AI Magazine Volume 8 Number 4 (1987) (© AAAI)

Recognizing Address Blocks on Mail Pieces: Specialized Tools and Problem-Solving Architecture

Sargur N. Srihari, Ching-Huei Wang, Paul W. Palumbo, and Jonathan J. Hull

An important task in postal automation technology is determining the position and orientation of the destination address block in the image of a mail piece such as a letter, magazine, or parcel. The corresponding subimage is then presented to a human operator or a machine reader (optical character reader) that can read the zip code and, if necessary, other address information and direct the mail piece to the appropriate sorting bin Analysis of physical characteristics of mail pieces indicates that in order to automate the address-finding task, several different image analysis operations are necessary Some examples are locating a rectangular white address label on a multicolor background, progressively grouping characters into text lines and text lines into text blocks, eliminating candidate regions by specialized detectors (for example, detecting regions such as postage stamps), and identifying handwritten regions. Described here are several such operations, their utility as predicted by statistics of mail piece characteristics, and the results of applying the operations to a task set of mail piece images A problem-solving architecture based on the blackboard model of problem solving for appropriately invoking the tools and combining their results is described

A typical sectional center facility of the U.S. Postal Service, such as that in Buffalo, New York, handles about 1 million pieces of mail each working night. The mail consists of several categories, for example, letters, flats (which are large envelopes, magazines, and other flat pieces), and irregular parcels and pieces (IPPs). Most of the effort in handling mail is sorting it into bins according to zip code.

Currently, about 50 percent of the letters are automatically sorted by a postal optical character reader (OCR). Given a stack of correctly faced letters, the OCR visually reads the zip code on each letter and drops it, via a letter transport mechanism, into the appropriate zip code-labeled bin or the reject bin. A letter is rejected by the OCR for reasons such as nonstandard location of the address block or handwritten address. Most of the letters that are not automatically sorted, as well as flats, are semiautomatically handled; that is, the pieces are presented by a transport mechanism to a human operator who manually keys in the destination zip code. In order to improve the performance of postal OCRs and the speed of semiautomatic handling of mail pieces, the task of automatic identification of the position and orientation of the address block has been identified as important (Srihari et al. 1985).

The role of an address block location subsystem (ABLS) in a zip code recognition system is shown in figure 1. The subsystem takes several types of images of a single mail piece as input and produces as output a list of candidate blocks (image regions), their orientations, and the degrees of support that could be associated with the destination address block. In principle, the images can be multispectral and can be one or more of the following: gray scale, color (three images corresponding to red, green, and blue filters), infrared, or color under ultraviolet illumination. The subimage corresponding to each candidate block is presented as a high-resolution, gray-scale image (approximately 300 pixels per inch) to either an OCR or a human reader that attempts to locate and read the zip code within the address block.

The Document Image Understanding Group at the State University of New York at Buffalo has been studying this problem of address block location for letters, flats, and IPPs with a view toward developing a robust address block location system. This article describes the methodology and preliminary results of a new system for address block location. The first section describes some of the types of knowledge that could be exploited. Computational tools required to identify and exploit several different image characteristics of a given mail piece are described in the following section. Integration of the various tools into a problem-solving architecture is then discussed. The article concludes with a presentation of performance issues; this section includes examples of processing, the current rate of success on a test set, and software-hardware implementation issues.

Knowledge Needed for Robust Address Block Location

A typical mail piece has several regions or blocks that are meaningful to mail processing, for example, address blocks (destination and return), postage (meter mark or stamp) as well as extraneous blocks

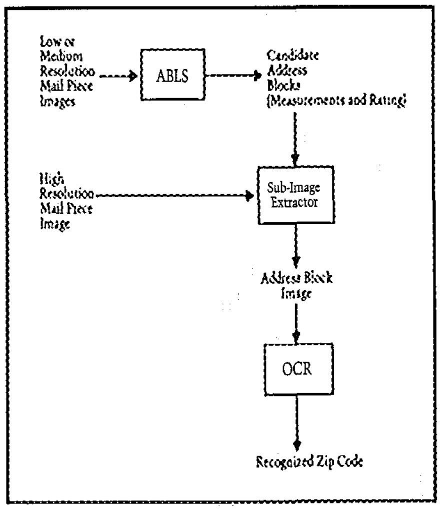
Low or Medium Resolution Mail Piece Images
ABLS
Candidate Address Blocks (Measurements and Rating)
High Resolution Mail Piece Image
Sub-Image Extractor
Address Block Image
OCR
Recognized Zip Code

Figure 1. The Role of an Address Block Location Subsystem in Address Recognition

such as advertising text, logos, and graphics. Several mail pieces with different levels of complexity in determining the address block are shown in figure 2. The significant blocks have certain characteristic properties and relationships that can be derived from images and used to assist in the location of the destination address in other images. Therefore, we conducted a knowledge-acquisition phase to determine the relevant characteristics of a sample set of mail pieces.

Four databases of physical characteristics of mail pieces (GTRI 1985-86) corresponding to the categories of IPPs, flats, letter mail, and manual letter mail were utilized. Each database corresponds to 3000 pieces, representing a wide sample for each case. Each database was analyzed using the statistical package for the social sciences (SPSS) and compiled into a mail statistics database (MSD) (Srihari et al. 1986). MSD was examined to extract heuristics, some of which follow. The probabilities referred to in the rules were computed as frequency ratios.

R1: In the case of letter mail, the address block is in the lower two-thirds of the mail piece—Today, commercial letter mail sorting machines largely assume a standard position for the address block (Srihari et al. 1985). This assumption is appropriate in the case of carefully prepared letter mail (such as shown in figure 2A). The location of the address block on correctly oriented letter mail is as follows: If the image of a letter mail piece is divided into nine equal rectangular parts (using a 3 x 3 grid), with probability .997, the center of the destination address block is in the lower two-thirds of the mail piece. Using the following grid location on the images, we found the center of the destination address block with the probability specified within each grid square:

.001	.002	.000
.070	.475	.056
.179	.191	.025

R2: Letter mail orientation can be determined by detecting postage fluorescence—This statement is based on three facts: (1) a letter mail piece has postage in the upper right corner with probability .99, (2) postage is fluorescent with probability .95, and (3) the number of fluorescent fields is small (no fields with probability .05, one field with probability .7, two fields with probability .22, and three fields with probability .03).

R3: In the case of flats, the color of the address background is often different from the color of the item—The color of the address background differs from the item color with the following probabilities: letters .2, flats .65, and IPPs .54. An item has a white address background with the following probabilities: letters .71, flats .84, and IPPs .73. Thus, a tool that extracts regions of a given color (typically, white) is extremely useful for extracting candidate regions in color images of flats and IPPs.

R4: Handwritten and machine-printed addresses have different characteristics—Destination address features on a mail piece are vastly different when the address is printed by hand as opposed to machine. If the address is hand generated, then the text is larger, appears in an unconventional font, is printed directly on the mail piece, and tends to have skewed address lines. Similarly, the arrangement of the address is also different when the address is printed by hand versus machine. If the destination address is printed by hand, then the address lines most likely have a zip code on a separate line; are not left justified; and do not have the city, state, and zip code all on the same line.

R5: Postage type is correlated to whether an address is handwritten or machine printed–A method that is useful in determining whether an address is printed by hand or machine is examining the type of postage, which usually appears in the upper right corner of the piece. Most mail pieces with a machine-printed address have a meter mark as postage; mail pieces with a hand-generated address usually have a stamp as postage. The probability of a letter bearing a meter mark is .337; the probability of a stamp appearing is .612. Conversely, for a letter with a machine-printed address, the probability of a meter mark is .7, and the probability of a stamp is .24.

Tools

The heuristics listed in the previous section suggest that the design of ABLS consist of several specialized tools that are appropriately deployed. Rule R2 suggests the need for a tool to detect postage fluorescence, rule R3 a tool for isolating blocks of a certain color, rule R4 for discriminating between handwriting and print, and so on. The process for developing a tool is as follows: (1) The MSD is examined to identify the objective of the tool. The context in which the tool is most useful is also determined. (2) An algorithm for the tool is developed. It is implemented such that its invocation and completion parameters are appropriate for the overall control structure. (3) The tool is experimentally run under various conditions. Estimated cost and parameter settings under various conditions are compiled into knowledge rules. (4) The tool performance is evaluated under various conditions. Combining this evaluation with the performance predicted by MSD yields a utility measure for the tool.

Several types of input images for a given mail piece can be operated on by the image analysis tools, including gray scale, color, infrared, and ultraviolet illuminated images. Examples of tools that can be incorporated into ABLS are (1) gray-level adaptive

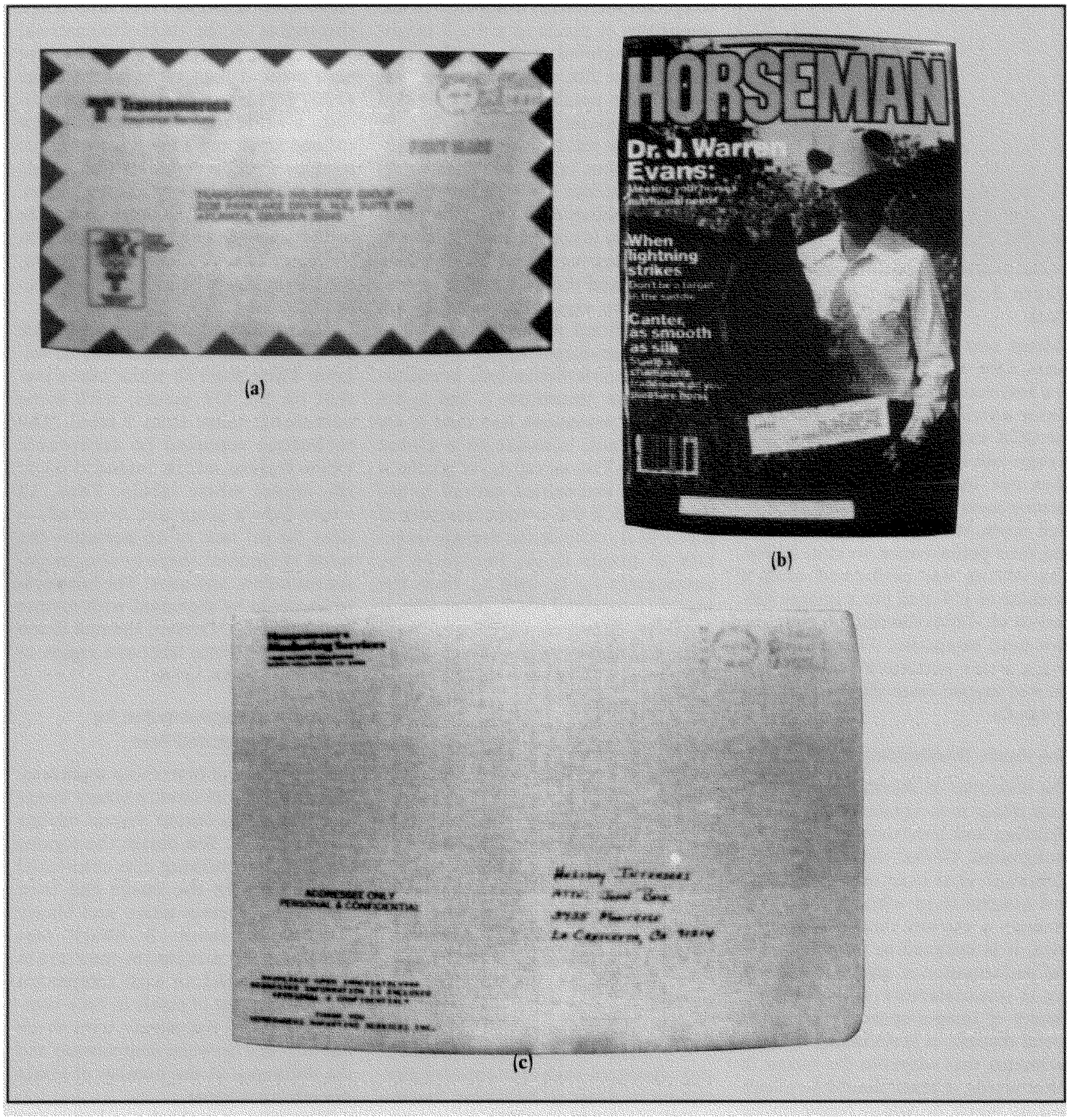

Figure 2. Examples of Mail Images with Different Levels of Complexity in Locating the Destination Address Block.
(A) A standard structure; (B) A randomly placed destination address; (C) An intermediate form where a hand-generated destination address coexists with several machine-generated text blocks

thresholding to convert a gray-level image into a binary image using local contrast; (2) color thresholding to extract white labels in a colored image; (3) a bottom-up segmenter to group machine-generated characters into words, lines, and blocks; (4) a regularity analyzer that discriminates between machine printing and handwriting; (5) an icon detector to detect the rectangular postal icon; (6) a bottom-up segmenter to group hand-generated thin strokes into lines and blocks; (7) a regularity analyzer that detects the likely regions of hand-generated text; (8) a shape analyzer to measure how rectangular a region is; (9) a texture discriminator for distinguishing the address block from miscellaneous text blocks; (10) a text reader, which maps iconic forms of text into their symbolic forms; and (11) an address syntax parser (Palumbo and Srihari 1986), which parses

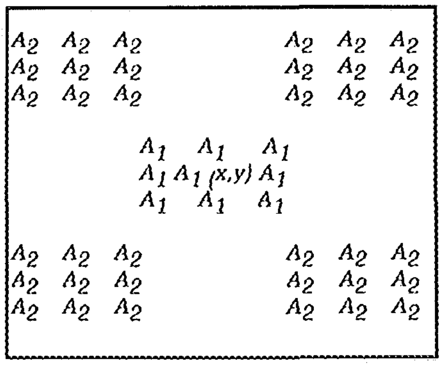

Figure 3. Adaptive Thresholding Mask.

address text into categories such as street, city, state, and so on. The use of a text reader and an address syntax parser entails a feedback path from the OCR subsystem to the ABLS subsystem (see figure 1). Currently, ABLS does not include OCR capability. Descriptions of the first five specialized tools, their parameter settings, and their performance (in situ) follow. The testing was performed using a database of 174 mail piece images (Srihari et al. 1986) distributed over the four mail categories. The images are not a strict statistical sample but skewed toward cases that are difficult to handle.

Gray-Scale Thresholding

The objective of thresholding a gray-scale image is to separate dark objects (printing and handwriting) from the background. *Global thresholding* is a technique that examines each pixel and assigns it to white if its value exceeds a certain threshold; otherwise, it is assigned to black. Because the performance of global thresholding is unsatisfactory with the wide variety of images encountered in the postal domain, it is necessary to use a technique that adapts to the image. In other words, a pixel should be classified as black or white based on its immediate surroundings rather than on global considerations. In the case of printed text images, one should not simply compare the value of a pixel to the local average around it in an n x n window but try and contrast text pixels and background pixels.

The adaptive thresholding algorithm (Palumbo, Swaminathan, and Srihari 1986) is as follows. The intensity $I(x, y)$ of a point (x, y) in the input image is functionally compared to the intensities of pixels in a 9 x 9 neighborhood centered at the pixel, and a binary value $Z(x, y)$ is output (see figure 3). The neighborhood is divided into two regions: A_1 and A_2. Area A_1 contains the pixels in the 3 x 3 neighborhood about (x, y), and an area A_2 contains the four 3 x 3 neighborhoods diagonally adjacent to A_1. This mask arrangement is constructed in order to capture foreground (character) pixels in area A_1 and the background in area A_2 and vice versa. Pixels in A_1 are most significant in the comparison to determine an output value.

In the implementation, predetermined static thresholds t_1 and t_2 are set so that intensities less than t_1 are always black (similar to a global threshold). The subset A_{2t} of pixels in A_2 whose intensities exceed t_2 are determined. If the average intensity of pixels in A_2 exceeds the average intensity of pixels in A_1 (weighted by parameters t_3, t_4, and t_5, then the pixel is considered black; otherwise, it is white. Based on experimentation with mail images, the values of the thresholds were set as follows:

$t_1 = 20$, $t_2 = 20$, $t_3 = .85$, $t_4 = 1$, and $t_5 = 0$. The algorithm is formally,

```
begin
  if (I(x, y) < $t_1$) then Z(x, y) := black ;
  else
    begin
      $a_1$ := average of pixels in area $A_1$ ;
      $A_{2t}$ := { (x, y) | (x, y) in $A_2$ and
          I(x, y) > $t_2$ } ;
      $a_2$ := average of pixels in area $A_{2t}$ ;
      if ( ($t_3$ * $a_2$ + $t_5$) > ($t_4$ * $a_1$) ) then
          Z(x, y) := black ;
      else Z(x, y) := white ;
    end ;
end ;
```

A comparison of the performance of this algorithm with conventional algorithms is given in Palumbo, Swaminathan, and Srihari (1986).

Color Thresholding

Paste-on address blocks on colorful magazine covers are usually identifiable because of their white color and size. Approximately 1.5 percent of all manual letters have labeled redirection address blocks that are yellow. The objective of color thresholding is to find labels of a certain color and size. Our method is to use fixed thresholds in the three-dimensional space of red, green, and blue values for each pixel. A high value (in the range of 70-255) for red, green, and blue indicates a white region. A high value for red and green and a low value for blue indicate a yellow region. A size criterion is used to accept or reject a region. For white regions, the height has to be in the range .5" to 6" and the width in the range 2" to 6". For yellow regions, the height has to be in the range .5" to 2".

The color thresholding performance on the test set of images was as follows: There were 75 white labels present on the 174 pieces, with some containing more than 1 label. The technique reported 76 acceptable white regions which included 63 of the actual white labels. Thus, 12 white labels were not detected—a miss (error) rate of 16 percent—the result of unusual shape or size, inconsistent colors, and glare. The false positives could be decreased with further shape analysis. Overall, the tool found the relevant white label as a region in 83 percent of the cases.

Bottom-Up Segmentation for Machine-Generated Text

The objective of bottom-up segmentation is to take as input a binary image and produce as output several regions of text blocks. You obtain the regions by first determining the connected components in the image and then performing certain unary and binary component tests. A unary test involves a minimum-maximum height and width of each component and the number of pixels in each component. Binary component tests check the distance between components and the difference in the number of pixels in each component.

These data are added to a data structure, a region adjacency graph (RAG), that associates a node with each primitive connected component. This process entails linking components of appropriate (unary condition) and similar size (binary condition) that are less than a fixed distance away from each other. Subsequently, each nonconnected RAG in the image is associated with a block concept in the conceptual data structure. This block information is added to this data structure as another node containing the block extent, the number of black pixels in the block, the number of components in the block, and the average skew of lines inside the block. Line information (a sequence of horizontal components inside a block) is also detected and added to the data structure. Line features include the line extent, the number of black pixels in the line, the number of components in the block, and the skew of the components in the line. In the building of the data structure, not only are the RAG relations between components maintained but also the conceptual relations between blocks and lines and between lines and components. This resulting (generally) hierarchical data structure is accessed by the higher-level control structure to determine which block is the destination address block. The control structure could also improve the conceptual data stored in the data structure.

Segmentation parameters are a minimum height of 1/12" (6 point), a minimum width of 1/15" (5 point), a maximum height of 1/4" (18 point), and a maximum width of 1/3" (24 point). The maximum space between components is 1/4" (18 point), the minimum pixels in a component. The minimum number of pixels in a component is equal to $p^2/1000$ and the maximum number of pixels in a component is equal to $p^2/2$, where p is the number of pixels per inch in image digitization.

The utility of such a technique is determined again by looking at the MSD. The probability that an address is formed of conventional nonlarge, nonlight, nonsmeared characters, parallel address lines, and nonhandwritten lines is as follows: letters .58, flats .8, and IPPs .61. When color thresholding and bottom-up segmentation are combined, the projected percentages of cases that can be handled increases to 64 percent for letters, 88 percent for flats, and 76 percent for IPPs.

The performance of the bottom-up segmentation tool was tested on 174 pieces thet consisted of 136 machine-generated destination address pieces and 38 hand-generated destination address pieces. For machine-generated destination address pieces, 101 were correctly segmented, 14 were segmented correctly but were missing the zip code, and 21 were incorrectly segmented. The failure on machine-generated destination address pieces was as a result of one of the following: light printing, no destination address, or large-size destination address. In the case of hand-generated destination address pieces, 9 pieces were correctly segmented, 3 pieces were correctly segmented but were missing the zip code, and 26 pieces were incorrectly segmented.

An analysis of the failure of the bottom-up segmentation tool revealed that an error was the result of hand-generated destination addresses in 77 percent of the cases for letters, 30 percent for flats, and 36 percent for IPPs. If the tool is able to handle hand-generated cases, then its projected success rate is 94 percent for letters, 98 percent for flats, and 84 percent for IPPs. This analysis indicates that color thresholding and bottom-up segmentation can potentially handle a high percentage of cases.

Handwriting—Machine Printing Discrimination

A useful tool is one that accepts a block of binary image data as input and determines whether that block contains predominantly machine-printed text by making certain global measurements on the block. (An example of a global measure is one that judges the regularity of black-white transitions in the horizontal or vertical direction of a text block.) This tool can be invoked in several different situations. First, it can be used for getting evidence on whether a given block contains handwriting, rather than machine printing. If it can be determined that the block contains handwriting, then it is highly likely to be an address block. Second, when it is determined that a given region contains machine printing or handwriting, then this information is useful for setting parameters for a closer bottom-up analysis (connected component grouping) of the block.

The discrimination technique involves the computation of three histograms. Each histogram is a plot of the number of occurrences of an event that occurs in the image versus a parameter n which is associated with determining the presence of the event. Each occurrence of the event is said to be computed by a filter. The first event consists of a 0-1 (black-to-white) transition horizontally separated from another 0-1 transition by n pixels and is sensitive to the width of each character. The second event is a 0-1 transition vertically separated from a 0-1 transition by n pixels and is sensitive to the spacing between each line of text. The third event is a 1-0 (white-to-black) transition vertically separated from a 1-0 (white-to-black) transition by n pixels and is sensitive to the height of each character.

The three histograms are jointly used to classify the region into one of two categories: machine print or handwriting and graphics. This classification is done using a three-dimensional feature space obtained by converting each histogram into a scalar value, as follows: A straight-line fit is computed for each histogram function, and the mean squared error value of the line is computed. A high error implies strong fluctuations in the histogram, which is characteristic of machine print. An example of the three histograms and their line fits for a block of machine-printed text is shown in figure 4.

The method was trained on 100 regions extracted from a (trial) database of 18 images. The method was tested on regions segmented from the database (174 images) by the bottom-up segmentation tool. The performance of the tool was 83 percent correct, 17 percent error. The majority of the errors consisted of machine-printed regions being classified as handwritten, an error that was in part the result of significant noise present in the images.

Icon Detection

The objective of icon detection is to find rectangles in the image. These rectangles can be postal icons, for example, a postage paid permit mark, a window that contains the address,

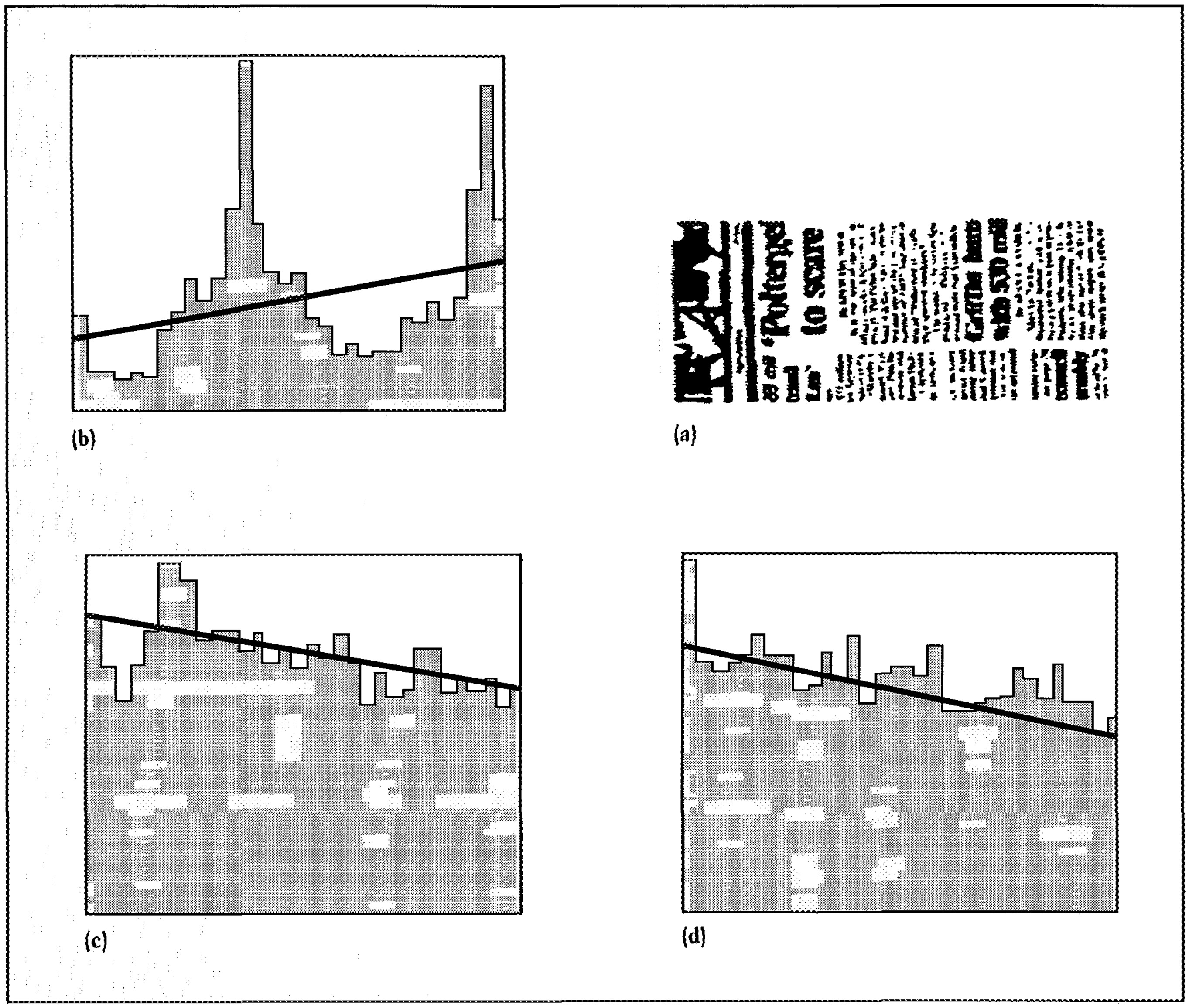

Figure 4. Print versus Nonprint Discrimination.
(A) Input image; (B) Filter 1 histogram; (C) Filter 2 histogram; (D) Filter 3 histogram Based on three scalar features, one corresponding to each histogram, the strength of classification is machine print 16, machine print 90 orientation at 90 72, handprint or graphics 12

or a box which is used to highlight text. Rectangle information can be used in several ways: to determine the orientation of the mail piece (because a postal icon is usually placed in the upper right corner) or to focus attention on the boxes that might be meant for finding the address. The algorithm consists of three steps. The first is edge detection of the gray-scale image, which involves a 5 x 5 Laplacian operator. The output is a black-and-white edge image. This image is then examined by a modified Hough transform in the second step. This transform detects line segments in the image and remembers the end points of them. The result of this step is a list of line segments found. The list contains length, line specifications in polar coordinates (Rho, Theta), and the coordinates (x,y) of the two end points. The third step uses this information to group parallel and perpendicular line segments into rectangles. The rectangles formed are reported.

The icon-detection tool was tested on images of 40 flats in the test database. Fifty three rectangles were present on the 40 flats, with some flats having no rectangles and others having one or more rectangles. The technique reported 41 rectangles present, of which 38 were actually in the images. These results mean a success rate of 72 percent (the likelihood of finding a rectangle) and a reliability rate of 92 percent (the confidence that a rectangle found is actually present in the image). All the errors occurred in 32 percent of the images; the technique is sensitive to image quality and consistency, which accounts for errors in these images. Further improvements are possible in this tool in regard to criteria for accepting lines as rectangles (broken sides, missing sides) and finding line segments (size of line, amount of noise acceptable).

Problem-Solving Architecture

When a large number of complex tools are present, it is necessary to judiciously plan their use. Because many image-processing tools are computationally intensive and slow, it is not feasible to let the system invoke all available tools to obtain evidence. Some tools are interdependent and cannot be invoked randomly.

An effective method in artificial intelligence to address the problem of integrating knowledge from various sources is the blackboard model of problem solving (its origins are in the HEARSAY-II speech understanding system (Erman et al. 1980); a tutorial survey is in Nii (1986)). The basic idea is to divide a complex problem into loosely coupled subtasks; each subtask is attacked by a specialized tool that is called a knowledge source (KS). A blackboard model is usually composed of three major components: KSs, the blackboard data structure, and the control mechanism (see figure 5). The KSs consist of a community of independent experts or tools that incrementally contribute information to the overall solution on the blackboard. Communication between KSs is exclusively through the blackboard. The blackboard data structure is a global database representing the current state of problem solving that also serves as a communication center between KSs. The control mechanism controls the invocation of KSs and is responsible for integrating information on the blackboard, which was contributed by various KSs, to determine if the current state of problem solving is sufficient to constitute a solution.

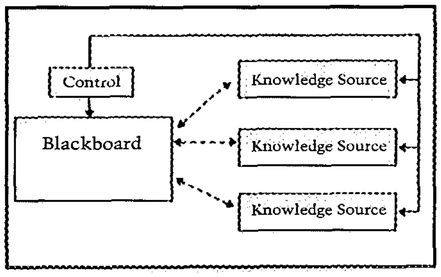

Figure 5. The Blackboard Model.

The blackboard of ABLS (see figure 6) contains a three-level hierarchy; spatial relationships between objects are on the highest level. At least three distinct object classes exist on the highest level: a machine-generated text block, a hand-generated text block, and graphic icons. A machine-generated text block is formed by grouping character components on the lowest level into text lines in an intermediate level and then grouping text lines into machine-generated text blocks. A hand-generated text block is formed by grouping thin strokes on the lowest level into a text line and then grouping text lines into hand-generated text blocks. Graphic icons have no levels below them and represent the regions that contain postal permit marks (circles, squares), postal icons, address pointers, or advertising graphics.

Components of ABLS

ABLS is composed of six major parts (see figure 7): MSD, a rule-based inference engine, the control data, the control mechanism, a blackboard, and a tool box. MSD contains the statistics of the geometric features of labels on mail pieces. The rule-based inference engine is used for forward or backward reasoning in various rule modules. The control data provide information for the control mechanism. They con-

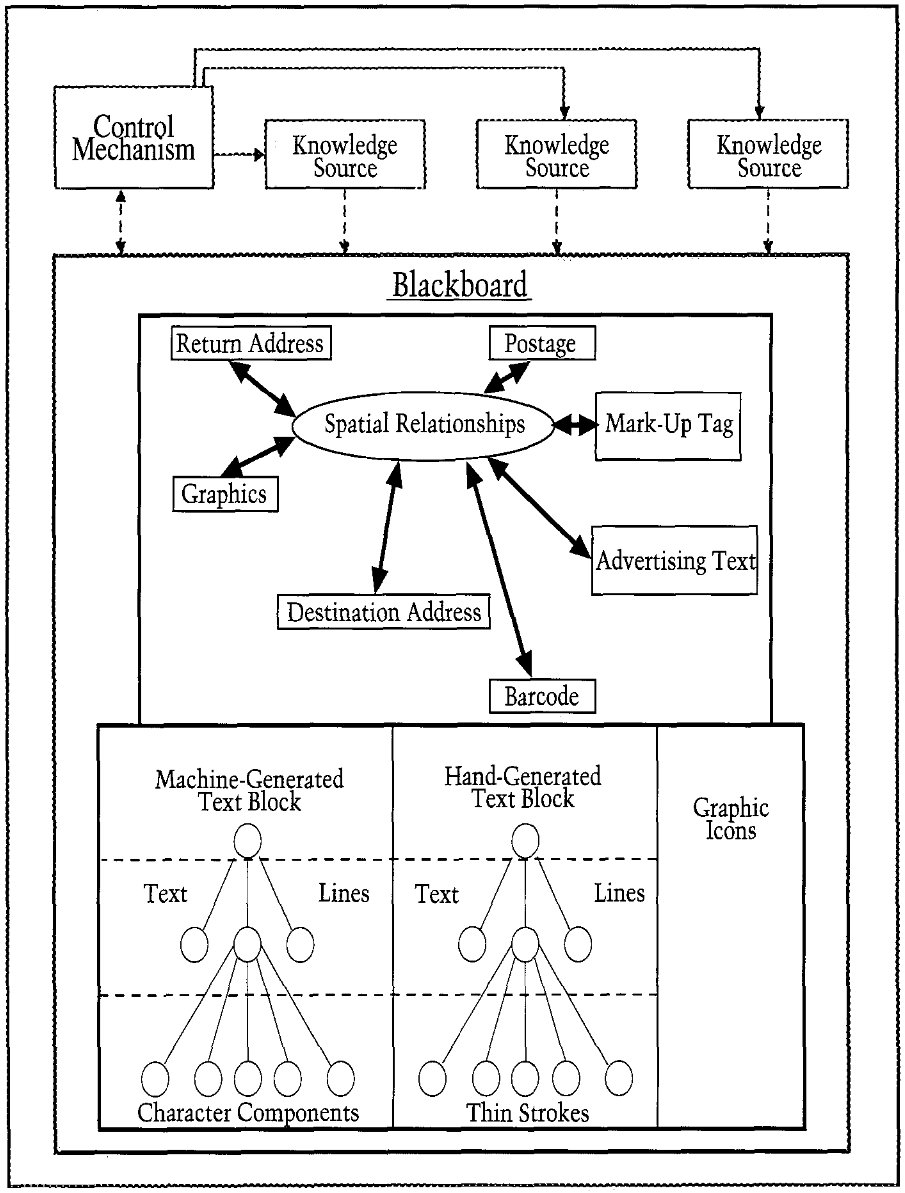

Figure 6. The Blackboard Framework of ABLS.

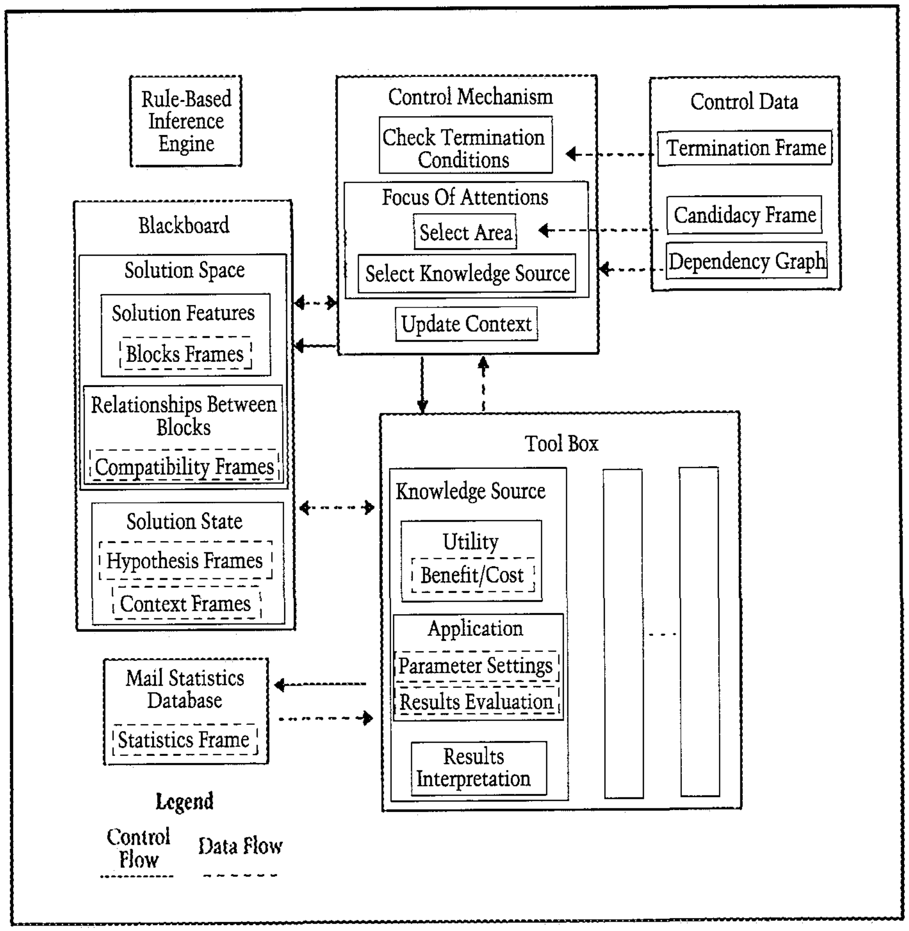

Figure 7. ABLS Organization as a Composition of Six Major Components

tain information about the interdependency between the KSs and criteria for accepting a block as the destination address or the destination address candidate. The control mechanism is responsible for checking the termination condition, selecting the KS, combining new evidence, and updating context. KS selection is based on the benefit-cost estimate of each KS. The control mechanism relies on the utility rule module of each KS and the dependency graph, which specifies the interdependency among KSs, to estimate the utility of each KS in the current context. Initially, the utility of each KS is zero. The control mechanism then sequentially invokes the utility rule modules of those KSs whose nodes are triggered in the dependency graph to estimate their utilities. The KS with the maximum utility is chosen as the KS to be applied next, and its utility is reset to zero before it is applied.

The blackboard contains the geometric attributes of blocks extracted from low-level image processing, the degree of support of labeling hypotheses, and the current context. The tool box contains a collection of KSs, many of them image processing related. Each KS contains rules for estimating the benefit and cost of using it, selecting parameters, evaluating results, and interpreting results.

Knowledge Representation

A hybrid of frame and rule-based knowledge representation is used to model knowledge for coordinating tools and computing degrees of support of labeling hypotheses. The following subsections describe how knowledge is represented in the control data, blackboard, and KSs.

Control Data. Knowledge used in the control data is modeled by the termination frame, the candidacy frame, and the dependency graph. The termination frame is used to represent criteria for accepting a block as the destination address. The criteria in the termination frame are usually strict in order to reduce the chance of mislabeling. When no candidate block meets these termination criteria, the system needs to apply additional tools to generate more evidence. The candidacy frame contains the minimum requirement for a block to remain as a candidate. Its purpose is to rule out highly unlikely candidates for the destination address.

The dependency graph (see figure 8) is a directed graph for specifying the order of applying KSs as well as for minimizing the reestimation of the utility of each KS. A node in a dependency graph is triggered if one of the arcs entering the node is activated. Each node in the dependency graph represents whether the utility of the associated KS should be reevaluated during the KS selection. The control mechanism does not invoke the utility rule module of a KS unless its associated node in the dependency graph is triggered. The selection of a KS causes the following changes to its associated node in the dependency graph: (1) all the arcs entering this node are deactivated, (2) this node is untriggered, and (3) the activation of outgoing arcs of this node is controlled by the rules in the results interpretation rule module of the selected KS.

Blackboard. Knowledge in the blackboard is stored in the block frame, the compatibility frame, the hypothesis frame, and the context frame.

The block frame is used to represent the results of applying KSs to an image. For each possible different feature that can be extracted by a KS from an image, a corresponding slot exists in the block frame to record this feature value. An example of a block frame used in ABLS is shown in the inset on the following page (an attribute with an unknown value is filled with a nil).

The compatibility frame models knowledge about the two-dimensional layouts of labels on an image. It stores information about the degree of compatibility of a spatial relationship between labels; for example, the degree of compatibility between the postage block and the destination address is high if the destination address is below and on the left-hand side of the postage block. The importance of spatial relationship knowledge is twofold: It allows checking for consistency in labeling blocks and predicting the existence of blocks.

The hypothesis frame is used to record the degree of support of labeling hypotheses. Because ABLS has five possible labels, each hypothesis frame has five labeling hypotheses. For each possible labeling hypothesis there is a slot in the hypothesis frame that represents the degree of support. The context frame is used to represent the current situation. It is composed of two parts: candidate blocks and performance parameters. The candidate blocks are those blocks that remain under consideration, given the evidence accumulated so far. The performance parameters represent an estimate of the difference between the current context and the goal, that is, the difference between the termination condition and the current situation.

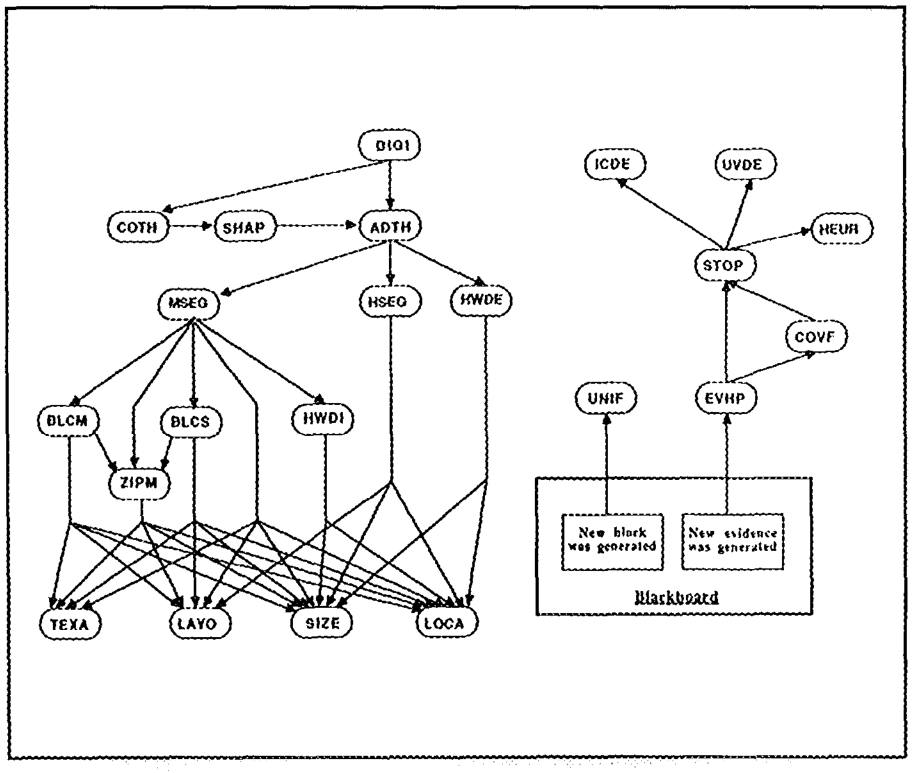

Figure 8 Dependency Graph for Specifying the Order of Applying Specialized Tools.
The label inside each node in the graph represents an abbreviation of a specific tool name The descriptions of tool name abbreviations appear in table 1 ADTH and COTH are triggered by DIGI UNIF is triggered by the blackboard when a new block is generated. EVHP is also triggered by the blackboard when a new block or new evidence is generated All the other nodes are triggered as described in the knowledge representation subsection

Tool Box. The tool box contains a collection of tools. Knowledge about the selection and utilization of each tool is divided into three rule modules in each KS. The rationale for this division is to group relevant rules into modules so that the rule-based inference engine can run efficiently by incrementally loading or removing rule modules during run time. Inside the KS, there are three rule modules: utility, application, and result interpretation.

The utility rule module contains knowledge about the intended purpose of a tool. It uses the information on the blackboard to estimate the expected gains and cost of using a tool in the current context. Besides estimating utility, the utility rule module also selects the area to which the tool later applies. The application rule module models the knowledge about the influence of parameter settings on the results. It contains rules to set the parameters and invoke the tool. The result interpretation rule module contains rules to evaluate and interpret results.

For most image-processing-related tools, a fixed set of parameters might not cover all the input cases, even though the parameters were fine tuned to cover most of the cases. In order to cover those residual cases which require a different set of parameters, the parameter setting and result evaluation are tightly coupled so that a tool can have the flexibility to try different parameter settings. At the

```
(!block
^id 4                   ;unique id for this block.
^minx 258               ;minx, miny, maxx, and maxy
^miny 109               ;define the
^maxx 362               ;rectangular enclosure of
^maxy 148               ;this block.
^area 1132              ;the # of black pixels in a block.
^skew 1.24              ;the orientation of a block.
^lines 4                ;the # of text lines in a block.
^comps 48               ;the # of components in a block.
^grid 5                 ;which grid this block lies on?
^alignment nil          ;are text lines left or right aligned?
^color white            ;the background color.
^texture nil            ;formed character or dot matrix printed?
^print-method machine   ;machine-generated or hand-generated?
^UV-hot nil             ;orange in ultra-violet image?
^shape nil              ;is this block rectangular? )
```

Tool	Descriptions
DIGI	produces digitized images (RGB and gray level) of the original picture.
ADTH	performs adaptive thresholding to convert a gray-level image into a binary image using local contrast.
COTH	provides color thresholding to extract regions of specified color in RGB images.
CCLA	labels connected components in a binary image; it is automatically invoked either after the use of the color thresholding tool or before the use of any seg mentation tools.
SHAP	measures how rectangular a region is.
MSEG	is the bottom-up segmenter that groups machine- generated characters into words, lines, and blocks.
HSEG	is the bottom-up segmenter that groups hand-generated thin strokes into lines and blocks.
HWDI	is the regularity analyzer that distinguishes machine- generated versus hand-generated address blocks.
HWDE	is the regularity analyzer that detects the likely regions of hand-generated text.
TEXA	is the texture discriminator that discriminates address block-type characters from nonaddress block-type characters.
ICDE	is the postal icon detector that detects rectangular postal icons.
UVDE	is the postage detector that detects postage location (stamp, meter mark) in ultraviolet images.
ZIPM	merges a small zip code block located in the lower right with the destination address candidate.
BLCM	merges machine-generated text blocks that are parallel and in close proximity.
BLCS	splits a too high or wide machine-generated text block into several smaller text blocks.
SIZE	uses block features, for example, aspect ratio, size, length, height, number of text lines, and number of components, to determine the likelihood of a block being the destination address, return address, or advertising text.
LAYO	examines the layout of text lines in a block.
LOCA	uses the location of a block to determine the likelihood of this block being the destination address, return address, or postage.
UNIF	unifies the block features between blocks generated by different KSs.
COVF	verifies the consistency of labeling hypotheses among neighboring blocks.
STOP	does evidence combination and decides whether to halt processing.
EVHP	pools the evidence generated by various KSs and then uses it to generate label ing hypotheses.
HEUR	uses heuristics to guess the destination address from a list of candidates.

Table 1. Functional Descriptions of Knowledge Sources.

start, a default parameter setting, which usually covers most of the input cases, is chosen. If the result evaluation shows that a different set of parameter settings is needed for the current input case, the result interpretation rule module notifies the control mechanism to reapply this tool. The new results obtained by reapplying a tool are interpreted by rules in the result interpretation rule module in order to generate new evidence. Each new piece of evidence is associated with a confidence value that represents the degree of support for a particular labeling hypothesis. Table 2 illustrates rule modules.

Control Strategy

The control strategy of ABLS is an integration of both bottom-up and top-down processing. Initially, one of the thresholding tools is chosen and applied to the entire mail piece image. The binary image is then segmented into blocks using the bottom-up segmentation tool. The physical attributes of a segmented block are then interpreted to generate evidence to either support or refute a block as the destination address. ABLS assumes there are only four possible global orientations for a mail piece with rectangular shape: correct global orientation or rotated by 90 180 , or 270 .

After the interpretation of segmented blocks, control is as follows: If only one segmented block satisfies the termination criteria, the destination address is considered found. If no candidate block satisfies the candidacy criteria, another thresholding tool is chosen and applied. Then, the bottom-up segmentation tool is again used to generate candidate blocks; otherwise, the tool manager selects and applies one of the specialized tools on the candidate blocks to generate more evidence to support or refute a candidate block as the destination address. Further details of the blackboard architecture, control strategy, and evidence combination method are described in Srihari and Wang (1986).

Performance

The complex mail images in Figure 2B and 2C illustrate the invocation of different tools within ABLS. Figure 2B is the photopic image of a colorful magazine cover. First, knowing that the mail piece is a flat, ABLS uses the color thresholding tool (COTH). This tool performs several operations in sequence: The image is thresholded in color space to obtain a binary image (see figure 9A), extracts connected white regions, computes the bounding rectangle of each white region, and outputs each region with reasonable size as a candidate. The shape analysis (SHAP) tool is then applied to check the rectangularity of each white region. Only one rectangular white region remains as a candidate (see figure 9B). The adaptive thresholding (ADTH) and machine-generated text segmentation (MSEG) tools are then applied to the rectangular white region; one destination address candidate and one zip code are segmented out (see figure 9C and 9D). Finally, the zip code merging (ZIPM) tool detects the separation of the destination address candidate and the zip code block and merges them into a complete destination address (see figure 9E). The located skewed destination address is then extracted and rotated to the standard orientation (see figure 9F).

Figure 2C is a photopic image of a flat mail piece. Again, the color thresholding (COTH) tool is applied first. In this case, no white regions of acceptable size are detected. This detection leads to the application of the adaptive thresholding (ADTH) tool, followed by the machine-generated text segmentation (MSEG) tool, to the entire image. As a result, several textual blocks are segmented out (see figure 10). The result interpretation rules of MSEG detect that (1) blocks 2,

Utility Module

Rule ID	IF	THEN
MSEGU1	1. The MSEG tool has not yet been used with the entire image.	1 Results in the MSEG tool having utility value 1.0 2. Selects the entire image as the area to be processed by the adaptive thresholding and MSEG tools.

Application Module

Rule ID	IF	THEN
MSEGA1	1. The selected area has never been segmented by the MSEG tool.	1. Uses the default parameters to segment the images.
MSEGA2	1. Too many small blocks were segmented out in the last segmentation.	1. Resegments the image Application with a larger maximum space between components threshold.
MSEGA3	1. Too many large blocks were segmented out in the last segmentation.	1. Resegments the image with a smaller maximum space between components threshold.

Results Interpretation Module

Rule ID	IF	THEN
MSEGR1	1. Given block A iscreated by the MSEG tool, block A's aspect ratio, length, and height aree within the acceptable range for the machine-generated destination address. 2. The number of components and text lines for block A are within the acceptable range for the machine-generated destination address.	1. Generates a piece of size evidence to confirm block A as the destination address to the degree 0.4, return address to the degree 0.3, and advertising text to the degree 0.2.
MESEGR2	1. No address block candidate is found. 2. A cluster of small blocks created by the MSEG tool are in close proximity. 3. The length and height of each small block in the cluster are below the acceptable range for the machine-generated destination address.	1. Merges these small blocks together. 2. Notifies the control mechanism to apply the handwriting segmentation tool around the newly merged block.
MSEGR3	1. No address block candidate is found. 2. Over two-thirds of the blocks created by the MSEG tool have their length and height below the acceptable range for the machine-generated address.	1. Notifies the control mechanism to resegment the the image with a larger maximum space between components threshold.
MSEGR4	1. No address block candidate is found. 2. Over two-thirds of the blocks created by the MSEG tool have their length and height above the acceptable range for the machine-generated address.	1. Notifies the control mechanism to resegment the image with a smaller maximum space between components threshold.

Table 2. Rule Modules and Rules of the Machine-Generated Text Segmenter (MSEG).

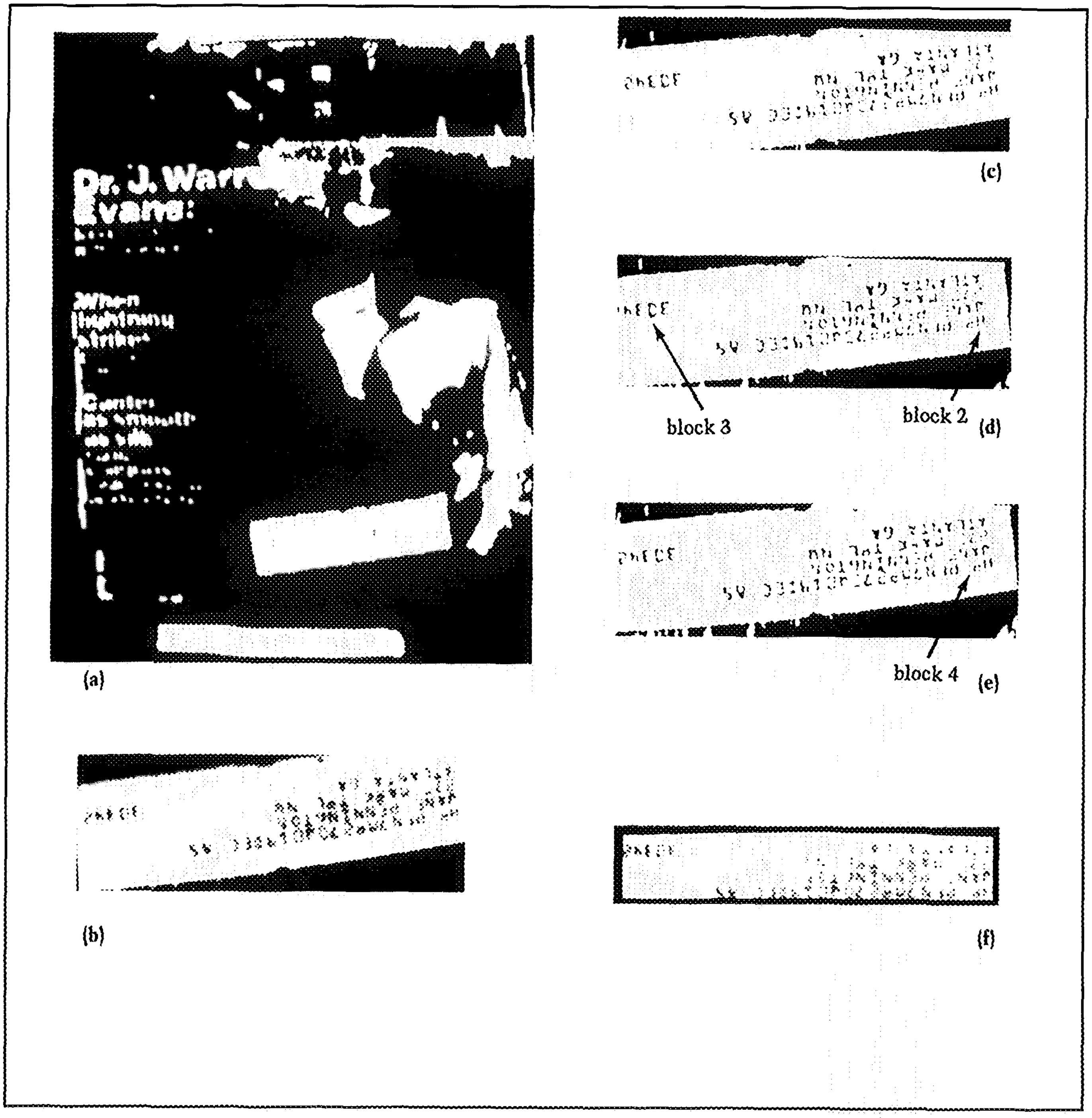

Figure 9. (A) Color thresholding results of Figure 2B; (B) The extracted white region (block 1); (C) The gray-level thresholding results of the white region; (D) The segmentation results of the white region; (E) The results of merging the destination address candidate (block 2) and the zip code candidate (block 3); (F) The extracted destination address in unskewed orientation.

9, and 10 are candidates for a machine-generated address and (2) the size of block 7 is too large to be a machine-generated destination address but is adequate to be a hand-generated destination address. MSEG then notifies the control mechanism to use the hand-writing and machine printing discrimination tool (HWDI) to examine the contents of block 7. The HWDI tool concludes that block 7 is hand-generated text. Thus, block 7 receives an additional piece of confirming evidence as the destination address because a hand-generated text block is more likely to be the destination address than a machine-generated text block. Because block 7 has not yet collected enough evidence to be declared the destination address, the location (LOCA) tool is called on to examine the location of each text block. Finally, block 7 is declared the destination address because its size, content, and location make it more likely to be the destination address than any other segmented text block.

Success Rate

In order to test the performance of ABLS, experiments were conducted with 174 mail piece images. Success was defined as the actual address block being chosen as the top-rated output. The success rates for each of the mail categories were as follows: letters 86 percent, flats 74 percent, IPPs 73 percent, and manual letters 84 percent. The high success rate for letter mail was the result of its structure. For the other categories, the major cause of failure was the difficulty in segmenting a hand-generated address. The region-based segmentation tool is effective on machine-generated addresses but less effective on hand-generated addresses. The major cause of error in the case of flats and IPPs was the presence of many text blocks with an appearance similar to the destination address. The color thresholding tool that extracts the white address label is effective in such cases, but it only works when there is a clear contrast between the label and the background. Another way to locate a randomly placed destination address is to utilize the icon detection tool (ICDE) or ultraviolet detection tool (UVDE) to determine the postage location (stamp, meter mark, permit mark, postal icon, and so on) and then use the spatial relationship between the postage and the destination address to locate the actual destination address.

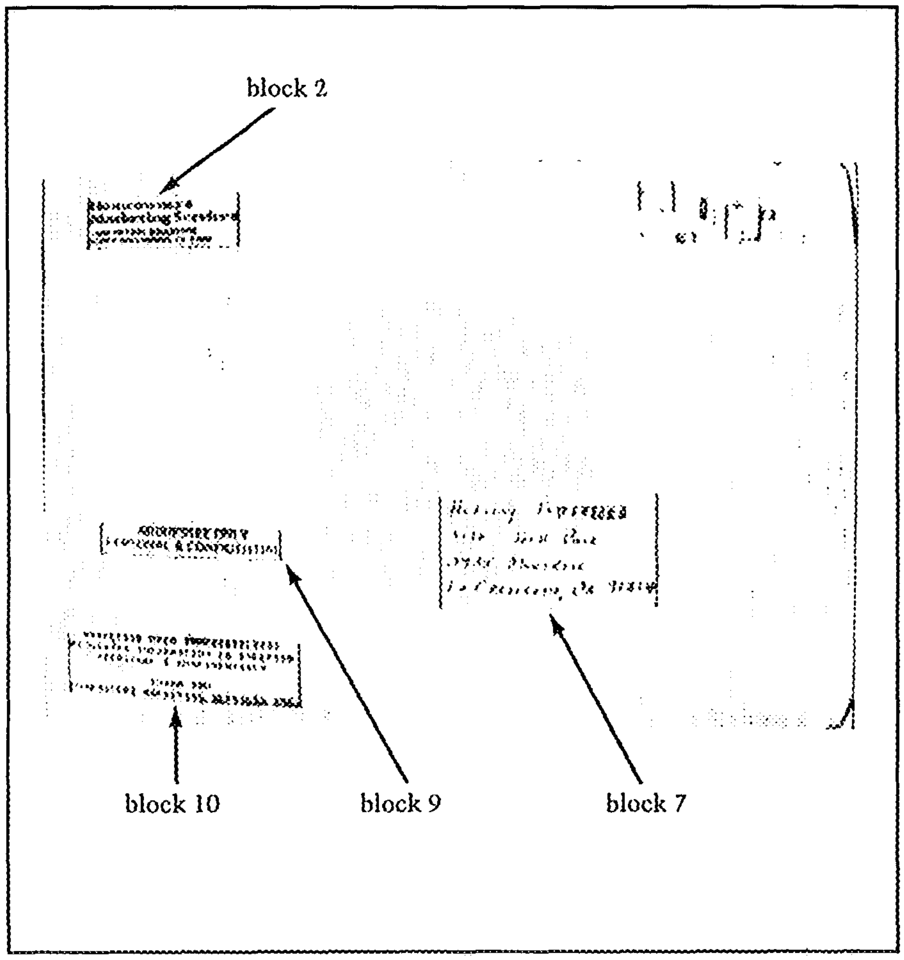

Figure 10 The Segmentation Results on the Thresholded Image of Figure 2C.

Speed. In the current implementation of ABLS under UNIX, the top-level control structure is written in Franz Lisp, which invokes the specialized tools written in C. ABLS currently runs on single-CPU systems (Sperry 7000, Digital VAX-785, and SUN-3 workstations).The average processing time is about 7 to 8 minutes for an IPP or manual letter, and 20 minutes for a flat mail piece. With the current system reaching an acceptable location rate of 79 percent, and some of the specialized tools requiring as much as 20 minutes of run time, we are now focusing on methods to reduce the processing time per mail piece by several orders of magnitude. To achieve this goal, we are exploring code optimization, the use of special processors on single-CPU systems, for example, floating-point processors such as an FPA or 68881 chip, as well as the use of distributed-processing systems. In particular, we are considering an implementation on a parallel machine where several processors are connected according to the topology of a hypercube.

Summary

The variability of the address block on mail pieces requires a systematic analysis of mail piece characteristics in order to determine those factors which are important for an automated address block location system. We described several specialized tools and their performance, in isolation, on a database of test images. An architecture for invoking the tools in order to locate the destination address block in a vast variety of mail piece images was presented. The framework is flexible enough to incorporate as many tools as possible into the system if experimental results can establish the usefulness of these tools. The performance rate of the entire system on a database, including many difficult cases, is about 79 percent. ABLS is a system under development. Future work includes the incorporation of more tools into the system, the continuing refinement of existing tools, and efficient implementation using parallel architectures.

Acknowledgments

The authors would like to thank Dr Rick Glickman, Mr Gary Herring, and Mr. Herb Schiller of the Office of Advanced Technology of the U S. Postal Service for the support of this work as well as for many suggestions and encouragement This work was supported under USPS Task Order 104230-85- M3349

References

Erman, L D ; Hayes-Roth, F ; Lesser, V R.; and Reddy, D R 1980 The Hearsay-II Speech-Understanding System: Integra- Knowledge to Resolve Uncertainty *ACM Computing Surveys* 12(2): 213-253.

GTRI 1985-1986 Automated Processing of Irregular Parcel Post: IPP, Letter, and Flat Statistical Database, Technical Report, A-3734, Electronics and Computer Systems Lab, Georgia Institute of Technology

Nii, H. P 1986. Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures, Part One *AI Magazine* 7(2): 38-53

Palumbo, P, and Srihari, S. N 1986 Text Parsing Using Spatial Information for Recognizing Addresses in Mail Pieces In Proceedings of the Eighth International Conference on Pattern Recognition, 1068-1070 Los Angeles, Calif.: IEEE Computer Society press.

Palumbo, P.; Swaminathan, P.; and Srihari, S N. 1986. Document Image Binarization: Evaluation of Algorithms In Proceedings of the SPIE Symposium on Applications of Digital Image Processing IX, 278-285. Bellingham, Wash : The Society of Photo-Optical Instrumentation Engineers

Srihari, S. N ; Hull, J J.; Palumbo, P. W ; and Wang, C H. 1986 Address Block Location: Evaluation of Image and Statistical Database, Technical Report, 86-09, Dept of Computer Science, State Univ of New York at Buffalo

Srihari, S N ; Hull, J J.; Palumbo, P W.; Niyogi, D.; and Wang, C H. 1985. Address Recognition Techniques in Mail Sorting: Research Directions, Technical Report, 85-09, Dept of Computer Science, State Univ of New York at Buffalo

Srihari, S N., and Wang, C H 1986 A Blackboard Architecture for Object Recognition: Locating Address Blocks on Mail Pieces. In Proceedings of the SPIE Symposium on Automated Inspection and Measurement, 224-232. Bellingham, Wash : The Society of Photo-Optical Instrumentation Engineers.

Wang, C. H., and Srihari, S N 1986 Object Recognition in Structured and Random Environment: Locating Address Blocks on Mail Pieces In Proceedings of the Fifth National Conference on Artificial Intelligence, 1133-1137. Los Altos, Calif : Morgan Kaufmann.

New Dates for St. Paul:

AAAI -88
Conference Dates
Sunday- Friday
21-26 August

Tutorials
Sunday-Monday
21-22 August

Technical Sessions
Tuesday-Friday
23-26 August

Exhibits
Tuesday-Thursday
23-25 August

Halbrecht Associates, Inc.
Executive Search Consultants

SERVING THE ARTIFICIAL INTELLIGENCE COMMUNITY

Halbrecht Associates, founded in 1957, was the first search firm in the United States to specialize in Information Systems Management, Operations Research, Management Sciences, Telecommunications, Computer Sciences and Decision Support Systems. We are now also in the forefront of the Artificial Intelligence field. Because we have been so extensively involved in this emerging technology, our portfolio of people includes extensive numbers of AI professionals at all levels, from the United States and abroad.

Our client list includes companies coast to coast, large and small, including start-ups with equity opportunities. Accordingly, depending on your needs, we are in a unique position to assist you rapidly with any of your requirements.

Our current assignments range downward from several Chief Scientists, or those who can provide the technical leadership to head up major AI programs, to Senior LISP Programmer. If you would like to be considered to be part of our AI data base, please send us your credentials. This will enable us to give you first notice of appropriate opportunities. All inquiries are in strictest confidence.

Contact: Daryl Furno, Senior Associate

1200 Summer Street • Stamford, CT 06905 • 203-327-5630

For free information, circle no. 193