# Holistic Verification of Handwritten Phrases

Sriganesh Madhvanath, *Member*, *IEEE*, Evelyn Kleinberg, and Venu Govindaraju, *Senior Member*, *IEEE*

**Abstract**—In this paper, we describe a system for rapid verification of unconstrained off-line handwritten phrases using perceptual holistic features of the handwritten phrase image. The system is used to verify handwritten street names automatically extracted from live U.S. mail against recognition results of analytical classifiers. Presented with a binary image of a street name and an ASCII street name, holistic features (reference lines, large gaps and local contour extrema) of the street name hypothesis are "predicted" from the expected features of the constituent characters using heuristic rules. A dynamic programming algorithm is used to match the predicted features with the extracted image features. Classes of holistic features are matched sequentially in increasing order of cost, allowing an ACCEPT/REJECT decision to be arrived at in a time-efficient manner. The system rejects errors with 98 percent accuracy at the 30 percent accept level, while consuming approximately 20/msec per image on the average on a 150 MHz SPARC 10.

**Index Terms**—Word verification, holistic approaches, word shape matching, handwritten word recognition, address interpretation.

## 1 Introduction

Handwritten Word Recognition (HWR), also called *Isolated* Handwritten Word Recognition, deals with the problem of machine reading of handwritten words. The location and segmentation of handwritten words from their surroundings is a complex task for most real applications (reading legal amounts on bank checks, responses on forms, and addresses on mail pieces), and a research problem in its own right. Most work on isolated HWR assumes that the handwritten word has been segmented by an algorithm appropriate to the application domain prior to being presented to the HWR algorithm.

Approaches to the task of HWR have traditionally been classified as *analytical* and *holistic*. The *analytical* approach treats a word as a collection of simpler subunits such as characters and proceeds by segmenting the word into these units, identifying the units and building a word-level interpretation using the lexicon. The *word-based* or *holistic* approach, on the other hand, treats the word as a single, indivisible entity and attempts to recognize it using features of the word as whole. The latter approach is inspired by psychological studies of human reading [1], [2] which indicate that humans use features of word shape such as *length, ascenders,* and *descenders* in reading.

In this paper, we use the term "word verification" to refer to the task 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. While word recognition is a classification task with as many possible responses as there are word classes, verification is the task of deciding whether or not a pattern could belong to a given word class. Independent verification is useful in overcoming limitations of a particular set of features or HWR algorithm, in terms of spotting errors and improving robustness with respect to a variety of writing styles.

Section 2 describes some related work in the area of using holistic features and verification. In Section 3, we discuss some challenges to the purely shape-based verification of U.S. street names. In Section 4, we describe HOVER, a system for rapid holistic verification of street name images using coarse shape features.

Section 5 describes the evaluation of the system. A summary of this effort and a plan for future work is presented in Section 6.

### 1.1 Background

Address blocks classified as handwritten are processed by a Handwritten Address Interpretation (HWAI) system. After the HWAI identifies and recognizes a five-digit zip code and the street number in a delivery line, street name recognition is required to extend the zip code to a Delivery Point Code (i.e., nine digits). The image of the street name is obtained by extracting the part of the street line to the right of the street number. Postal directories are queried to obtain all possible street names corresponding to the specific zip code and street number pair, and the resulting set of street names serves as the lexicon for the recognition task. Each street name in the lexicon comes in a standard form, and tagged with the corresponding four-digit add-on, which when appended to the ZIP constitutes the encode string for the address. Each street name is expanded into a restricted number of common variants that are encountered in practice, all of which are tagged with the original add-on.

Word Recognition Control (WRC) is part of HWAI. It is responsible for directing the recognition of street name images, and arriving at an ACCEPT/REJECT decision about a mail piece based on recognition confidences [3]. An ACCEPT decision is accompanied by the add-on corresponding to the recognized street name, as shown in

- *S. Madhvanath is with the IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120. E-mail: srig@almaden.ibm.com.*
- *E. Kleinberg and V. Govindaraju are with the Center for Document Analysis and Recognition, Department of Computer Science, State University of New York, Buffalo, NY 14228. E-mail: {govind,evie}@cedar.buffalo.edu.*

*Manuscript received 16 Oct. 1998; revised 1 Oct. 1999.*
*Recommended for acceptance by R. Plamondon.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 108057.*

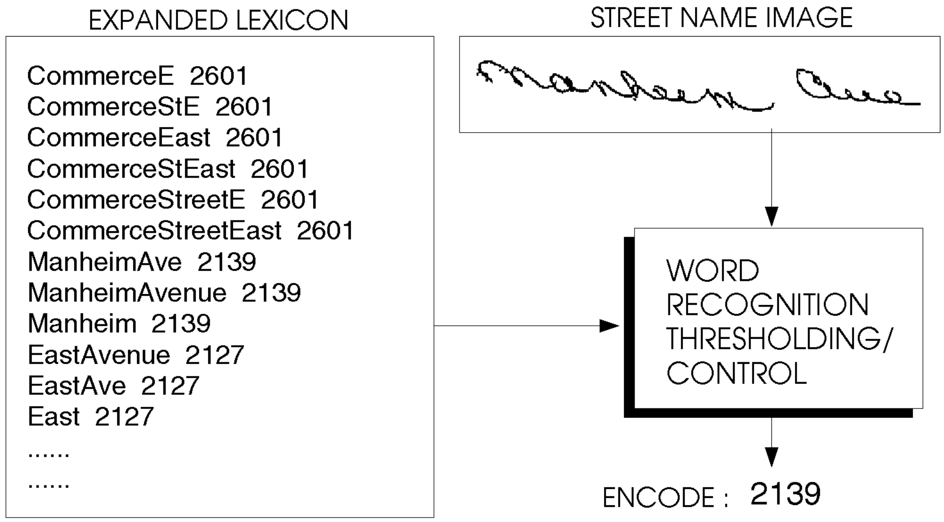



Fig. 1. Input-output behavior of Word Recognition Thresholding-Control (WRTC) module.

Fig. 1. It employs two handwritten word classifiers WMR [4] and CMR [5]. Each classifier takes as input the binary image of the street name and the expanded lexicon of street name candidates, computes a confidence for each street name candidate, and ranks the lexicon by decreasing confidence. Both classifiers are based on the *analytical* paradigm of over-segmentation of the given binary word image into segments, and use of OCR to determine the identities of groups of segments. However, they differ in the segmentation algorithm, features used by OCR, and the extent to which the lexicon is integrated into the recognition process. An error is defined in the context of this task as the event that the add-on associated with the top choice of a classifier is not the correct add-on for the mail piece.

### 1.2 Motivation

The context of interpretation of handwritten addresses (HWAI) is the prime motivation for the development of a fast word verification system based on the holistic paradigm. Streets recognized with high confidence by either classifier are accepted and those with poor confidences are rejected by WRC. Street names recognized with confidences in the "gray-area" of both classifiers are passed to a decision combination stage which looks for agreement between the two classifiers and accepts the agreed upon top-ranked street name if their combined confidence exceeds a threshold $T_{acc}$. In general, when neither WMR nor CMR is very confident, but both agree on the top-ranked street, it is more likely than not that the street is correctly recognized. Experiments indicate that nearly 84 percent of the cases of agreement rejected in the decision combination stage due to low classifier confidences are, in fact, correctly classified. A word verification system may be used to salvage some of the correctly classified cases from the pool of "agreement-rejects" by verifying that the street name image is that of the agreed upon street name.

Clearly, the correctly classified cases are not separable from the error cases based on CMR and WMR confidences, and the verifier must use a different set of features. The verification must also be rapid to meet the real-time requirements for the HWAI task. 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.

We have used holistic methods as a lexicon reduction step and analytical recognizers as the second stage [6]. However, such a strategy is more suitable in applications such as recognition of prose where the initial lexicon size is very large ($\geq 1,000$). The HWAI application presents an average lexicon size of 10. Hence, we choose to use the holistic paradigm to verify the choice(s) made by the analytical recognizers. We do use a way of weighting the recognizers by the choice of the confidence thresholds. The more reliable recognizer's results are accepted at a lower threshold. This is indirectly enforcing a weighting scheme. In our application, on average, WMR is the most reliable followed by CMR and the holistic recognizer (HOVER).

## 2 Previous Work

Moreau [7] extracted vertical and horizontal strokes, loops, i-dots, and t-bars from the offline image to obtain a string descriptor, and compared the descriptor with unique prototypes of words found in French check amounts and their more common orthographic deviations using Dynamic Programming.

Leroux et al. [8] extracted ascenders, descenders, loops, i-dots, and unattached t-bars from the contours of connected components, and obtained a string descriptor. Word length was estimated as the number of letter segments obtained as a by-product of a separate analytical subsystem. The Levenshtein metric was used to compare the test string with reference strings obtained from training corresponding to a small lexicon of check amounts.

Camillerapp et al. [9] labeled singular vertices (end-points, crossings, and points of local curvature) in the skeletonized gray-level image and obtained a tree of stroke primitives. Each tree node was described by the type of primitive, vertical word zone position, and its relative horizontal position within the word. Each lexicon word was coded as a similar tree of primitives, except that each node

TABLE 1
Examples of Nontextual Differences Between Image Contents and Verification String

| Category | Image contents | Standard form |
|---|---|---|
| Punctuation | City Line Ave, | City Line Ave |
| | W. Oakland Pl. | W Oakland Pl |
| | Hurley's Lane | Hurleys Lane |
| Word Splitting | Greenmount Ave | Green Mount Ave |
| | Sweetbriar Court | Sweet Briar Court |
| Case | n.w. 43 Court | NW 43 Court |
| | TURKEYFOOT RD. | Turkeyfoot Rd |
| | HAZElNUt Ct. | Hazelnut Ct |
| | McPherson ave | Mcpherson St |

could describe a set of primitives covering variations that may be expected at that point. The presence of extensions and a length estimate were used to reject irrelevant candidates. A similarity score was using a two-dimensional adaptation of edit distance with only insertions and deletions. In addition, a dissimilarity score was computed based on the primitives present in the word but forbidden in the model. The likelihood of a word model was computed as the difference between the similarity and dissimilarity scores.

Dodel and Shinghal [10] describe a hybrid analytical-holistic method for offline words which uses a decision tree (akin to '20-questions') to identify the correct class from a static lexicon of 31 words. Aspect ratio (horizontal extent/mid zone width), and relative positions of ascenders and descenders are used to achieve direct recognition of some words such as 'Eight,' and partial recognition of others.

Holistic features have been used for the verification of handwritten British postal addresses [11], [12]. The authors report that for some choice of thresholds, the verification procedure allows 43 percent of the postcodes to be read with an error rate of 1.5 percent. Because the object of the system is verification of the postcode using the rest of the address, the authors have not attempted to quantify the success of the system at verifying individual words in the address. The execution time per address of the verification process is also not discussed.

The term "verification" is encountered most frequently in the context of signatures [13]. Signature verification is markedly different from the application we have outlined in at least two respects: 1) The classes are static, and usually a large number of exemplars of each class may be assumed available, and 2) there is very little variability in the appearance of the handwritten signal compared to the general case of omni-scriptor, unconstrained handwritten words.

Recently, there has been some investigation into the task of spotting keywords in handwritten documents [14]. These approaches are typically based on global word shape and require models of the keywords to be constructed from training. This naturally limits the keywords that may be searched for to those with sufficient presence in the data set. The task of verification as we have described it, on the other hand, works by synthetically constructing word models from character models, since the "keyword" (phrase to be verified) is not known a priori.

Clearly, our task calls for the application of the holistic paradigm to a *dynamic lexicon scenario* [15], wherein the lexicon is not known a priori. Each zip code and street number pair presents a unique lexicon. Thus, each street name image is linked with a unique lexicon. As the zip code and street numbers vary among mail pieces, so do the lexicons. Such dynamic and large lexicon scenarios represent a challenge for holistic approaches since traditional training of a classifier from a large population of sample images of each word class is not possible. In [16], we developed a methodology of coarse holistic features and heuristic prediction of such features from ASCII that allows the application of holistic features to large and dynamic lexicon scenarios.

## 3 Challenging Issues

The name of a street as it appears on the mail pieces is often a variant of the standard form present in the postal directories. In order to use holistic features for verification of street names, it is desirable that the particular variant of the street name passed to the verifier correspond exactly to the manifestation on the mail piece. The differences may be classified as *nontextual* and *textual*.

Nontextual differences result primarily from the lack of a standard regarding the writing of street names, and may be further classified as differences in case, punctuation, and word splitting. Punctuation such as periods, commas, and apostrophes are sufficiently small relative to the text components to be discarded in a preprocessing step.

TABLE 2
Examples of Textual Differences Between Image Contents and Verification String

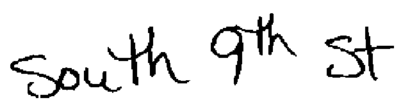
| **Category** | **Street image** | **Standard form** |
|---|---|---|
| Segmentation (left) | 52 Judith Drive | Judith Drive |
| Segmentation (right) | S. 4 st #e | S Fourth |
| | Sequoia Ct #15 | Sequoia Court |
| | THICKET #102 | Thicket Lane |
| Typographical | HARTWOOD DR | Heartwood Dr |
| | CONNETICUT AVE, N | Connecticut Ave NW |
| | West County Lane RD | West County Line Rd |
| | Sweetbrior Court | Sweet Briar Court |
| | Forest Ave | Forrest Ave |
| Missed directionals | CONNETICUT AVE, N | Connecticut Ave NW |
| | Merrimac Road | N Merrimac Road |
| Spurious directionals | N. Oceana Blvd. | Oceana Blvd |
| | Canterbury ne | Canterbury Lane |
| Transposed directionals | W. Marlton Pike | Marlton Pike W |
| Missed suffixes | Lincoln | Lincoln St |
| Suffix | Roosevelt Avenue | Roosevelt Blvd |
| | Dungan St | Dungan Rd |
| Uncommon expansions | no.Main St | North Main |
| | So Poplar St | S Poplar St |
| | GODDARD BULVD | Goddard Blvd |
| | Innisbrook la. | Innisbrook Ln |
| | S. West End Bl | S West End Blvd |
| | DELRAN PKWY | Delran Pky |

However, word breaks and differences in case greatly influence the overall shape of the phrase. Typographical errors are tolerated by holistic approaches to the extent that they preserve word shape. Examples of nontextual differences are shown in Table 1.

Textual differences spring from various sources: System errors in automatic segmentation of the street name from the mail piece, patron errors in directionals and suffix, typographical differences, and so on (Table 2).

Since the image of the street name is presently obtained by identifying the street number and extracting the portion of the street line to its right, secondary information such as suite and apartment numbers are often erroneously included in the street name. Errors in location of the street number may cause a portion of the street number to be included in the street name image, or vice versa.

Typographical differences stem not so much from carelessness on the part of the patron as from phonetically equivalent ways of spelling proper names. Missing and spurious directionals are indicative of differences between the common and correct names of streets. Transpositions of directionals are, strictly speaking, more related to form than content, but are included here since they alter the sequence of characters in the street name image. Common suffixes such as "Avenue" are frequently substituted by other common suffixes such as "Drive." Finally, in order to restrict the size of the expanded lexicon, only a limited number of common expansions of street prefixes and suffixes are applied during the expansion process, and "uncommon" expansions are another source of textual differences between image contents and lexicon string.



(a)

| | |
|---|---|
| S9thSt | SNinthStreet |
| South9thSt | SouthNinthStreet |
| 9thSt | NinthStreet |
| SNinthSt | S9th |
| SouthNinthSt | South9th |
| NinthSt | 9th |
| S9thStreet | SNinth |
| South9thStreet | SouthNinth |
| 9thStreet | Ninth |

(b)

Fig. 2. (a) street name image and (b) expanded lexicon for the image.

Many of the images differ from the officially valid form of the street name in more than one way. For example, "Canal street" differs from "Canal Ave SW" in case, and features suffix substitution, and missed directionals. Missed suffixes and predirectionals are so common that the lexicon is expanded to include these as street name variants. Fig. 2 shows the variants of "South 9th St" that are included in the expanded lexicon to account for the patron's errors and form of writing. In spite of this, a word recognizer, such as WMR, may pick a street name variant with a suffix over the variant without the suffix, typically when there is some noise or secondary information following the street name. For example, missing suffix combined with secondary information ("THICKET #102") causes the word recognizer to select the variant with a suffix ("Thicket Lane"). Secondary information following the street name ("E. Kings Hwy Suite 2," "S. 4 st #e") causes the word recognizer to select a variant with a longer suffix or middle part ("E. Kings Highway," "S Fourth").

The performance of a holistic verification scheme is limited by both textual and nontextual mismatches between the street name image and the variant of the street name presented for verification. The former is influenced by the patron and the segmentation procedure employed by the system. In the application of HWAI, the verification string is the output of an analytical classifier, the segmented street name image, and idiosyncrasies of the classifier. Our experiments have shown that mismatches between the form and content of the image and verification string limit the fraction of correctly classified cases that may be verified using shape features to approximately 60 percent.

## 4 Methodology

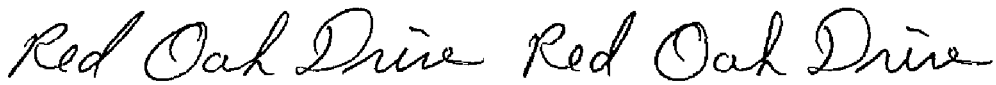
In this section, we introduce HOVER, a system for rapid HOlistic VERification of handwritten phrases such as street name images using coarse word shape features. The input to HOVER is a binary image of a handwritten phrase and a *verification lexicon* containing one or more ASCII verification strings. The objective of the system is to verify that the image is that of any one of the verification strings using coarse word shape features.

In our application, the input image is that of a handwritten street name as extracted by the HWAI system, and the lexicon is comprised of the street name variants of the concurrent highest ranked street from WMR. The design of HOVER, however, is independent of application domain. We will use the term "lexicon phrase" interchangeably with verification string to refer to strings from the verification lexicon.

The major components in HOVER are described in detail in the sections that follow. Processing is composed of a bottom-up *feature extraction* followed by a top-down *evaluation phase*.

The salient features of HOVER 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 [17].
2. *Lexicon-driven detection of features:* In lieu of purely bottom-up, image-driven detection of positional features such as interword 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. This is analogous to a lexicon-driven "oversegmentation" strategy of using elastic matching to select the best alignment of segmentation points corresponding to a lexicon string [4].
3. *Variable feature grid:* Positions of features such as gaps, ascenders, and descenders are computed with respect to a variable grid determined by the reference lines and lower contour minima [18]. The positions are real-valued and expressed in terms of *Holistic Segment Distance*.
4. *Heuristic prediction of lexicon features:* Features of the verification string are predicted from ASCII using coarse models of "ideal" characters.
5. *Hierarchical evaluation:* Lexicon evaluation is organized into stages. The sequence of features used is:



(a) (b)

Fig. 3. Phrase image: (a) Original; (b) Following slant normalization.

1) length, 2) gaps, 3) ascenders, and 4) descenders. Each stage is composed of a feature matching step and a decision step enabling rapid rejection of poor matches.

6. *Elastic 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.
7. *Models of positional distortion:* Models of how ideal shape features are distorted in practice are implicitly employed to determine the best match between the ideal lexicon features and feature candidates from the image.
8. *Attributes of writing style:* Style parameters computed during the matching process are used to compensate for particular writing styles by rescaling confidence measures prior to thresholding.
9. *Training:* Confidences for positional features such as gaps, ascenders, and descenders are computed from logistic models obtained using Logistic Regression from manually truthed training samples.

## 4.1 Preprocessing

The input image is converted to a contour representation and normalized with respect to slant. Baseline skew is computed and reference lines determined at the skew angle. Gaps, local maxima on the upper contours of components, and minima on the lower contours are identified. Confidences associated with these features are designed to reflect the likelihood that they are truly word gaps, ascenders, and descenders, respectively. Hard decisions regarding the identity of the features extracted from the image are postponed to the evaluation phase, where they may be made reliably in a lexicon-driven manner.

The binary image is scanned from top to bottom and right to left, and transitions from white (background) to black (foreground) are detected. The contour is then traced counterclockwise (clockwise for interior contours) and expressed as an array of contour elements. Each contour element represents a pixel on the contour and contains fields for the x,y coordinates of the pixel, the slope or direction of the contour *into* the pixel, and auxiliary information such as curvature.

### *4.1.1 Noise Removal*

Thresholds on minimum component area and dimensions are used to discard small connected components corresponding to salt and pepper noise during the process of chain code generation. In this step, an attempt is made to detect larger, extraneous components corresponding to dots of i's and punctuation marks such as periods and commas, using their size and position relative to the reference lines.

### *4.1.2 Slant Normalization*

Contours of connected components are traversed and the mean character slant is computed from relatively vertical stretches of contour. Normalization is performed by means of a shear transformation applied to the contour elements. Discontinuities introduced into the contours as a result of the transformation are mended and the contours are smoothed [17], as shown in Fig. 3.

## 4.2 Feature Extraction

This section describes the detection of coarse holistic features suitable for the purpose of verification of word and phrase strings.

### *4.2.1 Determination of Upper and Lower Contours*

It is useful to divide an exterior contour into an upper and lower part since it helps localize features such as ascenders and descenders. The segmentation of an exterior contour into upper and lower contours involves the detection of two "turnover" points on the contour—the points at which the lower contour changes to upper contour and vice versa [19]. The "true" turnover points are known only when the word is written cursively with leading and end ligatures—the left end of the leading ligature and the right end of the ending ligature then constitute the turnover points. For mixed-style writing, turnover points may be approximated by the left and right extremes of the contour within the middle zone, and can be computed by a single traversal of the contours (Fig. 4).

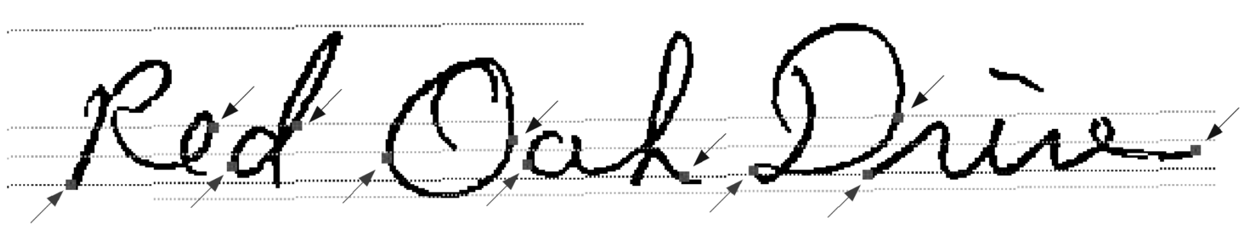


Fig. 4. Turnover points segment exterior contours into upper and lower contours. Here, they are approximated by the left and right extremes of the contour within the middle zone.

Fig. 5. Angular reference lines at estimated baseline skew: The center-line bisects the zone bounded by the baseline and the halfline.

#### 4.2.2 Estimation of Reference Lines

The technique for reference line estimation calls for an initial estimate of baseline skew. Local minima from the lower contours of words may be used as an indication of the implicit baseline. These are extracted from the sections of exterior contours demarcated by the left and right extremes and the standard deviation of their row coordinate is used to reject minima corresponding to descenders. Baseline skew is estimated by linear regression over the remaining minima.

Given the estimated skew $m_b$, intersections of image contours with lines of the form $y = m_b x + c$ are accumulated for different values of offset $c$, from a single traversal of the contours. The resulting *angular histogram* of contour crossings is used to determine the baseline, halfline, and top and bottom lines [19], as shown in Fig. 5.

#### 4.2.3 Determination of Local Extrema

Local y-maxima on the upper contour are candidate ascenders and local y-minima on the lower contour are potential descenders. The algorithm used for detecting local extrema from the contours is detailed in [19]. A number of heuristics are used to detect and reject spurious extrema resulting from jaggedness and fragmentation in the contour.

The local minima on the lower contour divide the image vertically into segments, as shown in Fig. 6. Segments are numbered from 0 to $n-1$, and $n$ is used as the estimate of the length of the given phrase.

Positions of features are specified in terms of *Holistic Segment Distance* (HSD). Each position is of the form *x.y*, where $x$ is the segment number and $y$ is the offset into the segment computed as a fraction of the width of the segment (Fig. 7). HSD preserves continuity of position across segment boundaries, while affording greater precision than the segment number alone.

#### 4.2.4 Determination of Word Gap Candidates

Since exterior contours are sorted from left to right, every transition between adjacent contours is a potential word gap. The gap width (distance between bounding boxes of exterior contours to the left and right of the gap) is normalized by the mean horizontal separation between minima. This is motivated by the observations that 1) the width of true word gaps is proportional to the mean width of characters, and 2) the mean character width is proportional to the "pitch" of the handwritten signal, which in turn is approximated by the mean separation between lower contour minima.

Since word gaps are often succeeded or preceded by ascenders [20], the presence of ascenders in the neighborhood is factored into the confidence computation. The displacements from the baseline of the maxima to the left and right are computed and normalized by the width of the middle zone. The confidence of the gap $c_g$ is computed as a logistic function of the normalized width $w$ and displacements of flanking maxima $h_{left}$ and $h_{right}$:

$$c_g = \frac{1}{1 + e^{G_0 + G_1.w + G_2.h_{left} + G_3.h_{right}}}. \tag{1}$$

The weights $G_i$, are computed by logistic regression over a training set. Only the $K$ largest gaps are retained as candidate word gaps. A $K$ of five was found sufficient for the street name verification task since lexicon phrases seldom have more than three words (Fig. 8).

### 4.3 Feature Prediction

In this phase, each lexicon phrase is evaluated in turn. The coarse holistic features of the phrase are predicted and matched with the feature candidates extracted from the image. Goodness of match scores are used to determine whether to accept or reject the lexicon phrase. The image itself is rejected in the absence of a satisfactory match with *any* of the lexicon phrases. The processing of a lexicon phrase is composed of: 1) feature prediction from ASCII, and 2) hierarchical evaluation of features.

The expected holistic features of lexicon phrases are predicted from ASCII using a set of heuristic rules and models of "ideal" lower and upper case characters.

1. *Character models:* Character models provide coarse descriptions of the shape of features in terms of the number of minima and positions of ascenders and

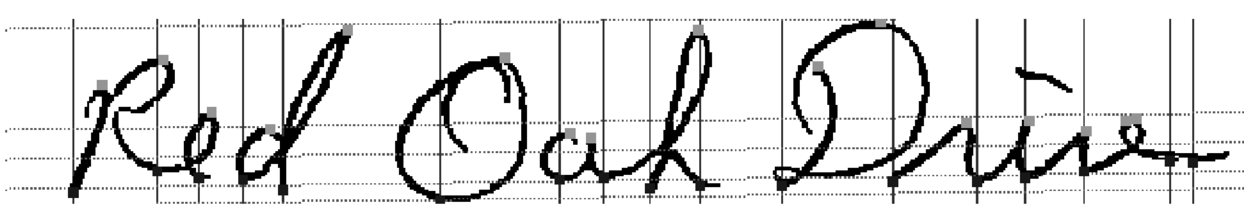


Fig. 6. Vertical grid imposed by lower contour minima.

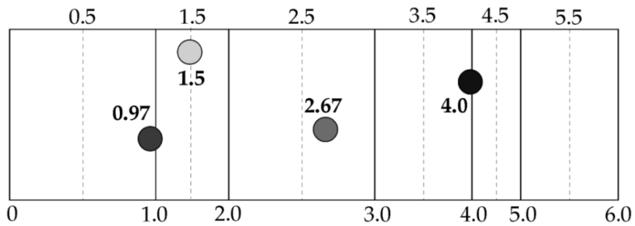



Fig. 7. Positions of features specified in terms of Holistic Segment Distance. The distance is linear in between consecutive grid lines. The grid itself is variable and determined by local minima on the lower contours of the handwritten phrase.

descenders relative to the minima. In addition to the features that may be normally expected (based on our observation of thousands of handwritten characters collected from mail pieces) of a handwritten character, the character models also specify optional features of the character (Fig. 9).

Optional features allow modeling of alternate ways of writing the same character. For example, 'G' and 'f' may be written with or without descenders—the descenders on these characters are said to be optional. Optional features also facilitate modeling of the additional ascenders created when strokes fail to connect, for example, when the two vertical strokes of an 'A' fail to touch at the apex. Finally, optional features allow compensation for spurious features resulting from fragmentation in the image, untidy writing, and inaccuracy of reference lines.

2. *Phrase models:* The features of the lexicon phrase as a whole are phrase length and word gaps. The phrase length is simply the sum of the ideal lengths of constituent characters, while word gaps are assumed at transitions from lower to upper case within the lexicon phrase. The phrase "RedOakDrive," for example, is assumed to have two word gaps and three words.
3. *Word models:* For each word in the phrase, the positions of ascenders and descenders are derived from the positions of these features of the constituent characters, as illustrated in Fig. 10. The word length is computed as the sum of the ideal lengths of characters in the word.

Fig. 8. Five largest gap candidates in handwritten phrase image.

The output of feature prediction is composed of the phrase length, the positions of gaps in the phrase and for each word, word length, and the positions of normal and optional ascenders and descenders within the word. All positions are expressed in terms of the Holistic Segment Distance described earlier. Normal features are assigned a confidence of 1.0, and optional features a confidence of 0.5.

### 4.4 Feature Evaluation

Holistic features of the lexicon phrase at the phrase and word level are matched against the corresponding candidates from the image sequentially in order of increasing computational cost. Each matching step is followed by thresholding to facilitate early rejection of poor matches. A lexicon phrase is said to be verified if it survives the entire sequence of thresholding stages without being rejected. Verification of a lexicon phrase leads to a decision of accept being immediately returned.

While scalar features such as phrase and word lengths may be directly compared, positional features (gaps, ascenders, descenders) are matched by a dynamic programming algorithm (POSMATCH) using multiple models of positional distortion of ideal features. The algorithm yields lexicon-driven detection of the true word gaps, ascenders, and descenders in the image when the lexicon phrase being considered is the truth, and poor match scores otherwise. Attributes of the matched image features are treated as attributes of the author's writing style and used to rescale match scores prior to thresholding.

The holistic features of an "ideally written phrase" occur in the ideal positions and have ideal confidences of unity. In practice, the same features are found to have shifted to the left or right (positional distortion) and to be different in size (scale distortion). Central to POSMATCH is the notion of separability of positional and scale distortion of positional features.

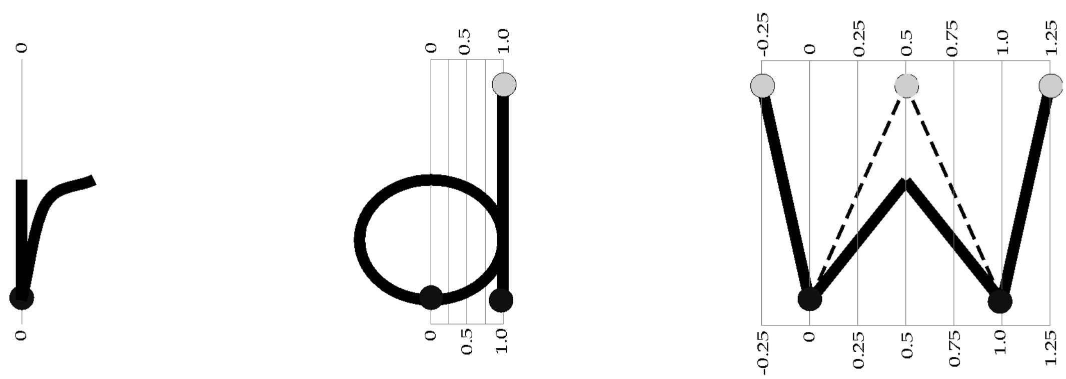



Fig. 9. Character models showing ideal positions of features in HSD. Optional features are shown by dotted lines.

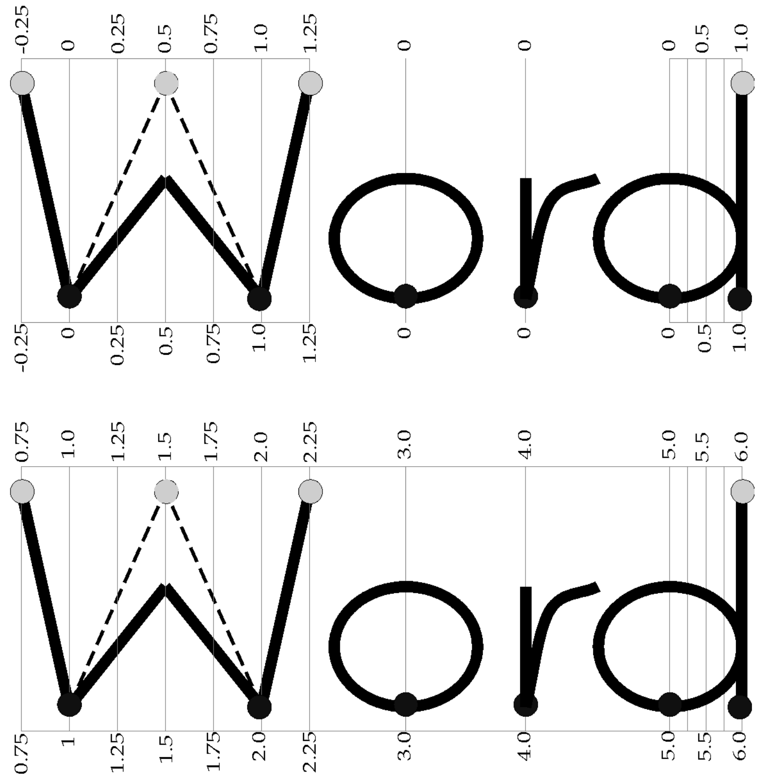



Fig. 10. The positions of ascenders and descenders in the word are derived by "concatenating" models of the constituent characters.

Two models of positional distortion, the *Independent Position Model*, and the *Tied Position Model* are described, both based on the underlying principle that ideal features have ideal positions from which they may shift in either direction within limits. Positional features are, therefore, represented as solid spheres attached to their ideal positions along the word by tightly coiled springs. Since the object is to model positional distortion exclusively, the spheres are only allowed one degree of freedom—they may move to the left or the right along the length of the word (Fig 11).

In the *independent-position* model, distortion of individual holistic features are assumed to be independent of one another. In the *tied-position model* on the other hand, features are tied to one another and to the ends of the word, and consequently, their distortions are not independent. The positional distortion incurred in matching a set of image features with the ideal features captured by the lexicon is measured in terms of the sum total of the "tension or compression" in the springs.

These models are implemented implicitly by the function $pfit(p, q)$, which measures the degree of positional fit between two given positional features $p$ and $q$.

POSMATCH models three ways in which extracted image features may differ from the ideal lexicon features: 1) missed image features, 2) spurious image features, and 3) features distorted in position and scale [18].

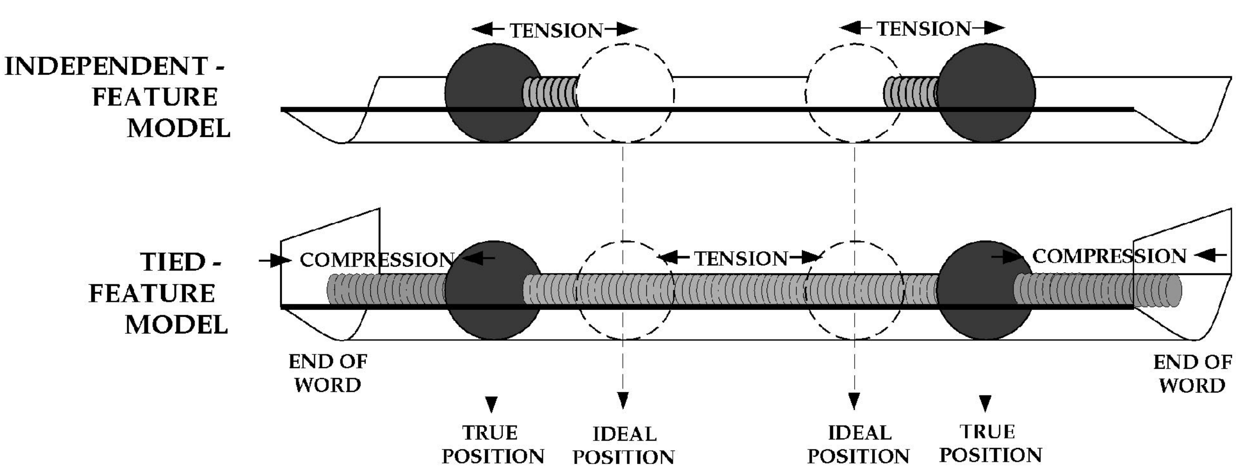



Fig. 11. Two models of positional distortion.

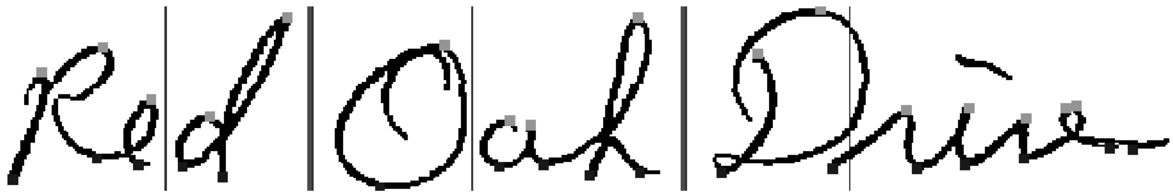


Fig. 12. Results of matching lexicon gaps using POSMATCH. Gap candidates that were matched with lexicon gaps and those which were left unmatched are indicated by bold and thin lines, respectively.

The practice of "overextraction" ensures that valid features are seldom missed altogether. Hence, matching reduces to a process of selecting the subset of image features that provides the best bipartite match with the lexicon features. An unmatched lexicon feature leads to immediate rejection of the lexicon phrase. The degree of match of a given image feature with a given lexicon feature is computed as a function of their positional fit and the associated confidences.

Normal and optional lexicon features are matched in separate passes over the image: Once the normal features of the lexicon phrase have been matched with the image features, the matched image features are tagged, and only the remaining image features are made available for matching with optional features. Once the best match has been determined, the following scores are computed and used for subsequent thresholding.

1. *Position-score (pscore)*: The position score is computed as the mean positional fit over matches of image features with normal lexicon features. The positional fit of a pair of positional features varies inversely with the positional distortion incurred in matching them.
2. *Confidence score (cscore)*: The confidence score is computed as the mean confidence associated with image features that matched normal lexicon features.
3. *Unmatch score (unmatch)*: The unmatch score is computed as the maximum confidence over unmatched image features.

It is to be noted that optional features are ignored in the score computation. Their sole purpose is to match spurious image features and prevent them from distorting the *unmatch* score.

The scores pscore, cscore, and unmatch are used for thresholding.

### 4.4.1 Phrase Level Features

1. *Phrase length:* The absolute value of the difference between the image and lexicon phrase lengths is used to eliminate lexicon phrases which are dissimilar in length.
2. *Word gaps:* POSMATCH is used to match image gap candidates with the gaps in the lexicon phrase. This procedure effectively accomplishes segmentation of the handwritten phrase into words, with the difference that the segmentation of the image so obtained is particular to, and driven by, the lexicon phrase under consideration. The true word gaps are found much more reliably in this manner than by using any purely image-based technique (Fig. 12). The values of *pscore*, *cscore*, and *unmatch* are used to reject poor matches, that is, lexicon phrases wherein the spatial arrangement of gaps differs considerably from that of significant gaps in the image.

### 4.4.2 Word Level Features

Knowledge of the positions of the true gaps in the image is used to partition the candidate ascenders and descenders extracted during the feature extraction phase by the word they belong to. The lengths of each of the image words also become available for the first time.

The primary attribute associated with a candidate ascender is its height above the baseline. In practice, ascenders are often written such that they are not prominent relative to the global reference lines, but stand out from their immediate surroundings. In an attempt to

Fig. 13. Lexicon-driven matching of ascenders using POSMATCH. Matches with normal and optional lexicon ascenders are shown in different shades.

Fig. 14. An untidily written handwritten street name where the assumption that ascenders are taller than other maxima is violated. Note also the mixing of case, a condition that is generally impossible to detect without OCR.

capture a sense for the "local" reference lines in the neighborhood of the candidate ascender, the heights above the baseline of the maxima to its immediate left and right are also determined. All three measurements are normalized by the middle zone width. In the event that the maximum is the first or last maximum within the word, the "missing" maximum is assumed to be aligned with the half line.

The ascender confidence $c_a$ is computed as a logistic function of the normalized heights $h$, $h_{left}$, and $h_{right}$:

$$c_a = \frac{1}{1 + e^{A_0 + A_1.h + A_2.h_{left} + A_3.h_{right}}}. \quad (2)$$

The weights $A_i$, are computed from logistic regression on training data. The computation of $c_a$ is postponed until after the segmentation of the image into words because of the special treatment required for the first and last maxima within words. A similar procedure is followed to assign confidences to descender candidates.

Feature matching at the word level is performed for each pair of image and lexicons, and scores evaluated after each word to allow early rejection. The features of words matched are word length, ascenders, and descenders:

1. *Word length:* The degree of agreement of lengths of a pair of corresponding image and lexicon words is computed as a function of their difference and their arithmetic mean, and is averaged over all words to obtain the overall length score.
2. *Ascenders and descenders:* For each word in the phrase, POSMATCH is used to select the maxima from the word image that best match the lexicon ascenders (Fig. 13). The mean unnormalized height of these maxima is used to rescale the value *cscore* and *unmatch* prior to thresholding. As mentioned earlier, the scores are evaluated for each word, and the lexicon phrase rejected in the event of a poor match. In addition, the overall match scores are computed from the scores at the word level as follows:
   - Overall pscore: arithmetic mean of word level values
   - Overall cscore: arithmetic mean of word level values
   - Overall unmatch: maximum of word level values

   Further thresholding is performed on the overall values when all pairs of image and lexicon words have been processed. The matching of descenders follows the same pattern as ascenders. The lexicon-driven approach to detection of ascenders and descenders provides for the correct lexicon phrase, reliable detection of the true ascenders and descenders, when compared to purely image-based techniques. Matches with other phrases are typically poor owing to the differences in spatial arrangement of features.

TABLE 3
Accept and Error Rates with and without Verification

| System | Total | Verified | Error |
|---|---|---|---|
| No verification | 3119 | 3119 (100%) | 579 (18.6%) |
| HOVER-2 | 3119 | 944 (30%) | 23 (2.4%) |

#### 4.4.3 Thresholds

The thresholds used for rejection of the lexicon phrase at various points during its evaluation are parameters which determine overall system behavior and performance. Higher thresholds lead to more conservative behavior. Fewer images are verified, and fewer of those verified are in error. Lowering thresholds has the opposite effect: It increases the numbers of both the images verified and the errors among them. Reasonable values for thresholds depend on desired system behavior, and may be determined by experimentation with a training set of phrase images.

### 4.5 Limitations of Methodology

The methodology of extraction and lexicon-driven matching of holistic features such as gaps, length, ascenders, and descenders has several limitations:

1. Phrases handprinted in either upper or mixed case cannot be handled, since they have no reliable holistic features (besides length). An attempt is made to detect and reject such images based on the separation between the half and top lines, but is only partially successful given the unconstrained nature of writing styles encountered. In particular, words printed in uppercase but with some characters larger than others are impossible to reject without OCR.
2. HOVER expects that words are generally well written, with ascenders and descenders clearly standing out from the body of the word. These assumptions do not always hold for the handwritten phrases encountered in practice, and the image may then be rejected, or worse, verified against an erroneous lexicon phrase (Fig. 14).

TABLE 4
Mean Execution Time per Image for Unoptimized Code Running on a 150 MHz Sun SPARC 10

| System | Image processing | Lexicon evaluation | Total |
|---|---|---|---|
| HOVER V 1.3 | 11.2 msec | 9.5 msec | 20.7 msec |

## 5 Experimental Results

A test deck of 3,119 street name images automatically extracted by the HWAI system was available for evaluating the performance of HOVER. HOVER is presented with the agreement-rejects (low-confidences instances of agreement of the top choices of WMR and CMR) and corresponding lexicon variant(s) of the agreed upon street name selected by WMR and CMR.

Since the agreement-rejects comprise a relatively small fraction of the total stream processed by WRC, the verification performance of HOVER was evaluated instead on the entire set of 3,119 street name images using a singleton lexicon composed of the top ranked street variant from WMR. This stream differs from the one HOVER would see in practice in that it includes both images that WMR or CMR would recognize with a high confidence (and WRC would accept upstream) and images that WMR or CMR would recognize with a very poor confidence (and WRC would reject upstream). In practice, HOVER would see only the cases with confidences in the "gray region" for both classifiers.

Of the 3,119 test cases, 2,540 (81.4 percent) have corresponding lexicons with the top choice of word recognition correct and 579 (18.6 percent) have wrong ones. However, for the reasons mentioned earlier, a correct top choice does not ensure that the variant of the street name in the lexicon is exactly that in the image. Nor does it preclude the presence of secondary information such as apartment and suite numbers following the street name in the image. For example, for an image of "N. Elmwood St Apt 4", the lexicon phrase "North Elmwood Street" is the top choice, but is not an exact match. It was estimated earlier that only 60 percent of the cases selected as the top choice in the lexicon are exact matches, capable of being verified by HOVER.

The accuracy of one version of HOVER, obtained from a particular choices of thresholds, on the test set is tabulated in Table 3. The error rate is defined as the fraction of erroneous verifications amongst the verified images. The mean execution time per image for image processing (excluding connected component analysis) and lexicon evaluation is shown in Table 4.

In general (based on a test set of about 10,000 handwritten mail piece images), the street name recognizer was called on about 40 percent of the images. The remaining images fall into the following two categories: 1) address is not a street name type of address, e.g., a PO Box address, and 2) the ZIP code or the street number were not recognized with a strong confidence. About 10 percent of the images where street name recognizer is called, (WMR and CMR) agree on the top choice but with low confidences. It is this pool of "gray-area" images that are passed on to the HOVER. This strategy effects the error rate as evident from Table 3.

## 6 Summary

HOVER allows rapid verification of handwritten phrases using holistic features such as gaps, length, ascenders, and descenders. The detection of the true word gaps, ascenders, and descenders is lexicon-driven. In general, "over-extraction" of candidates combined with lexicon-driven detection improves the reliability of feature detection. The confidences of image features are derived from multiple attributes using logistic functions computed from training samples using Logistic Regression.

Features of verification strings (lexicon phrases) are predicted from tabulated features of constituent characters using heuristic rules. A dynamic programming algorithm is used for matching positional feature candidates from the image with predicted features of the lexicon phrase.

Image processing in chain code and a hierarchical treatment of features maximize computational efficiency. The hierarchical strategy also facilitates rapid rejection of poor matches and increases system throughput.

Increasing the robustness of feature extraction is an important concern. Spurious extrema typically lead to additional ascenders and descenders. The writing of many authors is marked by an extending below the baseline of the leading vertical stroke of the starting letter in each word. Optional features help to compensate for these, but must be used with caution since they may match genuine image features. The elimination of spurious minima is therefore an important research direction. Moreover, the confidences computed for image features are highly dependent on the reference lines. The use of separate sets of reference lines for each word in the phrase [19] instead of a single set of global reference lines for the entire phrase is highly desirable.

More powerful character models which incorporate statistics from training is the important next step after ideal features. Improved modeling of optional features, better ways of combining results from distortion models, and better thresholding schemes are other important directions for further research.

## References


[1] R.N. Haber, L.R. Haber, and K.R. Furlin, "Word Length and Word Shape as Sources of Information in Reading," *Reading Research Quarterly*, vol. 18, pp. 165-189, 1983.

[2] S.J. Soltysiak, *Visual Information in Word recognition: Word Shape or Letter Identities?*, 1994.

[3] S. Madhvanath, E. Kleinberg, and V. Govindaraju, "Empirical Design of a Multi-Classifier Thresholding/Control Strategy for Recognition of Handwritten Street Names," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 11, no. 6, 1997.

[4] G. Kim and V. Govindaraju, "A Lexicon Driven Approach to Handwritten Word Recognition for Real-Time Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 4, pp. 366-379, Apr. 1997.

[5] J.T. Favata and S.N. Srihari, "Off-Line Recognition of Handwritten Cursive Words," *Proc. SPIE Symp. Electronic Imaging Science and Technology,* San Jose, Calif., 1992.

[6] S. Madhvanath and V. Govindaraju, "Serial Classifier Combination for Handwritten Word Recognition," *Proc. Third Int'l Conf. Document Analysis and Recognition (ICDAR 95),* Montreal, Canada, pp. 911-914, Aug. 1995.

[7] J. Moreau, "A New System for Automatic Reading of Postal Checks," *Proc. Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-2),* Bonas, France, pp. 121-132, Sept. 1991.

[8] J.C. Salome, M. Leroux, and J. Badard, "Recognition of Cursive Script Words in a Small Lexicon," *Proc. First Int'l Conf. Document Analysis and Recognition (ICDAR 91),* Saint-Malo, France, pp. 774-782, Sept./Oct. 1991.

[9] J. Camillerapp, G. Loreatte, G. Menier, H. Oulhadj, and J.C. Pettier, "Off-Line and On-Line Methods for Cursive Handwriting Recognition," *From Pixels to Features III: Frontiers in Handwriting Recognition,* S. Impedovo and J.C. Simon, eds. pp. 273-288, North-Holland, 1992.

[10] J.-P. Dodel and R. Shinghal, "Symbolic/Neural Recognition of Cursive Amounts on Bank Cheques," *Proc. Third Int'l Conf. on Document Analysis and Recognition (ICDAR),* Montreal, Canada, pp. 15-18, Aug. 1995.

[11] A.C. Downton, R.W.S. Tregidgo, C.G. Leedham, and Hendrawan, "Recognition of Handwritten British Postal Addresses," *From Pixels to Features III: Frontiers in Handwriting Recognition,* S. Impedovo and J.C. Simon, eds., pp. 129-144, North-Holland, 1992.

[12] Hendrawan and A.C. Downton, "Verification of Handwritten British Postcodes Using Address Features," *Fundamentals of Handwriting Recognition,* S. Impedovo, ed., pp. 313-317, Springer-Verlag, 1993.

[13] R. Plamondon and G. Lorette, "Automatic Signature Verification and Writer Identification—The State of the Art," *Pattern Recognition,* vol. 22, pp. 107-131, 1989.

[14] S. Kuo and O.E. Agazzi, "Keyword Spotting in Poorly Printed Documents Using Pseudo 2-D Hidden Markov Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 8, pp. 842-848, Aug. 1994.

[15] J. Bertille, M. Gilloux, and M. Leroux, "Recognition of Handwritten Words in a Limited Dynamic Vocabulary," *Proc. Third Int'l Workshop Frontiers in Handwriting Recognition (IWFHR III),* Buffalo, N.Y., pp. 417-422, 1993.

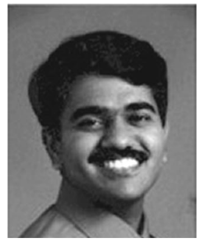
[16] S. Madhvanath, "The Holistic Paradigm in Handwritten Word Recognition and its Application to Large and Dynamic Lexicon Scenarios," PhD thesis, State Univ. of New York at Buffalo, 1997.

[17] S. Madhvanath, G. Kim, and V. Govindaraju, "Chaincode Processing for Handwritten Word Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 9, Sept. 1999.

[18] S. Madhvanath and V. Govindaraju, "Holistic Lexicon Reduction," *Proc. Third Int'l Workshop Frontiers in Handwriting Recognition,* Buffalo, N.Y., pp. 71-81, May 1993.

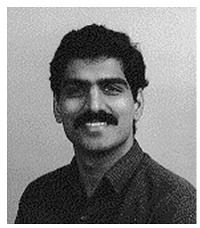
[19] S. Madhvanath and V. Govindaraju, "Contour-Based Image Processing for Holistic Handwritten Word Recognition," *Proc. Fourth Int'l Conf. Document Analysis and Recognition (ICDAR 97),* Ulm, Germany, Aug. 1997.

[20] G. Kim and V. Govindaraju, "Handwritten Phrase Recognition as Applied to Street Name Images," *J. Pattern Recognition,* vol. 31, no. 1, pp. 41-51, 1998.


**Sriganesh Madhvanath** received the BTech degree in computer science from the Indian Institute of Technology, Bombay, in 1990, and the MS and PhD degress in computer science from the State University of New York at Buffalo in 1993 and 1997, respectively. He worked as a research assistant at the Center of Excellence for Document Analysis and Recognition (CEDAR) from 1991 to 1996, and is presently a member of the research staff with the Image and Multimedia Systems Group at the IBM Almaden Research Center, San Jose, California. His research interests include classifier decision combination, genetic algorithms and document analysis, and recognition. He is a member of the IEEE.

**Evelyn Kleinberg** is a senior research scientist at the Center of Excellence for Document Analysis and Recognition (CEDAR), at State University of New York at Buffalo. Since 1990, she has participated in a variety of projects at CEDAR involving handwriting recognition and address interpretation. She has served as technical manager on several projects for the U.S. Postal Service, including real-time handwritten address interpretation and remote computer reading. These projects are now functional in the field.


**Venu Govindaraju** received the PhD degree in computer science from the State University of New York, at Buffalo in 1992. Currently, he is the associate director of Center of Excellence for Document Analysis and Recognition (CEDAR) and concurrently holds the research associate professorship in the Department of Computer Science and Engineering, State University of New York at Buffalo. Presently, he is leading multiple projects on postal applications. He has coauthored more than 90 technical papers in various international journals and conference proceedings and has one U.S. patent. He is the associate editor of the *Journal of Pattern Recognition* and the area chair of the IEEE SMC technical committee for pattern recognition. Dr. Govindaraju has been a coprincipal investigator on several federally sponsored and industry sponsored projects. He is a senior member of the IEEE.