PERGAMON

Pattern Recognition 32 (1999) 2021–2028

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/pattcog

Local reference lines for handwritten phrase recognition

Sriganesh Madhvanath, Venu Govindaraju*

Center of Excellence for Document Analysis and Recognition (CEDAR), Department of Computer Science, State University of New York at Buffalo, Amherst, NY 14228-2567, USA

Received 19 February 1998; received in revised form 9 October 1998; accepted 9 October 1998

Abstract

Reference line information has been used for diverse purposes in handwriting research, including word case classification, OCR, and holistic word recognition. In this paper, we argue that the commonly used global reference lines are inadequate for many handwritten phrase recognition applications. Individual words may be written at different orientations or vertically displaced with respect to one another. A function used to approximate the implicit baseline will not be differentiable or even continuous at some points. We have presented the case for local reference lines and illustrate its successful use in a system that verifies street name phrases in a postal application. © 1999 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Image processing; Chain code; Handwriting recognition; Reference lines; Feature extraction

1. Introduction

Handwritten text in english and other European languages is organized as lines written from left to right. In the absence of guiding lines on the medium, the skilled writer aligns words on an approximately horizontal, implicit "baseline". At the word level, the baseline is one of four *reference lines* that divide the word area into "upper", "middle" and "lower" zones (Fig. 1). These reference lines are present on the medium when instruction in writing is provided at school; the skilled writer uses implicit counterparts of these in freeform writing. Reference line determination is, in one sense, the task of making these implicit reference lines explicit.[1]

Reference line information has been used for diverse purposes in handwriting research [1,2], including word case classification, in OCR to distinguishing letters which differ only in scale or vertical position (e.g., "O" and "o", "P" and "p") and the detection of perceptual features such as ascenders and descenders [3]. They have also been used to provide a general characterization of writing style for author-specific learning [4], image quality assessment and graphology. In this paper we describe many other applications of reference lines, specially those that pertain to holistic word recognition [3], such as, determining the word length, presence of ascenders and descenders, and also segmentation of words into characters. For our purposes, a reference line is a closed-form function $r(x)$ of the column position, and approximates the corresponding implicit reference line. A *global* approach to reference line determination attempts to approximate the implicit reference lines of the entire handwriting sample by a single function $r(x)$. A *local* approach, on the other hand, attempts to approximate local sections of the handwriting sample. The resulting sets of reference lines are known as global and local reference lines respectively.

In this paper we argue that the commonly used global reference lines are inadequate for many handwritten phrase recognition applications. We have presented the case for local reference lines and illustrate its successful use in a system that verifies street name phrases in a postal application.

* Corresponding author.
E-mail address: govind@cedar.buffalo.edu (V. Govindaraju)

[1] The word "line" is used in this context to mean its common connotation, rather than the Euclidean definition. The implicit reference lines and their explicit approximations are not constrained to be "straight" lines.

PII: S0031-3203(98)00166-6

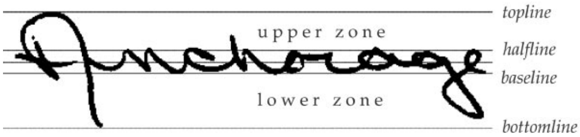

Fig. 1. Reference lines divide the word into three zones.

2. Previous work

A time-tested technique for global reference line determination is based on the vertical histogram of pixels [5]. The method makes the implicit assumption that the word was written horizontally and scanned in without skew. Consequently, the method fails for words with significant baseline skew. Unfortunately, baseline skew is present to varying degrees in most freeform handwriting.

Baseline skew can be corrected by estimating the skew of the word and applying a rotation or shear transformation on the image.

An alternative to skew correction is the use of reference lines of the form $r_i(x) = m \times x + c$, where m is the skew angle and c is the offset of the corresponding reference line from the x-axis. Angled reference lines may be computed from the angular histogram at the skew angle.

Knowledge of baseline skew is a prerequisite for both skew correction and angular reference line determination. In this sense, baseline determination is the central problem in reference line determination. The baseline must therefore be estimated independently, without recourse to the angular histogram.

The method described in this paper is motivated by the fact that the implicit baseline, by definition, is used to align the bases of characters. The bases of characters may in turn be approximated by local minima on the lower outer contour of the word (Fig. 2a). The baseline is estimated as the least-squares regression line through these local minima (Fig. 2b).

The best-fit line through the minima points may be determined by a least-squares linear regression procedure. Minima that do not fall in the vicinity of the implicit baseline either correspond to descenders, or are spurious. Detection and elimination of minima corresponding to descenders and spurious minima (i.e. minima that do not lie on the implicit baseline) prior to regression results in better estimates, and may be accomplished by heuristics based on local spatial constraints. The baseline determined is of the form $r(x) = m \times x + c$ (Fig. 2b).

The halfline may be determined as the regression line through upper contour maxima. However, upper contour maxima are often poorly aligned, and spurious points are difficult to detect and remove, especially in more discrete writing (non-cursive) styles [6]. Consequently, the resulting halfline is often erroneous. Reference lines computed from the angular histogram at the skew angle of the baseline have proved to be more reliable (Fig. 2c).

3. Motivation

Global baselines fail on larger handwriting samples when the implicit baseline cannot be approximated satisfactorily by a straight line. Individual words may be written at different orientations or vertically displaced with respect to one another (Fig. 3). It is clear that a function $r_i(x)$ used to approximate the implicit baseline will not be differentiable or even continuous at some points.

Consider Fig. 4. Given a handwritten phrase image, local contour minima are extracted, the baseline computed and the global reference lines determined using the methods described later in this section. These reference lines are approximate at best.

These issues may be resolved by approximating the implicit baseline by a set of *local* baselines rather than a single global baseline.

Fig. 5a shows the actual results of the global and local baseline techniques for several address images as described in this paper. The local approach determines superior approximations to the implicit baseline compared to the global approach.

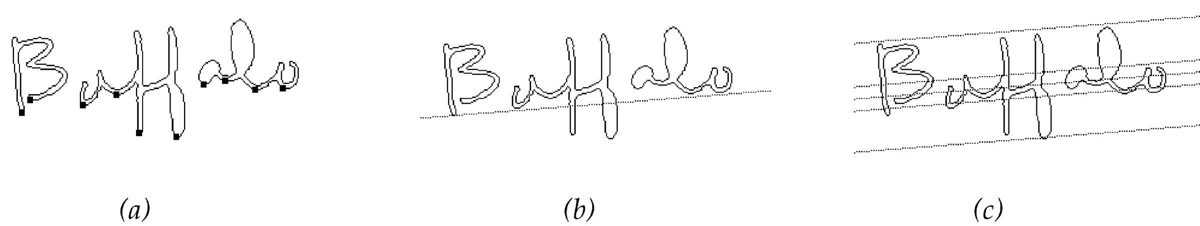

Fig. 2. (a) Exterior contour of word showing lower-contour minima. (b) Baseline determined as regression line through minima. (c) Angular reference lines from angular histogram at skew angle.

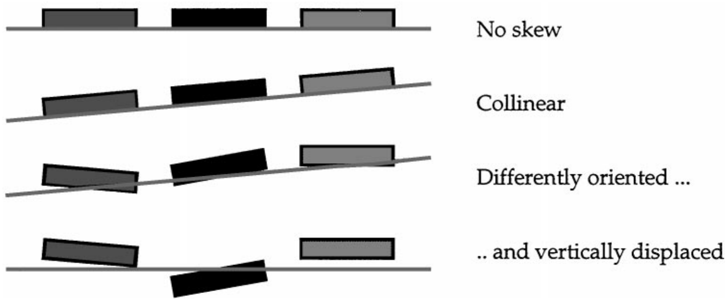

Fig. 3. Individual words in phrases may be written at different orientations and/or vertically displaced.

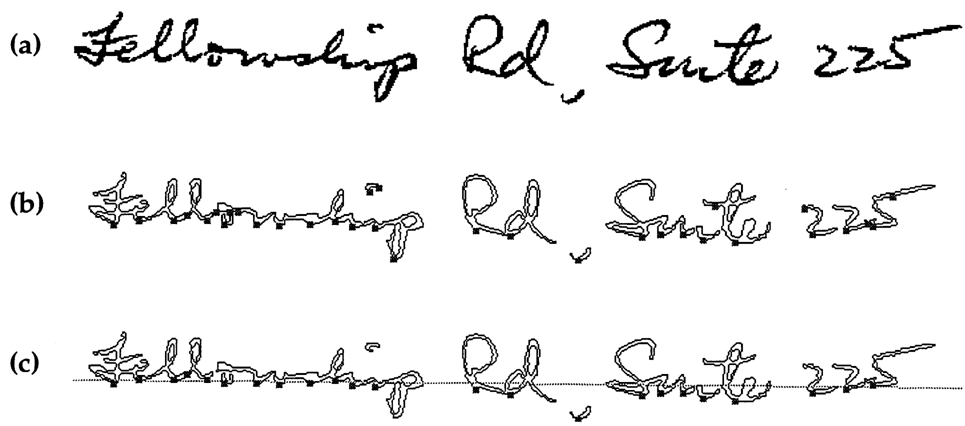

Fig. 4. Global reference lines in handwritten phrases prove inadequate: (a) original image, (b) after contour tracing and detection of local extrema, and (c) computation of global baseline following filtration of spurious minima.

Fig. 5. (a) Global and local baselines for five address lines. (b) Local reference lines from global histogram of contour offsets relative to local baselines.

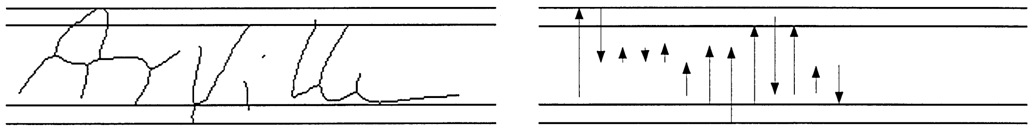

Fig. 6. Upward strokes and downward strokes computed from reference lines for word image *Arville*.

3.1. Reference line applications

Reference line information has been widely used in handwritten word recognition applications for a variety of reasons.

Fig. 6 shows how one particular word recognizer [7] uses the start and end position of strokes relative to the reference lines as features. It matches precomputed feature vectors of lexicon entries with features derived from the image.

Fig. 7 illustrates the use of holistic features of the word contour such as ascenders and descenders to distinguish handwritten words. *Center* and *Spring* are easily distinguishable because while *Spring* has two descenders corresponding to the "p" and the "g", *Center* has none. The presence of ascenders and descenders is detected by first determining the reference lines.

Another holistic feature that has proved useful is the length of the word. In one method of estimating word length, a center line is drawn half way between the base and halflines and the number of times that the script traverses this center line is counted. The word length is estimated as the ratio of this number to a statistic representing the number of traverses of the center line per letter of the average English word [8].

The segmentation of a word into characters is performed by detecting the ligatures between characters in cursively written words. Only those ligatures which lie within the middle zone of the reference lines are considered (see Fig. 8). Thus making the determination of reference lines important in the analytical word recognizers as well [6,9,10].

4. Local reference line determination

An efficient algorithm for the determination of local baselines is presented here. This approach is motivated by the observation that the implicit baseline is locally linear, that is, it may be modelled locally by straight line segments. Local minima of the contour of a phrase (word) image are used as indicators of the position of the implicit baseline. Local minima are clustered based on *Good Continuation* criteria into "local baseline segments" (or simply "segments"), which are then suitably extended to yield a set of local baselines.

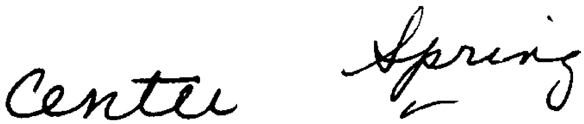

Fig. 7. Simple holistic features such as ascenders and descenders may be used to distinguish handwritten words.

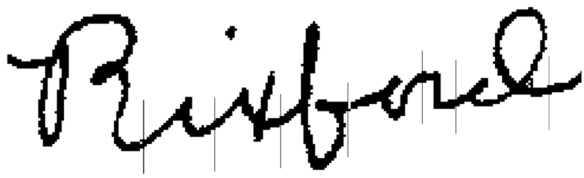

Fig. 8. Segmentation points must be on ligatures which tend to be in the middle zone of the word.

Given the image of a word sequence such as an address line, the algorithm for determination of local baselines is comprised of the following steps:

1. preprocessing to eliminate spurious minima,
2. clustering of minima into baseline segments based on Good Continuation,
3. detection and removal of spurious segments,
4. force-clustering of unlabelled minima,
5. computation of local baselines,

1. *Preprocessing:* Detection of local Y-extrema from the contour of phrases is trivial, and strict alternation of local maxima and minima follows implicitly from the definition of local extrema. The primary challenge is the detection and rejection of spurious and redundant contour extrema arising from irregularities in the contour, discreteness, fragmentation, noise from surrounding text, and "doubling-back" of the contour.

The process of scanning and binarization of offline images sometimes leads to irregularities or "jaggedness" on otherwise smooth stretches of contour, which in turn gives rise to spurious extrema (Fig. 9). Jaggedness is also introduced by the shear transformation used for slant normalization, and cannot be completely eliminated using contour smoothing techniques.

A combination of simple and complex heuristic filters may be used for elimination of such spurious extrema. These include checks on the distance of the detected

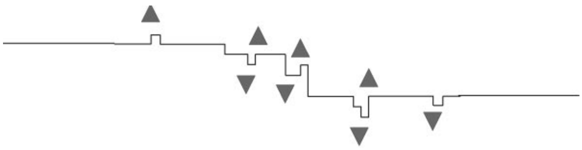

Fig. 9. Spurious extrema resulting from contour irregularities.

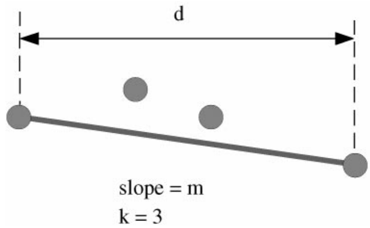

Fig. 10. "Good Continuation" criteria for clustering.

extremum from previous extrema detected along the contour, spatial postion relative to other extrema, the slope of the contour in the neighborhood of the extremum, and so forth.

2. *Clustering minima:* Clustering is driven by the Gestalt principle of Good Continuation, which is expressed in terms of two conditions for two minima that are k-apart to belong to the same segment (Fig. 10).

(a) *Proximity condition* ($C_P(k)$):

$$d < T_P(k).$$

Two minima that are k-apart are said to satisfy $C_P(k)$ if the x-separation d between them is within a certain threshold $T_P(k)$. $T_P(k)$ is a function of both k and the mean x-separation between consecutive points.

(b) *Slope condition* ($C_S(m_0)$):

$$\frac{(m - m_0)}{1s + - m \,.\, m_0} < T_S(m_0).$$

Two minima are said to satisfy $C_S(m_0)$ if the slope m of the line segment joining them is equal to a specified slope m_0 within limits. The maximum deviation allowed from m_0 is expressed as a threshold T_S on the tangent of the angular difference between the two slopes.

Clustering is initiated from the leftmost minimum p_0. A new segment S is initiated when in a left-to-right scan of the minima, a pair (p_i, p_{i+k}), $0 < k < = K$ satisfying both $C_P(k)$ and $C_S(m_g)$ is encountered, where m_g is the slope of the global baseline (Fig. 11).

Once a segment S is initiated, the indices of the first and last minima in the segment, denoted respectively by *S.first* and *S.last*, are maintained and suitably updated whenever a point is added to the segment. The segment is expanded to the right by adding points that satisfy $C_P(k)$ and $C_S(m_g)$ with respect to the current last point *S.last*.

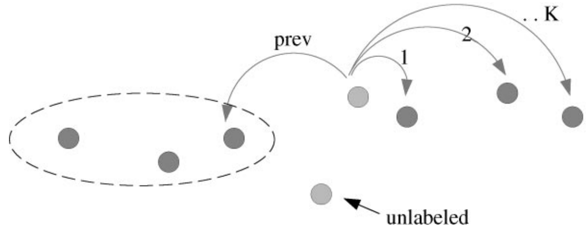

Fig. 11. Segments are grown to the right by considering up to k minima from the current right end.

When a segment cannot be grown any longer, the search for a new segment is initiated from $S.last + 1$. This process is continued until all p_i have been scanned.

Clustering Algorithm Psuedocode

```
foreach i in [0, ..., n − 1] do
  if GoodCont(S.last, p_i) then
      add_point_to_segment(p_i, S)
      mark_active_member(p_i, S)
      S.last = p_i
  else
      Done = FALSE
      foreach k in [1, ..., K] and not Done do
          if GoodCont(p_i, p_{i+k}) then
              S = new_segment(p_i, p_{i+k})
              mark_active_member(p_i, S)
              mark_active_member(p_{i+k}, S)
              S.first = p_i
              S.last = p_{i+k}
              foreach j in [i + 1, ..., i + k − 1] do
                  mark_inactive_member(p_j, S)
              endfor
              Done = TRUE
          endif
      endfor
  endif
endfor
```

K is a crucial clustering parameter that allows invalid minima to be "skipped over" during clustering. Clustering is thus achieved by a greedy strategy in a single pass over the minima. This is highly desirable from an efficiency point of view.

3. *Detecting and removing spurious minima:* Spurious segments can result when spurious points occur together and are grouped (as can happen when there is heavy fragmentation) or when valid points are skipped. These segments tend to be small (typically two or three points) and are vertically displaced with respect to the true implicit baseline. The frequency of occurrence of spurious segments depends on the success achieved in the removal of spurious points during preprocessing. These segments can usually be detected by using local spatial constraints at the segment level, in a single pass over the segments.

4. *Forced clustering:* In this pass, valid points that remain unlabelled after the first clustering pass are reclaimed by dropping the proximity criterion. Unlabelled points to the left of the first segment S_0 are assigned to the first segment. Each point is labelled active or inactive depending on their satisfying the slope criterion $C_S(m_0)$. The proximity condition is not applied. Similarly, the unlabeled points to the right of the last segment are assigned to the last segment. Unlabelled points in between segments are assigned to one or the other segments based on a decision tree that determines the optimal partition.

5. *Local reference line*: The local baseline corresponding to segment S_j is computed by minimum-square linear regression on the active points of S_j. The horizontal span of the baselines (start and ending columns) are determined such that every column in which there is a black pixel is covered by at least one local baseline.

5. Experimental results

The domain under consideration (handwritten addresses in the US mailstream) presents a new twist to the problem of word recognition. In the central part of the address interpretation process lies the task of reading street names [1]. Street names almost always are phrases comprising of two to four words. This makes it more likely for writers to have a non-collinear baselines for the words in a phrase.

Furthermore, addresses are often carelessly written. A variety of writing styles from cursive to touching discrete to discrete and mixtures of these are represented. The images are poor in quality, and on occasion exhibit a high degree of fragmentation from suboptimal thresholding. Individual lines may contain fragments from the lines above and below as a consequence of splitting lines during address line separation.

5.1. Background on phrase verification

We use the term "phrase verification" to refer to the task of verifying that a given image of a phrase is that of a given ASCII string, or one of a given set of ASCII strings. While phrase recognition is a classification task with as many possible responses as there are phrase classes, verification is the task of deciding whether or not a pattern could belong to a given phrase class.

The context of interpretation of handwritten addresses (HWAI) is the prime motivation for the development of a fast phrase verification system based on the holistic paradigm. A street name recognized with confidence in the "gray-area" is submitted to a holistic verifier (Fig. 12). The verification must be rapid to meet the real-time requirements. Holistic features are a logical choice for this purpose both because of their orthogonality to the features used by the analytical classifiers, and the speed of feature extraction and matching they make possible.

The salient features of the phrase verifier are the following.

1. *Chain-code based image processing:* Preprocessing operations such as slant normalization and feature extraction tasks such as detection of local extrema and determination of reference lines are implemented using a chain-coded representation of the binary image for greater computational efficiency.
2. *Lexicon-driven detection of features:* In lieu of purely bottom-up, image-driven detection of positional features such as inter-word gaps, ascenders and descenders, multiple candidates are extracted from the image and the predicted features of the verification string are used to determine the true image features among the extracted candidates.
3. *Hierarchical matching:* A dynamic programming algorithm is used to match positional features such as gaps, ascenders and descenders of the verification string with the corresponding candidates from the image, the object being to determine the best match for the predicted lexicon features from among the extracted image feature candidates.

Thus local reference lines are an integral part of the system to verify street name phrases with ASCII lexicon

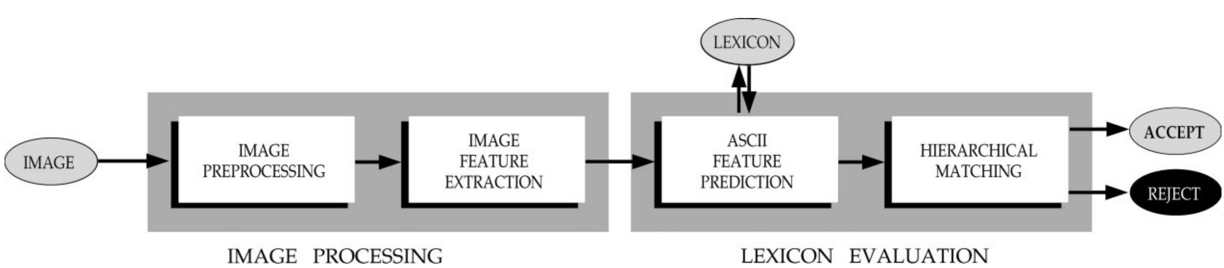

Fig. 12. Overview of the Phrase Verification System.

Red Oak Drive
R Oak Drive
Red Oak Dr
R Oak Dr
Red Oak
R Oak

(a) (b)

Fig. 13. Verification task inputs: (a) a street name phrase image and (b) expanded lexicon entries for the image; Output is a Accept or Reject decision.

entries. The system extracted the length of phrases, and presence and prominence of ascenders and descenders.

The task of the system is that of verifying that a given image of a word or phrase is that of a given ASCII string (or one of a given set of ASCII strings generated on a variety of possible variations in which the phrase can appear), frequently the result of another recognition algorithm.

Fig. 13 describes the complexity of the process.

5.2. Results

Training and test data sets of street name images were extracted from live mailpiece images. The street name images are essentially unconstrained with respect to writing style, and contain artifacts such as noise and fragments resulting from suboptimal binarization.

Table 1
Accept and error rates with and without verification

System	Total	Verified	Error
No verification	1041	1041 (100%)	194 (18.6%)
Phrase verification system 1	1041	335 (32.2%)	5 (1.5%)
Phrase verification system 2	1041	393 (37.8%)	9 (2.3%)

Table 2
Mean execution time per image for unoptimized code running on a 150 MHz SUN SPARC 10

System	Image processing (ms)	Lexicon evaluation (ms)	Total (ms)
Phrase verification system 1	11.2	9.5	20.7

A set of 2428 street name images were used for training. The single lexicon associated with each image was correct in 1693 (81.5%) cases, and is in error in 385 (18.5%) cases.

The test set comprised of 1041 street name images with corresponding singleton lexicons. The proportions of correct and error cases in the test set are similar to those in the training set.

The test-set accuracy of two versions of the system, obtained from different choices of thresholds, is tabulated in Table 1. The error rate is defined as the fraction of erroneous verifications amongst the verified images. The mean execution time per image for image processing and lexicon evaluation is shown in Table 2.

6. Summary

We have presented a method for finding local reference lines in handwritten phrases. We have demonstrated the need for such reference lines as well as the effectiveness of our method by experiments on postal applications.

References

[1] G. Kim, V. Govindaraju, Handwritten phrase recognition as applied to street name images, Pattern Recognition 31 (1) (1998) 41–51.
[2] G. Kim, V. Govindaraju, Bankcheck recognition using cross validation between legal and courtesy amounts, Int. J. Pattern Recognition Artificial Intell. 11 (4) (1997) 657–674.
[3] S. Madhvanath, V. Govindaraju, Holistic word recognition. Int. Workshop on Frontiers in Handwriting Recognition, 1993.
[4] C.C. Tappert, C. Suen, T. Wakahara, The state-of-the-art in on-line handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. 12 (8) (1990) 787–808.
[5] L.D. Ernest, Machine recognition of cursive script, Proc. IFIP Congress, Vol. 62, 1992, 462–466.
[6] G. Kim, V. Govindaraju, A lexicon-driven approach to handwritten word recognition for real-time applications, IEEE Trans Pattern Anal. Mach. Intell. 19 (4) (1997) 366–379.
[7] V. Govindaraju, R.K. Krishnamurthy, Holsitic handwritten word recognition using temporal features derived from off-line images, Pattern Recognition Lett. 17 (1996) 537–540.
[8] E.R. Brocklehurst, P.D. Kenward, Preprocessing for cursive script recognition, NPL Report, 1988.
[9] F. Kimura, M. Sridhar, N. Narsimhamurthi, Lexicon directed segmentation – recognition procedure for unconstrained handwritten words, 3rd Int. Workshop on Frontiers in Handwriting Recognition, 1993, pp. 122–131.
[10] J. Simon, Off-line cursive word recognition, Proc. IEEE 80 (7) (1992) 1150–1161.

About the Author—SRIGANESH MADHVANATH received his B.Tech. in Computer Science from the Indian Institute of Technology, Bombay, India 1990. He obtained his M.S. in Computer Science in 1993 and Ph.D. in 1997 from the State University of New York at Buffalo. He worked as a Research Assistant at the Center of Excellence for Document Analysis and Recognition (CEDAR) from 1991 and 1996, and is presently with the Document Analysis and Recognition group at IBM Almaden Research Center, San Jose, USA. His research interests include document analysis and understanding, handwritten word recognition, and classifier decision combination.

About the Author—VENU GOVINDARAJU is the Associate Director of the Center of Excellence for Document Analysis and Recognition at SUNY Buffalo. He is the project manager and senior research scientist for the CEDAR Handwritten Address Interpretation Project. He holds the Research Associate Professorship in the Department of Computer Science at SUNY Buffalo. Govindaraju received his B.Tech. (Hons.) in Computer Science from the Indian Institute of Technology at Kharagpur in 1986, and his M.S. and Ph.D. in Computer Science from SUNY Buffalo in 1988 and 1992, respectively.

Dr. Govindaraju has co-authored more than 70 research papers in various conferences and journals in the fields pattern recognition and computer vision.

Dr. Govindaraju is currently the associate editor of The Pattern Recognition Journal and Chair of the IEEE SMC subcommittee on pattern recognition.