Compréhension du texte écrit a la mains structurée spatiale
Understanding Spatially Structured Handwritten Text

Edward Cohen, Jonathan J. Hull, and Sargur N. Srihari

RÉSUMÉ -

Comprendre un block de text écrit à la main c'est l'en mettre en correspondence avec une représentation sémantique. Nous décrivons une approche pour lire un block de texte écrit à la main quand il y a certaines constraintes vagues dans la disposition typographique et le syntax du texte. Un système pour lire des addresses postales écrit à la main est décrit comme une instance.

Mots clés :

Reconnaissance de caracteres, reconnaissance de l'écriture à la main, systèm des classifieurs.

ABSTRACT

A method for mapping a block of handwritten text into a symbolic representation is described. It is assumed that certain loose constraints are placed on the spatial layout and syntax of the text. Early recognition of primitives guides the location of syntactic components. The instance of reading handwritten postal addresses is described, where the symbolic representation is a digit string (ZIP Code).

Keywords :

Handwriting Recognition, Character Recognition, Classifier Systems

Center for Document Analysis and Recognition
State University of New York at Buffalo
226 Bell Hall, Buffalo, New York 14260
telephone: (716) 636-3191 fax: (716) 636-3966

Internet Addresses:
edcohen@cs.buffalo.edu
hull@cs.buffalo.edu
srihari@cs.buffalo.edu

1. Introduction

Understanding the message conveyed by the image of a block of text involves both recognition and comprehension. The two processes can be disjoint for machine-printed text and highly constrained handwriting because recognition is sufficient to disambiguate uncertainties due to the small degree of variability in the input patterns. Constraints that greatly reduce handwriting variability can take various forms: preprinted boxes to limit the size and location of characters or digits, guidelines to specify location of words, suggestions for forming or joining letters, the presence of ligatures in cursively written words, rigidly fixed syntax, no spelling errors, etc. Strengthening constraints makes the recognition process easier and reduces its dependence on comprehension.

Comprehension must be used when the writing is unconstrained (or very loosely constrained). When handwritten text is unconstrained, no restrictions are placed on the writing style or implements used. A system that reads unconstrained writing must compensate for many factors that affect text appearance (e.g., variations in writing styles, size of text, writing implement used, writing surface, digitization, and thresholding).

We can explore the role of context and comprehension experimentally, by building systems in which the domain rather than the handwriting style is constrained. By placing restrictions on spatial layout, the problem of using comprehension to extract the information from the text becomes tractable. Examples of constrained domains in which handwriting is often present are: a form, an address on an envelope, a bank check, a credit card slip, a drug prescription, etc. Our research explores issues in comprehending unconstrained handwriting in constrained domains by describing a general approach and applying it to the domain of United States handwritten postal addresses.

This paper is organized as follows. Section 2 discusses previous work in reading handwriting in a constrained domain. Section 3 describes a general approach to "understand" unconstrained handwritten text in certain limited domains. Sections 4 and 5 describe the postal address domain and how the general approach was implemented to extract ZIP Code information from handwritten addresses. Section 6 presents a detailed example of how the system processes a handwritten address example. Section 7 discusses performance and refinements to the system.

2. Previous Work

Computational reading of handwritten words and digits is an important but difficult task that has a large literature; a tutorial and key papers can be found in [8]. However, most efforts have not utilized substantial amounts of available context. Typically, context is used only in the form of spelling information to compensate for errors in character recognition. While this type of context can offer substantial improvements (over not using it), much more context is often available. Global context has been used in the domain of reading machine printed chess games [1], but many issues unique to reading handwritten text need not be addressed in reading machine printed text. Two other research projects that use contextual information (which included more than spelling information) for analyzing handwritten text are described in [3, 4]. These projects were both concerned with reading hand-printed FORTRAN coding sheets and are described below.

In [4], all the character-recognition information is processed bottom-up. For each character in the image, the character-recognition algorithm provides one or more choices with confidence values for each choice [6]. Using this character-recognition information and connected component location information, contextual analysis is applied to locate and correct recognition errors. Combinatorial explosion that is possible with multiple recognition values is avoided by making assumptions about the data; e.g., that character recognition is fairly accurate (the correct

character recognition result was always included in the list of possible character values) and that the number of characters is known from positional data. The authors state that much more elaborate techniques are required for very poor data (such as unconstrained handwritten text).

Bornat, Brady, and Wielinga [2,3] discuss control issues for computer interpretation of handwritten FORTRAN coding sheets and describe techniques for combining character-recognition information with high level FORTRAN information. Input is a FORTRAN coding sheet image with certain restrictions on size, neatness, and programming correctness.

Frames are used to store character-recognition information, including several hypotheses for each character. Frames keep track of information obtained, information desired, and whether or not a conclusion is reached. A conclusion is reached if most character feature requirements are satisfied or if syntax or domain knowledge can identify the character. Much of their work involves improving character recognition by reexamining previous recognition results (e.g., if a *T* was not recognized because its strokes did not touch and additional evidence indicated that the character was a *T*, the strokes would be extended to form a recognizable *T*). They also use confidence information.

Our approach does not re-recognize characters. Instead, available character-recognition results and context are used to identify words; e.g., if context rules out all but one choice for a word, then word recognition has succeeded even if most of its characters have not been identified. Both FORTRAN coding sheet research projects constrain the writing style which reduces the need for context. Our approach is to let the writing style be unconstrained, but to use contextual information from the domain to assist in extracting the desired semantic information from the text.

We describe a system that can use substantial amounts of context. Even though the system is only partially implemented, we have demonstrated with a core system substantial improvements over previous approaches. We believe the methods described in this paper are applicable to a variety of domains where handwriting style is unconstrained.

3. General Approach

Experiments with unconstrained handwriting in handwritten address blocks have confirmed that straightforward bottom-up approaches are insufficient to solve the reading problem. A high-performance system must compensate for the following difficulties:

(1) Unreliable character recognition results.

(2) Non-text information that may appear in the image (e.g., underlining).

(3) Uneven spacing between text lines and overlapping or touching components from adjacent lines.

(4) Word isolation uncertainties (i.e., selection of image components so that the component set contains one complete word with no extraneous components is not always accurate).

(5) Syntax variations hinder correct parsing of text lines; some writers use suggested syntax only as a rough guideline (e.g., punctuation is often omitted or confused with image noise).

(6) Semantic uncertainties can make the choice of proper meanings difficult (e.g., 14215 may be a ZIP Code, box number, street number, etc.).

Our approach first develops a global description of the text and then verifies the hypotheses created from the global description. This results in a bottom-up and top-down approach (see Figure 1). Bottom-up information is used to create a global description and top-down information focuses attention on areas of the image that need further investigation.

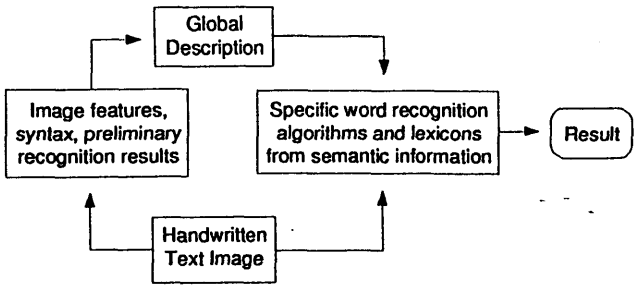


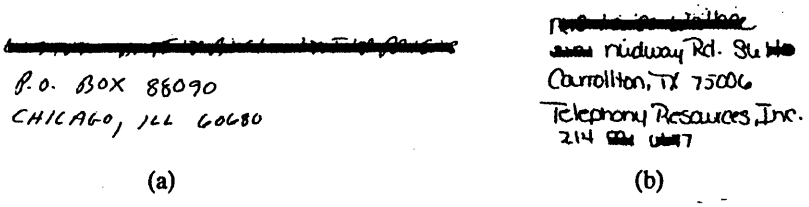
Figure 1. Conceptual control flow of reading process.

This approach builds a global description of a handwritten image by using spatial information, recognition of isolated characters and digits, and syntax. The global description contains all the high level information that is currently "known" about the image. Using the global structure, we can predict where words (which may not have been recognized) are expected to be found in the image. This recognition is driven top-down, creating hypotheses of the word types and then using signals from the image to confirm or deny the hypotheses. Verification may identify the words or simply determine that the words have the proper size and shape characteristics.

4. General Approach Applied to Handwritten Addresses - Background

Using this approach, we have developed a system that determines ZIP Codes for handwritten addresses. The present system uses recognition results for the ZIP Code in the image and the state name to determine a 5-digit ZIP Code (we also use punctuation, word position, and rough estimates of word shape to determine the syntax of the address image text). The post office box number and the street address are used to determine the 4-digit add-on (which creates a more specific ZIP Code).

Handwritten ZIP Code Recognition (HZR) is difficult due to wide variations in characters and digits as well as in address structure. Some characteristics are illustrated by the address block examples taken from live mail shown in Figure 2. Figure 2(a) shows a neat hand-printed address block in which the ZIP Code digits are prototypical. However, the address contains a P.O. box number which is also a string of digits. Figure 2(b) is a hand-printed address block where the ZIP Code is not in the last line, which makes the ZIP Code location problem non-trivial.



(a) (b)

Figure 2. Examples of handwritten address blocks extracted from live mail. Addressee names are intentionally blacked out.

An address has several natural constraints that can be exploited in locating and recognizing the digits of the ZIP Code. If the state name is known, we can determine the first digit of the ZIP Code and restrict the second digit to at most five choices. If the street address is also known, we can precisely determine the complete nine-digit ZIP Code for that address. An address also has a spatial syntax that can be used to locate function words (e.g., street, city, state) [7]. The following section describes the details of our current HZR system (HZRS).

5. HZRS Description

The control flow in the current HZRS is based upon our general theory of reading unconstrained handwritten text in limited domains. There are ten operational modules in the system.

A. Preprocessing;

B. Line segmentation based on positional information;

C. Feature identification;

D. Text line segmentation (combining B & C);

E. Word segmentation;

F. Word classification;

G. Grading match of word classification and syntax;

H. Recognizing words that have high syntax-classification match;

I. Using semantics to constrain word recognition;

J. If a reliable semantic evaluation is found, deriving an interpretation, otherwise, returning to step H until no reliable syntax-classification matches are found;

A brief overview of the entire system is now given. The HZRS takes a grey-level address block image as input. The image is binarized using a thresholding technique. Borders and underlines are removed from the image.

An initial line segmentation is performed. Connected components are placed into text lines based only on positional information. If connected components cannot be placed into single lines due to insufficient information and/or due to components from adjacent lines touching, connected components are assigned to a set of possible text lines.

Next, each connected component is identified (based on its size and shape) as either a dash, a comma, a disconnected 5-hat (the vertical bar on the digit 5 may be disconnected from the rest of the digit), or none of these. Each connected component is also identified as a digit or a non-digit. A connected component is identified as a digit if it is within size thresholds, and is recognized with sufficient confidence by the digit recognition algorithms. Note: there may be multiple

identifications (e.g., a comma and the digit 1 may have identical shapes), but these multiple identifications will be reduced after positional information is considered.

When positional information and identification are combined, a more accurate interpretation of the text line structure in the address is achieved. At this time, connected components can be moved to different lines (e.g., a comma may be moved from the line to which it is spatially closest to the line above). Connected components that span more than one text line may be placed into a single text line or the connected components may be split into two or more components and placed in separate text lines. At this point, we will assume that all components have been placed in their proper lines; however, some flexibility in the syntax (described later) allows us to compensate for improper line placement.

Positional information and identification are combined to determine multiple word segmentation hypotheses. Each text line can be divided into one to five words, and multiple hypotheses can be created for each number of words (i.e., a text line can be divided into two words several different ways depending on the location of the hypothesized word break in the line).

Each hypothesized word is tentatively classified. Words can presently be classified as *city*, *text*, *state-abbreviation*, *digits*, *9-digit ZIP+4 Codes*, *digits-dash* (digits followed by a dash), *P.O.*, the word "*box*" (for P.O. box numbers), *noise*, and *barcodes*. Words can have multiple classifications (e.g., a word can be identified as *state-abbreviation*, *text* and *3-digits*).

All hypothesized words and word classifications are examined to find consistent classifications that match the expected syntax. For instance, if a line contained 3 words identified as *city*, *text*, and *5-digits* respectively, the words in the line would match the *city* + *state* + *ZIP Code* syntax.

Based on statistical analysis of information location in mail pieces [5], we have determined the likely syntactic constructions that are found in handwritten addresses. Although the most likely syntax of an address has the bottom line containing the city, state, and ZIP Code, many other address syntaxes are possible. For instance, line 1 (our line numbering begins at the bottom of the address) may contain a ZIP Code and line 2 may contain the city and state name. Similarly, line 1 may contain an attention line (e.g., "Attn: Joe Smith"), line 2 may contain state and ZIP Code, and line 3 may contain the city. Matches between syntax, word groupings, and word classifications are determined and the most consistent matches are examined in order to find likely ZIP Code and state name candidates.

Words that match the syntax of a ZIP Code are selected as ZIP Code candidates. Depending on the number of consistent word classifications, zero or more ZIP Code candidates are selected. The system processes up to three candidates to determine the ZIP Code of the address.

To recognize a candidate, the ZIP Code segmentation program divides it into either five or nine digits. Four digit recognition algorithms are then applied and their results are combined to determine the identity of each segmented digit (due to segmentation, these identities may be different than the original connected component identities). After the ZIP Code candidate is processed, the state name candidate is recognized. Our current character recognition algorithms only recognize hand-printed (vs. cursive) state names and state abbreviations. State name recognition information is used to compensate for ZIP Code recognition uncertainties.

Once the ZIP Code is recognized, we apply semantic information to determine the recognition reliability. If all digits are identified with sufficient confidence, the ZIP Code candidate is assigned a confidence value based on the recognition confidence of the individual digits and the amount of segmentation required. Furthermore, if the candidate has nine digits, the confidence is increased, because any candidate with nine recognized digits is likely the correct ZIP Code. If the state name is recognized and the first two ZIP Code digits are not recognized with high confidence, a ZIP Code directory (containing state name listings) is used to assign high

confidence digit recognition (consistent with the state name) to replace the low confidence ZIP Code digit recognition results. If any of the digits is still not recognized, the ZIP Code candidate is assigned a confidence value of –1. The confidence of the syntax match is also used to compute an overall confidence in the ZIP Code candidate. In addition, if recognized digits are not in a directory of valid US ZIP Codes, the ZIP Code candidate is assigned a confidence value of –1.

Up to three ZIP Code candidates are examined. The first candidate found whose confidence is greater than a threshold is selected as the ZIP Code for the address block.

If a valid 5-digit ZIP Code is found, more sophisticated semantic information can be used. For instance, reading the street address line allows us to convert a 5-digit ZIP Code to a 9-digit ZIP+4 Code. The extra four digits give a more accurate semantic evaluation of the delivery address for the mailpiece.

The street address line is located by searching for a line in the address that contains *digits* followed by *text*. By recognizing the street number and the 5-digit ZIP Code, we can determine a dictionary of street names. The dictionary and the street name image are used to rank the dictionary entries. By taking the top-ranked dictionary entry, we can get its associated 4-digit add-on and determine the ZIP+4 Code. Our current system is able to locate and recognize over 20% of the street address lines correctly.

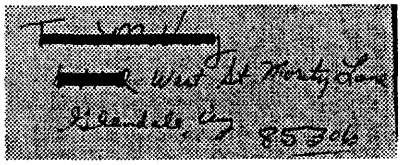
6. An Example of HZRS on Address Images

This example of the complete system's operation (Figures 3) shows how the system can recover after initial ZIP Code location fails. The grey-level image is shown in (a), and the thresholded image in (b). The underline removal algorithm is run, but results in no change. Connected component analysis is performed to segment the address into text lines and words.

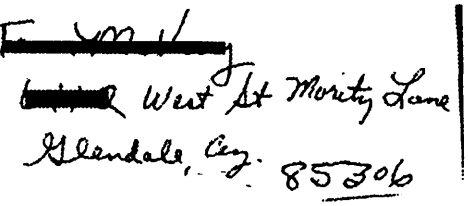
In this example, the first match to the *city-state-ZIP* syntax is incorrect. Figure 3(c) shows that the state name includes part of the ZIP Code. This city-state-ZIPCode syntax was chosen because the digits *5* and *3* in the ZIP Code are touching, which causes the digit estimator to be in error. Note that the touching digits are large compared to the other digits in the ZIP Code. Since the touching digits are not recognized during the connected component analysis (only inter-line segmentation has been performed), the digit estimator determines that this two-digit connected component contains 3 digits. At the same time, we note that the confidence assigned to the *5-digits* classification for the ZIP Code is only 0.24. However, the high classification confidences for city and state abbreviation match this word grouping to the *city-state-ZIP* syntax.

This ZIP Code candidate is segmented (d) and the digit recognition results are shown in (e). The segmentation results show the digit locations that required forced segmentation (i.e., in this example segmentation was required between the first three digits of the image). This ZIP Code candidate is rejected since one of the digits was not recognized (state name recognition does not succeed on the cursive state name), so the HZRS selects the next best candidate.

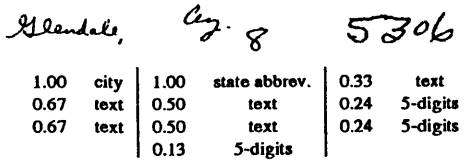
The second best match to the *city-state-ZIP* syntax is shown in (f). This was selected second since the last word, *6-digits*, is not a good match to an expected ZIP Code (no ZIP Codes have 6 digits). However, the system realizes that a digit estimation error may have occurred and allows the syntactic match to proceed. The ZIP Code candidate is selected and segmented in (g). The digit recognition and segmentation results are shown in (h). The segmentation results show that forced segmentation was required between the second and third digits of the ZIP Code image. Still, the digit recognition confidence is high and the ZIP Code is valid compared to the USPS (United States Postal Service) ZIP Code directory, so the system returns *85306* as the result.



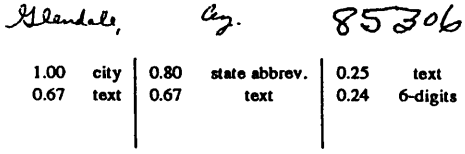
(a)



(b)

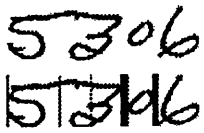


1.00	city	1.00	state abbrev.	0.33	text
0.67	text	0.50	text	0.24	5-digits
0.67	text	0.50	text	0.24	5-digits
		0.13	5-digits		

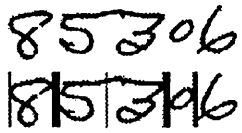


1.00	city	0.80	state abbrev.	0.25	text
0.67	text	0.67	text	0.24	6-digits

(c)



(f)



(d)

5 0.74
Reject
3 0.50
0 0.99
6 0.99

Segment Results
1 2 3

(e)

(g)

8 0.99
5 0.97
3 0.50
0 0.99
6 0.99

Segment Results
2 3

(h)

Figure 3. Steps in processing a handwritten address: (a) grey-level image, (b) thresholded image, (c) first match for segmented bottom line to *city-state-ZIP* syntax, (d) first ZIP Code candidate and its segmentation, (e) digit recognition results of first ZIP Code candidate shown as pairs (recognized digit, confidence) and segmentation results, (f) second match for segmented bottom line to *city-state-ZIP* syntax, (g) second ZIP Code candidate and its segmentation, (h) digit recognition and segmentation results of the second ZIP Code candidate. Note: areas have been blacked out to preserve confidentiality.

7. Discussion

We have presented a systematic approach for extracting semantic information from unconstrained handwritten text in limited domains. This approach uses bottom-up information to develop a global description and then uses the global description to suggest hypotheses that can be verified.

The HZRS research is ongoing and system performance is being regularly improved. As of March 1991, the system correctly recognizes 75.6% of the ZIP Codes that are present on mail

pieces with an error rate of 1.6%. These experiments were preformed on a set of 508 live mail images chosen by the USPS. Furthermore, by reading street addresses and P.O. box numbers, we are able to increase the number of 4-digit add-ons assigned from 4% to 25%.

We are actively developing more sophisticated handwritten character recognition algorithms. Additional character and word recognition information will allow us to use much more of the semantic information available in the addresses (such as city names). Other research efforts will include applying our approach to other domains, such as bank checks.

Acknowledgment

The authors thank Carl O'Connor of the Office of Advanced Technology of the USPS and John Tan of Arthur D. Little Inc. for their guidance and support of this work. In addition, we would like to thank Alan Commike, Peter Cullen, Chih-Chau Kuan, and John Favata for contributing ideas and code towards portions of the HZR project. This work was supported by the USPS Office of Advanced Technology under Task Order 104230-86M-3990.

REFERENCES

1. H. S. Baird and K. Thompson, "Reading chess", *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 6 (June 1990), 552-559.
2. R. Bornat and J. M. Brady, "Using knowledge in the computer interpretation of handwritten FORTRAN coding sheets", *International Journal of Man-Machine Studies 8*, 1 (Jan 1976), 13-27.
3. J. M. Brady and B. J. Wielinga, "Reading the writing on the wall", in *Computer Vision Systems*, A. R. Hanson and E. Riseman (editor), Academic Press, New York, 1978, 283-299.
4. R. O. Duda and P. E. Hart, "Experiments in the recognition of hand-printed text: Part II context analysis", *AFIPS Conference Proceedings 33* (1968), 1139-1149.
5. J. J. Hull, D. Lee and S. N. Srihari, "Characteristics of handwritten mail addresses: A statistical study for developing an automatic ZIP code recognition system", Technical Report 88-06, Department of Computer Science, State University of New York at Buffalo, Mar 1988.
6. J. H. Munson, "Experiments in the Recognition of Hand-printed Text: Part 1 - Character Recognition", *Proceeding of the Fall Joint Computer Conference 33* (1968), 1125-1136.
7. M. Prussak and J. J. Hull, "A multi-level pattern matching method for text image parsing", *IEEE Computer Society Conference on Artificial Intelligence Applications*, Miami Beach, FL, Feb 1991, 183-189.
8. S. N. Srihari, *Computer text recognition and error correction*, IEEE Computer Society Press, Silver Spring, Maryland, 1984. (ISBN 0-8186-0579-0).