider the character \( m \) depicted in Figure 1. Taking horizontal single pixel-width scan lines down the image, it is clear that the pattern is characterized by observing a few lines with long runs of ON pixels, followed by a large number of lines with three short runs of ON pixels. When scanned vertically, a similar generalization can be made about the patterns: three sets of lines with long single runs of ON pixels are separated by two sets of lines with short single runs. The HMM appears to be a natural way of expressing this profile of image features—the presence of a feature depends upon those immediately prior to it, but not on features much earlier in the profile (a low-order Markov dependency).

Feature extraction

The assumption outlined thus far is that a character image can be represented by the profile generated by observing a sequence of horizontal or vertical scan lines through the image. Furthermore, for a \( 64 \times 64 \) pixel binary image (for example) it is unlikely that there will be \( 2^{64} \) possible scan lines observed in either the horizontal or vertical profiles generated for a set of images. The patterns observed in the scan lines therefore lend themselves to quantization, such that a pattern is represented by one entry from a codebook of possible patterns, where the entry is chosen as the closest to the pattern, in some sense. The codebook design is discussed further in the next section.

It is not possible to directly cluster a training set of scan lines using a simple distance measure. Consider two patterns taken from scans through an image. They may be identical, save for a one pixel offset in the direction of the scan. This offset introduces a large distance between the patterns, which is counter-intuitive to the assumption of how the scan lines represent the image. It is therefore desirable to represent the scan lines in a way which is independent of the absolute location of the character within the image window. By transforming the scan line to produce the magnitude of its complex 1-dimensional Fourier Transform (using a Blackman window and a 64 point FFT), the resulting representation is shift invariant, and due to the symmetry of the spectrum has halved the number of dimensions of the vector—a 64 pixel scan line becomes a 32 dimensional spectrum. If the features extracted from the vertical scan line were completely shift-invariant, however, the characters \( b \) and \( p \) would be very similar in feature space. Therefore, a measure of the centre-of-gravity (COG) relative to the running average of the COG of previous scan lines is incorporated as the 33rd dimension of the feature vector. This is designed to emphasize large changes of COG with respect to previous lines, whilst reducing the effect of small differences which may be due to noise.

The feature vector for scan line \( i \) is therefore:

\[
F_i(k) = (R_i(k)^2 + I_i(k)^2)^{\frac{1}{2}} \quad k = 0, \ldots, 31
\]

\[
F_i(32) = \left(\frac{\text{COG}^2_i - \text{COG}_{i-3} - \text{COG}_{i-1} - \text{COG}_i}{3}\right)^{\frac{1}{2}}
\]

(1)

where \( R_i(k) \) and \( I_i(k) \) are the real and imaginary components of the complex spectrum of scan line \( i \), and COG is its centre of gravity. Figure 1 depicts the feature extraction process for horizontal and vertical scan lines of a character image.

Vector quantization

The result of the feature extraction defined in equation (1) is a continuous 33-dimensional vector representing each vertical and horizontal line through the image. The HMMs are used to recognize a sequence of discrete symbols, so it is necessary to quantize the feature vectors such that each one can be represented by a symbol representing the closest vector contained in a codebook.
The codebook must first have been created to closely approximate the feature vectors in a training set. The training set is clustered according to the Post Transfer Advantage rule of Kittler and Pairman19. Clusters are represented as normal distributions in 3D and cluster sizes are taken into consideration. This gives an improved partition of the data set when compared to a simple k-means approach. The result of this clustering is a codebook which represents each cluster by its mean vector, covariance matrix and number of elements. During the recognition process, the feature vector calculated for each scan line of the test image is used as the key to a linear search of the codebook—an observation symbol is produced according to which cluster is most likely to contain the vector. This likelihood is computed from the cluster parameters, using the number of elements to estimate the prior probability of the vector being contained in the cluster.

Model training and scoring

A set of example images is required for the training of the models for each of the character classes to be recognized. Features are extracted as described above, resulting in a set of sequences of symbols representing the horizontal and vertical profiles of the training images. A HMM can then be trained (using the Baum-Welch reestimation procedure4) to represent the statistics of the process which generates the training sequences. This means that each class is represented by two models—one representing possible vertical profiles, and one representing horizontal profiles.

The HMM models a sequence as the output of a finite state automaton, in which a transition from state i to state j occurs according to a probability \(a_{ij}\) at each time instant, and an output is observed based on a probability distribution \(b_j(x)\) associated with the state \(j\) the model is in at that time instant. Figure 2a shows a typical state transition diagram. The Markov dependency is between state transitions—the probability of a transition from state \(i\) to state \(j\) depends only upon the values \(i\) and \(j\) (a first-order Markov assumption) and not on any previous transitions. The result of training on vertical scan lines can be envisaged as shown in Figure 2b—the model will partition the sequence of observations between its states such that in state 1 symbols representing long single runs of ON pixels are more likely to be observed, whereas in state 2 short single runs are more likely.

During recognition, a Viterbi scoring method is used to find the likelihood score \(L(q)\) that each model \(C_q\) could have produced the given observation sequence. Each model is matched against the observation sequence beginning at sample number \(t = 1\). Initialization is:

\[
\delta_1(1) = [b_j^n(O_1)]
\]

Recursion is for \(2 \leq t \leq T, 1 \leq j \leq N:\)

\[
\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i)a_{ij}] \cdot [b_j^n(O_t)]
\]

Termination occurs when:

\[
L(q) = \delta_T(N)
\]

The matching of the observation sequence to a character model can exhibit time-warping properties as it is possible for a short observation sequence to be generated by the model of a character which normally produces a long sequence. This is due to the long forward state transitions as depicted in Figure 2a. Rabiner and Levinson reduce this effect in their spoken word recognition system13 by heuristically incorporating the probability of word duration. In a similar fashion, we model printed character width as a normal distribution, deriving \(\mu_q\) and \(\sigma_q\) (in number of pixels) from training data.

The probability of model \(C_q\) matching an observation sequence of \(T\) samples is:

\[
P_{\theta}(T) = \frac{\exp(-\frac{(T-\mu_q)^2}{2\sigma_q^2})}{\sqrt{2\pi\sigma_q}}
\]

The likelihood score \(L(q)\) is modified to account for the character width probability \(P_{\theta}(T)\):

\[
\tilde{L}(q) = L(q)(P_{\theta}(T))^{\gamma}
\]

where \(\gamma\) is a weighting factor which was optimized experimentally. Various values of \(\gamma\) were tried during a small set of word recognition tests and \(\gamma = 3.0\) appeared to give the best results, but the results were not particularly sensitive to the value of \(\gamma\).

COMBINATION OF RECOGNIZERS

By using the scoring mechanism described above, a likelihood can be computed for each class \(q\) that the model \(C_q\) could have produced a given input pattern. Strictly speaking, there are two likelihoods, \(\tilde{L}_v(q)\) and
\( L_U(q) \), relating to the models of the vertical and horizontal profiles, respectively. It will be seen experimentally that choosing the maximum likelihood model from either of these ‘experts’ produced a good recognition rate, but clearly combination of the evidence from both experts is likely to produce a better overall result. One solution to this problem is to compute:

\[
L_V(q) = L_V(q) L_U(q) \tag{7}
\]

Equation 7 is only true given that \( L_V(q) \) and \( L_U(q) \) are statistically independent. Since both recognizers work well in isolation, the evidence from one expert allows us to predict the label of the input pattern. Then we can predict the output of the other expert. Thus the two experts are dependent. However, it will also be shown later that in practice equation 7, an approximation to the real situation, can be used to combine the likelihoods from the two experts. This results in an improved recognition rate greater than either could have achieved in isolation. Clearly, a better combination rule, which exploits the dependence between the two experts, is a subject of future investigation.

As all three likelihoods, \( L(q) \), \( L_V(q) \) and \( L_U(q) \) are computed for each class \( q \), it is possible to rank the likelihood (whether from an individual expert or from the combined result) to produce a number of ranked alternative recognition results. A top-\( k \) recognition result can therefore be computed for this method. This is the percentage of test cases where the correct class label appears in the first \( k \) choices output by the recognizer. This figure indicates how well the recognizer would perform given further contextual information—if, for example, the top-3 performance is significantly better than the top-1, then a bottom-up contextual knowledge source (such as the probabilities of transitions between letters) is expected to improve the performance of the recognizer on running text.

**Noise model**

Baird’s model of document image defects facilitates production of artificial images which are corrupted by typical document image noise. Each output image takes as its basis a noise-free input image which is then corrupted by noise characterized by a number of model parameters. These parameters can themselves vary from image to image according to a pseudo-random process. The model has ten parameters quantifying Resolution, Blur, Threshold, Sensitivity, Jitter, Skew, Width, Height, Baseline and Kerning. The Mahalanobis Distance (MD) of the model parameters from the mean of the defect distribution can be used as a single measure of image quality—the larger the MD the more deformed the image. Note that Baseline, Jitter and Kerning are not included in the MD calculation.

The model was implemented to reflect the description by Baird as closely as possible. The GNU project PostScript interpreter ‘GhostScript’ was used to generate a bitmap image for each character at 300 dpi resolution. The Skew, Width and Height variations were then applied to the bitmap. This was done by means of an inverse transformation—each pixel in the output image was mapped back to its location in the original image under the inverse transform. In general, this point would lie between four pixels on the original image, so the output pixel level was found by linear interpolation. The resulting image was then oversampled by a factor of 8, and the output pixel centres were computed according to the Jitter and Kerning parameters. The over-sampled image was then blurred by a circularly symmetric Gaussian filter with a standard error \( \sigma_{f_{\text{blur}}} \) set by the Blur parameter. The extent of the filter was limited to \( \pm \sigma_{f_{\text{blur}}} \). The filter was centred on each output pixel centre, and the result of the convolution was normalized by dividing by the sum of the weights computed for a filter with the mean Blur of 0.7. The output was adjusted according to the Sensitivity parameter, and was finally binarized by comparison with a threshold. A value of 0.43 was selected as optimal for a Blur of 0.7. It was noted that for a Blur parameter selected from a normal distribution with \( \mu_{\text{blur}} = 0.7, \sigma_{\text{blur}} = 0.3 \), it is necessary to truncate the distribution in order that the Blur value is positive. In fact, it was observed that Blur must be \( \geq 0.37 \) for the output image to be non-empty.

The MD, used as an overall measure of image quality, can be related to the population upon which the model was based by the graph of Figure 3. If the defect model represents all possible defective document images, then the graph enables us to predict what proportion of defects we are considering if we limit the MD to a given range. The percentage coverage of the distribution was found experimentally by successively generating model parameters, and calculating the proportion of the distribution covered by an MD in the range \([0.0, x]\) as the number of generated parameter sets produced such that 1000 of the sets had an MD in the range \([0, x]\), divided into 1000. The range \([0.0, 2.0]\)
can be seen to cover the best quality 35% of the population and the range [0.0,2.6] covers 70% of the population.

Results

The experimental procedure consists of three main stages. A codebook must first be generated – this is the most time consuming stage. Then, the codebook is used to extract feature sequences from a set of training images, and(157,316),(878,552) these sequences are used to train the HMMs representing each class. Finally, features are extracted from a test set of images using the same codebook, and the models are scored against the test sequences to produce a ranked recognition result. The details of the experiments can be summarized as follows:

- **Training data set**: 200 examples of each character a-za-z0-9 were generated by the noise model with MD limited to the range [0.0,2.0].
- **Test data set**: 100 examples of each character were generated by the noise model for each of the 13 MD ranges [0.2,0.4], [0.4,0.6],...,[2.6,2.8].
- **Codebook**: generating a codebook from the entire training set would be far too computationally expensive, so the noise model was used to produce 10 images of each character in each of the MD ranges [0.0,0.5], [0.5,1.0], [1.0,1.5], [1.5,2.0].
- **HMMs**: models were trained over a maximum of 30 iterations using eight states. (These figures have been optimized experimentally.) Two models per class represented horizontal and vertical profiles, respectively.

Recognition was attempted for both the training and test sets, using three methods: horizontal profile model only; vertical profile model only; and the combined result of these experts according to equation (7). As the training set was generated with a wide range of noise parameters, its recognition rate represents an ‘average’ figure, whereas recognition rates for the test sets apply to narrow bands of noise parameters. Therefore, the peak performance on the test set (for low noise conditions) is better than the performance on the training set, which is averaged over a wide range of noise conditions. The results are summarized in Table 1.

In all cases, the peak recognition rate of the test set corresponds to an MD of the noise parameters in the range [0.4,0.6]. It is interesting that this does not correspond with the absolute lowest noise levels – this reflects the fact that the models are trained over a wide range of noise.

Figure 4 shows how the three methods compare over a wide range of noise conditions. The recognition rate is calculated for each noise level in the test set, and is plotted on the graph against the corresponding MD range of the model. For example, a test set generated with an MD in the range [2.6,2.8] has a larger noise content than a test set generated with an MD in the range [0.2,0.4], as the noise parameters are a greater distance from the population mean. Therefore, it is reasonable to expect the classification accuracy of any recognizer to decrease as the MD increases. It can clearly be seen that the result of combining the horizontal and vertical profile recognizers is consistently better than either in isolation. It is also interesting to note that performance of the horizontal profile recognizer (which takes vertical scan lines through the image) is considerably better than that of the vertical profile recognizer. It could be suggested that as we read text in a left/right direction, the discriminant information in our alphabet is greater in the horizontal profile than the vertical.

It is possible to draw some conclusions on the generalizing power of the recognizer. Although trained on the noise range [0.0,2.0] which covers the top 35% of the population, when tested over the range

<table>
<thead>
<tr>
<th>Scan direction</th>
<th>Test set (peak % correct)</th>
<th>Training set (% correct)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>92.9</td>
<td>86.6</td>
</tr>
<tr>
<td>Vertical</td>
<td>97.8</td>
<td>95.4</td>
</tr>
<tr>
<td>Combined</td>
<td>99.5</td>
<td>98.0</td>
</tr>
</tbody>
</table>
[0.2, 2.8], with over twice the population coverage, the recognition rate declined gracefully, suggesting good generalization.

The recognizer also exhibits promising behaviour for the recognition of language text using contextual information—if the top choice is not correct then the correct answer is likely to be ranked highly (Table 2).

Given the font under test, it is not surprising that confusions such as those between the characters 1 and I (ell. EYE, one) occur. The top-k performance suggests that a context driven recognizer would be able to recognize printed words with a much higher success rate than that of the isolated character recognizer. Figure 5 shows in more detail how the top-k performance is affected by the noise-level.

**EXTENSION TO WORD RECOGNITION**

The likelihood score of equation (4) is a result of matching a HMM against a sequence of discrete symbols derived from a character image. This method of model matching has been extended12 in the application of speech recognition to show how a number of models may be matched, still in the maximum likelihood sense, to an observation sequence derived from a number of patterns. This method, known as the Level Building Search procedure, is applicable to recognizing printed words, where the characters have become connected or fragmented due to noise. It is possible for the horizontal profile of the characters constituting a printed word to be evaluated prior to segmentation by taking vertical scan lines through the word image. As has been demonstrated, the horizontal profile alone can provide a good recognition rate, and confusions that occur are generally between characters of similar width such as w and V. Therefore a level building search based on matching the horizontal profile models to the observation sequence can be used to hypothesize about the letter segmentation points and the corresponding letter identities. It is then possible to construct the vertical profile of the letters to confirm or amend the hypothesis.

Figure 6 shows how the connected character recognizer is used for printed word recognition without first segmenting the word into isolated characters. The box below the image highlights the results of the matching. The #chars column shows the number of models that the recognizer is attempting to match against the sequence, up to a user-defined maximum. The Score column is a measure of the likelihood of result. As this is on a logarithmic scale it lies in the range \([-\infty, 0]\), with the most likely result being the one with the score closest to 0. The Result field indicates the output, which is the sequence of models which have best matched the observation sequence.

It is not trivial to segment this image into characters. Considering the amount of noise present the letter ‘m’ in the word ‘holmes’ could plausibly be the bigram ‘rn’. This confusion is reflected by the recognizer scores for the 6 and 7 letter matches (‘holmes’ and ‘holrnes’, respectively). The relatively small difference between the scores indicates the problems that would be faced by a segment-then-recognize strategy.

**CONCLUSIONS**

In this paper, a novel method has been outlined for the recognition of printed characters. The method is based on finding ‘profiles’ of an image by approximating raster scans of the image to represent the shape of a character, using Hidden Markov Models.

Our method, although not optimized for speed, can recognize characters at a rate of approximately one per second. The main computational requirement is for the Vector Quantization. The VQ process and also the
scoring of the HMMs are highly parallel in nature. A commercially feasible implementation would therefore be possible through the exploitation of such parallelism or through the use of hardware accelerators for the computationally expensive tasks.

The use of the models for classifying test images has been studied in the presence of typical document imaging defects, enabling a performance figure to be measured given set noise conditions. A model of imaging defects has been used such that the level of noise in the test condition is itself quantifiable.

Finally, an extension of the recognition method has been outlined to demonstrate one of its great advantages. The method is capable of recognizing printed words, in which the individual letters may become fragmented or merged, without its prior segmentation into letters. Indeed, the method provides the maximum of the joint likelihood of a correct segmentation and recognition.

ACKNOWLEDGEMENTS

Thanks to J Kittler for help with the vector quantization. Also to P Hoad, M Petrou and G Matas for advice on the implementation of Baird’s Defect model.

REFERENCES