

TOWARDS AUTOMATIC TRANSCRIPTION OF ESTRANGELO SCRIPT

WILLIAM F. CLOCKSIN & PREM P.J. FERNANDO

Department of Computing
Oxford Brookes University
Wheatley, Oxford OX33 1HX,
U.K.

&

Computer Laboratory
University of Cambridge
Cambridge CB3 0FD
U.K.

ABSTRACT

This paper surveys several computer-based techniques we have developed for the automatic transcription of Estrangelo handwriting from historical manuscripts. The Syriac language has been a neglected area for research into automatic handwriting transcription, yet is interesting because the preponderance of scribe-written manuscripts offers a challenging yet tractable medium between the extremes of type-written text and free handwriting. The methods described here do not need to find strokes or contours of the characters, but exploit characteristic measures of shape that are calculated by geometric moment functions. Both whole words and character shapes are used in recognition experiments. After segmentation using a novel probabilistic method, features of character-like shapes are found that tolerate variation in formation and image quality. Each shape is recognised individually using a discriminative support vector machine with 10-fold cross-validation. We describe experiments using a variety of segmentation methods and combinations of features. Images from scribe-written historical manuscripts are used, and the recognition results are compared with those for images taken from clearer 19th century typeset docu-

ments. Recognition rates vary from 61–100% depending on the algorithms used and the size and source of the data set.


1. INTRODUCTION

Syriac manuscripts dating back to before the 6th century CE are available in large quantities and are undergoing the process of manual transcription into machine-readable form for scholarly analysis, commentary, and publication. Manual transcription and keyboarding is a tedious and laborious task that few are willing and qualified to undertake. Syriac scholars would welcome a computer-based system that is able to provide transcriptions into machine-readable form with a reasonable accuracy. Any errors made by the automatic transcriber could then be corrected manually as part of on-line proofreading. Syriac is a useful vehicle for automatic handwriting transcription research because many sources are carefully written by scribes. Therefore, as far as the designers of optical character recognition (OCR) algorithms are concerned, Syriac manuscripts present a large corpus that is intermediate in difficulty between type-written text and unconstrained handwriting. OCR of clearly typewritten Roman-style text is essentially solved, and OCR of unconstrained handwriting will continue to be a challenging research problem far into the foreseeable future. By contrast, in scribe-written texts there is sufficient regularity for the OCR problem to be tractable, while there is sufficient variation to require the development of techniques more sophisticated than standard OCR methods. This rationale has also motivated our previous work in automatic transcription of scribe-written Arabic [14, 6]. Syriac is one of the simpler early Semitic languages, lacking the grammatical complexity of classical Arabic and the unpredictability of biblical Hebrew. Although the system described in this paper does not have comprehensive competence, the relative simplicity of Syriac offers motivation for further development of a complete system for Syriac handwriting transcription. Of the several script forms in use, here we focus on Estrangelo, found in the oldest manuscripts, also later widely used in Europe for printed books.



Fig. 1. The word ܩܢܡܐ qnoma 'person, self' from MS.

No previous work has been published on automatic recognition of Syriac handwriting, but this work falls into the general category of off-line cursive script recognition, an area in which there has been much effort [23, 21, 2]. However, from a character recognition perspective, Syriac is similar to Arabic, and the existing research in Arabic character recognition has been comprehensively surveyed [15] recently. The system described in this paper implements a standard statistical classification framework [12]. Figure 2 shows the components of the system. In the training mode, a model is constructed using the input data as training data. In the recognition mode, the model is used to classify the previously unseen input data.

The results described below were obtained from a handwritten manuscript source (MS) and a typeset source (TS). Both sources were written in Estrangelo. The MS is a leaf¹ taken from Peter of Callinicum's *Adversus Damianum*, a 6th century commentary on the Trinity [8]. The TS consists of the 36 pages of Mark's Gospel taken from Burkitt's 1904 edition [3] of the *Evangelion Da-Mepharreshe* typeset in the late 19th century. Pages were scanned at 300 dpi and saved as 8-bit greyscale images. Any editorial apparatus (brackets, verse numbers, footnotes) was removed manually. Figure 1 shows an example of the word)  *qnoma* 'person, self' from MS.

The trials described in Section 4 are mainly concerned with recognizing characters within the word. However, for comparison purposes, a few trials on the recognition of whole words are also described. Practically, word recognition [23] or 'word spotting' [19] techniques are less useful for Syriac because it is a highly inflected language: Spellings change according to grammatical function, and almost all grammatical functions are written as word prefixes or suffixes instead of as separate words. Therefore, a combinatorially large lexicon would be required to support a word recognition approach. For this reason, we focus on a character recognition approach and remain attentive to relevant insights arising from the word recognition approach.

2. IMAGE PROCESSING

Given a page image from a source, image processing proceeds as follows. First, the connected components of the image are ex-

¹ British Library Add. MS 7191, Folio 100va-101rb, which contains the end of Chapter XXIV and the beginning of Chapter XXV of Book III.

tracted using the standard two-pass algorithm [13] in which a label is assigned to each pixel in the first pass, with label equivalence based on pixel connectivity with its eight neighbours. Equivalence classes are determined, and a second pass updates each pixel in a connected component with a label unique to the component. This algorithm has a running time of approximately $O(N)$ in the number of pixels. The bounding boxes of each component are then determined. Next, words are found by calculating the frequency distribution (histogram) of the horizontal separation between neighbouring bounding boxes. The idea is that the distance between words tends to be larger than the distance between components within a word [22]. The minima between two maxima in the histogram is located to determine a threshold above which inter-component separations are interpreted as inter-word spaces (Figure 4). In the data we have considered, there is a clear gap between modes of the histogram, leading to successful use of this method on both MS and TS sources (see Figure 3).

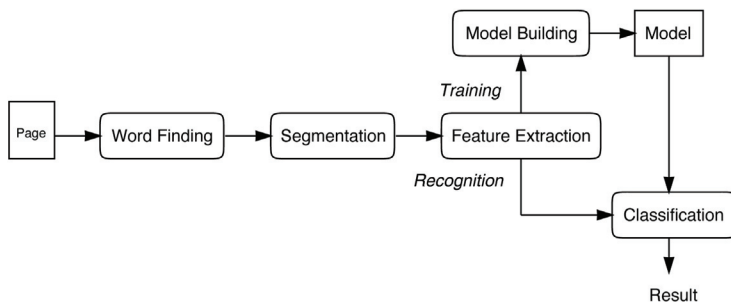


Fig. 2. Block diagram of recognition system. The system operates in training mode or recognition mode. Recognition mode requires that a model is available; the model is built during training mode.



Fig. 3. Portion of MS showing bounding boxes around words spotted automatically.

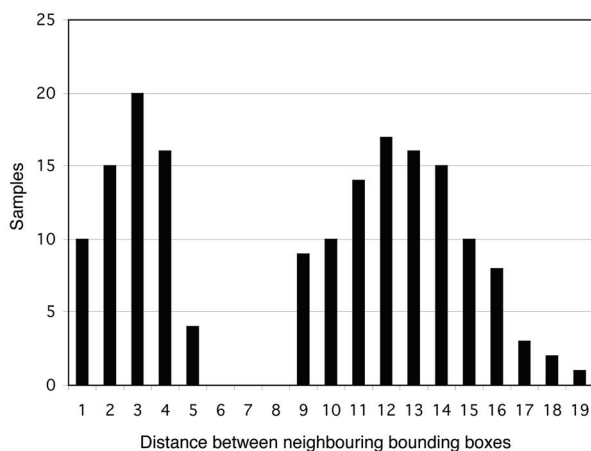


Fig. 4. A frequency distribution of the horizontal separation between neighbouring bounding boxes of connected components.

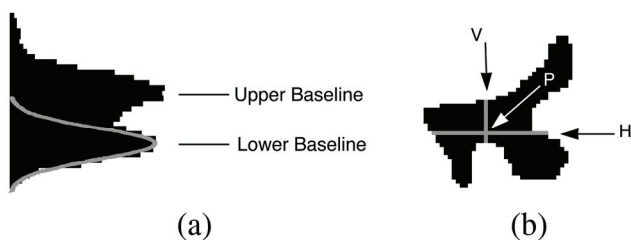


Fig. 5. Illustration of projections used by the segmentation algorithm. (a) The horizontal projection from the word sample of Figure 1, showing the upper and lower baseline. The normal density estimated from data near the lower baseline is superimposed in grey. (b) At point P within a shape, the vertical run V and horizontal run H through P are shown. The number of pixels in these runs gives the respective run lengths.

2.1. Character Segmentation

One of the main difficulties in cursive word recognition comes from segmentation of the connected characters within the word. In most cases the precise point of segmentation is indeterminate, and in some cases segmentation points can be ambiguous without using higher level contextual information such as the spelling of a word.

$$p_{seg}((r, c)|W) = p_{base}(r|W)p_{hrun}((r, c)|W)(1.0 - p_{vrun}((r, c)|W)). \quad (1)$$

Our approach is to score each pixel in a word with a likelihood of being a valid segmentation point based on general principles.

Because segmentation points lie on horizontal strokes near the baseline, pixels are given a score based on the distance from the lower baseline and approximations to the thickness and direction of the stroke. All measurements are efficiently calculated from horizontal and vertical projections and run lengths of the pixels in the image (Figure 5). For the purposes of definition, let pixels in a word image be represented as the array $W[r, c]$ having rows 1 to R and columns 1 to C ; the lower left corner pixel is $W[1, 1]$. The likelihood that a pixel at r, c is a segmentation point $p_{seg}((r, c)|W)$ may be conveniently modelled as

$$p_{seg}((r, c)|W) = p_{base}(r|W)p_{hrun}((r, c)|W)(1.0 - p_{vrun}((r, c)|W)). \quad (1)$$

The baseline likelihood $p_{base}(b|W)$ is estimated by using the horizontal projection h of the whole word

$$h_i = \sum_{j=1}^C W[i, j] \quad (2)$$

for each row i , then normalising h so that $\sum_{i=1}^R h_i = 1$. Syriac words tend to be formed so that h has two modes: one at the upper baseline and one at the lower baseline. The data between the lower mode and a point halfway between the modes is used to estimate the mean μ and variance σ^2 of a normal density modelling the horizontal projection of the word near the baseline. The likelihood of a pixel at row r being on the baseline is therefore

$$p_{base}(r) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(r - \mu)^2}{2\sigma^2}\right). \quad (3)$$

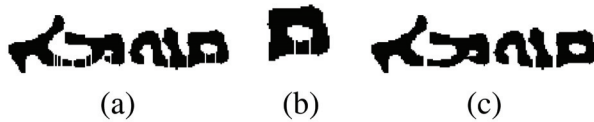


Fig. 6.(a) Segmented word showing oversegmentation. (b) Detail of the two spurious cuts made within the rightmost letter ܐ (qoph). (c) Segmentation corrected by eliminating 'nested' segmentations.

The horizontal and vertical run lengths of the pixels connected to r , c are measured and normalised by dividing by C and R respectively:

$$p_{hrun}(r, c) = \text{'horizontal run length through } r, c' / C \quad (4)$$

$$p_{vrun}(r, c) = \text{'vertical run length through } r, c' / R \quad (5)$$

Equation 1 therefore expresses the dependency of segmentation upon proximity to the baseline, and width of the horizontal and vertical run lengths of the neighbouring pixels. The probability of segmentation is maximised when a point is closest to the baseline, within a narrow horizontal stroke. This probability is maximised, for example, at the trough of a 'V' shape, which explains why some segmentation techniques use curvature (e.g. [2]).

Pixels where p_{seg} is maximal are chosen as segmentation points, and a vertical cut is made is at the point, stopping when the background is reached (Figure 6). The result is usually oversegmented in a systematic way. To correct this, spurious 'nested' segmentations are detected in the following way. First, the bounding boxes of segments are found. If a bounding box is entirely enclosed by another bounding box, the segmentation points given by the inner box are ignored. Also, single cuts within an enclosing bounding box are also ignored.

The segmentation method fails in two particular cases: for the unconnected letter \curvearrowright (*num*) because it crosses the baseline, and for the letter \mathfrak{H} (*Heth*) because it contains two places resembling plausible points of segmentation. The segmentation algorithm finds usable segmentations for about 70% of the characters. For the purposes of constructing a database of segmented characters to which classification trials could be applied, the remaining 30% were corrected manually. Curiously, this is the same over-segmentation rate as recently reported using a very sophisticated segmentation algorithm on neat italic English handwriting [2].

2.2. Feature Extraction

It is useful to represent character image data as a small set of features, partly to reduce the size of the model, and partly to characterise the data in ways that are invariant to typically encountered transformations and deformations. Geometric moments invariant to a variety of transformations are widely used in computer vision [13]. We have considered several alternative methods using moment functions. The first method follows the well known approach of using a set of predefined moment functions (e.g. [17]). The sec-

ond method starts from the generalized moment functions (GMFs) recently introduced by Chang and Grover [4].



Fig. 7. Image of the letter α and its size normalised polar map.

2.2.1. Pre-defined Moment Functions

We use the feature set $\{\bar{x}, \bar{y}, M_2, M_3, M_4\}$ defined as follows. Given an image function $f(x, y)$ with mass $N = \sum_x \sum_y f(x, y)$, the normalised central moments are

$$m_{p,q} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \quad (6)$$

where

$$\bar{x} = \frac{1}{N} \sum_x \sum_y x f(x, y) \quad \bar{y} = \frac{1}{N} \sum_x \sum_y y f(x, y) \quad (7)$$

Following [17], selected moment functions are

$$M_2 = \frac{(m_{20} - m_{02})^2 + 4m_{11}^2}{(m_{20} + m_{02})^2} \quad (8)$$

$$M_3 = \frac{(m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2}{(m_{20} + m_{02})^3} \quad (9)$$

$$M_4 = \frac{(m_{30} - m_{12})^2 + (m_{21} - m_{03})^2}{(m_{20} + m_{02})^3} \quad (10)$$

In the experiments described in the next section, these moments are applied to images of several kinds: the whole image of the character, subimages of overlapping and non-overlapping windows, and

a polar transformed image (with windows). The polar transformation, similar to the log-polar transform [25] widely used in computer vision research, is a conformal mapping from points in image $f(x, y)$ to points in the polar image $g(\xi, \eta)$. We adapt this by defining an ‘origin’ $O = (o_x, o_y)$ given by the centroid ($o_x = \bar{x}, o_y = \bar{y}$). Where d is the maximum distance between O and all pixels in f , the mapping is described by

$$\xi = \frac{\sqrt{(x - o_x)^2 + (y - o_y)^2}}{d} \quad (11)$$

$$\eta = \arctan\left(\frac{y - o_y}{x - o_x}\right) \quad (12)$$

We map f onto a polar image of size 64×64 , giving a representation that is size invariant and for which rotations have been transformed to translations (Figure 7). Because the resampling is dense and data is reduced, there is also a certain degree of smoothing of shape distortions. We have used this adaptation of the polar image previously for Arabic handwriting recognition [6], with comparable results.



Fig. 8. Four probing functions used by Chang and Grover (here redrawn from [4]). The leftmost function gives a result equivalent to the mass centroid. Each function is used in both the x and y directions.

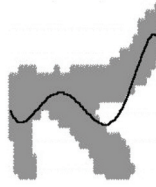


Fig. 9. Six degree polynomial signature ψ_x^a superimposed onto the letter α (alph).

2.2.2. More Generalized Moment Functions

The method of Chang and Grover [4] convolves the object with up to four different predefined ‘probing’ functions (better described as basis functions $\psi(x)$) as shown in Fig. 8. The basis functions are one-dimensional, so following [9] are combined using a complex convolution to scan the input image $f(x, y)$, so that a moment $m = f(x, y) \otimes (\psi(x) + i\psi(y))$. Within a window, convolution of each basis function with the image will result in a distinct generalized centroid (G-centroid) at the convolution’s zero-crossing point. Chang and Grover pair G-centroids into phasers that are used as features. However, we further generalise the GMF method by *generating a set of basis functions from each character in a training set* instead of using predefined basis functions. Furthermore, because our basis functions are not necessarily symmetric about an origin, the concept of a G-centroid is not justified, so we must use a pair of basis functions, ψ_x and ψ_y , and use the moment value as a feature. Thus the ‘more generalized’ generalized moment m_ψ over the image function $f(x, y)$ is defined $m_\psi = f(x, y) \otimes (\psi_x(x) + i\psi_y(y))$.

Using our More Generalized GMF method (MGGMF), a model is defined by selecting one sample from each of the 25 letter shapes. A pair of basis functions ψ_x^k and ψ_y^k is generated for each shape in the model, giving a total of 50 basis functions. The function ψ_x^k is found by regressing an n -degree polynomial in x to the pixels of image $g(x, y)$ of character k interpreted as unit weighted points in an x - y scatterplot, as shown in Fig. 9. Function ψ_y^k is similarly found using image $g(y, x)$ of character k . The justification for this approach lay with the basis polynomial representing a ‘signature’: a representation of the distribution of the mass of the character as functions of x and of y . The fitting method mini-

mizes the squared mean error. Goodness of fit is not really an issue, as the resulting curve is intended as simply a discriminable signature of the shape, and not a faithful copy of the shape.

Given a character, a feature vector of length 25 is found by convolving the character image with each $(\psi_x^k + i \psi_y^k)$ or $k = 1, \dots, 25$. We have experimented with polynomials of degree $n = 6$, and have also experimented with increasing the resolution of the method by finding signatures for the four quarters of the bounding box of each character. This increases the number of values in the feature vector to 100.

3. CLASSIFICATION

Each letter in the alphabet is associated with one or more classes. Some letters are associated with more than one class because their variants are quite different shapes.

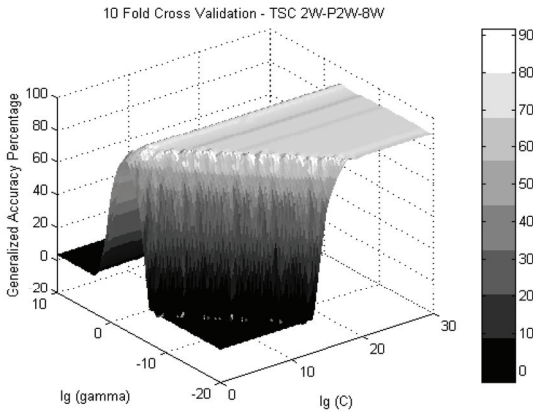


Fig 10. Recognition rate (in percent) obtained with tenfold cross-validation for tabulated values of (C, γ) for the trial FW2-PW2-FW8 on source TS.

For example, the letter *mim* is associated with two classes, one for each variant \mathfrak{P} and \mathfrak{M} . We use the ‘one against one’ approach in which for k classes, $k(k - 1)/2$ classifiers are constructed and each classifier trains data from two different classes. Each classifier is a support vector machine (SVM) [7, 26], in which d -dimensional training vectors are mapped into a sufficiently high dimensional space where linear separation exists. In practice, a separating hyperplane may not exist, for example in cases of high noise level. Therefore, slack variables can be introduced in order to relax classi-

fication constraints at a risk of misclassification. By using a kernel function, it is possible to compute the separating hyperplane without explicitly mapping into the higher dimensional space. We use the radial basis function (RBF) kernel, defined for patterns \mathbf{x}_i and \mathbf{x}_j : $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2)$. Given n training patterns \mathbf{x}_i with associated labels $y_i = \pm 1$, the SVM algorithm solves a dual quadratic optimisation problem to find Lagrange expansion coefficients α that specify the separating hyperplane [20]:

$$\max_{\alpha} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (13)$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C, i = 1, \dots, n \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0. \quad (14)$$

Those patterns whose α_i are non-zero are called support vectors. This leads to the nonlinear decision function (classifier)

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^n y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_j) + b \right). \quad (15)$$

The classifier tends to be very efficient because most α_i become 0, so the support vectors are the only ones needed. The SVM model we use has two parameters: the kernel ‘spread’ $\gamma = 1/(2\sigma^2)$ and the relaxation cost trade-off C . Model selection is performed by enumerating values of the parameter pairs (C, γ) to find the pair that gives the highest cross-validation accuracy for each fold of a 10-fold cross-validation procedure (CV-10). In the CV- m procedure, the samples are randomly divided into m disjoint sets; the classifier is trained m times, each with a different set held out as a validation set. The estimated performance is the mean of these m errors. Figure 10 shows an example of the recognition rate as a function of (C, γ) . Note the ridge along which the highest recognition rates are obtained, suggesting correlation between C and γ .

Feature Set	MSC			TSC			TSW
	28/10	25/10	25/20	28/10	25/10	25/20	99/10
F	61.0			73.0			76.5
FW2	80.5	81.5	90.0	85.0	89.0	94.5	93.5
PW2	80.5	87.5	82.0	93.5	93.5	95.5	86.5
FW8	84.0	84.5	90.5	82.0	84.0	91.5	96.5
PW8	81.5	76.0	74.0	90.5	81.5	87.5	81.0
FS2W8	85.5	84.0	93.5	79.0	84.0	91.5	97.0
PS2W8	86.5	86.5	88.5	91.5	94.0	89.5	97.0
FS4W8	84.0	84.0	93.0	80.0	83.5	92.0	96.0
PS4W8	85.5	86.0	91.5	90.5	92.5	92.5	96.0

Table 1. Results (in percent recognition rate) of 28- and 25-class trials using features of character and word samples. Each column is headed with c/s, indicating the class size c and sample size s per class. Results for the feature set F were considered too unpromising to be included in later trials

Once a satisfactory (C, γ) is found, the one-against-one method is used for training a k -class discrimination problem in which all $k(k-1)/2$ classifiers use the same (C, γ) model.

4. RESULTS

A database of character images was obtained from both MS and TS sources. Character images were size-normalised to 64×64 pixels, and a polar transformed image was also obtained. Several classification trials were carried out, variously using the image, polar image, and regions within the images. To take into account the context-sensitive variations of character shape in Syriac, the model built during the training mode used 28 classes, depending on how the training set was constructed. For example, the variants of the letter *mim*, ܡ and ܡܐ, were assigned different classes during training. Most variants are distinguished by having a longer baseline (e.g. ܡ and ܡܐ), and these variants were assigned to the same class because the segmentation tended to trim the baseline to an approximately uniform length. This study did not use the vowels, as the sources were not vowelised.

The classification trials are identified as follows:

F The five features were obtained from the 64×64 pixel character image, giving a feature vector of length 5.

F/PW2 The character/polar image was divided into 2 non-overlapping windows of 64 rows and 32 columns, and the five features ob-

tained from each window resulting in a feature vector of length 10.

F/PS2W8 The character/polar image was divided into 29 regions of 8×8 pixels overlapped by 6 pixels, and the five features obtained from each window resulting in a feature vector of length 145.

F/PS4W8 The character/polar image was divided into 15 8×8 pixels overlapped by 4 pixels, and the five features obtained from each window resulting in a feature vector of length 75.

Feature Set	MSC 28/10	TSC 28/10
FW2-PW2	88.5	93.5
FW2-FW8	85.5	86.0
FW2-PS2W8	87.0	81.0
FW2-PS4W8	85.5	84.0
FW2-PW2-FW8	91.0	90.5

Table 2. Results (in percent recognition rate) for composite feature vectors. Comparing like trials in Table 4, a composite vector gives the highest recognition rate for the MSC source.

Table 4 shows the results from the trials carried out. Columns MSC and TSC refer to character samples taken from the manuscript source and the typeset source respectively. Under each source are columns showing results for different sample sizes and class sizes. The first trial used ten samples of each character. A second character recognition trial was undertaken using a different association of character shapes to classes. In this trial 25 classes were defined by merging classes having insignificant differences according to the previous trial. A larger sample set used for the third trial was constructed by duplicating the original sample size.

We also evaluated the classifier on a word recognition task, for which character segmentation is unnecessary. Column TSW of the table refers to trials carried out on a sample of 990 word images taken from the typeset source TS. The sample consisted of 10 examples of each of the 99 most frequent words in TS. This trial was done only for comparison to other cursive word recognition studies [6], and the recognition rates are comparable. A relatively high word recognition rate is expected because of the uniform quality of the TS sample and the inherent more pronounced distinctions between word shapes relative to character shapes. A word recognition

trial was not carried out for the MS source because an insufficient sample of each word was available.

Table 2 illustrates character recognition trials in which long feature vectors were generated by concatenating the vectors obtained from previous trials. The trials using concatenated feature vectors, such as FW2-PW2-FW8, show higher recognition rates, possibly because these trials use both the character image and the polar transformed image in the same feature vector, as well as a combination of window sizes. Despite the longer feature vectors for these trials, the peaking phenomenon [11] is not in evidence. With a few exceptions, the recognition rate in Table 4 increases as the number of samples is increased, even if the new samples are simply duplicates. In other trials not shown in the table, the recognition rate reached 100% when the number of samples per character was replicated to 200 (i.e. still only 10 unique samples). This result should be treated with caution because of two sources of bias when sample size is increased. First, because cross-validation constructs the training set essentially by sampling without replacement, it is more likely that the training set of a larger sample size represents more diversity within the sample, even if the proportion held out is unchanged. Second, if the classifier shows poor generalisation, then a small increase in the diversity of the training set might cause a disproportionately higher recognition rate. The cross-validation procedure is designed to limit bias [12], but some combination of these effects may account for an increase in recognition rate in certain trials.

Table 2 illustrates character recognition trials in which long feature vectors were generated by concatenating the vectors obtained from previous trials. The trials using concatenated feature vectors, such as FW2-PW2-FW8, show higher recognition rates, possibly because these trials use both the character image and the polar transformed image in the same feature vector, as well as a combination of window sizes. Despite the longer feature vectors for these trials, the peaking phenomenon [11] is not in evidence. With a few exceptions, the recognition rate in Table 4 increases as the number of samples is increased, even if the new samples are simply duplicates. In other trials not shown in the table, the recognition rate reached 100% when the number of samples per character was replicated to 200 (i.e. still only 10 unique samples). This result should be treated with caution because of two sources of bias when sample size is increased. First, because cross-validation constructs the training set essentially by sampling without replacement, it is more likely that the training set of a larger sample size repre-

sents more diversity within the sample, even if the proportion held out is unchanged. Second, if the classifier shows poor generalisation, then a small increase in the diversity of the training set might cause a disproportionately higher recognition rate. The cross-validation procedure is designed to limit bias [12], but some combination of these effects may account for an increase in recognition rate in certain trials.

We then considered a situation where the classifier was trained on the typeset source TS, then the resulting model used for character recognition on the manuscript source MS (Table 3). The motivation for this was to test the performance of the system on a multi-font problem in which no training data were obtained from the test source. Although classification repeatability is confirmed by the high recognition rate when the model is tested with samples taken solely from the training set, a low rate is shown when the model is tested against samples from the manuscript source. A number of factors may account for this. First, the uniformity of the characters in the TS source provide insufficient variation needed for the model to have good generalization behaviour. Second, there are systematic differences in design between the characters in the MS and TS. In general, the MS characters have a thicker stroke width and a lower width/height ratio. Also, individual characters have slight differences in shape. These factors suggest that the system is unable to treat the TS and MS sources as interchangeable, and that further work will be required to design a system with multi-font capability.

Test Using	Samples	Rate	Samples	Rate
Trained Model	200	100.0	10	97.6
MSC	50	14.0	10	27.0

Table 3. Results of recognition trials on MSC using model obtained from characters from TS source.

Feature	MS	TS
Geometric [16]	90.0	94.5
Hu [10]	89.0	93.0
Legendre [24]	88.5	93.7
MGGMF 6	86.0	87.5
MGGMF 6Q	94.0	98.0

Table 4. Results (in percent recognition rate) of trials using features of character samples from the manuscript source (MS) and typeset source (TS). To provide a basis for comparison, training and recognition was also performed with ten geometric moment features [16], seven Hu features [10], and ten Legendre polynomial features [24]. The MGGMF method used a 6-degree polynomial signature on whole character image; MGGMF 6Q used a 6-degree signature on each of four quarters of the character image. All recognition trials used twenty samples of each character from each source.

The final experiments concern the use of our More General Generalized Moment function, comparing the performance of well-known non-generalized moment functions. Table 4 shows the results from the trials carried out. Columns MS and TS refer to character samples taken from the manuscript source and the typeset source respectively. Under each source are columns showing results for different moment functions. As one might expect, recognition rate is better for the typeset source than the manuscript source, no doubt owing to the regularity of the TS. The performance of the MGGMF method applied to the whole character image suggests that the signature is insufficiently discriminative. However, when signatures are found for each quarter of the character image, a dramatic improvement is noticed. One explanation is that signatures are thereby more closely identified with separate strokes of the character.

5. CONCLUSION

This paper has described a system for recognising cursive Syriac text (Estrangelo) from ancient scribe-written and early modern typeset sources. Given a document, the system finds words and then segments each word into characters. These preliminary stages require some manual intervention to remove editorial apparatus and to correct certain systematic oversegmentations. Each character is then recognized using a trainable classifier constructed using a support vector machine. Recognition rates vary from 61% to 100% based on the method used and the source of text. Some trials may exhibit methodological bias, and these results should be treated with caution. Excluding these, the highest recognition rate on scribe-written manuscript samples, 94%, has been obtained using a the MGGMF 6Q feature vector of length 24. The support vector classifier has been tested using a 10-fold cross-validation procedure, which has provided a high accuracy of classification. Because the number of support vectors is minimised during the training

stage, recognition is more efficient than the Hidden Markov Model classifier used by our previous work on similar sized data sets [6].

It is important to stress that the system described here is at a most preliminary stage of development. It has been a useful laboratory research tool, but is not ready to be used on arbitrary documents, nor may it be conveniently used by people other than the developers. The entire system is essentially ‘knowledge free’ in the sense that no knowledge of characteristic Syriac letter shapes or statistics has been used in the system design. Future work should concentrate on improving the segmentation algorithm, and extending the system to deal with articulation marks and punctuation. Steps can also be taken to improve the robustness of the system on documents that have been badly reproduced. Both these areas of work might benefit from building in knowledge of Syriac from the letter-formation level to the morphological and lexical levels [18]. At the letter-formation level, matching flexible templates might be a productive approach instead of geometric moment functions, and a start in this direction has been recently reported for Arabic [1]. However, that method treats each character as an isolated shape, thus presuming some type of segmentation will have been applied. Finally, because Syriac is written in several forms, it would also be useful to investigate whether the system could be trained and tested equally well on the East Syriac and Serto (West Syriac) forms, as well as font-specific variants within the main script systems.

ACKNOWLEDGMENTS

We thank Chih-Jen Lin of National Taiwan University for assistance in using his LIBSVM library. P.P.J. Fernando is supported by a studentship from the Bishop’s Conference of Sri Lanka. We are grateful to Sebastian Brock of the University of Oxford, Rifaat Ebied of the University of Sydney and George Kiraz of Beth Mardutho: The Syriac Institute, for valuable advice, source manuscripts and encouragement. This paper is an expanded version of [5].

BIBLIOGRAPHY

- Al-Shaher A. and E.R. Hancock. Arabic character recognition with shape mixtures. In Proc. 13th British Machine Vision Conference, Cardiff, Wales, September 2002.
- Arica N. and F.T. Yarman-Vural. Optical character recognition for cursive handwriting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):801–813, 2002.

- Crawford Burkitt, F. *Evangelion Da-Mepharresbe*. Cambridge University Press, 1904.
- Chang, S. and C.P. Grover. Generalized moment functions and conformal transforms. *Proceedings of SPIE*, 4790:102–113, 2002.
- Clocksinn, W.F. and P.P.J. Fernando. Towards automatic recognition of Syriac handwriting. In *Proceedings of the IEEE International Conference on Image Analysis and Processing*, Mantova, Italy, September 2003.
- Clocksinn, W.F. and M. Khorsheed. Word recognition in Arabic handwriting. In *Proc. 8th Int. Conf. on Artificial Intelligence Applications*, pages 271–279, Cairo, February 2000.
- Cortes, C. and V. Vapnik. Support-vector network. *Machine Learning*, 20:273–297, 1995.
- Ebied, R.Y ., A. Van Roey, and L.R. Wickham. *Petri Callinicensis Patriarchae Antiocheni: Tractatus contra Damianum*, volume 32 of *Corpus Christianorum, Series Graeca*. University of Louvain Press, Louvain, 1996.
- Freeman, M.O. and B.E.A. Saleh. “Optical location of centroids of non-overlapping objects.” *Applied Optics*, 26(14):2752–2759, 1987.
- Hu, M.K. “Visual pattern recognition by moment invariants.” *IRE Trans. Information Theory*, IT-8:179–187, 1962.
- Jain, A.K. and B. Chandrasekaran. “Dimension and sample size considerations in pattern recognition practice.” In P.R. Krishnaiah and L.N. Kanal, editors, *Handbook of Statistics*, pages 835–855. North-Holland, Amsterdam, 1982.
- Jain, A.K., R.P.W. Duin, and J. Mao. “Statistical pattern recognition: A review.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1): 4–37, 2000.
- Jain,R., R. Kasturi, and B.G. Schunck. *Machine Vision*. McGraw Hill, New York, 1995.
- Khorsheed, M. and W.F. Clocksinn. “Structural features of cursive Arabic script.” In *Proc. 10th British Machine Vision Conference*, pages 422–431, Nottingham, England, 1999.
- Khorsheed, M. “Off-line Arabic character recognition – a review.” *Pattern Analysis and Applications*, 5(1):31–45, 2002.
- Kim, J.H., K.K. Kim, and C.Y. Suen. “An HMM-MLP hybrid model for cursive script recognition.” *Pattern Analysis and Applications*, 3(4):314–324, 2000.
- Kiraz, G.A. “Syriac morphology: From a linguistic model to a computational implementation.” In R. Lavenant, (ed.), *VII Symposium Syriacum*, Rome, 1996. Orientalia Christiana Analecta.
- Manmatha, R. Chengfeng Han, and E.M. Riseman. Word spotting: A new approach to indexing handwriting. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 631–637, San Francisco, June 1996.
- Müller, Klaus-Robert, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf. “An introduction to kernel-based learning

- algorithms.” *IEEE Transactions on Neural Networks*, 12(2):181–202, 2001.
- Plamondon, R. and S.N. Srihari. “On-line and off-line handwriting recognition: A comprehensive review.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.
- Seni, G. and E. Cohen. “External word segmentation of off-line handwritten text lines.” *Pattern Recognition*, 27(1):41–52, 1994.
- Steinherz, Tal, Ehud Rivlin, and Nathan Intrator. “Offline cursive script word recognition – A survey.” *International Journal on Document Analysis and Recognition*, 2(2/3):90–110, 1999.
- Teague, M.R. “Image analysis via the general theory of moments.” *Journal of the Optical Society of America*, 70(8):375–397, 1980.
- Tistarelli, M. and G. Sandini. On the advantage of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):401–410, 1993.
- Vapnik, V. *Statistical Learning Theory*. Wiley, New York, 1998.