Postprocessing of Recognized Strings Using Nonstationary Markovian Models

Djamel Bouchaffra, *Member*, *IEEE*, Venu Govindaraju, *Senior Member*, *IEEE*, and Sargur N. Srihari, *Fellow*, *IEEE*

Abstract—This paper presents Nonstationary Markovian Models and their application to recognition of strings of tokens. Domain specific knowledge is brought to bear on the application of recognizing zip Codes in the U.S. mailstream by the use of postal directory files. These files provide a wealth of information on the delivery points (mailstops) corresponding to each zip code. This data feeds into the models as *n-grams*, statistics that are seamlessly integrated with recognition scores of digit images. An especially interesting facet of the model is its ability to excite and inhibit certain positions in the n-grams leading to the familiar area of Markov Random Fields. The authors have previously described elsewhere [2] a methodology for deriving probability values from recognizer scores. These probability measures allow the Markov chain to be constructed in a truly Bayesian framework. We empirically illustrate the success of Markovian modeling in postprocessing applications of string recognition. We present the recognition accuracy of the different models on a set of 20,000 zip codes. The performance is superior to the present system which ignores all contextual information and simply relies on the recognition scores of the digit recognizers.

Index Terms—Nonstationary hidden Markov models, zip code recognition, postprocessing, class conditional probability, Markov random fields.

1 Introduction

Researchers in all fields have shown the value of using contextual information to enhance the recognition of patterns. Speech recognition community was the first to model the contextual knowledge present in the English language [12]. However, the signal and language information were used in two distinct but successive phases. It has been shown since that it is possible to integrate these bodies of information in a single phase [3], [4]. Besides, Shinghal and Toussaint [20] address the benefits of using transition probablities that are dependent on both the word position and length in recognition of machine-printed text. In both applications, the use of Markovian models is well justified, albeit for different reasons. Speech recognition involves processing the sound patterns in real-time. This naturally imposes a temporal aspect on the problem. It is a well-known fact in speech analysis that the human voice apparatus introduces the *co-articulation effect*, where the sound of a syllable is affected by the previous syllable uttered. In fact, most real world applications where a stream of patterns is presented to a recognizer in a sequential manner would lend themselves to Markovian models. At any instant of time, the recognition of the pattern in question is dependent on what immediately preceedes it, known as the history of the event. The order of the Markovian process would dictate the extent of the history which has influence on the pattern at hand.

Unlike the temporal realm of speech recognition, the task of machine-printed text recognition does not have a natural temporal component. An entire word image is presented for recognition as a list of character images. Nevertheless, researchers [3], [15], [16], [17] have consistently assumed the Markovian model. Studies indicate the distributions of certain pairs (bigrams) and triples (trigrams) of letters being more frequent than others. In information theoretic terms [11], the language provides an abundance of redundancy (about 50 percent).

The single most important issue that is addressed in this paper is whether what is true with the English language holds true with numeric strings. Let us consider all the zip codes in the United States. If the source of all the zip codes is Markovian, then, once again, the identity of any numeric symbol at a given position will be confined to its neighborhood. This conjecture has not been verified either empirically or theoretically. Each numeral in a string is created in isolation, independent of the neighboring numerals in the string. For that matter, it would not be unusual for a writer to write the numerals in the string in an arbitrary order (write the last two digits and then the first three to its left). The coocurrence of different pairs of numerals at different positions has to be studied. Given that digits at any positions in a zip code can be related all the time, it would be hard to imagine that a strict Markovian property is continually preserved. However, that is exactly the subject of investigation of this paper.

We are interested in finding what underlying structure best models all the zip codes in the United States. First, we will empirically investigate various statistical models. Alternatively, we will arrive at the model that best reflects our understanding of zip codes and the relationship between the digits of the zip code, the goal being one of

• *The authors are with the Center of Excellence for Document Analysis and Recognition (CEDAR), Department of Computer Science, State University of New York at Buffalo, Amherst, NY 14228.*
E-mail: {bouchaff, govind, srihari}@cedar.buffalo.edu.

Manuscript received 16 Oct. 1998; revised 17 May 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 108058.

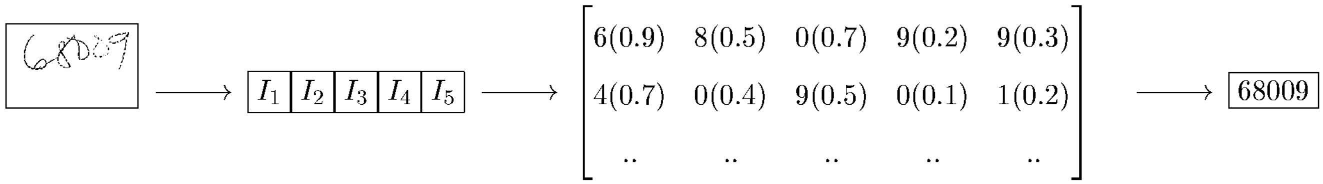


Fig. 1. I/O behavior of the postprocessing module. The input is a *trellis* of classifier choices assoicated with probability values. The ouput is the best zip code candidate based on the probabilities of recognition as well as the Markovian modeling as described in this paper.

achieving the best recognition accuracy of handwritten zip codes.

Throughout this paper, we draw from the application of zip code recognition. We wish to emphasize at this point that there is no loss of generality in our approach. In fact, the method would be equally valid for machine-printed strings. Further, the investigation of an entire gamut of models would pave the way for conducting a similar investigative study for applications with other numeric strings, such as telephone numbers, social security numbers, form IDs, etc.

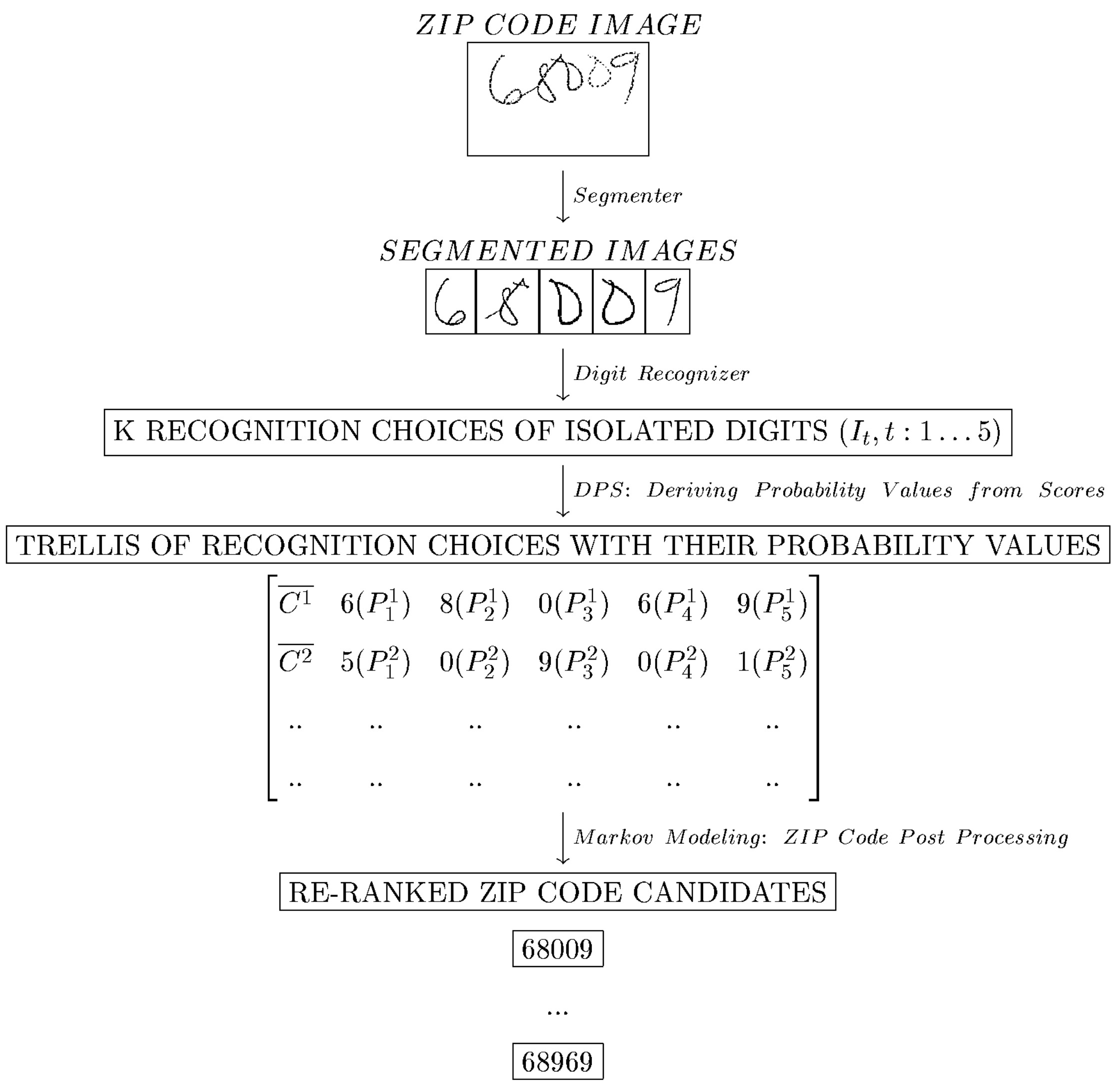


Fig. 2. Starting with an image of a zip code (numeric string of length $L = 5$), segmentation and digit recognition provide K class choices (C_t^k) for each isolated digit ($t : 1 \ldots 5$). Each class choice C_t^k is associated with a score of S_t^k. A procedure called DPS converts the scores into probability values P_t^k. *Postprocessing* by Markov modeling, which is the focus of this paper, takes as input the trellis of class choices with associated probabilities to output a ranked list of zip code candidates.

We are assuming that recognition results of symbols (digit images in case of zip codes) with confidence scores are provided to us for *postprocessing*. The modeling of the knowledge is to improve the quality of the recognition results. To this end, we treat the digit recognizers as *black boxes*.

Fig. 2 describes the flow-down of the various processes involved. Starting with an image of a zip code, the very first step is to isolate five digit images by a process of segmentation. Next is the task of recognizing each digit image by a digit recognizer. The recognizer returns a ranked list of classes for each digit along with its confidence associated with each choice. While the confidence measures returned by recognizers are adequate for most applications where recognition is the final stage of the application, there is a need for true probabilistic measures when the the scores of the recognizer must be integrated with subsequent stages of recognition, as is the case with postprocessing in zip code recognition (Fig. 2). We adopt a procedure called DPS (Deriving Probability from Scores), which we have described in detail elsewhere [2], that simply maps the confidence values to probability values.

This leads us to the subject matter of this paper: *Postprocessing* by Nonstationary Markov Modeling. The input is the trellis of class choices with associated probabilities and the output of the postprocessing is a ranked list of zip code candidates.

Our objective can be simply stated as developing a postprocessing model that produces the maximally accurate recognition of a string of symbols given a *trellis* of symbols and their associated scores by a recognizer. In the case of zip code images, a succint description of the problem is illustrated by Fig. 1.

2 Previous Work

Recognition of handwritten numeric strings has been tackled by several researchers in the literature. Notably, [21] and [22] describe a "segment and then recognize" approach for the application of postal zip code recognition. In [12], [13], the authors described the problem of character segmentation in words and numeric strings, respectively. In [10], the use of context in recognition is described and, in [12], we find a description of the methods of recognizing numeral strings using segmentation. Other methods using a hidden Markov model to scan the entire string and extract feature vectors to perform recognition have also been proposed [9]. However, many of these approaches are dealing with raw recognition and not with postprocessing as known in the natural language or speech community. While there are applications of numeric string recognition other than zip codes, such as recognizing courtesy amounts on bank checks [12], [16], the zip codes have the distinguishing feature of having a fixed length. Besides, each position of a digit in a zip code has a physical meaning. Much of the positional (temporal) aspect of our model depends on this very fact. Choosing the right model and of the right order has been discussed in the context of bank check recognition [17]. However, we believe that a systematic study as presented in this paper has been lacking in the literature.

3 Postprocessing Using a Nonstationary Markov Chain

In this section, we present our approach of postprocessing the digit recognition results. However, we first need to describe the environment that interacts with our task.

3.1 Background

In order to place the *postprocessing* work on zip code recognition using Markov Models in the proper context, we will digress briefly to the other modules of Fig. 2. Present zip code recognition procedures [21], [22], [23], [25], [24] routinely perform the task of locating the zip code within the address, followed by segmenting and recognizing the zip code digits.

Given the image of a string of digits, a segmentation algorithm partitions the image into regions, each containing an isolated digit to be submitted to an isolated digit recognizer. Since adjacent digits can be touching and some of the digits might be broken into more than one component (e.g., 5-hats), the number of digits in a digit string is not simply a count of the number of components in the field.

Improper segmentation tends to leave some of the newly separated digits with artifacts, such as a digit which might end up losing a piece of its stroke to the adjacent digit or might end up with an additional ligature. Either way, subsequent recognition of the digit is rendered inaccurate. Furthermore, finding the precise splitting path that separates the touching digits is nontrivial. The segmenter [19] we use has a correct segmentation rate of 92.9 percent.

We use a Gradient Structural Concavity (GSC) recognizer which is described in [2]. It uses symbolic multiresolutional features. **G**radient of the image contour captures the local shape of a character. The Gradient features are extended to **S**tructural features by encoding the relationships between strokes. **C**oncavity features capture the global shape of characters. Features at the three levels, **G**, **S**, and **C** are combined in a k-nearest neighbor classification method to produce a multiresolution character recognizer. **G** features are the finest and the **C** features are the coarsest. The accuracy of the recognizer on isolated digits is 96 percent.

3.2 Mathematical Formalism

Let S_i represent the measurement (a score, a confidence value, or a proximity measure) made on an input pattern image I_i, where $\overline{I} = [I_1, I_2, \dots I_L]$ is a sequence of subimages (e.g., sequence of subimages of isolated digits). In some sense, the true image pattern is "hidden" from our modeling exercise and we are left to deal with the measurements alone. The classifier's kth class choices are denoted by the vector $\overline{C}^k = [C_1^k, C_2^k, \dots C_L^k]^T$ based on the measurement vector $\overline{S}^k = [S_1^k, S_2^k, \dots S_L^k]^T$, which corresponds to the subimages of $\overline{I}$.

Considering all the class choices with their measurements, we obtain a rectangular trellis $\mathcal{C}$ composed of nodes $\langle C_t^k, S_t^k \rangle$ $(t : 1 \dots L,\ k : 1 \dots c)$ represented in Table 1. The parameter k is the choice level of the recognizer and c is the total number of classes. The pair $\langle C_t^k, S_t^k \rangle$ is the kth recognition choice with its measurement in column t of the *trellis*.

We denote the set of all paths through the trellis by $\mathcal{P}$. Let $\overline{\langle C,S\rangle} = \{\langle C,S\rangle_1, \langle C,S\rangle_2, \ldots, \langle C,S\rangle_L\}$ be a particular path in $\mathcal{P}$ composed of classes and their respective measurements (or nodes). We also denote a path of classes as $\overline{C}$ and a path of measurements as $\overline{S}$. Our objective is to maximize $P(\overline{C} \mid \overline{I})$. Because of the existence of several input subimages I_t, $(t = 1, \ldots, L)$ which produced a grid of recognition scores, our problem is transformed into the search of the optimal path $\overline{\langle C,S\rangle}$ among all possible paths in the trellis. Mathematically, the problem can be stated as:

$$\max_{\overline{\langle C,S\rangle}\in\mathcal{P}} \left[P(\overline{\langle C,S\rangle})\right] \equiv \max_{\overline{C}} P(\overline{C},\overline{S}) = \max_{\overline{C}} \left[P(\overline{S} \mid \overline{C}) \times P(\overline{C})\right]. \quad (1)$$

The task at hand is one of maximizing $P(\overline{S} \mid \overline{C}) \times P(\overline{C})$ over all possible paths $\overline{C}$ in the trellis. Typically, digit recognizers return just the top two or three recognition choices. Let us say the maximum is three. This leads to 3^5 zip code candidates that can be generated by the 3×5 *trellis*.

Maximizing the expression $P(\overline{S} \mid \overline{C}) \times P(\overline{C})$ over $\overline{C}$ would be equivalent to maximizing any monotonically increasing function of the same expression. Hence, our objective could be restated as

$$\max_{\overline{C}} [\log P(\overline{S} \mid \overline{C}) + \log P(\overline{C})]. \quad (2)$$

If we were to assume that the measurements $\overline{S}$ are all independent, the confidence scores $\overline{S}$ would be independent as well and (2) can be further simplified. However, in real life, adjacent handwritten digits in a numeric string can touch. This leads to an equivalent of the *co-articulation* effect that is witnessed among adjacent syllables in speech recognition. To keep the model simple, we will, at this time, ignore the impact of touching digits.

For the sake of brevity, we denote a path as $(\overline{C},\overline{S}) = \langle C_t, S_t\rangle$ for $(t = 1 \ldots L)$ composed of classes C_t and their measurements (or scores) S_t. The assumption of independence allows the following simplification of (2):

$$\max_{C_t} \left[\sum_{t=1}^{t=L} (\log P(S_t \mid C_t)) + \left(\log P(C_1) + \sum_{t=1}^{t=L} \log P(C_t \mid C_1, C_2, \ldots, C_{t-1}) \right) \right]. \quad (3)$$

Each of the two terms of (3) bear diverse bodies of knowledge.

The first term, $P(S_t \mid C_t)$ is the probability of a measurement of a class at a particular node in the path $\overline{C}$ given the class C_t at this node. Since each node $\langle C_t, S_t\rangle$ in a path of the trellis $\mathcal{C}$ is a pair $\langle C_t^k, S_t^k\rangle$, where $(k = 1, \ldots, c)$ is the class choice and $(t = 1, \ldots, L)$ the position of this class in the path, the maximization problem of (3) involves conditional probabilities $P(S_t^k \mid C_t^k)$. We assume that this conditional probability is independent of time (or position): $(P(S_t^k \mid C_t^k) = P(S^k \mid C^k))$.[1] It is the probability of a measurement returned by a digit recognizer given the class of the image. Computing this probability is nontrivial. Indeed, computing its Bayes derivative, $P(C^k|S^k)$ is equally difficult. We have described a methodology of computing this probability elsewhere [2] in sufficient detail. For subsequent integration with the postprocessing module, it is crucial that the scores be converted to probability values (P_t^k) so that the Bayesian framework can be adopted, thus putting the methodology on firm mathematical grounding. This is exactly what we achieve in [2].

We use the nonstationarity property of a Markov chain in the language part:

$$\log P(C_1) + \sum_{t=1}^{t=L} P(C_t \mid C_1, C_2, \ldots, C_{t-1})$$

by highlighting the temporal aspect of the class distribution. In fact, the position (or temporal aspect) of a class (or digit) is a discriminative factor in zip codes. In other words, $P(C_{t_{j_1}}^k) \neq P(C_{t_{j_2}}^k)$ if $j_1 \neq j_2$, where $C_{t_j}^k$ is the class C^k at time t_j.

These facts allow the following substitution in (3):

$$\max_{C_t^{i_t}} \left[\sum_{t=1}^{t=L} \log \frac{P(C^{i_t} \mid S^{i_t})}{P(C^{i_t})} + \log P(C_1^{i_1}) + \sum_{t=1}^{t=L} \log P(C_t^{i_t} \mid C_1^{i_1}, C_2^{i_2}, \ldots, C_{t-1}^{i_{t-1}}) \right]. \quad (4)$$

Equation (4) shows that the language part is essentially a combination of n-gram probabilities, where $1 \leq n \leq L$.

The algorithm used to compute the optimal path is the "brute force" algorithm. That is, all possible paths were computed from the rectangular trellis and the path resulting in the largest probability value was chosen. The total number of paths in the trellis with k rows is k^5. Each path resulted in five probability searches, each of O(1) (big O). Therefore, the total number of probability searches is $5 \times k^5$. The complexity of this algorithm is $O(k^5)$, where k is the number of rows in the trellis. In our application, $k = 2$, therefore, there were 32 possible paths and 160 probability searches.

We can also use the Viterbi algorithm [20] in cases where the dimension of the trellis is large. Equation (4) describes a nonstationary hidden Markov model. In fact, a state is a class of digit and an observation emitted from a state is the score assigned by the recognizer to this class.

3.3 Language Parameters Estimation

The nonstationary transition probabilities are estimated using likelihood functions [1]. For simplicity, we show the ML estimation for n = 3 (trigrams). Suppose the Markov chain has c possible states (or classes) and we observe L sets of successive transitions. Suppose there are $Y_{C_1^{i_1}}$ number of times we observe class $C_1^{i_1}$ at time $t = 1$ (or number of individuals), and the $\{Y_{C_1^{i_1}}\}$ are multinomially distributed

1. This assumption is not always true since digit images at either end of a string contain less ligature and thus pose different complexity to recognizers when compared to digit images in the middle of a string.

TABLE 1
2D-Trellis Representation of Class Candidates with Their Scores

	$\boxed{I_1}$	$\boxed{I_2}$	$\boxed{I_3}$	$\boxed{I_4}$	..	$\boxed{I_L}$
$\overline{C^1}$	$\langle C_1^1, S_1^1\rangle$	$\langle C_2^1, S_2^1\rangle$	$\langle C_3^1, S_3^1\rangle$	$\langle C_4^1, S_4^1\rangle$	..	$\langle C_L^1, S_L^1\rangle$
$\overline{C^2}$	$\langle C_1^2, S_1^2\rangle$	$\langle C_2^2, S_2^2\rangle$	$\langle C_3^2, S_3^2\rangle$	$\langle C_4^2, S_4^2\rangle$	..	$\langle C_L^2, S_L^2\rangle$
$\overline{C^3}$	$\langle C_1^3, S_1^3\rangle$	$\langle C_2^3, S_2^3\rangle$	$\langle C_3^3, S_3^3\rangle$	$\langle C_4^3, S_4^3\rangle$	..	$\langle C_L^3, S_L^3\rangle$
..	..	..	..	..	..	
$\overline{C^c}$	$\langle C_1^c, S_1^c\rangle$	$\langle C_2^c, S_2^c\rangle$	$\langle C_3^c, S_3^c\rangle$	$\langle C_4^c, S_4^c\rangle$	..	$\langle C_L^c, S_L^c\rangle$

with probabilities $\eta_{C_1^{i_1}}$ and sample size $m = \sum_{C^{i_1}=1}^{C^{i_1}=c} Y_{C_1^{i_1}}$. Finally, let $Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, C_t^{i_t}}$ be the number of times we observe class C^{i_t} at time t, class $C^{i_{t-1}}$ at time $t-1$, and class $C^{i_{t-2}}$ at time $t-2$, and let $Y_{C_t^{i_t}}$ be the number of times we observe the class C^{i_t} at time t. Let $P(C_t^{i_t} \mid C_{t-1}^{i_{t-1}}, C_{t-2}^{i_{t-2}})$ be the conditional probability of being in class C^{i_t} at time t, given class $C^{i_{t-1}}$ at time $t-1$ and class $C^{i_{t-2}}$ at time $t-2$ so that:

$$\sum_{C_{i_t}} P(C_t^{i_t} \mid C_{t-1}^{i_{t-1}}, C_{t-2}^{i_{t-2}}) = 1 \qquad \forall t,\ \forall C_{t-1}^{i_{t-1}}, C_{t-2}^{i_{t-2}}. \tag{5}$$

Therefore, for fixed values of $C_{t-1}^{i_{t-1}}, C_{t-2}^{i_{t-2}}$, the conditional distribution of $Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, C_t^{i_t}}$ for $C_t^{i_t} = 1, \ldots, c$, given $Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, +(t)}$ (where $+(t)$ means any class at time t) is:

$$\frac{Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, +(t)}!}{\prod_{C_t^{i_t}=1}^{c} Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, C_t^{i_t}}!} \times \prod_{C_t^{i_t}=1}^{c} P(C_t^{i_t} \mid C_{t-1}^{i_{t-1}}, C_{t-2}^{i_{t-2}})^{Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, C_t^{i_t}}}. \tag{6}$$

If all transitions are mutually independent, then the joint probability distribution of the $Y_{C_{t-2}}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, C_t^{i_t}$ and the $Y_{C_1^{i_1}}$ is:

$$\left[\frac{m!}{\prod_{i_1=1}^{c} Y_{C_1^{i_1}}!} \prod_{i_1=1}^{c} \eta_{C_1^{i_1}}^{Y_{C_1^{i_1}}}\right] \prod_{t=1}^{t=L} \left\{\prod_{i=1}^{i=c}\left[\frac{Y_{C_{t-1}^{i_{t-1}}}!}{\prod_{i_t=1}^{c} Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, C_t^{i_t}}!} \prod_{i_t=1}^{c} P(C_t^{i_t} \mid C_{t-1}^{i_{t-1}}, C_{t-2}^{i_{t-2}})\right]\right\}. \tag{7}$$

The likelihood function of $\eta_{C_1^{i_1}}$ and $\{P(C_t^{i_t} | C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}})\}$ is:

$$\left[\prod_{i_1=1}^{i_1=c} \eta_{C_1^{i_1}}^{Y_{C_1^{i_1}}}\right] \times \left[\prod_{t=1}^{t=L} \prod_{i_{t-2}, i_{t-1}, i_t=1}^{c} P(C_t^{i_t} \mid C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}})^{Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, C_t^{i_t}}}\right].$$

The maximum likelihood estimates of the nonstationary transition probabilities are:

$$\begin{cases} \hat{P}(C_t^{i_t} \mid C_{t-1}^{i_{t-1}}, C_{t-2}^{i_{t-2}}) = \frac{Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, C_t^{i_t}}}{Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, +(t)}}, \\ \hat{\eta}_{C_1^{i_1}} = \frac{Y_{C_1^{i_1}}}{m} \end{cases} \tag{8}$$

where $Y_{C_{t-2}^{i_{t-2}}, C_{t-1}^{i_{t-1}}, +(t)}$ is the number of times we observe any class C^{i_t} at time t preceded by class $C^{i_{t-1}}$ at time $t-1$ and class $C^{i_{t-2}}$ at time $t-2$. The computation of $P(C^{i_t} \mid S^{i_t})$ itself is performed by DPS, which is described in [2].

3.4 Markov Chain Training Set

The estimation of these transition probabilities has been computed from postal directories made available by the United States Postal Service. We have constructed a database from the postal directories. It contains two entries which correspond to all zip codes in the U.S. and the number of records in each zip code corresponding to the delivery addresses in the zip code. Ideally, we would need the statistics from the mailsbeam that reveals the frequency of mail to different zip codes. The number of records in a zip code is a reasonable substitute for the same. Statistics of interest hold over time. About 0.50 percent of zip codes are added and 0.50 percent deactivated (removed for a specific period of time) over a period of three months.

In this application, data in the postal directories depend on the period λ_{t_1,t_2} between the two times t_1 and t_2 where these directories are made available by the U.S. postal service. There is a very small proportion of zip codes that are added or deactivated (removed) over the time periods of 3, 7, and 21 months. Zip codes are deactivated when they do not have any records assigned to them anymore. There is also a translation (or change) in zip codes of 66 percent, 72.33 percent, and 85 percent, respectively, over 3, 7, and 21 months. These translations are perhaps due to new housing in a particular zip code area or a change in the delivery address in the zip code. Besides, the proportion of records added, deactivated, and translated is relatively small in the three periods of time. Therefore, in order to compute a consistent and reliable ML estimation, *training and testing should be performed in the same period of time*. The period of time λ_{t_1,t_2} that is needed for the postal service in order to provide postal directories is short. Even during this period of time, there are around 15 percent of zip codes that have changed. In conclusion, data sparseness is unavoidable in our application. Some zip codes encountered during testing may never have been encountered during training. Further, a zip code that is deactivated in a certain period of time may be activated at another time. We are using linear interpolation techniques [4], [5] in order to optimally estimate the transition probabilities of n-grams contained in the zip codes that are unseen in training.

3.5 The Trie Data Structure

The trie data structure was developed for the purpose of fast data retrieval. The trie structure is adapted for retrieving information in the form of a string. For our purposes, the strings were the n-gram numeral strings, where n is the length of the string. It is not important to store the temporal information because only one type of n-gram can occur at a certain time. Only a unigram can occur at time 1, only a bigram can occur at time 2, etc. The trie structure was constructed and maintained in an external file. A binary file was created to store all the probabilities and the file was searched externally. Since the trie has megabytes of information, it would take no longer to read the entire structure into RAM than it would to perform an external search. Since the probabilities were computed off-line, they are not a time issue. The time complexity for searching the trie is $O(n)$, where n is the length of the numeral string. However, since $1 \leq n \leq 5$ and 5 is a small number, the time complexity for searching the trie is O(1).

4 Modeling Contextual Knowledge

For the application of zip code recognition, the appropriate model that represents the "structure" of the contextual knowledge must be determined. There are two ways of approaching this task. First, we experimentally test all possible models (in terms of degrees of Markovian processes) and all their combinations as an empirical way of determining the most suitable model. A second approach would be one where we attempt to understand the structure of the zip code, how different digits relate to one another, and derive a model that matches this understanding. Of course, it would be intuitively satisfying if both approaches point to the same model. In the following section, we explore both scenarios.

4.1 Empirical Evaluation of Models

If the numeric strings are all of a fixed length (L in the discussion above), then the contextual information can be further qualified. By making the probabilities specific to their position within the string, one can better exploit the context. This adaptation is referred to in the literature as a *nonstationary* Markov Model [18]. The context that we are modeling is the probability, $P(C_t = \alpha)$ of a particular class, α in a particular position, t in the numerical string, *given* the occurrence of a particular class (*say*, β) in the *previous* position, $t-1$: $P(C_t = \alpha) \mid C_{t-1} = \beta)$. Note that the choice of the variable t for denoting the position in the string is intentional in that it is the counterpart of the time parameter in temporal Markov models.[2]

This notion of class conditionals ($P(C_t = \alpha \mid C_{t-1} = \beta)$) that we describe is similar to that of positional *n-grams* described in the literature [20]. The use of bigram, trigram, and even quadragram probabilities has been extensively used in text recognition.

- *Markovian Model of* 0*th order:* It represents the case where there is no contextual information shared between neighboring numerals in a string. Assuming $t = 3$ (central position shown by $\boxtimes$) and $L = 5$, the context is drawn from none of the neighboring positions.

$$\begin{bmatrix} \square & \square & \boxtimes & \square & \square \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \quad t = 3$$

  In the zip code application, there are $\binom{5}{1}$ possible position 1-grams. This is represented by five column vectors, one for each position (t) of the zip code. Each entry in the column vector, $U_t[i]_{i:0\ldots9}$, represents the *prior* probability $P(C_t = \alpha)$, the frequency of occurrence of digit α in position $t_{1\ldots5}$.

$$\begin{bmatrix} U_t[0] \\ .. \\ .. \\ U_t[9] \end{bmatrix} = \begin{bmatrix} P(C_t = 0) \\ .. \\ .. \\ P(C_t = 9) \end{bmatrix} t : 1 \ldots 5 \qquad (9)$$

- *Markovian Model of first order:* Conditional probabilities of the first order are represented by two-dimensional matrices $B_{t_1,t_2}[i,j]$ where $i,j : 0 \ldots 9; t_1, t_2 : 1 \ldots 5; t_1 \neq t_2$. It is to be noted that t_1 and t_2 need not be adjacent, i.e., t_2 is not necessarily equal to $t_1 \bigoplus 1$. Furthermore, we will distinguish between the ordering of t_1 and t_2. Thus, the entry $B_{t_1,t_2}[i,j] \neq B_{t_2,t_1}[i,j]$.

  In the zip code application, there are $2 \times \binom{5}{2}$ possible position pairs of t_1 and t_2. This would result in 20 matrices of size 10×10 each.[3]

2. In order to avoid defining special cases for the boundary conditions, we will assume the numerical string to be circular. *Modulo L* addition (denoted henceforth by the operator $\bigoplus$) will imply that the successive neighbor of the last digit in the last (Lth) position is the first digit ($L \bigoplus 1 = 1$).

3. Traditional Bigram tables would simply note the frequency of occurrence of each digit pair.

$$\begin{bmatrix} B_{t_1,t_2}[0,0] & . & . & B_{t_1,t_2}[0,9] \\ . & . & . & . \\ . & . & . & . \\ B_{t_1,t_2}[9,0] & . & . & B_{t_1,t_2}[9,9] \end{bmatrix} = \begin{bmatrix} P(C_{t_1}=0 \mid C_{t_2}=0) & . & . & P(C_{t_1}=0 \mid C_{t_2}=9) \\ . & . & . & . \\ . & . & . & . \\ P(C_{t_1}=9 \mid C_{t_2}=0) & . & . & P(C_{t_1}=9 \mid C_{t_2}=9) \end{bmatrix}$$
$$(t_1,t_2) \in \{1,2,\ldots,5\}^2; \quad t_1 \neq t_2. \tag{10}$$

Not all combinations of t_1 and t_2 are contiguous. In fact, of the $20 = 2 \times \binom{5}{2}$ combinations, only five (with circularity) of the combinations are contiguous in (10).

Assuming $t = 3$ (central position shown by $\boxtimes$) in the string, the context is drawn from one of the neighbors (shown by $\boxplus$).

$$\begin{bmatrix} \boxplus & \square & \boxtimes & \square & \square \\ t_2 & & t_1 & & \end{bmatrix} \quad t_1 = 3; t_2 = 1$$
$$\begin{bmatrix} B_{3,1}[0,0] & .. & .. & B_{3,1}[0,9] \\ .. & .. & .. & .. \\ .. & .. & .. & .. \\ B_{3,1}[9,0] & .. & .. & B_{3,1}[9,9] \end{bmatrix} = \begin{bmatrix} P(C_3=0 \mid C_1=0) & .. & .. & P(C_3=0 \mid C_1=9) \\ .. & .. & .. & .. \\ .. & .. & .. & .. \\ P(C_3=9 \mid C_1=0) & .. & .. & P(C_3=9 \mid C_1=9) \end{bmatrix}. \tag{11}$$

- *Markovian Model of second order:* Conditional probabilities of the second order are represented by three-dimensional matrices $T_{t_1,t_2,t_3}[i,j,k]$ where $i,j,k : 0\ldots 9; t_1,t_2,t_3 : 1\ldots 5; t_1 \neq t_2 \neq t_3$.

  In the zip code application, there are $2 \times \binom{5}{3}$ possible position triples of of t_1, t_2, and t_3. This would result in 40 matrices of size $10 \times 10 \times 10$ each.
- *Markovian Model of third order:* Conditional probabilities of the second order are represented by four-dimensional matrices $Q_{t_1,t_2,t_3,t_4}[i,j,k,l]$ where $i,j,k,l : 0\ldots 9; t_1,t_2,t_3,t_4 : 1\ldots 5; t_1 \neq t_2 \neq t_3 \neq t_4$.

  In the zip code application there are $2 \times \binom{5}{4}$ possible position quadrapules. This would result in 10 matrices of size $10 \times 10 \times 10 \times 10$ each.
- 5-grams information would correspond to a Markovian Model of the fourth order. In the zip code application, there is just one $\binom{5}{5}$ possible position 5-gram. The corresponding matrix $P_{t_1,t_2,t_3,t_4,t_5}[i,j,k,l]$ where $i,j,k,l : 0\ldots 9; t_1,t_2,t_3,t_4,t_5 : 1\ldots 5; t_1 \neq t_2 \neq t_3 \neq t_4 \neq t_5$ would be a five-dimensional matrix.

4.2 Markov Random Fields

Equation (10) shows a phenomenon that we call *triggering*. The assignment of a digit at a particular position *triggers* the assignment of a digit at any position to the right (front) of it, keeping in mind that the positions are in a circular list. The concept of triggering is more general than that of *chaining*, where the triggers are always in adjacent positions. It is easy to see that the ordering of the positions can be changed so that the noncontiguous combinations become contiguous (and vice versa) without altering the problem at hand. Triggering allows us to model the correlation between digits at any two positions in the string. Mathematically speaking, we are looking for the optimal "field" in this Markovian random process. The information content of the noncontiguous combinations could be equally if not more, informative. Unlike the case of text strings, numeric strings usually do not carry any special meaning in adjacent groups of digits. It is for the experiments to determine the true underlying structure.

4.3 Intuitive Modeling Using Knowledge of the Structure of Zip Codes

Statistical information can be readily collected from a database of numeric strings pertinent to the application at hand. In the application of recognizing zip codes, we have been able to collect the n-gram tables (matrices) from the U.S. Delivery Point File.

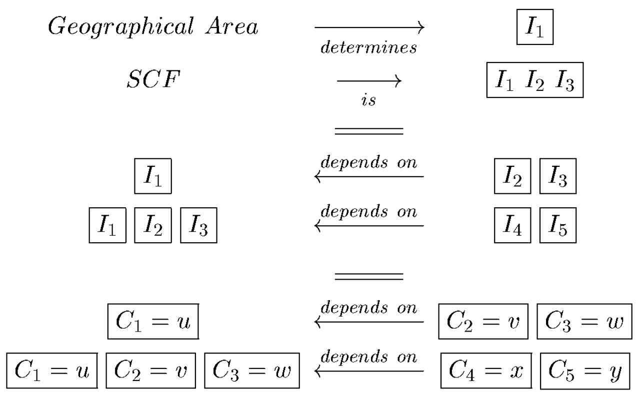
The zip code consists of five digits. The first digit designates a broad geographical area of the United States, ranging from 0 for the northeast to 9 for the far west. The following two digits pinpoint population concentrations and reflect Sectional Center Facilities (SCF) that act as hubs in the transportation networks (see Table 2). The final two digits designate small post offices. There are about 43,000 zip codes in the United States that are valid. Only 43 percent of all possible five-digit strings are valid. Of these, some zip codes are more common in the mailstream than others. Surely the volume of mail destined for Manhattan, New York, is far greater than the volume destined for Boise, Idaho. This information will be used to weigh the frequency of occurrence of different digits in different positions of the zip code. This weight is estimated throughout a fixed period of time by computing the proportion of mailpieces addressed to a particular area of the country in this fixed period of time.

An intuitive reading of the structure of the zip code would suggest that the second and third digits of the zip code are related to the first and the fourth and fifth digits depend on the first three digits (12).

4.4 Mathematical Modeling of the Zip Code Structure

Let us assume that the recognition choice of the zip code under consideration is: $[u\,v\,w\,x\,y]$, i.e., $[I_1 \equiv u, I_2 \equiv v, I_3 \equiv w, I_4 \equiv x, I_5 \equiv y]$.

TABLE 2
Structure of a Five-Digit Zip Code



$$\left[I_t : \begin{array}{ccccc} \boxed{I_1} & \boxed{I_2} & \boxed{I_3} & \boxed{I_4} & \boxed{I_5} \\ u & v & w & x & y \end{array} \right]$$

Then, the objective is simply one of evaluating the probability of the string $u\ v\ w\ x\ y$ (and eventually ranking the probabilities of all the strings). Given the observation made above on the dependence of the various digits in the zip code, it is to be noted that throughout this section we deal with only a single choice of the recognizer and, hence, we drop the superscript k from our notations, i.e., $P(C_t^k) = P(C_t)$.

$$\begin{aligned} &P(C_1 = u, C_2 = v, C_3 = w, C_4 = x, C_5 = y) = \\ &P(C_1 = u)\times \\ &P(C_2 = v, C_3 = w \mid C_1 = u)\times \\ &P(C_4 = x, C_5 = y \mid C_1 = u, C_2 = v, C_3 = w). \end{aligned} \tag{12}$$

However, using Bayes' rule, this is equivalent to:

$$\begin{aligned} &P(C_1 = u, C_2 = v, C_3 = w, C_4 = x, C_5 = y) = \\ &P(C_1 = u) \times P(C_2 = v \mid C_1 = u)\times \\ &P(C_3 = w \mid C_1 = u, C_2 = v)\times \\ &P(C_4 = x \mid C_1 = u, C_2 = v, C_3 = w)\times \\ &P(C_5 = y \mid C_1 = u, C_2 = v, C_3 = w, C_4 = x), \end{aligned} \tag{13}$$

which can be expressed as a function of the matrix elements:

$$\begin{aligned} &P(C_1 = u, C_2 = v, C_3 = w, C_4 = x, C_5 = y) = \\ &U_1[u] \times B_{2,1}[v, w] \times T_{3,1,2}[w, u, v]\times \\ &Q_{4,1,2,3}[x, u, v, w] \times P_{5,1,2,3,4}[y, u, v, w, x]. \end{aligned} \tag{14}$$

Because of the long history of the last term of (14), data sparseness techniques are being used.

4.5 Other Markovian Chains

Other different ways of modeling the zip code have been designed and experimented with. Before we discovered the real structure of the zip code, we related the last two digits as a group to the second and the third, which can be expressed:

$$\begin{aligned} &P(C_1 = u, C_2 = v, C_3 = w, C_4 = x, C_5 = y) = \\ &P(C_1 = u) \times P(C_2 = v, C_3 = w \mid C_1 = u)\times \\ &\times P(C_4 = x, C_5 = y \mid C_2 = v, C_3 = w) = \\ &U_1[u] \times B_{2,1}[v, w] \times T_{3,1,2}[w, u, v]\times \\ &Q_{4,2,3}[x, v, w] \times P_{5,2,3,4}[y, v, w, x]. \end{aligned} \tag{15}$$

This Markov chain does not take into account the first digit, so it has broken the SCF structure. The performance of this model was lower than the zip code structure model of (14).

Finally, the "backward-forward" model in which the history of any digit is composed of its left and right side digits has also been tested. However, this latter model has worsened the results.

TABLE 3
Results on Four Different Sets Consistently Indicate the Benefits of the Markov Modeling Postprocessing (ZPP) Described in This Paper Compared to the Method That Does Not Make Use of Contextual Information (GSC)

ZIP Code images		GSC Precision	ZPP Precision
Set 1:	5,407	4741 (87.683%)	4,842 (89.551%)
Set 2:	5,378	4,552 (84.641%)	4,656 (86.575%)
Set 3:	5,533	4,665 (84.312%)	4,779 (86.373%)
Set 4:	3,878	3,087 (79.603%)	3,142 (81.021%)

5 EXPERIMENTS

We have experimented with various combinations of *triggers* and different orders of the Markov modeling. Not surprisingly, the best results were obtained for the modeling based on (14). It is indeed intuitively satisfying that the model which reflects our understanding of the correlation of digits in the zip code does perform the best. ZPP (Zip code Post Processing) software was tested on four different sets of zip code images that were automatically extracted by the Handwritten Address Interpretation system [8]. These four sets of images were extracted in approximately the same period of time where training was performed using the Delivery Point file. GSC refers to the zip code recognizer that is simply based on the recognition confidences of the GSC digit recognizer [7].

Table 3 shows a consistent 2 percent improvement on GSC. In fact, if the top two choice candidates are considered, the improvement margin is at 4 percent. The zip code candidates can be cross-verified by other entities in an address, such as the state and city name. Thus, the impact of ZPP can be maximized.

In most real world applications, the accuracy of a recognizer is measured at a desired operating point that control the error rate. A zip code recognizer that makes 16 percent error is not acceptable. However, real world applications, such as the address interpretation task for the U.S. Postal Service demands a very tight control on the error rate. The error rate of 1-2 percent can be achieved by

TABLE 4
ZPP Proves to Have a Better 10E+R Value Compared to GSC

Method	Minimum 10E+R	Error rate	Reject rate	Accept rate
ZPP:	39.9	2.6%	14%	86%
GSC:	43.1	2.8%	14.8%	85.2%

rejecting some of the zip code images where the ZPP model is not confident of the results. Thus, at a lower accept rate, the target error rates can be achieved. In the literature [7], [8], the trade-off between Rejects (R) and Errors (E) has been described by the quantity "10E+R." This implies that the cost of 10 rejects is the same as that of one error. The point on the curve that yields the minimum 10E+R value is taken as the operating point. Thus, any two systems can be compared by evaluating their lowest 10E+R. The system with a lower 10E+R value is a better system under the assumptions described. ZPP gives a better 10E+R value (39.9), as seen in Table 4. At this operating point (its most optimal), it accepts 86 percent of zip code images and is 97.4 percent correct in recognizing them since GSC is a digit recognizer with optimal threshold values for each digit. If at least one digit is rejected during testing, we then consider this zip code as rejected. Similarly, in the case where all digits are accepted, if at least one digit has an error, we consider this zip code to be an error.

6 Summary

We have presented a nonstationary Markov model that merges images (signal) and context (language) in a fully Bayesian framework. The results obtained are promising. The Delivery Point file of the U.S. Postal Service is used to generate a list of every valid zip code paired with the number of records in the zip code. In the absence of live mail distribution statistics that should reveal the volume of mail "sourced" and "destined" to each zip code, we have resorted to the next best alternative. The number of records that exist in a zip code, we believe, indirectly reflect on the volume of mail "destined" for a zip code. Common wisdom suggests that a zip with more delivery points (in terms of houses, apartments, etc.) receives more mail.

It must be pointed out that achieving a 16 percent error rate in zip code recognition requires that the digit recognizer accuracy be at 98 percent. To make an overall 2 percent error rate improvement at the zip code level (as we have), without changing the recognizer but by merely postprocessing the digit recognizers results is quite significant. It amounts to an effective $log_5 2 \approx 0.25\%$ improvement in the accuracy of digit recognition, which is considered significant in the literature for a recognizer performing at accuracy levels in the high 90s on real world data.

As future work, we are trying to build nonstationary hidden Markov models at shapes (or strokes) level for each digit. The states of the HMM are different clusters of "similar" shapes. However, we are facing an important problem due to the nonrobustness of these states. Hidden Markov models provide probability values that can naturally be embedded within this probabilistic postprocessor in a fully Bayesian framework.

References

[1] T.W. Anderson and L.A. Goodman, "Statistical Inference about Markov Chains," *Ann. Math Statistics,* vol. 28, pp. 89-110, 1957.
[2] D. Bouchaffra, V. Govindaraju, and S. Srihari, "A Methodology for Determining Probability of Correctness of Recognizer Scores," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* Santa Barbara, Calif., June 1998.
[3] D. Bouchaffra, E. Koontz, V. Kripasundar, R.K. Srihari, and S.N. Srihari, "Integrating Signal and Language Context to Improve Handwritten Phrase Recognition: Alternative Approaches," *Proc. Sixth Int'l Workshop Artificial Intelligence and Statistics,* Ft. Lauderdale, Fla., Jan. 1997.
[4] D. Bouchaffra, E. Koontz, V. Kripasundar, and R.K. Srihari, "Incorporating Diverse Information Sources in Handwriting Recognition PostProcessing," *Int'l J. Imaging Systems and Technology,* vol. 7, no. 4, June 1996.
[5] P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, and R.L. Mercer, "Class-Based n-Grams Models of Natural Language," *Computational Linguistics,* vol. 18, no. 4, pp. 467-479, 1992.
[6] R.L. Dobrushin, "The Description of Random Fields by Means of Conditional Probabilities and Conditions of Its Regularity," *Theory Probability Applications,* vol. 13, pp. 197-224, 1968.
[7] J. Favata and G. Srikantan, "A Multiple Feature/Resolution Approach to Handprinted Digit and Character Recognition," *Int'l J. Imaging Systems and Technology,* vol. 7, pp. 304-311, 1996.
[8] V. Govindaraju, A. Shekhawat, and S. Srihari, "Determining Delivery Point Code in US Mailstream," *Proc. IWFHR III,* 1993.
[9] T.M. Ha and H. Bunke, "Off-Line Handwritten Numeral String Recognition," internal report, Univ. of Berne, Institut für Informatik und Angewandte Mathematik Neubrückstr, Switzerland, 1997.
[10] J.J. Hull, "Incorporating Language Syntax in Visual Text Recognition with a Statistical Model," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 12, Dec. 1996.
[11] A. Jessop, *Informed Assessments.* Ellis Horwood, year?
[12] 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, 1997.
[13] G. Kim and V. Govindaraju, "Bankcheck Recognition Using Cross Validation of Courtesy and Legal Amount Fields," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 19, no. 4, pp. 366-379, 1997.
[14] F. Kimura and M. Sridhar, "Handwritten Numerical Recognition Based on Multiple Algorithms," *Pattern Recognition,* vol. 24, no. 10, pp. 969-983, 1991.
[15] A. Kundu and Y. He, "On Optimal Order in Modeling Sequence of Letters in Words in Common Language as a Markov Chain," *Pattern Recognition,* vol. 24, no. 7, pp. 603-608, 1991.
[16] M.Y. Chen, A. Kundu, and J. Zhou, "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, pp. 481-496, 1994.

[17] C. Oliver et al., "Optimal Order of Markov Models Applied to Bankchecks," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 19, no. 4, 1997.

[18] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, no. 2, pp. 257-286, 1989.

[19] Z. Shi and V. Govindaraju, "Segmentation and Recognition of Connected Handwritten Numeral Strings," *Pattern Recognition,* vol. 30, pp. 1,501-1,504, 1997.

[20] R. Shingal and G. Toussaint, " Experiments in Text Recognition with the Modified Viterbi Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, 1979.

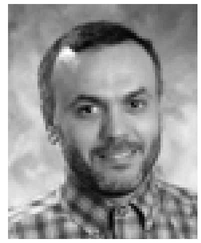
[21] M. Sridhar and A. Badreldin, "Context-Directed Segmentation Algorithm for Handwritten Numeral Strings," *Image and Vision Computing,* vol. 5, no. 1, pp. 3-9, 1997.

[22] L. Lam and C.Y. Suen, "Structural Classification and Relaxation Matching of Totally Unconstrained Handwritten Zip Code Numbers," *Pattern Recognition,* vol. 21, no. 19, pp. 19-31, year?

[23] C.Y. Suen et al., "Computer Recognition of Unconstrained Handwritten Numerals," *IEEE Proc.,* vol. 80, pp. 1,162-1,180, 1992.

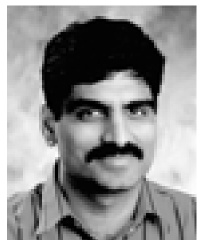
[24] R. Yee, M. Teo, and R. Shinghal, "A Hybrid Classifier for Recognizing Handwritten Numerals," *Proc. Fourth Int'l Conf. Document Analysis and Recognition,* Ulm, Germany, Aug. 1997.

[25] J. Zhou, Q. Gan, and C. Suen, "A High Performance Hand-Printed Numeral Recognition System," *Proc. Fourth Int'l Conf. Document Analysis and Recognition,* Ulm, Germany, Aug. 1997.



Djamel Bouchaffra received a master degree in mathematics from the University of Science and Technology of Algiers and a "DEA" and a PhD in computer science from the University of Grenoble (France). While working on his PhD, he was also working on the problem of parts of speech tagging using hidden Markov models. He held a postdoctoral position in Montreal, Canada, where he worked on "the theory of classifiers." He has been a research scientist at CEDAR since August 1995. His research is focused on statistical pattern recognition. He is currently working on character recognition. During his first two years at CEDAR, he worked on linguistic postprocessing using Bayesian models. He has had papers on designing and developing several language models published in peer conferences and journals (visit: www.cedar.buffalo.edu/~bouchaff). He was the technical lead on several federally sponsored projects.

Dr. Bouchaffra chaired the session "Object Classification" at the International Conference on Imaging Science, Systems and Technology (CISST'99). He is a member of the IEEE.



Venu Govindaraju received a PhD in computer science from the State University of New York at Buffalo in 1992. He has coauthored more than 90 technical papers in various International journals and conferences and has one U.S. patent. He is currently the associate director of CEDAR and concurrently holds the research associate professorship in the Department of Computer Science and Engineering, State University of New York at Buffalo. 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 researcher on several federally sponsored and industry sponsored projects. He is presently leading multiple projects on postal applications. He is a senior member of the IEEE.

Sargur N. Srihari received a BE in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1970, and a PhD in computer and information science from the Ohio State University in 1976. He is presently a University Distinguished Professor of SUNY. He was acting chairman of the Computer Science Department of SUNY at Buffalo during 1987-88. He is presently a faculty member in the Computer Science and Engineering Department. He is the founding director of CEDAR, Center of Excellence for Document Analysis and Recognition, located at the State University of New York at Buffalo. The center's work is in developing computer systems for reading of text (handwriting and print) and document understanding. The center has had several hundred individuals participate in its program in more than a decade. The work of CEDAR has led to systems depoloyed for the United States Postal Service, Australia Post, and the Internal Revenue Service. Dr. Srihari is a fellow of the IEEE, an editor-in-chief of the *International Journal of Document Analysis and Recognition*, and general chair of the Fifth International Conference on Document Analysis and Recognition held in September 1999.