

A Framework for Object Recognition in a Visually Complex Environment and its Application to Locating Address Blocks on Mail Pieces*

CHING-HUEI WANG and SARGUR N. SRIHARI

Department of Computer Science, State University of New York at Buffalo, Buffalo, NY 14260

Abstract

A computational framework for recognizing an object of interest in a complex visual environment is described. Arising from the problem of finding the destination address block on a mail piece, a general framework for coordinating a collection of specialized image-analysis tools is described. The resulting system is capable of dealing with a wide range of environments—from those having a high degree of global spatial structure (e.g., letter mail envelopes that conform to specifications) to those with no structure (e.g., magazines with randomly pasted address labels). The problem-solving architecture accounts for uncertainty in the imaging environment by using the blackboard model. This paper discusses systematic derivation of a set of object recognition heuristics (knowledge base), specialized image analysis tools for extracting those features that are called for by the heuristics, and a control structure for integrating evidence and managing tools. Experimental results with a database of difficult cases demonstrating the promise of the methodology are presented.

1 Introduction

We are concerned here with the problem of visually finding an object of interest in a complex environment. It is assumed that we have an image of the environment that contains the object. The object environment is assumed to be one over which not much control can be exercised and therefore the image may be either structured or cluttered.

Our objective is to develop a computational solution to handle two sources of problem uncertainty—*environmental* and *imaging*. Environmental ambiguity is due to wide variations in the physical environment being imaged. This environment includes:

1. *a variable number of objects*;
2. *variable object attributes*—for each object, attributes such as location, orientation, width, height, image attributes (e.g., contrast between object and background), etc., have a high variance and there may be many objects with the same attributes; and
3. *both structure and randomness*—in some cases, spatial relationships hold between the object

of interest and other objects in the environment; in other cases the object may occur in a random spatial position and orientation.

Ambiguity from imaging arises from electro-optic sources such as uneven illumination, coarse image digitization, and early image processing operations used to extract objects from the image (e.g., thresholding and segmentation). Initial segmentation may miss important regions of the object of interest. The computational solution should provide for merging missing regions using knowledge, for repairing segmentation results, or for rethresholding/resegmenting a portion of an image by using different parameters if the initial segmentation is found to over- or under-segment the object.

When the distinctive features of the object of interest are clearly specified, the *pattern recognition* approach, i.e., segmentation-feature extraction-classification, can be appropriate. In this ap-

*This work was supported by the Office of Advanced Technology of the United States Postal Service Under Task Order 104230-85-M3349.

proach, objects are extracted first, relevant features of *every* such object are computed, and finally each object is classified based on the features. There are several disadvantages with this approach in practice. First, it is basically a bottom-up process that provides no feedback for using the results of higher-level processes at lower levels. Second, it is nonopportunistic, it may not be necessary to extract all objects and classify each of them, i.e., in every case.

When the image is known to exhibit well-defined structure, the model-based approach—one where model knowledge is used to reason about identities of regions [3,4]—can be useful. Model knowledge typically includes *attributes of different objects*—e.g., size, length, height, contrast, location, texture, intensity, etc.—and *spatial structure*—i.e., spatial relationships between objects. Examples of specific systems using the model-based approach are ACRONYM [4], which is domain independent, SPAM [17], which interprets aerial images of airports, SIGMA [16], which interprets aerial images of suburban areas, and VISIONS [11], which interprets outdoor scenes. The effectiveness of model-based reasoning depends on the completeness and certainty of model knowledge. For an environment with a different structure, a different model has to be built and stored. The model-based approach is appropriate when the image does not vary too much from the model. The problem at hand is to account for randomness that renders model-based spatial reasoning ineffective, while not ignoring the spatial relationships that frequently hold.

Recognizing a specific object is in one sense a specific case of the problem of interpreting a complex scene. Object recognition in such problems has led researchers to propose the use of multiple sources of image data and knowledge [5,11,21,24, 29]. Ambiguity arising from lack of perfect information from a given source is reduced by having partially redundant information from other sources.

ADDRESS BLOCK LOCATION PROBLEM

Our computational framework for object recognition is a domain-independent generalization of a solution for locating address blocks on images of

mail pieces. The object of interest in this case is the destination address block and the environment is the mail piece face that contains it. Several mail pieces with different levels of problem complexity in locating the destination address block are shown in figure 1. Assuming correct global orientation, the destination address is usually below and on the left-hand side of postage or meter mark, and the return address is above and on the left-hand side of a destination address, etc., (see figure 1(a)). In some cases, the object can occur randomly anywhere on the background (see figure 1(b)). The number of objects on a mail piece face is variable.

In previous papers [25,28], we introduced the address block location problem, a general description of our computational framework, and the image analysis tools invoked by the framework. This paper describes a generalization of the architecture and performance of a refinement of the system, with an image database consisting of many difficult cases. Section 2 describes a blackboard framework for object recognition. Section 3 describes the Address Block Location System (ABLS) based on this framework. Section 4 describes the implementation and section 5 is a description of experimental results and analysis.

2 A Framework for Object Recognition

The key to handling both structure and randomness in the input is not to put too much emphasis on a single source of knowledge while acquiring evidence from diverse sources of knowledge. It is important not to depend too heavily on location and spatial relationships until there is a certain degree of confidence that the input conforms to a well-defined structure. Also, it is not necessary to interpret every object in the input. When it happens that certain unique features of the object are detected in a particular candidate, the system should halt processing regardless of whether every element in the input image has been recognized.

We approach the problem as one that needs (1) a set of good features (for the object of interest and other objects that help locate it); (2) a collection of specialized tools to extract objects and compute

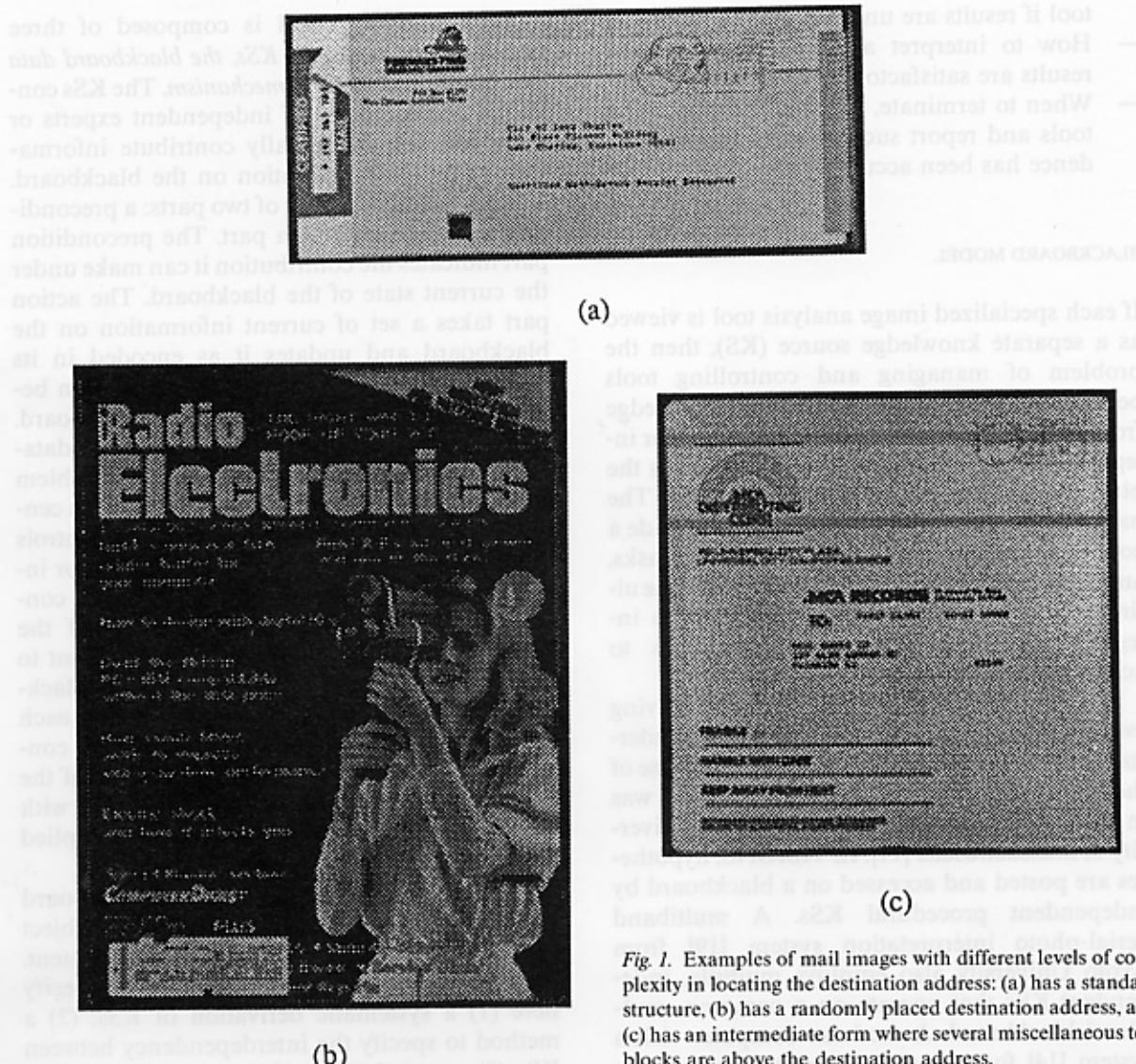


Fig. 1. Examples of mail images with different levels of complexity in locating the destination address: (a) has a standard structure, (b) has a randomly placed destination address, and (c) has an intermediate form where several miscellaneous text blocks are above the destination address.

those features, interpret extracted features using domain knowledge, and generate evidence; and (3) a control structure to integrate evidence and manage the tools so that effort expended in gathering evidence is in accordance with input complexity.

When a large number of specialized image analysis tools are used, it is necessary to judiciously plan their usage. In order to arrive at a plan, it is necessary to know the following:

- Where to use a tool, i.e., the applicable area of image.
- When to use a tool, i.e., the appropriate time to use it.
- Why use a tool, i.e., given several tools for a task, which is the best one under a given circumstance.
- How to use a tool, i.e., what parameters must be set before invocation.
- How to change parameters and reapply the

tool if results are unsatisfactory.

- How to interpret as new evidence when results are satisfactory.
- When to terminate, i.e., when to stop using tools and report success when enough evidence has been accumulated.

BLACKBOARD MODEL

If each specialized image analysis tool is viewed as a separate knowledge source (KS), then the problem of managing and controlling tools becomes a problem of integrating knowledge from various sources. An effective method for integrating knowledge from various sources is the blackboard model of problem solving [7,20]. The basic idea of the blackboard model is to divide a complex problem into loosely coupled subtasks, and then each subtask is attacked by a KS. The ultimate goal of the blackboard model is to integrate knowledge from various sources to achieve a common goal.

The blackboard model of problem solving evolved out of the needs of the speech-understanding task in HEARSAY-II [7]. An early use of the blackboard model in the vision domain was in the VISIONS systems developed at the University of Massachusetts [11]. In VISIONS, hypotheses are posted and accessed on a blackboard by independent procedural KSs. A multiband aerial-photo interpretation system [19] from Kyoto University also employs multiple, independent KSs that operate on a common, multilevel blackboard. A modular computer vision system [14] for picture segmentation and interpretation developed at McGill University also utilizes a collection of analysis processors, each of which is specialized for a particular task. A query-oriented vision system [1] developed at the University of Rochester also incorporates a collection of mapping procedures whose characteristics are similar to the independent KSs in the blackboard model. Each mapping procedure is specialized to one particular object, having a precondition, a postcondition, a priori reliability, and an expected cost. An executive procedure is used to examine these characteristics to select the mapping procedure that best fulfills its needs.

A blackboard model is composed of three major components: *the KSs*, *the blackboard data structure*, and *the control mechanism*. The KSs consist of a community of independent experts or tools which incrementally contribute information to the overall solution on the blackboard. The KS usually consists of two parts: a precondition part, and an action part. The precondition part indicates the contribution it can make under the current state of the blackboard. The action part takes a set of current information on the blackboard and updates it as encoded in its specialized knowledge. The communication between KSs is exclusively through the blackboard. The blackboard data structure is a global database to represent the current state of problem solving, and also serves as a communication center between KSs. The control mechanism controls the invocation of KSs, and is responsible for integrating on the blackboard information contributed by various KSs, to determine if the current state of problem solving is sufficient to constitute a solution. The solution on the blackboard is built one step at a time. During each cycle of operation, each KS indicates the contribution it can make to the current state of the blackboard. The control then selects the KS with the maximum utility as the KS to be applied next.

We will specialize the general blackboard model into a blackboard *framework* for object recognition in a visually complex environment. As components of this framework, we specify here (1) a systematic derivation of KSs; (2) a method to specify the interdependency between KSs; (3) a control mechanism to deal with varying degrees of complexity in the input; (4) a formalism for estimating the utility of a KS; (5) the organization of KSs which allows the image-analysis-related KSs to be reapplied if the initial results are inadequate; and (6) a formalism to combine confidence values associated with the evidence.

2.1 Systematic Derivation of Knowledge Sources

The first step in developing KSs is to define the scope of domain knowledge. The scope of do-

main knowledge and the KSs needed to acquire evidence varies over different application domains. Domain knowledge generally includes not only the intrinsic features of objects of interest, but also the intrinsic features of nontarget objects and the spatial relationships between objects. The image data may include gray-level (photopic) images, RGB colored images, infrared, color under ultraviolet illumination, or other appropriate images. The scope of domain knowledge can be established by collecting application domain statistics. These statistics form a statistical database from which a KS can be developed according to the following steps:

1. The statistical database is examined to identify an objective of a KS. The context in which the KS is most useful is also determined.
2. An algorithm for the KS is developed. Its implementation is such that its invocation and completion are appropriate for the overall control structure.
3. The KS is experimentally run under various conditions. Estimated cost and parameter settings under various conditions are compiled into the knowledge rules.
4. The KS performance is evaluated under various conditions. Combining this with the performance predicted by the statistical database will yield a utility measure for the KS.

2.2 Interdependency between Knowledge Sources

In the ideal blackboard model, KSs are supposed to be totally independent of each other, however, in practice, some KSs are interdependent and cannot be randomly invoked. In order to specify this kind of interdependency among KSs and to avoid reestimating the utility of each KS after each KS application, a formalism is necessary to specify KS interdependency.

A *dependency graph* is a directed graph that specifies the order of applying KSs. Each KS is associated with a node to represent whether the utility of a KS (see section 2.4) should be evaluated during the KS selection. The utility value of a KS is evaluated if its associated node in

the dependency graph is triggered.¹ The relationships between nodes and arcs are defined as follows: If node M depends on node N , an arc A is connected from node N to M . In this case, arc A is an outgoing arc to N and is an incoming arc to M . A node can have multiple incoming arcs and outgoing arcs. A node in the dependency graph is triggered if one of the arcs entering the node is activated.

The control mechanism will not evaluate the utility value of a KS unless its associated node in the dependency graph is triggered. The selection of a KS will cause the following changes in its associated node: (1) all the arcs entering this node are deactivated; (2) this node is untriggered; (3) the activation of outgoing arcs from this node is meaningful results are produced during the application of the selected KS.

2.3 Dealing with Varying Degrees of Input Complexity

An input image might have the object of interest located in either a standard or a random position and orientation. The image may contain non-target objects with characteristics similar to the object of interest. The control mechanism has to detect the complexity of the input, and adjust overall system behavior so that the effort expended in gathering evidence is in accordance with the complexity. Image complexity can be determined by counting the number of candidates for the object of interest, and checking the differences between them.

The system can utilize one of several tools at any different time while processing an image. When several tools are used, it is necessary to quantify the strength of evidence gathered from the application of each tool and to combine the evidence generated by different tools.

¹The term *utility* in this work is a measure to express the relative usefulness of KSs in the blackboard framework. It is not related to the traditional utility theory in probability and statistics.

CONFIDENCE VALUES

A general rule has the form

IF condition THEN action

where *condition* represents the testing portion of the rule, which must match with the current state of the blackboard; the *action* part of the rule is executed if the conditional part is true. An object is hypothesized to have a particular label if there exists evidence that shows some of the object's features satisfy the conditional part of a rule, and the action part of that rule confirms that this object has the particular label.

Each new piece of evidence generated by the application of a specialized tool is associated with a *confidence value* to represent the degree to which it supports or refutes a particular labeling hypothesis. An example of including a confidence value is

IF aspect ratio(b) = x and size(b) = y

THEN b is the object with confidence value z where b is a particular object being tested, x and y are rule parameters, and z is the probability associated with the confidence of the rule results. The confidence value z represents the a posteriori probability $P(l'|e)$, where e is the evidence (i.e., the condition or "IF" part of a rule), and l' is the hypothesis associated with the object, i.e., it represents the probability of assigning label l' to object b given evidence e . In the above example, evidence e is the aspect ratio and size of the test object, while label l' corresponds to the object of interest.

There are several ways to estimate the a posteriori probability $P(l'|e)$. The first is to use Bayes' theorem, i.e.,

$$P(l'|e) = \frac{p(e|l')P(l')}{\sum_j P(l^j)p(e|l^j)}$$

to calculate $P(l'|e)$ from the class-conditional probability density functions $p(e|l^j)$ and a priori probabilities $P(l^j)$. The second is to use experimental results for estimation. The third method is subjective estimation, which is an educated guess without empirical support. Bayes' theorem and

subjective estimation represent two extremes in estimating a posteriori probability $P(l'|e)$. The former is statistically sound, and the latter relies on a guess.

TERMINATION CRITERIA

The control mechanism needs criteria to determine whether to halt processing or to invoke KSSs for gathering additional evidence. There are two sets of termination criteria as follows:

Let B be the set of candidate objects and L be the set of possible labels. Let $S(b,l)$ be the confidence value of assigning object b the label l . Let l^b be the label with the highest confidence value for block b , i.e., $\max \{S(b,l)|l \in L\} = S(b,l^b) = S^b$. Let b_o be the object with the highest confidence value for the object-of-interest label o , i.e., $\max \{S(b,o)|b \in B\} = S(b_o,o) = S_o$.

The first set of criteria is related to the confidence value of different labels for the same candidate object. The conditions for candidate object b to pass the first set of criteria are:

1. $S(b,o) = S^b > T_1$, and
2. $S^b - \max \{S(b,l)|l \in L - l^b\} > T_2$

where T_1 and T_2 are predefined thresholds.

The second set of criteria is related to the confidence of each object-of-interest label among all objects. The conditions for candidate object b to pass the second set of criteria are:

1. $S(b,o) = S_o > T_1$, and
2. $S_o - \max \{S(b,o)|b \in B - b_o\} > T_3$

where T_1 and T_3 are predefined thresholds.

The objective of the termination criteria is consistent with the spirit of expending just enough effort according to the complexity of the input. For input with less complexity, the termination criteria will be satisfied after just applying a few necessary KSSs. For input with higher complexity, the termination criteria will not be easily satisfied unless a large amount of evidence is gathered after applying many KSSs.

CONTROL STRATEGY

The control strategy is an integration of both bottom-up and top-down processing. Bottom-up processing segments the image into objects. After the segmented objects are interpreted, the control strategy can be summarized as follows:

- If only one segmented object satisfies the termination criteria, the object is considered found.
- If no candidate object satisfies the candidacy criteria, another bottom-up segmentation tool, which has not yet been used, is applied to generate more candidates.
- Otherwise, the control mechanism will select and apply one of the unused KSs on those candidates. This will hopefully generate more pieces of evidence to support a candidate as being the object of interest.

GLOBAL ORIENTATION

Global orientation is the overall orientation of the image. It is important to know the correct global orientation prior to the interpretation of any segmented object, since the global orientation can affect use of spatial relationships. The location of certain uniquely identifiable objects may be able to help determine the correct global orientation. If correct orientation cannot be determined prior to the interpretation of segmented objects, and assuming a rectangular frame for the environment, we will interpret each segmented object in all four different global orientations. The correct global orientation is then assumed to be the orientation in which a segmented object obtains the maximum confidence value to be the object of interest.

2.4 Estimating Knowledge Source Utility

The precondition part of a KS contains either procedures or rules to estimate the utility of invoking a KS under the current state of the blackboard. The utility estimation of each KS is very important since it can affect the sequence of

invoking KSs, which in turn affects overall efficiency.

The strategy of KS selection in the blackboard framework is determined by the dependency graph and the utility value of each KS. The dependency graph specifies the interdependency between KSs and dictates which KSs are eligible for utility value estimation. After the utility value estimation, the KS with the maximum utility value is chosen as the KS to be used next. The utility value estimation of a KS is computed from the following five measures:

1. Efficiency of a KS, denoted by F , which is the ratio of the number of times the KS generates evidence to support the object of interest to the number of times the KS is used—if a KS to extract objects of interest generates k candidates in n invocations, then $F = k/n$.
2. Effectiveness of a KS, denoted by E , which is the ratio of the number of pieces of evidence that support the real object of interest to the number of pieces of evidence generated by this KS—if a KS to extract objects generates k candidates of which m are actually the objects of interest, then $E = m/k$.
3. Average processing time of a KS, denoted by T .
4. Proportion of the population with which a KS is designed to deal, denoted by P —if a KS is designed to extract white regions from the image and 60% of the images have the object of interest in white regions, then the proportion of the population for this KS is 0.6.
5. Special situation adjustment—those occasions that require the immediate attention of the system to invoke a particular KS. For example, if new pieces of evidence are generated after the application of a KS, the KS responsible for combining evidence has to be invoked immediately, this special adjustment is denoted by S .

The utility value U of a KS is computed as follows:

$$U = \frac{(F + WE)P}{\max(\log T, 1)} + S$$

where W is a weighting factor to adjust the relative

importance of KS effectiveness and KS efficiency. The utility-value formula is derived from the need to measure the relative usefulness of KSs from experimental results. The core elements of this formula are F and E ; their relative importance is expressed by W . The multiplication factor drastically decreases the utility value of a KS that is not applicable in most cases in the application domain. The result is divided by the logarithm of the average processing time so as to discount the utility value of a KS that is computationally expensive. The effect of the logarithm is to scale down absolute processing time. The smaller the base of the logarithm, the more important is processing time.

2.5 Organization of a Knowledge Source

In the blackboard model, knowledge about the selection and utilization of each KS is stored inside each KS. Two kinds of knowledge are needed in the selection and utilization of KSs. The first is knowledge about the nature of each KS, such as its intended purpose and the influence of its parameters on the results. The second is knowledge about the evaluation and the interpretation of results.

Each KS in the blackboard framework is organized to consist of three rule modules: *utility*, *application*, and *results interpretation*. The rationale for this division is to group relevant rules into modules so that the rule-based inference engine can run more efficiently by incrementally loading or removing rule modules during run time. The utility rule module contains knowledge about the intended purpose of a KS. The application rule module models knowledge about the influence of parameters settings on the results. The results interpretation rule module contains knowledge about how to evaluate and interpret the new results obtained from the KS application.

The *utility* rule module uses the information on the blackboard to estimate the expected gains and cost of using this KS in the current context. Besides estimating utility, the utility rule module also selects the area to which the KS will apply later. The *application* rule module models the knowledge about the influence of parameter set-

tings on the results. It contains rules to set the parameters, invoke the KS, and evaluate the results. For most image-analysis-related KSs, a fixed set of parameters may not cover all input cases even though the parameters were fine tuned to cover most of the cases. In order to cover those residual cases that require a different set of parameters, the parameter setting and result evaluation are tightly coupled in the application rule module so that a KS can have the flexibility to try different parameter settings. In the beginning, a default parameter setting, which usually will cover most of the input cases, is chosen. If the results evaluation shows that a different set of parameter settings is needed for the current input case, the application rule module will choose another set of parameters according to the result evaluation and then reapply the KS. For example, incorrect segmentation results can be caused by either source of uncertainty. Initial segmentation may miss some important regions of the object of interest. Therefore, provisions should be made to rethreshold or resegment a portion of an image by using a different set of parameters if the initial segmentation is found to oversegment (i.e., to find regions larger than desired) or to mis-segment (i.e., to partially include the desired object).

The new evidence obtained is interpreted by rules in the *results interpretation* rule module to generate new evidence. Each new piece of evidence generated by this rule module is associated with a confidence value to represent the degree of support for a particular labeling hypothesis.

2.6 A Formalism to Combine Evidence

The problem-solving approach of this framework is to partition the problem of object recognition into several small independent problems. Each small problem is then attacked by a specialized KS.

When there are many KSs generating evidence, there comes the problem of combining various pieces of evidence to correctly locate the object of interest. A correct formalism for combining evidence can help the system in the following ways:

1. Correctly locate the object of interest.

2. Make it easier to fine tune the confidence value associated with evidence.
3. Make the system run more efficiently by invoking a minimum number of KSSs.

Most existing application systems use ad-hoc procedures to combine confidence values associated with evidence generated by KSSs. The scheme to combine confidence values of evidence in the blackboard framework is based on Dempster-Shafer's rules of combination [2,23,10]. The choice of this evidence-combination scheme was based on the following:

- It has a sound mathematical foundation.
- It is commutative, i.e., it is independent of the order in which evidence is gathered.
- It has the ability to model the narrowing of the hypothesis set with the accumulation of evidence, a process that characterizes the problem-solving architecture of our blackboard framework for object recognition.

In Dempster-Shafer theory, all possible labels for objects form a set called the *frame of discernment*, Θ . The labels in Θ are mutually exclusive and exhaustive. All possible subsets of Θ , i.e., the $2^{|\Theta|}$ elements of the power set, are considered in evidence combination. The impact of each distinct piece of evidence on subsets of Θ is represented by a *basic probability assignment* m , which must satisfy the following three conditions:

1. $0 \leq m(A) \leq 1$ for all $A \subseteq \Theta$
2. $m(\emptyset) = 0$
3. $\sum_{A \subseteq \Theta} m(A) = 1$.

A belief function, Bel , corresponding to a specific basic probability assignment m , represents the total amount of belief committed to a subset of Θ and is defined as

$$\text{Bel}(A) = \sum_{B \subseteq A} m(B)$$

Given two belief functions, Dempster's rule of combination can compute a new belief function based on the pooled evidence. Let $m_1(A)$, $m_2(B)$, and $m(C)$ be the basic probability assignments for Bel_1 , Bel_2 , and Bel , respectively. Dempster's rule computes $m(C)$ as the *orthogonal sum* of $m_1(A)$ and $m_2(B)$:

$$m(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ \frac{\sum_{A \cap B = C} m_1(A)m_2(B)}{1 - \sum_{A \cap B = \emptyset} m_1(A)m_2(B)} & \text{if } C \neq \emptyset \end{cases}$$

To illustrate the rule, consider the frame of discernment $\Theta = \{D, R, P\}$, which consists of a set of labels. Let m_1 and m_2 be basic probability assignments corresponding to two pieces of evidence (or features) defined as follows:

$$\begin{aligned} m_1: \quad \emptyset &= 0.0, \{D\} = 0.2, \{R\} = 0.3, \{P\} = 0.0, \\ &\{D, R\} = 0.0, \{D, P\} = 0.0, \{R, P\} = 0.0, \\ &\Theta = 0.5 \\ m_2: \quad \emptyset &= 0.0, \{D\} = 0.3, \{R\} = 0.0, \{P\} = 0.0, \\ &\{D, R\} = 0.4, \{D, P\} = 0.0, \{R, P\} = 0.0, \\ &\Theta = 0.3 \end{aligned}$$

Therefore, $\text{Bel}_1\{D, R\} = 0.2 + 0.3 + 0.0 = 0.5$ and $\text{Bel}_2\{D, R\} = 0.3 + 0.0 + 0.4 = 0.7$. The basic probability assignment of the orthogonal sum $m(C) = m_1(A) \text{O} m_2(B)$ is defined by the matrix of table 1. From this table, we have the basic probability assignments for $m(C)$ as: $\emptyset = 0.09$,

$$\begin{aligned} \{D\} &= \frac{0.06 + 0.08 + 0.06 + 0.15}{1 - 0.09} = 0.38 \\ \{P\} &= 0.0 \\ \{R\} &= \frac{0.09 + 0.12}{1 - 0.09} = 0.23 \\ \{D, R\} &= \frac{0.20}{1 - 0.09} = 0.22 \\ \{D, P\} &= 0.0 \\ \{R, P\} &= 0.0 \\ \{R, P\} &= 0.0 \\ \Theta &= \frac{0.15}{1 - 0.09} = 0.17 \end{aligned}$$

Therefore $\text{Bel}\{D, R\} = 0.38 + 0.23 + 0.22 = 0.83$.

Table 1. Example of Dempster's rule of combination.

m_1	\emptyset	$ D $	$ R $	$ P $	$ D,R $	$ D,P $	$ R,P $	Θ
m_2	0.0	0.2	0.3	0.0	0.0	0.0	0.0	0.5
\emptyset	0.0	0.2	0.3	0.0	0.0	0.0	0.0	0.5
$ D $		$ D $	\emptyset					
0.3		0.06	0.09					
$ R $								
0.0								
$ P $								
0.0								
$ D,R $		$ D $	$ R $					
0.4		0.08	0.12					
$ D,P $								
0.0								
$ R,P $								
0.0								
Θ		$ D $	$ R $					
0.3		0.06	0.09					

3 Address Block Location System (ABLS)

3.1 Objectives and Assumptions

In the address block location problem the input is an image of the face of an arbitrary mail piece. The objective is to produce as output one or more candidate blocks, their orientations and degrees of support associated with being the destination address block. In principle, the image may be multispectral and include photopic (or gray-scale), color (three images corresponding to red, green, and blue filters), infrared, and color under ultraviolet illumination. The subimage corresponding to the located block would be presented as a high resolution gray-scale image (approximately 300 pixels per inch) to either a machine reader (OCR) or a human reader who will attempt to locate and read the zip code or other information within the address block. Since mail is usually presorted into the categories of letters, flats, and parcels, we assume that this information about the mail stream is available to the system.²

²Machines for automatically sorting letter mail have existed for several years. The process begins with an attempt to locate that portion of the mail piece face that carries the destination address. Present machines can only sort letter mail and sort

The ultimate decision as to whether a block is indeed the address block can only be based on reading the text in the address block and determining whether it has the syntax and semantics of an address. However, the assumption is that in almost all cases it is not necessary to recognize letters and words of text in order to recognize the address block.

Several solutions for the problem of locating address blocks on mail pieces have been recently proposed; by groups at the University of Maryland [15,30], Elsag [6], Ektron [13], Honeywell [22], and SRI [18]. All these solutions employ a two-phase operation—segmentation followed by classification. The first phase segments out the candidate objects (text blocks) and extracts the intrinsic features of each block. The second phase uses the features of the blocks to rank them by

about 55% of the mail presented to them [12,27]. The reasons for this relatively poor performance have been determined to be failure in locating the address block as well as in locating and reading the ZIP code within the address block. Since either a standard address location is assumed or a few features such as window reflectivity and high edge density are used to locate the address block, current letter mail sorting machines are confused by extraneous data on the letter face. Flats (e.g., magazines), manual letters (e.g., letters that are too thick for machine transport mechanisms), and irregular parcels and pieces (IPP) are not presently sorted automatically.

likelihood of being the destination address block. This two-phase operation is basically sequential. No provision is made to resegment an image or to use domain knowledge to refine the segmentation results. The knowledge used is limited to intrinsic address block features. Spatial relationships between the various blocks on the mail piece are rarely used.

In the remainder of this section we will describe our solution to the address block location problem, which is a specialization of the general principles described in section 2.

3.2 Identification of Knowledge Sources

KNOWLEDGE BASE OF ABLS

A statistical database of characteristics of 12,000 mail pieces [9] was utilized in developing our solution. The database was analyzed and compiled into a mail statistical database [26]. This database is divided into several categories, including letters' flats (magazines and other large flat pieces of mail), and irregular parcels and pieces. A study of the relevant characteristics of a sample set of mail pieces reveals the following characteristics:

- The number of logical blocks is variable; it ranges from simply structured first-class mail that contains three blocks, viz., postage, and return and destination addresses, to complex third-class mail with several regions of extraneous text and graphics (logos, icons, etc.). We assume five different kinds of logical blocks: destination address block, return address block, postage stamp or meter mark, extraneous text, and graphics.
- Logical blocks have certain distinct visual properties. These properties, called block features, include length, height, number of components, number of text lines, skew angle, text-line alignment, black/white pixels ratio, texture, color, shape, print method, and sensitivity in infrared or ultraviolet images, etc.
- Spatial relationships usually hold among different parts, e.g., assuming correct global orientation, the destination address block is

below and on the left side of postage, and the return address is above and on the left side of the destination address. Therefore, while the recognition of every image region is not of direct concern, knowledge of spatial relationships between labels is useful to guide label assignment.

Analysis of the database also indicates that several different image analysis operations are useful. Some examples are: locating a rectangular white address label on a multicolor background, progressive grouping of characters into text lines and text lines into text blocks, eliminating candidate regions by specialized detectors (e.g., normal postage stamps, airmail stamps, and meter marks are green, red-orange, and red, respectively, when viewed under ultraviolet light; windows with translucent covering can be detected by measuring reflectivity; certain inks are sensitive under infrared, etc.) and identifying handwritten regions.

The complete set of KSs is shown in table 2. The dependency graph of the KSs is shown in figure 2. In terms of purpose, KSs in ABLS can be divided into three categories: (1) candidate-generation tools for generating destination address candidates from the mail piece images, (2) discrimination tools for distinguishing the destination address from among the candidates, and (3) control-mechanism tools for unifying the block features between overlapping blocks, combining evidence, initiating new hypotheses, and determining when to halt processing. The utility value of each tool in ABLS is shown in table 3.

3.3 System Organization and Control

ABLS is composed of six major components (figure 3): a blackboard, the mail statistical data base, the control mechanism, the control data, a rule-based inference engine, and a tool box.

1. The blackboard contains the geometric attributes of blocks extracted from low-level image analysis, the degree of support of labeling hypotheses, and the current context; all information in the blackboard can be either ac-

Table 2. Functional descriptions of knowledge sources.

Category	Tool	Descriptions
Destination address candidate generation tools	DIGI	produces digitized images (RGB and gray-level) of the original picture.
	ADTH	adaptive thresholding to convert a gray-level image into a binary image using local contrast.
	COTH	color thresholding to extract regions of specified color in RGB images.
	SHAP	measures the degree of rectangularity of a blob.
	MSEG	bottom-up segmenter to group machine-generated characters into words, lines, and blocks.
	HSEG	bottom-up segmenter to group hand-generated thin strokes into lines and blocks.
	HWDE	regularity analyzer to detect the likely regions of hand-generated text.
	ZIPM	merges small zip code block located in the right bottom corner of a destination address candidate.
	BLCS	splits a too-high or too-wide machine-generated text block into several smaller text blocks.
	BLCM	merges machine-generated text blocks that are parallel and close in proximity.
Destination address candidate discrimination tools	HWDI	regularity analyzer for distinguishing machine-generated versus hand-generated address block.
	TEXA	texture discriminator to discriminate address-block type characters from nonaddress-block type characters.
	ICDE	postal icon detector to detect rectangular postal icons.
	UVDE	postage detector to detect the postage locations (stamp, meter mark) on UV illuminated image.
	SIZE	uses block features, e.g., aspect ratio, length, height, number of text lines, number of components, etc., to classify how likely a block is a destination address, return address, or extraneous 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.
	HEUR	uses spatial heuristics to guess the destination address block from a list of candidates, e.g., given two blocks the one on the lower right side is preferred.
	COVF	verifies the consistency of labeling hypotheses among neighboring blocks.
	UNIF	unifies the block features between blocks generated by different tools.
Control mechanism tools	EVHP	pools together the evidence generated by various tools and then uses them to generate labeling hypotheses.
	STOP	decides whether to halt processing or not.

cessed or modified by other components in the system.

- The mail statistical data base contains the statistics of the geometric features of all meaningful logical blocks in a large sample of mail pieces. This includes the probability that the destination address block and the return address block are in a particular location in a 3

by 3 grid on the image, the average and standard deviation of the aspect ratio, number of text lines, and the address block length of a typical hand and machine-generated destination address block.

- The control mechanism is responsible for checking the termination condition, selecting KSSs, combining new evidence, and updating

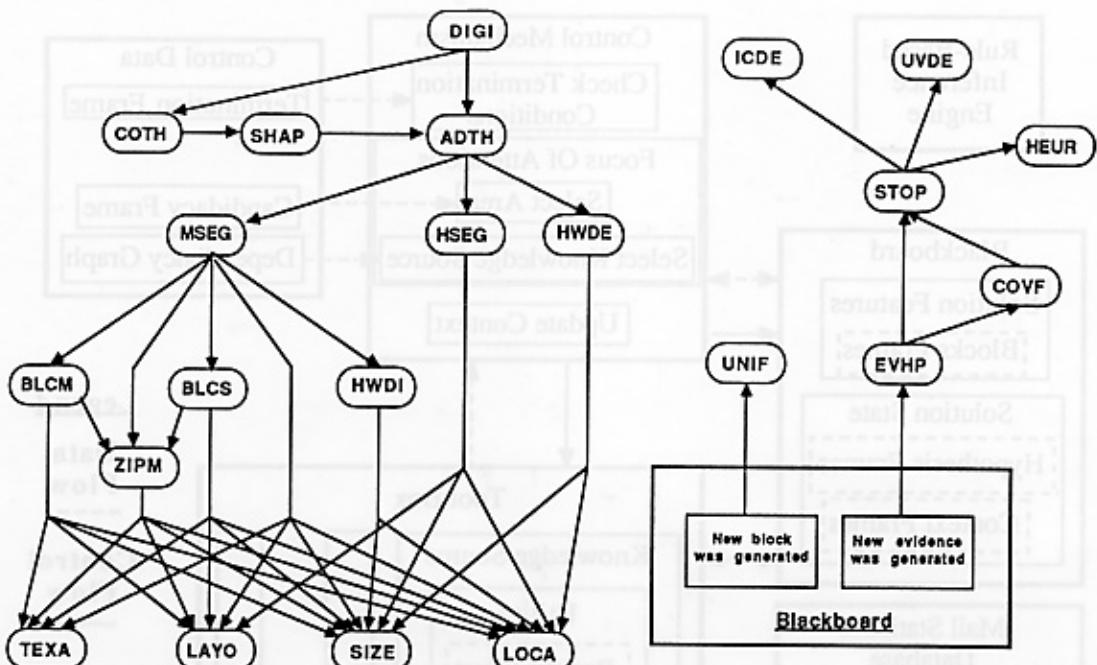


Fig. 2. Dependency graph for specifying order of applying specialized tools. The label inside each node in the graph represents an abbreviation of a knowledge source. The descriptions of the knowledge sources are in table 2. In the beginning, ADTH and COTH are triggered by DIGI. UNIF is triggered

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 other nodes are triggered as described in section 2.2

Table 3. Utility value of each tool in ABLS.

Tool	Efficiency	Effectiveness	Weighting Factor	Mail Population	Processing Time (sec.)	Special Situation	Utility Value
COTH	0.26	0.59	2.0	0.65	23.47	0.0	0.68
SHAP	0.46	0.70	2.0	0.65	11.85	0.0	1.14
MSEG	0.52	0.43	2.0	0.85	89.01	0.0	0.68
HSEG	0.53	0.56	2.0	0.15	97.87	0.0	0.12
HWDE	0.09	0.29	2.0	0.15	378.73	0.0	0.04
HWDI	0.33	0.50	2.0	0.15	118.67	10.0	10.10
ZIPM	0.14	0.70	2.0	0.85	9.24	10.0	11.29
BLCM	0.03	0.09	2.0	0.85	3.24	10.0	10.17
BLCS	0.33	0.50	2.0	0.85	2.33	10.0	11.13
TEXA	0.33	0.36	2.0	0.85	2.50	0.0	0.9
LAYO	0.16	0.32	2.0	0.85	0.80	0.0	0.68
SIZE	0.21	0.50	2.0	1.00	1.38	0.0	1.21
LOCA	0.68	0.66	2.0	1.00	2.39	0.0	2.00
COVF	0.53	0.59	2.0	1.00	4.06	10.0	11.69
HEUR	0.69	0.75	2.0	1.00	5.62	0.0	2.19
ICDE	0.11	1.00	2.0	0.35	185.28	0.0	0.05
ADTH	N/A	N/A	N/A	1.00	331.22	0.0	0.50
UNIF	N/A	N/A	N/A	1.00	1.27	10.0	10.00
EVHP	N/A	N/A	N/A	1.00	0.61	9.0	9.00
STOP	N/A	N/A	N/A	1.00	1.02	10.0	10.00

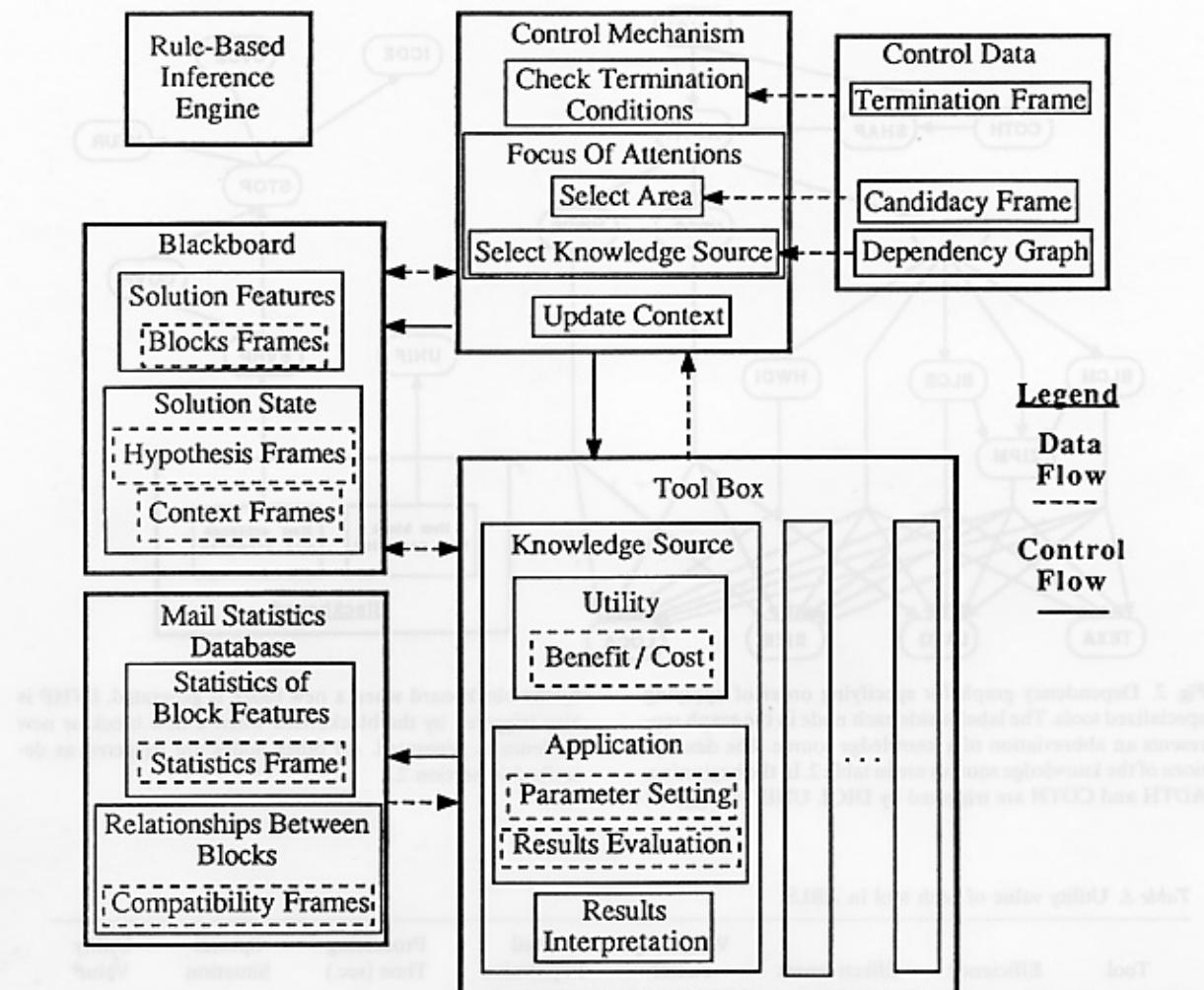


Fig. 3. ABLS organization has six components.

- context.
4. The control data provides information for the control mechanism. It contains information about the interdependency between the KSs, and the criteria for accepting a block as the destination address, or the destination address candidate.
 5. The rule-based inference engine is used for doing either forward or backward reasoning on various rule modules, which are stored with each tool. The inference engine acts as an interpreter of all the rules.
 6. The tool box contains a collection of KSs, many of which are image analysis related.

Each KS contains rules for estimating the benefit and cost of using it, selecting parameters, evaluating results, and interpreting results.

A hybrid of a *frame* and *rule-based* knowledge representation is used in ABLS to model knowledge used in coordinating KSs and in computing confidence values of various labeling hypotheses. The relationship between knowledge needed by ABLS and where that knowledge is actually represented is given in table 4. This section describes how knowledge is represented in the blackboard,

mail statistical database, and in the control unit.

3.3.1 Blackboard Data Structure. The blackboard data structure of ABLS contains a three-level hierarchy. Spatial relationships between objects are in the highest level of the hierarchy. There are at least three distinct object classes in the highest level of the hierarchy: machine-generated text blocks, hand-generated text blocks, and graphic icons. A machine-generated text block is formed by grouping character components in 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 in 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. Figure 4 shows the levels and KSS of ABLS.

Knowledge in the blackboard is stored in the *block frame*, the *hypothesis frame*, and the *context frame*. The *block frame* is used to represent the results of applying KSSs to an image. For every possible feature that can be extracted from an image by a KS, there is a corresponding slot in the block frame to record that feature value.

The *hypothesis frame* is used to record the degree of support of a labeling hypotheses for a candidate block. Since there are five possible labels, there are five labeling hypotheses in each hypothesis frame. For each possible labeling hypothesis, there is a slot in the hypothesis frame to represent 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 to be considered as destination address under the evidence accumulated so far. The performance parameters represent an estimate of the difference between the current context and the goal, i.e., the difference between the termination condition and the current situation.

Table 4. Knowledge representation in ABLS.

Knowledge needed	Implementation
Logical block's attributes	Statistics Frame
Spatial relationships among logical blocks	Compatibility Frame
Where to use a knowledge source	Candidacy Frame
When to use a knowledge source	Dependency Graph
Why use a knowledge source	Utility Rule Module
How to use a knowledge source	Application Rule Module
How to evaluate result	
How to interpret results	Results Interpretation Rule Module

3.3.2 Mail Statistical Database. Knowledge in the mail statistical data base is stored in the *statistics frame*, and the *compatibility frame*. The statistics frame stores the mean and standard deviation of the physical attributes of the destination address block, e.g., length, height, aspect ratio (block length divided by the block height), number of character components, and number of text lines. There are eight statistics frames in ABLS, two for each mail class. Among the two statistics frames for each mail class, one is for machine-generated addresses and the other is for hand-generated addresses.

The *compatibility frame* models knowledge about the two-dimensional layouts of block labels on an image. It stores information about the degrees of compatibility of spatial relationships between these block labels. The importance of spatial-relationship knowledge is two fold. First, it provides knowledge necessary to check the overall consistency of assigning block labels to each component of an image. Second, it provides clues to predict the existence of other blocks when there is ambiguity due to noise or unusual appearance, e.g., a destination address is usually broken into two separate text blocks in the segmentation process if there is excess space between the city-state names and the ZIP code. Since the ZIP code is usually located in the right bottom corner of a destination address, the ZIP code merging tool (ZIPM) can predict the existence of a destination address if it finds a small

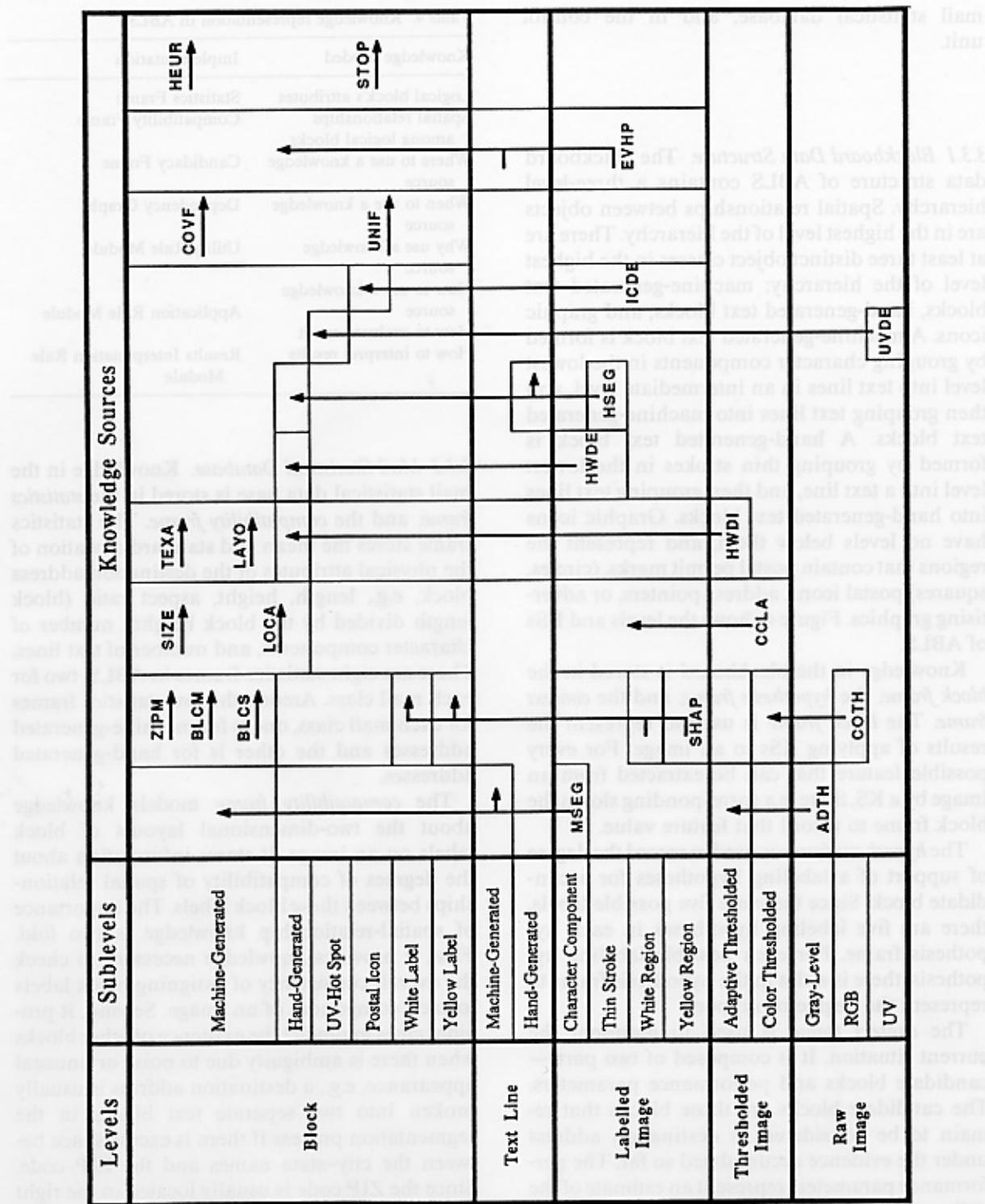


Fig. 4. The levels and knowledge sources of ABLS. Knowledge sources are indicated by arrows with the tail of arrow indicating the input level and the head of arrow indicating the output level.

text block containing roughly 5 or 9 components and close to the right bottom corner of a text block.

3.3.3 Control Mechanism and Control Data. In addition to bottom-up processing, ABLS occasionally uses top-down processing to search for the address block on a particular portion of the mail piece image. For example, if there are two segmentation tools, the system has to make a choice as to which tool should be used first. One segmentation tool is designed to extract a region with machine-generated text (MSEG) and the second is designed to extract hand-generated addresses (HSEG). Given this decision of running one of the two segmentation tools, ABLS selects the segmentation tool for machine-generated text first since (1) it runs much faster and (2) substantially more mail pieces have a machine-generated destination address than a hand-generated destination address.

In order to save processing time, the system usually needs to withhold the invocation of HSEG unless there are clues to suggest the existence of a hand-generated address. If the system can identify those clues from examining the results of the machine-generated text segmentation on hand-generated address area, it will be easy for the system to decide whether to invoke the hand-generated text segmentation tool on that area. Clues that can suggest the existence of a hand-generated address from the results of the machine-generated segmentation tool are (i) a cluster of neighboring small blocks, and (ii) a block whose size is within the acceptable range for a hand-generated address but too large for a machine-generated address.

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 the criteria for accepting a block as the DAB. The *candidacy frame* contains the minimum requirement for a block to remain as a candidate to be a destination address. Its purpose is to rule out highly unlikely candidates for the destination address block. The *dependency graph* (figure 2) is a directed graph to specify the order of applying a KS as well as to minimize reestimating the utility of each KS continuously.

3.4 Evidence Combination in ABLS

A label for a block can be a destination address, return address, postage, graphics, or extraneous text. The mail statistical database does not contain enough information to enable us to directly use Bayes' theorem, e.g., for the block's length, the mail statistical database provides the conditional probabilities $p(\text{length}|\text{destination address})$, and $p(\text{length}|\text{return address})$, but the other conditional probabilities $p(\text{length}|\text{postage})$, $p(\text{length}|\text{extraneous text})$, and $p(\text{length}|\text{graphics})$ are unavailable.

The approach we have taken is to use available statistics from the mail statistical database plus subjective estimation to initially estimate the a posteriori probability $P(l^i|e)$, and then use the experimental results (from an image database) to fine tune the initial estimate. This approach is by no means complete, or statistically sound, but it represents the best we can get from the available data.

EXAMPLE OF A POSTERIORI PROBABILITY

The following is an example of estimating the a posteriori probability $P(l^i|e)$ (confidence value), when h_i is the destination address labeling hypothesis, and e is evidence about five block features associated with a machine-generated address block—aspect ratio, length, height, number of text lines and the number of components. From the mail statistical database, the mean (μ) and standard deviation (σ) of each of these block features for the machine-generated address blocks can be obtained. Because of the wide variations of destination address blocks in the mail statistical database, acceptable ranges for the above five block features are first found so that most destination address blocks will fall within the selected feature range.

Given this acceptance range for features of a destination address, another factor we need to consider is that other blocks might also inadvertently satisfy the same feature range. For example, blocks such as the return address block, extraneous text, and graphics might also have their block features within the acceptable ranges of a destination address. Therefore, experimental

results are used to check all the blocks that have block features within the acceptable ranges, and determine what percentage of them are actually the destination address block. This percentage becomes the confidence value to support a block as the destination address block given the evidence that its features are all within the acceptable feature ranges. Of course, the acceptable ranges can always be shortened so that among all the blocks whose features are within the smaller acceptable ranges, the percentage of the destination address block is higher. The rule with an *action* part containing this smaller feature range will then have a higher confidence value in its *action* part.

EVIDENCE CATEGORIES

The evidence utilized by ABLS can be divided into eleven categories:

1. Size: block features such as aspect ratio, size, length, height, number of text lines, and number of character components
2. Layout: the layouts of text lines in a block
3. Color: the background color of a block
4. Shape: the rectangularity of a block
5. Postage: the likelihood of a block being a postal icon or meter mark
6. Location: the location of a block on a mail piece
7. Print method: machine-generated or hand-generated
8. Texture: dot-matrix print or formed-character print
9. Compatibility: the consistency of labeling hypotheses between neighboring blocks
10. ZIP: the existence of a ZIP code block in the right bottom corner of an address block
11. Heuristics: guesses or rules of thumb for picking the destination address block from a list of competing candidates.

Each category of evidence in ABLS can contain several attributes. For example, color evidence has two attributes: white background color, or yellow background color. The confidence values associated with different attribute evidence were obtained by collecting statistics after running the system through 174 training images (section 5 has

more descriptions about the training images). For example, the text evidence has a 0.43 confidence value for the destination address, 0.23 for the return address, 0.08 for the postage block, and 0.17 for extraneous text. This means that among the total number of pieces of text evidence generated after running through 174 training images, 43% of them are associated with the actual destination address, 23% with the return address, 8% with the postage block, and 17% with extraneous text.

EVIDENCE COMBINATION

The direct implementation of the Dempster-Shafer theory, as noted by Barnett [2], will result in exponential growth in computation time due to the need to enumerate all subsets of Θ . In order to save processing time in evidence combination, we assume that a rule can generate multiple pieces of evidence, and each piece of evidence assigns a number in the range of [0,1] to every singleton label of Θ such that the sum of numbers could equal 1 but does not exceed 1. If numbers assigned to labels do not sum to 1, the remaining belief, 1 minus the sum, is assigned to Θ . This assumption is appropriate in ABLS for two reasons: (i) most of the basic probability assignments for pieces of evidence are obtained from the mail statistical database and experimental results, which like the traditional probability density functions, assign belief to the singleton label of Θ only; and (ii) a block cannot be hypothesized to have multiple labels in ABLS because each block on a mail piece can only have a distinctive identity. An example of the evidence combination process is shown below: In the following example, O represents evidence combination and ∇ represents the direction of evidence combination results. Evidence 1 is the first piece of evidence associated with block 14, so a new hypothesis 1 is created. Hypothesis 1 has the same confidence values with those of evidence 1. Evidence 2 is the second piece of evidence and is combined with hypothesis 1 to produce new confidence values. These new confidence values are then combined with the confidence values in evidence 3 to produce the final confidence values in hypothesis 1.

```

(!evidence ^id 1 ^block 14 ^orientation 0 ^attribute text
           ^dest 0.43 ^return 0.23 ^postage 0.08
           ^extraneous 0.17 ^graphics 0.00)
  ▽
(!hypothesis ^id 1 ^block 14 ^orientation 0
           ^dest 0.43 ^return 0.23 ^postage 0.08
           ^extraneous 0.17 ^graphics 0.00)

○
(!evidence ^id 2 ^block 14 ^orientation 0 ^attribute alignment
           ^dest 0.55 ^return 0.21 ^postage 0.00
           ^extraneous 0.15 ^graphics 0.00)
  ▽
(!hypothesis ^id 1 ^block 14 ^orientation 0
           ^dest 0.67 ^return 0.18 ^postage 0.01
           ^extraneous 0.11 ^graphics 0.00)

○
(!evidence ^id 3 ^block 14 ^orientation 0 ^attribute zip
           ^dest 0.80 ^return 0.00 ^postage 0.00
           ^extraneous 0.15 ^graphics 0.00)
  ▽
(!hypothesis ^id 1 ^block 14 ^orientation 0
           ^dest 0.94 ^return 0.02 ^postage 0.00
           ^extraneous 0.04 ^graphics 0.00)

```

An example of the confidence values associated with each block is shown below. At this stage the process terminates.

```

(!hypothesis
  ^id 5
  ^block 7
  ^orientation 1
  ^label dest
  ^dest_candidate yes
  ^dest 0.9815
  ^return 0.006096
  ^postage 0.0004969
  ^extraneous 0.01131
  ^graphics 0.0)
(!hypothesis
  ^id 6
  ^block 13

```

```

  ^orientation 1
  ^label nil
  ^dest_candidate yes
  ^dest 0.7888
  ^return 0.07999
  ^postage 0.004557
  ^extraneous 0.1216
  ^graphics 0.0)
  (!hypothesis
    ^id 8
    ^block 31
    ^orientation 3
    ^label nil
    ^dest_candidate yes
    ^dest 0.8649
    ^return 0.06934
    ^postage 0.01546
    ^extraneous 0.03286
    ^graphics 0.0)

```

4 Implementation

In terms of the implementation, the tools in ABLS can be divided into two categories. The first category of tools is primarily implemented in the C programming language with some additional knowledge rules and functions written in the LISP programming language to do the interface between the control structure and these tools. Tools falling in this category are the ADTH, COTH, MSEG, HSEG, HWDE, HWDI TEXA, SHAP, and ICDE tools. Detailed descriptions of tools in the first category can be found in [25]. Tools in the second category are coded in knowledge rules with some additional LISP functions to implement those tasks not easily coded in the knowledge rules. Tools belonging to the second category include the BLCM, BLCS, ZIPM, LAYO, LOCA, SIZE, UVDE, COVF, HEUR, UNIF, EVHP, and STOP tools.

ABLS has been implemented on a SUN 3/260 under Sun UNIX 4.2. All the image analysis tools are written in C for speed and memory efficiency in UNIX environment. The control structure is implemented in Lisp to control and manage tools written either in C or LISP. An inference engine is included in the control structure. The inference engine is very similar to the OPS5 production system [8]. It does not run as efficiently as OPS5 but allows both forward and backward chaining and has the flexibility of separating a large set of rules into modules, and incrementally loading them during the run time. The control structure consists of 200 rules, and 130 Lisp functions for supporting those tasks that are difficult to implement as rules. The entire system consists of 10,000 lines of C code and 7,000 lines of Lisp code.

The average speed of locating an address block on a mail piece largely depends on the complexity of the mail piece. For letter mail, it usually takes less than 8 minutes but for the flats and manual letter mail, which tend to be more cluttered than letter mail, it might take over 20 CPU minutes. The average processing time to locate the destination address block on a mail piece is 10 minutes. Adaptive gray-scale thresholding is the most time-consuming step, it usually takes more than half of the total processing time.

In order to expedite the processing speed of

ABLS to the extent of real-time application, parallel processing and specialized hardware for image thresholding and segmentation are clearly needed. For this we are exploring code optimization, the use of special processors on single CPU systems, e.g., floating point processors such as the FPA or 68881 chip, as well as the use of distributed processing systems.

5 Experimental Results and Analysis

EXAMPLE OF RECOGNITION

Figure 1(c) is a photopic image of a flat mail piece. The color thresholding (COTH) tool is applied first but no white region with a reasonable size is detected. The adaptive thresholding (ADTH) and machine-generated text segmentation (MSEG) tools are then applied to the whole image and segment out several text blocks (figure 5(a,b)). The block splitting tool (BLCS) detects that one of the text blocks (block 5 of figure 5(b)) located in the center of the mail piece is too high to be the destination address. The block splitting tool (BLCS) further examines that block and detects that several short miscellaneous text lines in the top half of that block is the reason why the block is too high. In order to make that block a destination address candidate, the block splitting tool (BLCS) splits into a smaller block (figure 5(c)). The ZIP code merging tool (ZIPM) then detects a possible ZIP code block located in the right bottom corner of the newly split block, so it merges them and forms a complete destination address (figure 5(d)). The located destination address is then extracted as shown in figure 5(e).

5.1 System Performance and Analysis

In order to time the performance of ABLS, experiments were conducted using an image data base consisting of 174 complex training mail piece images. The purpose of the experiment was to determine the major causes of failure in locating the destination address. The competence of ABLS was then to be improved by either incorporating new knowledge sources or refining exist-

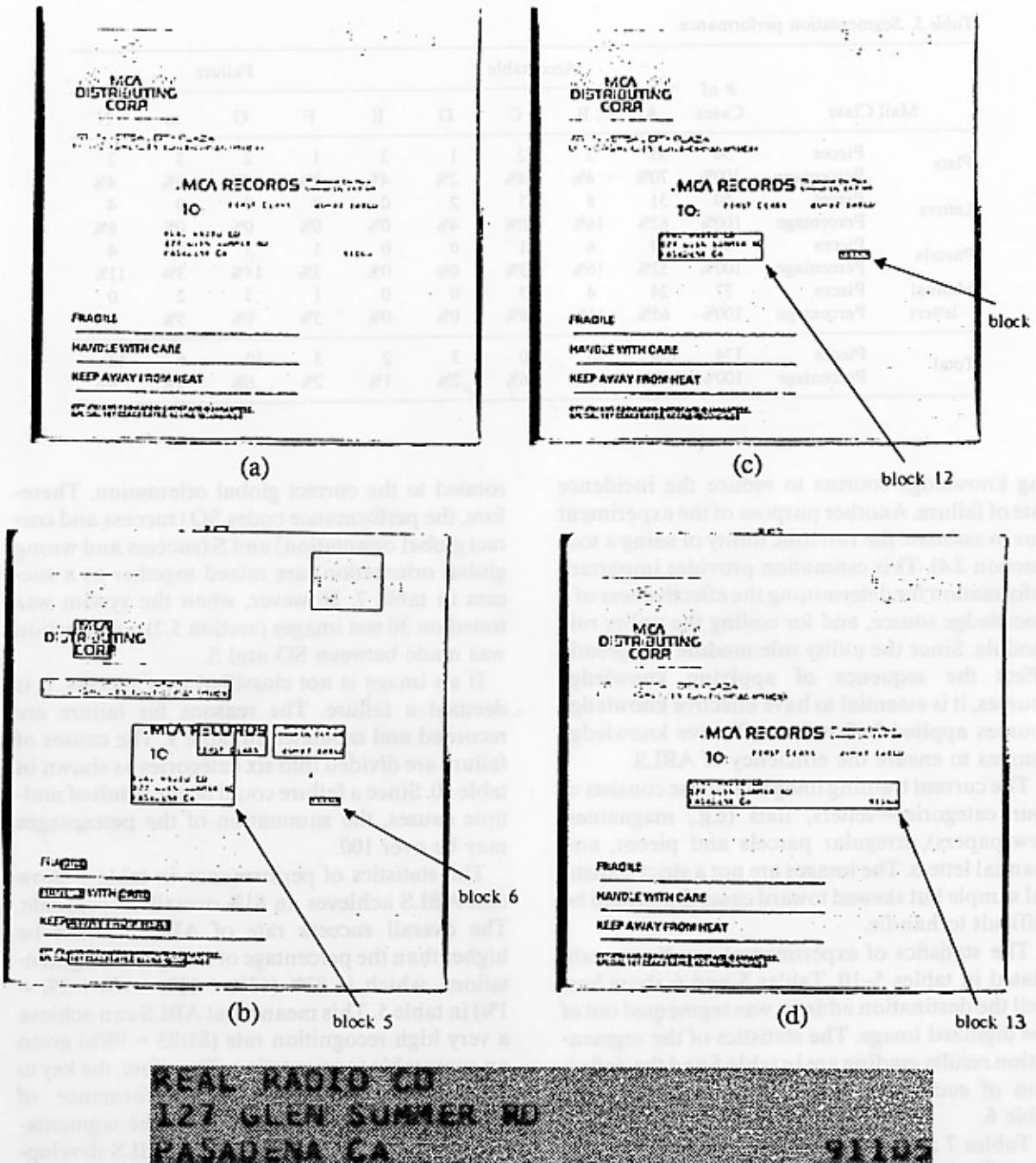


Fig. 5. (a) The gray-level thresholding results of figure 1(c). (b) The segmentation results. (c) The results of splitting block 5 into a small block (block 12). (d) The results of merging the

destination address candidate (block 12) and zip code candidate (block 6). (e) The extracted destination address.

Table 5. Segmentation performance.

Mail Class	# of Cases	Acceptable					Failure		
		A	B	C	D	E	F	G	m
Flats	Pieces	50	35	2	2	1	2	1	2
	Percentage	100%	70%	4%	4%	2%	4%	2%	4%
Letters	Pieces	50	31	8	5	2	0	0	0
	Percentage	100%	62%	16%	10%	4%	0%	0%	0%
Parcels	Pieces	37	19	6	1	0	0	1	5
	Percentage	100%	52%	16%	3%	0%	0%	3%	14%
Manual letters	Pieces	37	24	4	3	0	0	1	3
	Percentage	100%	65%	11%	8%	0%	0%	3%	8%
Total	Pieces	174	109	20	10	3	2	3	10
	Percentage	100%	63%	11%	6%	2%	1%	2%	6%

ing knowledge sources to reduce the incidence rate of failure. Another purpose of the experiment was to estimate the run time utility of using a tool (section 2.4). This estimation provides important information for determining the effectiveness of a knowledge source, and for coding the utility rule module. Since the utility rule module can greatly affect the sequence of applying knowledge sources, it is essential to have effective knowledge sources applied before less effective knowledge sources to ensure the efficiency of ABLS.

The current training image database consists of four categories—letters, flats (e.g., magazines, newspapers), irregular parcels and pieces, and manual letters. The images are not a strict statistical sample but skewed toward cases that would be difficult to handle.

The statistics of experimental results are tabulated in tables 5–10. Tables 5 and 6 show how well the destination address was segmented out of the digitized image. The statistics of the segmentation results grading are in table 5 and the definition of each segmentation grading code is in table 6.

Tables 7 and 8 show the statistics of performance. Each testing is classified as a success and correct global orientation (SO), a success and wrong global orientation (S), a partial success (P), a reject (R), or an error (E). The statistics of performance are shown in table 7 and the definition of performance codes are in table 8. In order to facilitate the development of tools in ABLS, every image in the 174 training images was

rotated to the correct global orientation. Therefore, the performance codes SO (success and correct global orientation) and S (success and wrong global orientation) are mixed together as a success in table 7. However, when the system was tested on 30 test images (section 5.2) a distinction was made between SO and S.

If an image is not classified as a success, it is deemed a failure. The reasons for failure are recorded and tabulated in table 9. The causes of failure are divided into six categories as shown in table 10. Since a failure could be the result of multiple causes, the summation of the percentages may be over 100.

The statistics of performance in table 7 show that ABLS achieves an 81% overall success rate. The overall success rate of ABLS cannot be higher than the percentage of acceptable segmentations, which is 83% (63% + 11% + 6% + 2% + 1%) in table 5. This means that ABLS can achieve a very high recognition rate ($81/83 = 98\%$) given an acceptable segmentation. Therefore, the key to substantially enhancing the performance of ABLS lies in the improvement of the segmentation results. In the early stage of ABLS development (June 1986), only one bottom-up segmentation tool (MSEG) was used, and only about 60% correct segmentation results were achieved. However, with the incorporation of more tools that either use different methods to segment an address block or to repair the segmentation results, ABLS now can achieve an 83% acceptable segmentation. The new tools incorporated into

Table 6. Definition of codes for segmentation performance.

Grades	Definitions
A	Perfect segmentation. The segment containing the destination address contains it exactly with no more or less information.
B	Slight oversegmentation. The segment containing the destination address contains a small amount of extraneous image data such as an attention line or some advertising. The majority of the segment is the entire destination address.
C	Undersegmentation, however, sorting is still possible. The largest segment of the destination address contains the street address, city, state and ZIP but part or all of the addressee name is missing.
D	Undersegmentation, however some sorting is still possible. The city, state, and ZIP are together in one segment but part or all of the street name and addressee name are separated.
E	Undersegmentation, however, there is enough of the address to effect at least a partial sort to city, state. The city and state are together in one segment but the ZIP code and possibly other parts of the address are missing.
F	Undersegmentation where there is not enough of the destination address remaining to sort. The largest segment of the destination address is missing significant parts of the city, state, or ZIP.
m	If there is undersegmentation and the destination address is broken up into 2 or more segments that could be merged to form a sortable destination address (beyond any automatic merging of segments that has already been done), this grade should be used in combination with another grade (e.g., Fm).
G	Gross oversegmentation. The total area of the segment containing the destination address is much larger than the size of the destination address and other blocks may be included.
N	None of the destination address is segmented out or labeled as a potential address block.
O	Other. Does not fall into any of the above categories

ABLS after the early development stage include: the HSEG and HWDE tools to segment hand-generated addresses, the ZIPM tool to merge the ZIP code and the address block, the BLCS tool to split a text block that is either too high or too wide to be an address block, and the BLCM tool to merge an address block with excess spaces between its parallel text lines.

Table 7. Object recognition performance.

Mail Class	# of Cases	SO				
		S	P	R	E	
Flats	Pieces	50	41	1	3	5
	Percentage	100%	82%	2%	6%	10%
Letters	Pieces	50	46	0	2	2
	Percentage	100%	92%	0%	4%	4%
Parcels	Pieces	37	23	4	3	7
	Percentage	100%	62%	11%	8%	19%
Manual letters	Pieces	37	30	2	1	4
	Percentage	100%	81%	5%	3%	11%
Total	Pieces	174	140	7	9	18
	Percentage	100%	81%	4%	5%	10%

Table 8. Definition of codes for object recognition performance.

Performance	Definitions
SO	Success and Correct Global Orientation: The destination address is the highest ranked block, and the segmented destination address contains enough address to correctly sort the mail piece (segmentation code (A-E)), and correct global orientation.
S	Success but Wrong Global Orientation: The destination address is the highest ranked block, and the segmented destination address contains enough address to correctly sort the mail piece (segmentation code (A-E)), but wrong global orientation.
P	Partial Success: The destination address is the highest ranked block, but the segmented destination address contains insufficient address to correctly sort the mail piece (segmentation code (F-O)).
R	Reject: System cannot recommend any block as the destination address (i.e., all figures of confidence were too low).
E	Error: The highest ranked block is not the destination address.

Poor image quality is the major cause of segmentation failures (column Q of table 9). The images were digitized via photographic film rather than directly, thereby introducing noise. Both the thresholding and segmentation tools can affect the outcome of the segmentation results. Our original adaptive thresholding tool (ADTH) and region-based segmentation tool

Table 9. Statistics of causes of failure.

Mail Class	# of Cases	Cause of Failure					
		H	Q	L	S	N	C
Flats	Pieces	9	2	3	2	1	4
	Percentage	100%	22%	33%	22%	11%	44%
Letters	Pieces	4	0	4	0	0	1
	Percentage	100%	0%	100%	0%	0%	25%
Parcels	Pieces	14	1	8	0	4	4
	Percentage	100%	7%	57%	0%	29%	29%
Manual letters	Pieces	7	2	2	1	2	2
	Percentage	100%	29%	29%	14%	29%	29%
Total	Pieces	34	5	17	3	7	3
	Percentage	100%	18%	50%	9%	21%	32%

Table 10. Definition of codes for causes of failure.

Code	Definition
H	Hand-generated destination address that caused a mis-segmentation or nonsegmentation of the destination address.
Q	Poor image quality that caused a mis-segmentation or nonsegmentation of the destination address.
L	The destination address located in an unusual location.
S	The size of the destination address is either unusually too large or too small.
N	The destination address is near other extraneous text blocks, which caused an oversegmentation.
C	Advertising or miscellaneous text blocks which are similar to the destination address and are in an usual location for a destination address.

(MSEG) are very effective in segmenting a machine-generated address block. However, in order to successfully segment a hand-generated address, some enhancements have been made (1) to modify the existing thresholding tool so that hand-generated thin strokes will not disappear after thresholding, and (2) to develop a more robust segmentation tool (HSEG) to group thin strokes into a complete hand-generated address block.

The experimental statistics (table 7) also show that ABLS achieved a high success rate on letter mail. This is because most letter mail pieces are structured and have the destination address in a

standard position. For the other three mail classes, generally, the major cause of failure, besides segmentation failure, is the presence of extraneous text blocks that are too close to the destination address (column N of table 9). This kind of failure is particularly common in flats, which usually have more extraneous text than other mail classes. The block splitting tool (BLCS) of ABLS is aimed at solving this kind of failure, and achieves limited success on a few cases.

5.2 Performance with Test Images

In July 1987, a set of 30 test images that were not part of the design set were used to make the final evaluation of ABLS. The 30 test images were chosen from thousands of mail pieces representing a cross section of mail pieces encountered in the mail stream. The performance on the 30 test images was: 77% (SO), 0% (S), 23% (P), 0% (R), and 0% (E). These results are consistent with the results of the previous experiments. The performance code for test images is slightly different from the definitions in table 8 because the definition of partial success (P) was relaxed to include every test image that does not have a satisfactory segmentation result. Therefore, there were no rejects and errors because all failure cases among the test images were due to poor segmentation results.

The rule-based system developed at the University of Maryland [30] for address block location also achieved a very high success rate, but only with letter mail and machine-generated addresses. The success rate of ABLS on letter mail with machine-generated addresses is comparable with that of the Maryland system. The Buffalo ABLS is, however, more versatile and is designed to deal with a much larger variety of mail piece images.

5.3 Chronology of Experimental Results

A short chronology of experimental results from June 1986 to December 1987 is shown in figure 6. Before January 1987, the experimental results were only classified into three categories—success, reject, and error. For experimental results later than November 1986, an additional category “partial success” was added to the performance categories. A complete description of each performance code is in table 8. The latest experimen-

tal results as of December 1987 have been tabulated in table 7.

As shown in figure 6, as more tools were incorporated into ABLS, the overall success rate is gradually improved from 50% to 81%, except for a slight dip in January 1987. The success rate for the entire 174 training images in the image database was 72% in January 1987, which is slightly lower than the 75% success rate for the first 97 images in November 1986. The reason for this dip is that there were 30 letter images (10 in Trial and 20 in Letters) in the first 97 images but only 20 letter images in the last 77 images. Since letter mail has the highest success rate among the mail classes, a higher volume of letter mail means a higher overall success rate.

6 Conclusion

A methodology for designing a system to recognize an object in an environment that may be

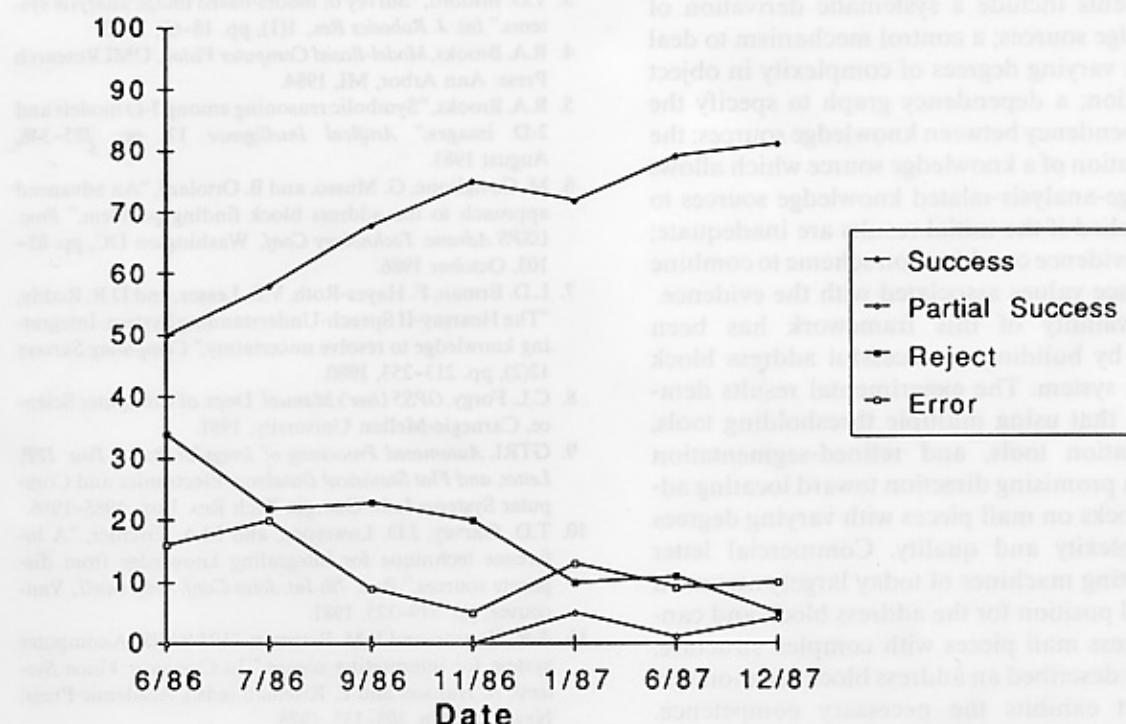


Fig. 6. Chronology of experimental results.

structured, partially structured, or random has been described. The approach has been to utilize specialized tools to generate several hypotheses (candidate objects) and to select the best hypothesis by performing certain tests. The method uses an opportunistic strategy in that model information is used only if necessary. Randomness in the input is handled by using certain tools whose success leads to by-passing model-based reasoning. The framework is flexible enough to incorporate many tools. Knowledge about the selection and utilization of each tool is kept independently and separately for each tool so that the addition, deletion, or modification of a tool will not cause side effects on other tools.

The blackboard model was used as a starting point, and then detailed descriptions for building an object recognition system was given. In addition to the three major components in the blackboard model, the *knowledge sources*, the *blackboard data structure*, and the *control mechanism*, several enhancements were incorporated into the framework to make it more suitable for locating objects in uncertain environments. Those enhancements include a systematic derivation of knowledge sources; a control mechanism to deal with the varying degrees of complexity in object recognition; a dependency graph to specify the interdependency between knowledge sources; the organization of a knowledge source which allows the image-analysis-related knowledge sources to be reapplied if the initial results are inadequate; and an evidence combination scheme to combine confidence values associated with the evidence.

The validity of this framework has been verified by building a successful address block location system. The experimental results demonstrate that using multiple thresholding tools, segmentation tools, and refined-segmentation tools is a promising direction toward locating address blocks on mail pieces with varying degrees of complexity and quality. Commercial letter mail sorting machines of today largely assume a standard position for the address block and cannot process mail pieces with complex structure. We have described an address block location system that exhibits the necessary competence. Work is needed for the system to exhibit real-time performance so as to be incorporated into postal

OCR machines.

ACKNOWLEDGEMENTS

We are indebted to Jonathan Hull, Paul Palumbo, Steve Tylock, Leonard Kuan, and Ed Cohen for valuable suggestions and assistance. We also would like to thank Dr. Rick Glickman and Mr. Gary Herring of the Office of Advanced Technology of the United States Postal Service for the support of this work as well as for encouragement and many suggestions.

REFERENCES

- D.H. Ballard, C.M. Brown, and J.A. Feldman, "An approach to knowledge-directed image analysis." In *Computer Vision Systems*, A. Hanson and E. Riseman (eds.), Academic Press: New York, pp. 271-281, 1978.
- J.A. Barnett, "Computational methods for a mathematical theory of evidence," *Proc. 7th Int. Joint Conf. Artif. Intell.*, Vancouver, pp. 868-875, 1981.
- T.O. Binford, "Survey of model-based image analysis systems," *Int. J. Robotics Res.*, 1(1), pp. 18-62, 1982.
- R.A. Brooks, *Model-Based Computer Vision*, UMI Research Press: Ann Arbor, MI, 1984.
- R.A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artificial Intelligence* 17, pp. 285-348, August 1981.
- M. Caviglione, G. Musso, and B. Ortolani, "An advanced approach to the address block finding problem," *Proc. USPS Advanc. Technology Conf.*, Washington DC, pp. 85-103, October 1986.
- L.D. Erman, F. Hayes-Roth, V.R. Lesser, and D.R. Reddy, "The Hearsay-II Speech-Understanding System: Integrating knowledge to resolve uncertainty," *Computing Surveys* 12(2), pp. 213-253, 1980.
- C.L. Forgy, *OPSS User's Manual*, Dept. of Computer Science, Carnegie-Mellon University, 1981.
- GTRI, *Automated Processing of Irregular Parcel Post: IPP, Letter, and Flat Statistical Database*, Electronics and Computer Systems Lab, Georgia Tech Res. Inst., 1985-1986.
- T.D. Garvey, J.D. Lawrence, and M.A. Fischler, "A inference technique for integrating knowledge from disparate sources," *Proc. 7th Int. Joint Conf. Artif. Intell.*, Vancouver, pp. 319-325, 1981.
- A.R. Hanson and E.M. Riseman, "VISIONS: A computer system for interpreting scenes." In *Computer Vision Systems*, A. Hanson and E. Riseman (eds.), Academic Press: New York, pp. 303-333, 1978.
- J.J. Hull, G. Krishnan, P. Palumbo, and S.N. Srihari, "Optical character recognition techniques in mail sorting: A

- review of algorithms," Tech. Rept. 214, Dept. of Computer Science, State Univ. of New York at Buffalo, June 1984.
- 13. A.F. Lehar, "The automatic detection and ranking of address blocks in unconstrained mail," *Proc. USPS Advanc. Technology Conf.*, Washington DC, pp. 132-146, October 1986.
 - 14. M.D. Levine and S.I. Shaheen, "A modular computer vision system for picture segmentation and interpretation," *IEEE Trans. PAMI-3*, 5, pp. 540-556, September 1981.
 - 15. A. Litcher, S. Antoy, P.S. Yeh, and A. Rosenfeld, "A rule-based system for address recognition," *Proc. USPS Advanc. Technology Conf.*, Washington DC, pp. 66-84, October 1986.
 - 16. T. Matsuyama and V. Hwang, "SIGMA: A framework for image understanding—Integration of bottom-up and top-down analysis," *Proc. Int. Joint Conf. Artific. Intell.*, Los Angeles, pp. 908-915, August 1985.
 - 17. D.M.J. McKeown, W.A. Harvey, and J. McDermott, "Rule-based interpretation of aerial imagery," *IEEE Trans. PAMI-7*, 5, pp. 570-585, September 1985.
 - 18. P.G. Mulgaonkar and A. Bergman, "Address block location: The SRI approach," *Proc. USPS Advanc. Technology Conf.*, Washington DC, pp. 161-178, October 1986.
 - 19. M. Nagao and T. Matsuyama, *A Structural Analysis of Complex Aerial Photographs*. Plenum Press: New York, 1980.
 - 20. H.P. Nii, "Part One, Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures," *AI Magazine* 7(2), pp. 38-53, Summer 1986.
 - 21. Y. Ohta, *Knowledge-Based Interpretation of Outdoor Natural Scenes*. Pitman Publishing: Marshfield MA, 1985.
 - 22. J.E. Orrock, J. Jelinek, and K. Schaper, "A system for automatic address block location," *Proc. USPS Advanc. Technology Conf.*, Washington DC, pp. 104-115, October 1986.
 - 23. G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
 - 24. L.G. Shapiro, "The role of AI in computer vision," *Proc. 2nd IEEE Computer Soc. Conf. Artif. Intell. Applications*, Miami Beach, FL, pp. 76-81, December 1985.
 - 25. S.N. Srihari, C.H. Wang, P.W. Palumbo, and J.J. Hull, "Recognizing address blocks on mail pieces: Specialized tools and problem solving architecture," *AI magazine* 8(4), pp. 25-40, Winter 1987.
 - 26. S.N. Srihari, J.J. Hull, P.W. Palumbo, and C.H. Wang, "Address block location: Evaluation of image and statistical database," Tech. Rept. 86-09, Dept. of Computer Science, SUNY at Buffalo, April 1986.
 - 27. USPS, Engineering Report on OCR Readability Guidelines, June 1984.
 - 28. C.H. Wang and S.N. Srihari, "Object recognition in structured and random environments: Locating address blocks on mail pieces," *Proc. 5th AAAI Nat. Conf.*, Philadelphia, pp. 1133-1137, August 1986.
 - 29. L.P. Wesley and A.R. Hanson, "The use of an evidential-based model for representing knowledge and reasoning about images in the VISIONS system," *Proc. of the Workshop on Computer Vision: Representation and Control*, Rindge, New Hampshire, pp. 14-25, August 1982.
 - 30. P.S. Yeh, S. Antoy, A. Litcher, and A. Rosenfeld, "Address location on envelopes," *Pattern Recognition* 20(2), pp. 213-217, 1987.