

AD-A087 684

HONEYWELL SYSTEMS AND RESEARCH CENTER MINNEAPOLIS MN

F/O 17/7

ADVANCED PATTERN-MATCHING TECHNIQUES FOR AUTONOMOUS ACQUISITION-ETC(U)

MAR 80 P M NARENDRA, J J GRABAU

DAAK70-79-C-0114

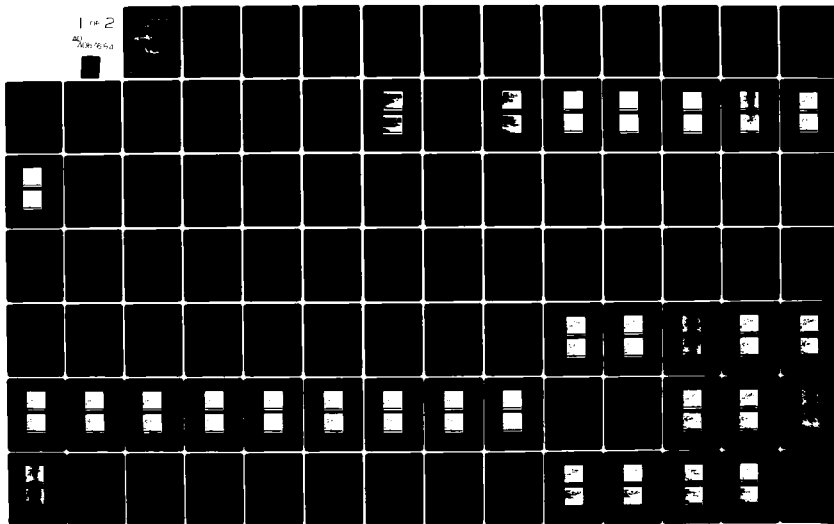
UNCLASSIFIED

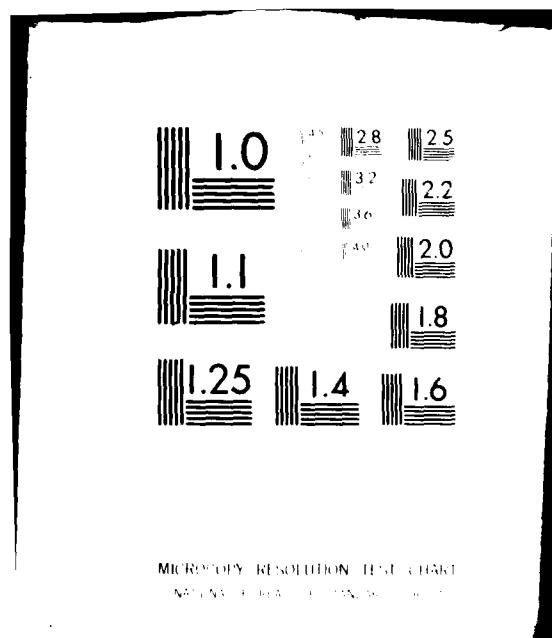
80SRC25

NL

1 of 2

40
MAR 80 5:4





LEVEL

A082038

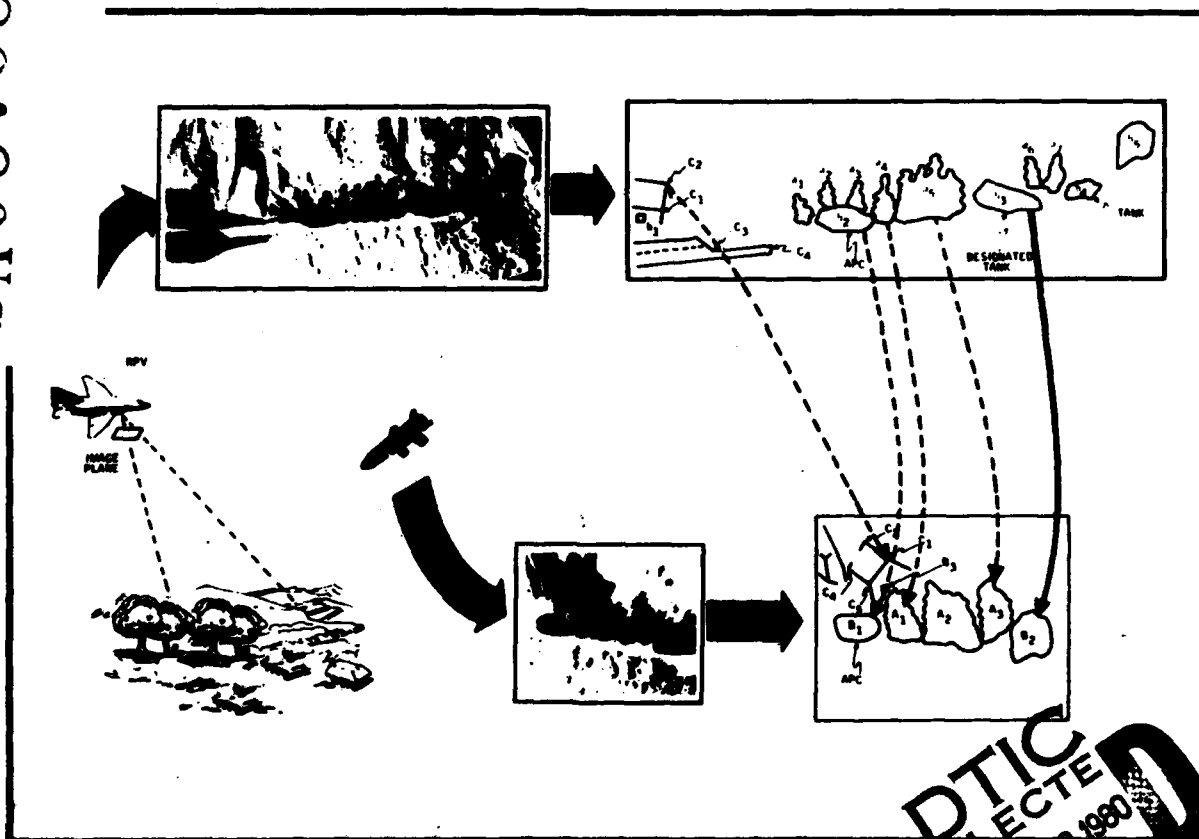
2

ADVANCED PATTERN-MATCHING TECHNIQUES FOR

AUTONOMOUS ACQUISITION

Second Quarterly Progress Report

ADA 087684



by

P.M. Narendra
J.J. Graben

Honeywell
SYSTEMS & RESEARCH CENTER,
2800 RIDGWAY PARKWAY
MINNEAPOLIS, MINNESOTA 55413

This document has been approved
for public release and sale; its
distribution is unlimited.

DTIC
ELECTE
AUG 8 1980

80 8 7 025

DDC FILE COPY

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
AD-A087684		
4. TITLE (and Subtitle) ADVANCED PATTERN-MATCHING TECHNIQUES FOR AUTONOMOUS ACQUISITION		5. TYPE OF REPORT & PERIOD COVERED Quarterly Progress Report 23 Nov 1979 to 23 Feb 1980
7. AUTHOR(s) P. M./Narendra J. J./Grabau		6. PERFORMING ORG. REPORT NUMBER 80SRC25
9. PERFORMING ORGANIZATION NAME AND ADDRESS Honeywell Inc., Systems and Research Center 2600 Ridgway Parkway Minneapolis, Minnesota 55413		8. CONTRACT OR GRANT NUMBER(s) DAAK70-79-C-0114
11. CONTROLLING OFFICE NAME AND ADDRESS Night Vision and Electro-Optics Laboratory Fort Belvoir, Virginia 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE March 1980
		13. NUMBER OF PAGES 105
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Unlimited		
This document has been approved for public release and its distribution is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Mr. C. Goehrig is NV&EOL Contract Monitor on this program.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Infrared	Target recognition	Symbolic processing
FLIR	Pattern recognition	Target tracking
Target cueing	Image processing	Scene analysis
Target screening	Artificial intelligence	Pattern matching
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
This is the second interim quarterly progress report on "Advanced Pattern-Matching Concepts," NV&EOL contract No. DAAK70-79-C-0114. It reports the results of work performed between November 23, 1979 and February 23, 1980. The key objective of this effort is the development of pattern-matching algorithms which can impart autonomous acquisition capability to		

DD FORM 1473

EDITION OF 1 NOV 55 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (continued)

precision-guided munitions such as Copperhead and Hellfire. Autonomous acquisition through pattern matching holds the promise of eliminating laser designation and enhancing fire power by multiple target prioritization.

The pattern-matching approach being developed under this program is based on a symbolic pattern-matching framework, which is suited for the autonomous acquisition scenario. It is based on matching a symbolic representation derived from the two images, and it can accommodate the stringent pattern-matching criteria established by the scenario: enormous differences in the scene perspective, aspect and range between the two sensors, differences in sensor characteristics and illumination, and scene changes such as target motion and obscuration from one view point to the other.

Accession For
NTIS GFA&I
DFO TAB
Unannounced
Justification

By _____
Distribution/

_____ Codes
_____ and/or
_____ special

List
A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

Section		Page
1	INTRODUCTION	1
	Overview of Pattern Matching	3
	Summary of Progress	4
	Report Organization	6
2	CANDIDATE MATCH SELECTION	7
3	BRANCH-AND-BOUND MATCHING ALGORITHM	23
4	CRITERION FUNCTIONS	39
	The Distinctiveness Component	40
	The Object Similarity Component	41
	The Configuration Component	41
5	SYSTEM SIMULATION AND RESULTS	49
	Example 1	52
	Example 2	69
	Example 3	69

CONTENTS (concluded)

Section		Page
6	ANALYSIS OF A PRIORI INFORMATION	76
7	MINIMAL SPANNING TREES	80
8	PATTERN-MATCHING DATA BASE	87
9	PLANS FOR THE NEXT REPORTING PERIOD	95

LIST OF ILLUSTRATIONS

Figure		Page
1	Pattern-Matching Overview	2
2	Block Diagram of Pattern-Matching Process	8
3	FLIR Images for Example	10
4	Objects Extracted for Images of Figure 3	12
5	Candidate Match Selection Example	13
6	Illustrates the Sensor Geometry for the Example in Figure 3	21
7	Example of a Branch-and-Bound Search Tree	25
8	Branch-and-Bound Algorithm Flow Chart	28
9	A Prioritized Branch-and-Bound Search Tree	37
10	An Example Showing How the Function f_{ms} Detects Bad Matches	43
11	Examples of Topologically Consistent and Topologically Inconsistent Matches	46
12	Block Diagram of Simulation Software	50
13	A Complete Search Tree for Example 1	54
14	The First 14 Matches Evaluated in Example 1	55
15	The First Four Matches Evaluated in Example 2	71

LIST OF ILLUSTRATIONS (concluded)

Figure		Page
16	A Minimal Spanning Tree for a Graph	81
17	A Minimal Spanning Mat Clusters When it is Broken	82
18	Minimal Spanning Trees in Images of Figure 3	83
19	Some FLIR Images from the AP Hill Data Base	88

LIST OF TABLES

Table		Page
1	Example of Editing Matches in Step 2 of the Candidate Match Selection Algorithm	11
2	An Example of Ranking Acceptable Matches	11
3	Example Candidate Match Selection	19
4	Candidate Match Selection	24
5	Branch-and-Bound Algorithm Parameters for Example 1	53
6	Candidate Match Selection for Example 2	70
7	Data for Example 3	75
8	Current Data Base Summary	94

SECTION 1

INTRODUCTION

This is the second interim quarterly progress report on "Advanced Pattern-Matching Concepts," NV&EOL contract No. DAAK70-79-C-0114. It reports the results of work performed between November 23, 1979 and February 23, 1980.

The key objective of this effort is the development of pattern-matching algorithms which can impart autonomous acquisition capability to precision-guided munitions such as Copperhead and Hellfire. Autonomous acquisition through pattern matching holds the promise of eliminating laser designation and enhancing fire power by multiple-target prioritization. However, this application imposes stringent performance requirements on the pattern-matching algorithm--it must be robust under perspective and aspect change, target motion, illumination change, sensor differences, image quality, and target obscuration. Conventional pattern-matching techniques are incapable of meeting the performance requirements in the autonomous acquisition scenario.

The pattern-matching approach being developed under this program is based on a symbolic pattern-matching framework which is suited for the autonomous acquisition scenario. It is based on matching a symbolic representation derived from the two images, rather than on numerical correlation. It can

accommodate the stringent pattern-matching criteria established by the scenario: enormous differences in the scene perspective, aspect and range between the two sensors, differences in sensor characteristics and illumination, and scene changes such as target motion and obscuration from one view point to the other.

Figure 1 shows a broad overview of the symbolic pattern-matching approach. The symbolic pattern-matching technique is based on matching a symbolic representation of the two images, not the gray levels of the individual picture elements themselves, as in conventional correlation approaches. A symbolic representation of an image consists of describing objects (or distinctive elements) in the image and their positions.



Figure 1. Pattern-Matching Overview

The symbolic matching technique operates on the symbolic image to find an optimal match which simultaneously ensures 1) that the matched objects are similar in their descriptors and 2) that the interobject configuration in the two matched sets of objects are consistent. This yields a robust pattern-matching algorithm which is insensitive to a number of variables, including variation in object descriptors.

OVERVIEW OF PATTERN MATCHING

Figure 1 shows the basic steps in symbolic pattern matching between the reference and sensed images. The first step is to extract object features from both images. Note that the result of this scene analysis combines both blobs (from a target screener segmenter) and edge-based features such as long lines and vertices as well. The next stage is the symbolic description of these objects (blobs and lines) as lists of object descriptors and their positions in the two FOV. Finally, symbolic pattern matching is performed between the two symbolic images to establish correspondence between the target designated in the reference image and the corresponding object in the sensed image.

The key to a robust symbolic matching is that in an optimal match both the object descriptions and the inter-object relationships in the two images should be consistent. Criteria for evaluating interobject configuration matches must be able to account for scene perspective, aspect, roll, and scale differences. The development of an efficient search algorithm to examine the most promising object and configuration matches as defined by the match criteria is the main thrust of this program.

SUMMARY OF PROGRESS

The following is a summary of the progress made in the second quarter toward the above program objective.

1. Algorithms for candidate match selection have been developed, implemented, and evaluated with FLIR imagery. Examples indicate that the candidate match selection process is able to limit the number of candidates to a few without discarding any correct object-to-object matches. The procedure is illustrated with an example taken from a pair of FLIR images.
2. Given the candidate matches, an efficient branch-and-bound enumeration scheme to find the best match of object subsets in the two fields of view has been developed and implemented. Prioritized and unprioritized versions of an enumeration scheme are illustrated with examples. The algorithm finds the best match without an exhaustive enumeration of all possible matches.
3. Several criterion functions have been specified for use in the branch-and-bound algorithm. Criterion functions are measures of the goodness of an object configuration match between the two images. The criteria include the similarity of individual object-to-object matches, geometric transformation consistency, and topological consistency. It is shown that the criteria are monotonic, guaranteeing that the branch-and-bound-algorithm yields the optimal solution among all possible solutions.

4. All components of the matching process—the object extraction, object feature extraction, candidate match selection, and the branch-and-bound algorithm (with the criterion functions)—have been installed in a complete system simulation at the Honeywell Image Processing facility. Two pattern-matching examples on FLIR imagery obtained with the system simulation are included. The examples demonstrate that correct pattern matching is feasible between two fields of view 50 deg apart, even when only a few objects are common to both fields of view.
5. A systematic analysis of how a priori information can be utilized to speed up the search process has been made. A computer program to predict the efficiency of the algorithm utilizing a priori information has been coded and tested.
6. A minimal spanning tree approach to identifying distinctive clusters of objects in the two fields of view has been developed. The technique appears to have the potential of speeding up the search process for the optimum match by finding good initial object-to-object matches, using similar distinctive object clusters.
7. The data base generation task continues; 36 frames of FLIR imagery have been digitized from FLIR video tapes at aspects of 0° , -35° , and $+50^\circ$ and spanning ranges from 8 kilometers to overflight. This data base has identified a number of key issues and is helping us quantify the performance of the approach under conditions that are typical of the expected conditions in the Copperhead and Hellfire scenarios.

REPORT ORGANIZATION

The body of this report is organized under the following headings:

- Section 2: Candidate Match Selection
- Section 3: Branch-and-Bound Matching Algorithm
- Section 4: Criterion Functions
- Section 5: System Simulation and Results
- Section 6: Analysis of A Priori Information
- Section 7: Minimal Spanning Trees
- Section 8: Pattern-Matching Data Base
- Section 9: Plans for the Next Reporting Period

SECTION 2

CANDIDATE MATCH SELECTION

The candidate match selection algorithm is presented in this section. As shown in Figure 2, the candidate match selection process operates on the output of the feature extraction process and selects a list of likely matches for the branch-and-bound algorithm. It effects a drastic reduction in the complexity of the problem that must be solved by the branch-and-bound algorithm. Part of this reduction is achieved by removing from consideration the nondistinctive objects in the two images. The distinctive objects are the objects that have high probability of having corresponding matches in the other image. The rest of the reduction is achieved by rejecting those object-to-object matches that are clearly dissimilar, as measured by the object features.

The candidate match selection algorithm in the current simulation exploits four object features. They are as follows:

1. Contrast with background
2. Area
3. $\text{Perimeter}/\sqrt{\text{area}}$
4. Minor axis length/major axis length

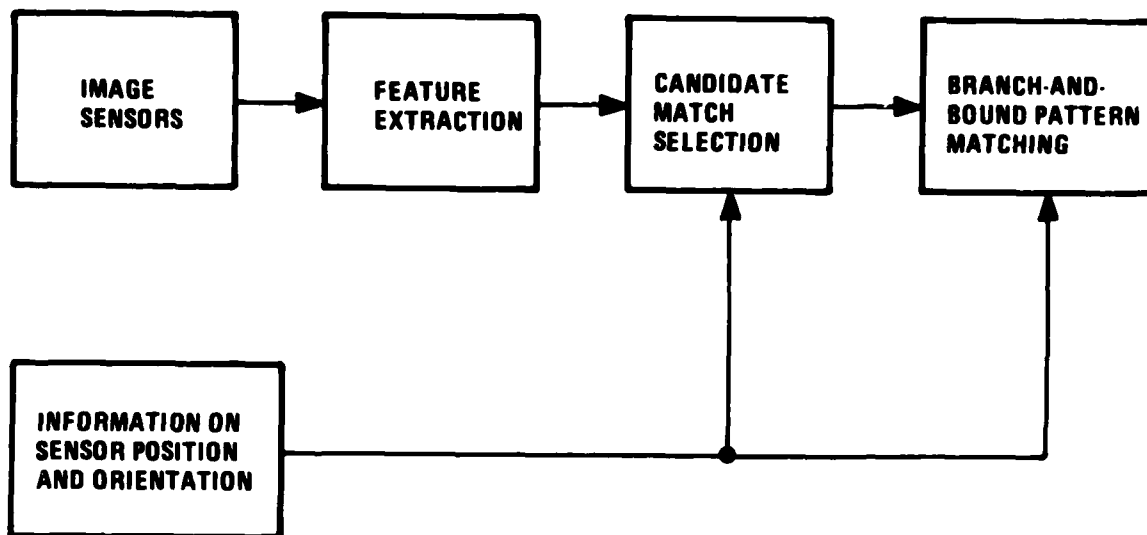


Figure 2. Block Diagram of Pattern-Matching Process

The candidate match selection is done in two steps:

1. Edit from the list of objects in the reference image all but the N_r most distinctive objects, using contrast as a measure of distinctiveness. Similarly, edit from the list for the sensed image all but the N_s most distinctive objects.
2. For each remaining object in the sensed image, select candidate matches in the reference image. First, remove from consideration those of the remaining objects in the reference image that have an area too large or too small or shape features too different to be matches for the object in the sensed image. Then use the area and shape information to rank those objects not edited. Retain the C_{max} best matches.

It is assumed that estimates of the relative ranges for the two images are available for use in evaluating the quality of area matches. N_s , N_r , and C_{max} are chosen large enough to include the correct matches with high probability.

An example of Step 2 for the image pair in Figure 3 is shown in Tables 1 and 2. The objects extracted by the segmentation algorithm are shown in Figure 4. The editing process of Step 2 is illustrated in Table 1. Measures of similarity of the 10 most distinctive objects in the reference image with an object in the sensed image are shown for the three features used in Step 2. The larger the measure, the larger the dissimilarity between objects. Very dissimilar matches are marked with asterisks, and the objects remaining after editing are circled. The ranking process is shown in Table 2. Ranks for similarity in the shape features are combined into a shape score. The shape score and the area similarity rank determine the overall rank of an object for match quality. The true match is circled.

An example will illustrate the level of performance of the candidate match selection algorithm. The candidate match selection for the segmentations of Figure 4 is shown in Figure 5 and in Table 3. There are four correct object-to-object matches in this image pair. They happen to be the four targets. The targets were among the five most distinctive objects in the sensed image, and their correct matches were ranked either first or second among the candidate object lists.



a. Sensed image



b. Reference image

Figure 3. FLIR Images for Example

TABLE 1. EXAMPLE OF EDITING MATCHES IN STEP 2
OF THE CANDIDATE MATCH SELECTION ALGORITHM

OBJECT IN REFERENCE IMAGE	CONTRAST RANK	DISCREPANCY MEASURES		
		AREA	$p/\sqrt{A}$	MINOR/MAJOR
142	1	.28	.58	.28
103	2	.75*	.23	.62
141	3	1.18*	1.76*	.04
138	4	.13	.72	.19
158	5	.66*	.21	.24
137	6	.58*	1.06	1.72*
162	7	1.12*	.25	.20
161	8	.11	.77	.00
125	9	.87*	1.00*	1.22*
100	10	.06	1.13	1.13

* INDICATES DISCREPANCY MEASURE LARGE ENOUGH
TO REJECT. ACCEPTED MATCHES ARE ENCIRCLED.

TABLE 2. AN EXAMPLE OF RANKING ACCEPTABLE MATCHES

	RANK FOR $p/\sqrt{A}$	RANK FOR MINOR/MAJOR	SHAPE SCORE	AREA RANK	COMBINED SCORE
101	3	1	4	1	1
138	2	2	4	2	2
142	1	3	4	3	3
100	4	4	8	4	4

CORRECT MATCH ENCIRCLED

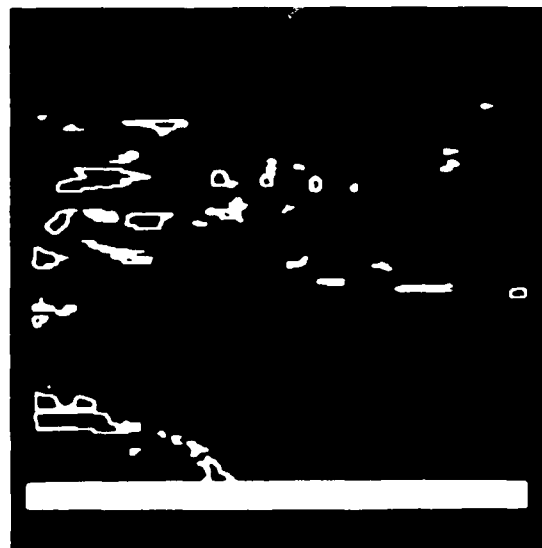
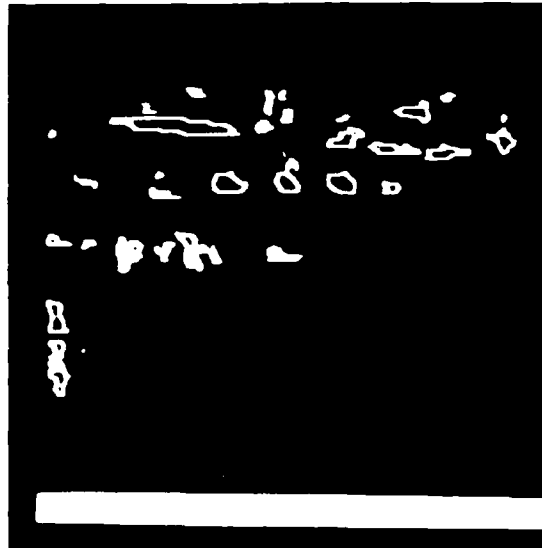


a. Sensed image



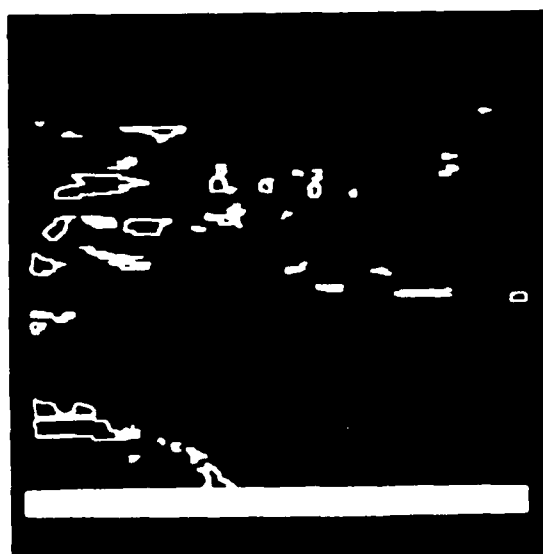
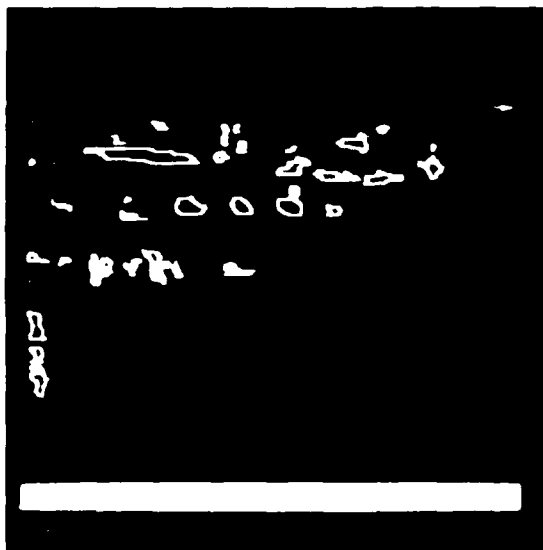
b. Reference image

Figure 4. Objects Extracted for Images of Figure 3



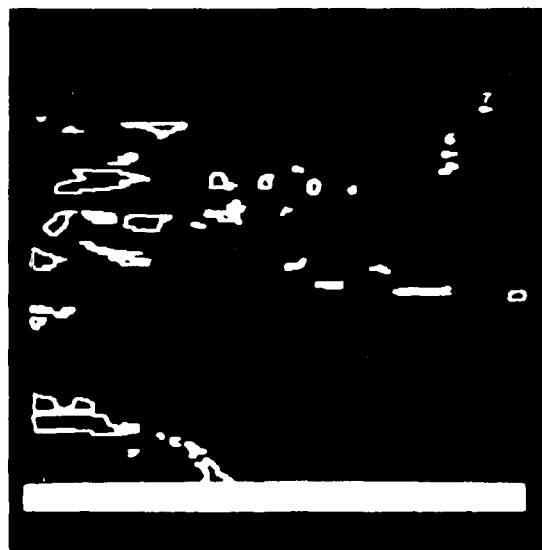
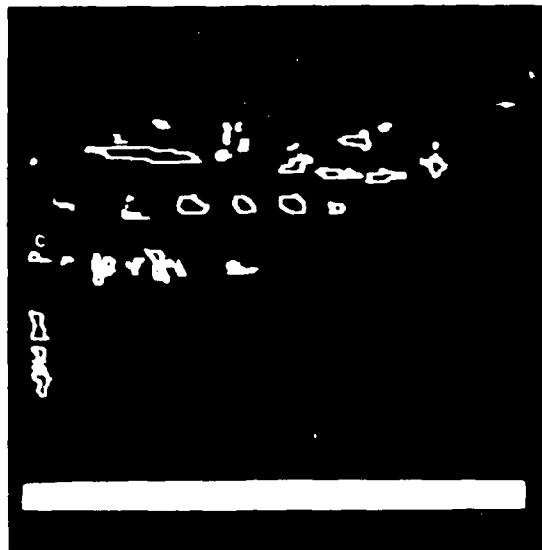
- a. Candidate matches for A are 1 and 2. (Object 1 is in a three-object group at the left.)

Figure 5. Candidate Match Selection Example



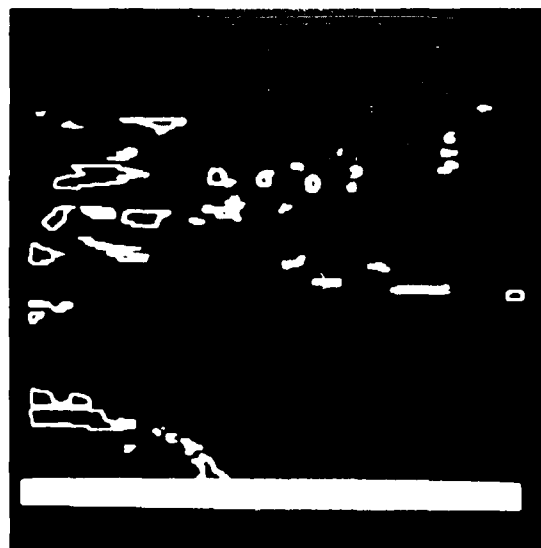
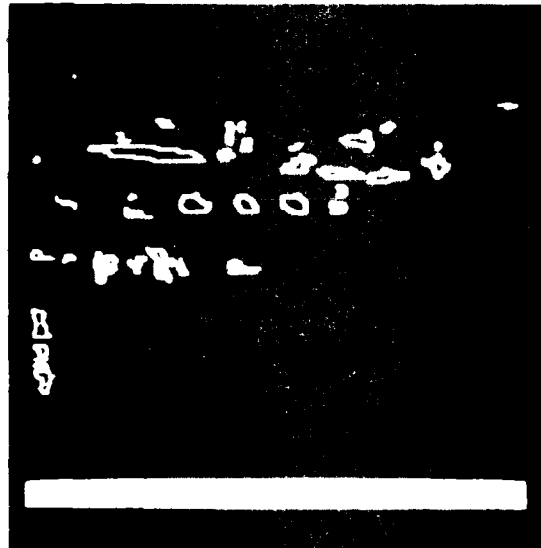
b. Candidate matches for B are 3 and 5

Figure 5. Candidate Match Selection Example (continued)



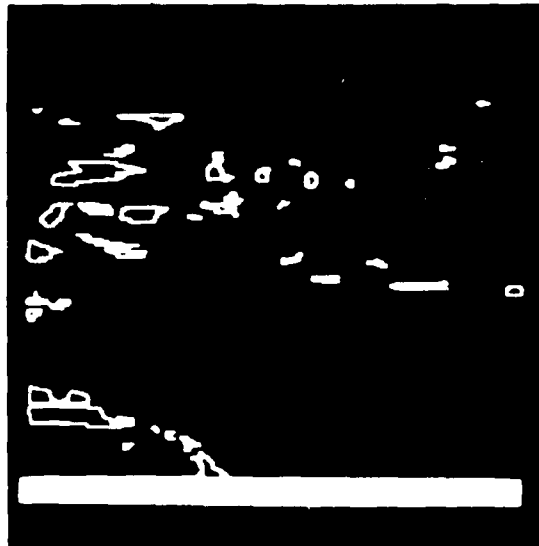
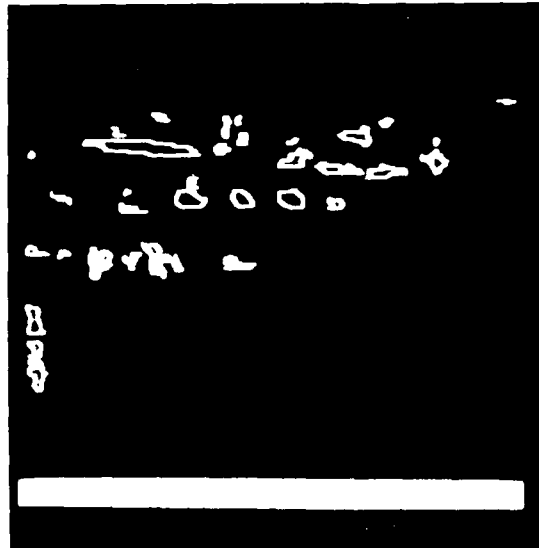
c. Candidate matches for C are 6 and 7

Figure 5. Candidate Match Selection Example (continued)



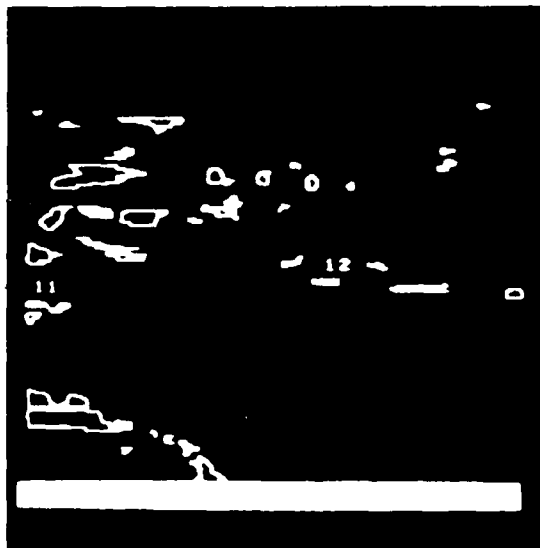
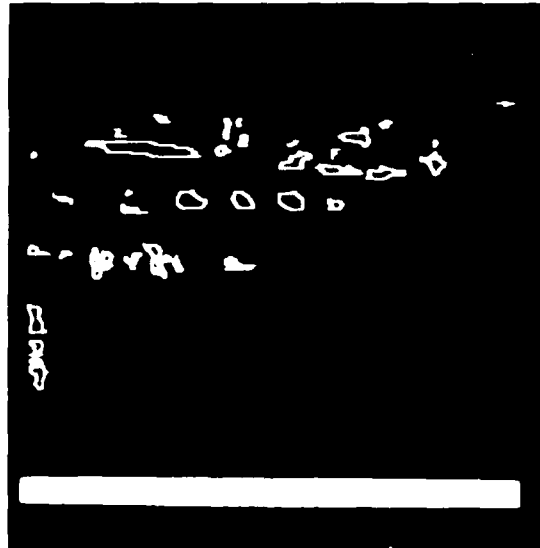
d. Candidate matches for D are G and H

Figure 5. Candidate Match Selection Example (continued)



- e. Candidate matches for E are 5 and 1. (Object 1 is in a three-object group at the left.)

Figure 5. Candidate Match Selection Example (continued)



f. Candidate matches for F are 11 and 12

Figure 5. Candidate Match Selection Example (concluded)

TABLE 3. EXAMPLE CANDIDATE MATCH SELECTION

Object in Sensed Image	Candidate Matches in Reference Image
A	1, 2
B	3, 5
C	6, 7
D	6, 9
E	5, 1
F	11, 12

In this example, retaining eight distinctive objects in the sensed image and four candidate matches per object would include the correct matches with high probability. For such a candidate match selection there are at most 17,920 four-object matches. The number of objects extracted by the segmentation algorithm was about 40 per image. This implies that before candidate-match selection there were about 2×10^{11} four-object matches. As will be seen in the next section, the branch-and-bound algorithm finds the correct match with about 400 match evaluations when such a candidate match selection is used. On the other hand, it is estimated that a hardware implementation would be able to do 300 match evaluations per frame. Thus, the present candidate selector performance appears very good, even with a conservative evaluation.

The example has raised a few key issues which should be addressed in detail. The first issue is the probability of finding corresponding objects in the two fields of view. In this example, the only distinctive objects that were common to both fields of view were the four targets. On the other hand, the background and foreground clutter has changed so drastically that it is impossible to identify corresponding clutter objects in the two fields of view. This is because the common ground covered by the two fields of view can be extremely small --due to the narrow fields of view, the low angle of elevation, and large aspect angle differences between the two sensors. In Figure 6, the narrow field of view and the low angle of elevation have conspired to make the "footprint" of the sensor field of view on the ground narrow and long. For aspect angle differences as large as 50° as in this example, the common footprint on the ground is confined to only the targets.

A further observation is that because of the low angle of elevation, clutter and targets which possess 3D relief from the ground (like trees, tanks, etc.) tend to be more invariant to aspect angle differences than planar clutter (roads, rivers, temperature gradients in the terrain, etc.). This issue will be studied in detail in the next reporting period to determine the limitations placed by the geometry of the sensors on the presence of corresponding distinctive objects in the two fields of view. To maximize the common footprint, it appears that a wide field of view look would be needed (if only from the acquisition sensor, i. e., the RPV FLIR).

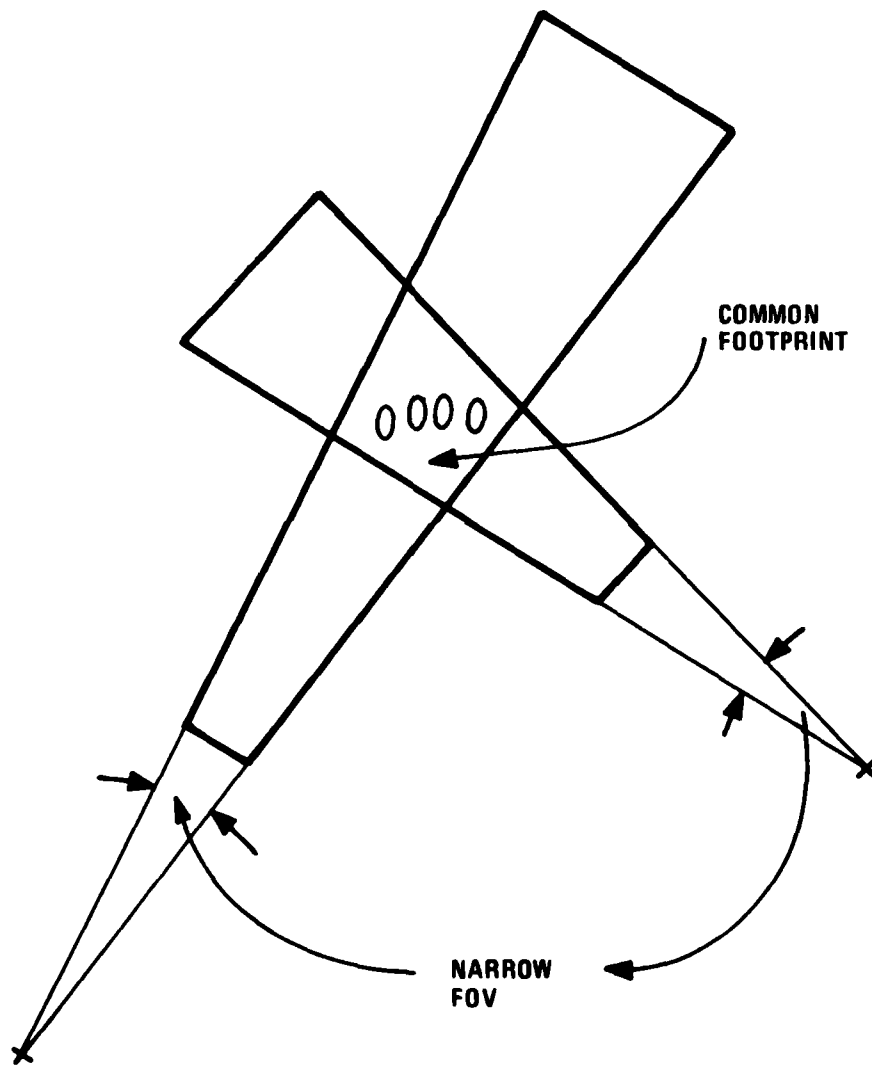


Figure 6. Illustrates the Sensor Geometry for the Example in Figure 3 (Note the small common footprint because of the low angle of elevation, narrow FOVs, and the aspect angle differences.)

The second issue that is illustrated by the example in this section is the need for configuration analysis. As we saw, the candidate match selector does not always rank the correct object match highest in similarity, based on the object-to-object similarity alone. However, the correct match is within the first few most similar objects for every object in the sensed image. This points to the need for the configuration matching. In its absence, it would be difficult to distinguish a specific target (from all other targets and distinctive clutter in the field of view) for target prioritization. This is needed to achieve rapid salvo fire, with successive projectiles being programmed to hit different targets in a target cluster. This example also points out the difficulty of truly autonomous acquisition, especially as it relates to the rejection of false alarms.

SECTION 3

BRANCH-AND-BOUND MATCHING ALGORITHM

The final search for the best multi-object match among the candidate object-to-object matches allowed by the candidate match selector is implemented with a branch-and-bound algorithm. It achieves efficiency by enumerating the set of matches in such a way that when a partial match fails a simple test, the entire subset of matches containing the partial match need not be evaluated. The algorithm is presented in this section and illustrated with examples.

The branch-and-bound algorithm finds the best n-object match that can be formed, using the object-to-object matches allowed by the candidate match selector. It is most naturally presented as an efficient way to search a tree.

A tree enumerating all one-, two-, and three-object matches for the candidate match selection of Table 4 is shown in Figure 7. The nodes of the tree specify object-to-object matches. An m-object match is specified by the nodes visited when m-consecutive branches are traversed, starting from the root.

In the branch-and-bound search for the best match, each path from the root is traced down until it is terminated. A path is terminated in two situations:

1. Terminal level: n branches have been traced, and a full n-object match has been specified.

TABLE 4. CANDIDATE MATCH SELECTION

Object in sensed image	Candidate matches in reference image
I	1, 2
II	3, 4
III	5, 6

2. Suboptimality Test: The algorithm has reached a node which is clearly suboptimal and cannot be a part of the best n -object match.

For example, if the suboptimality test were satisfied at nodes B and C in Figure 7, but not at A and D, a search for the best two-object match would have to trace the branches marked with bold lines.

The process of tracing a path corresponds to building up a partial match, one object-to-object match at a time, until a full n -object match is built or until it is clear that the best n -object match cannot be built by adding more object-to-object matches to a partial ($< n$) object match.

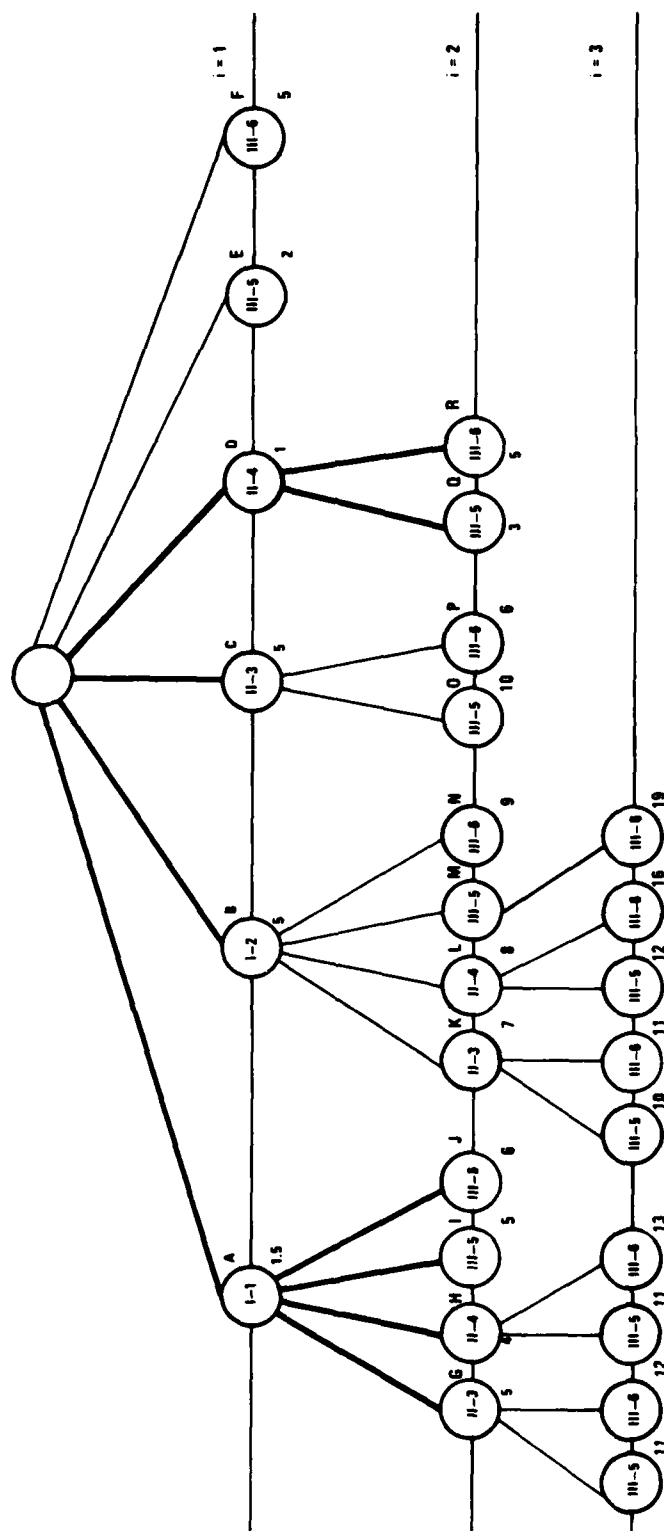


Figure 7. Example of a Branch-and-Bound Search Tree

When the suboptimality condition occurs at a node, the partial match specified at the node is rejected--it cannot be part of the best complete n-object match. Furthermore, when a partial match is rejected, all the complete n-object matches containing it are implicitly rejected. Rather than exploring further implicitly rejected matches subtended by the node, the algorithm backtracks to a node not previously explored. This is how the algorithm achieves efficiency.

For the suboptimality condition to be detectable, it is necessary that the criterion function that evaluates the goodness of a match satisfy monotonicity. Monotonicity is defined as follows: Let m be an object-to-object match allowed by the candidate match selector. Let a match containing i object-to-object matches be written.

$$M^{(i)} = (m_1, m_2, \dots, m_i).$$

Let $f(M^{(i)})$ be the function that evaluates the goodness of matches. Let $f(M_1^{(i)}) < f(M_2^{(i)})$, whenever $M_1^{(i)}$ is better than $M_2^{(i)}$. The function is defined for partial matches, as well as for complete n-object matches. Let us denote

$$M^{(i)} + m = (m_1, m_2, \dots, m_i, m).$$

The monotonicity condition is satisfied if

$$f(M^{(i)} + m) \geq f(M^{(i)}).$$

In other words, adding an object-to-object match to a partial match never decreases a monotonic criterion function.

Using the monotonicity condition to detect suboptimality is simple. A record is kept of the best n -object match found so far in the search. Each time a path is traced to a new node, the criterion function is evaluated for the partial match specified by the new path. If the value obtained is larger than the value of the criterion function for the best n -object match found so far, the monotonicity condition guarantees that all n -object matches specified by tracing further will be worse than the current best match and, hence, can be rejected without explicit evaluation. The best n -object match formed during the search is obviously the best n -object match that can be formed.

The branch-and-bound algorithm is given in steps 1 through 9 below. A flowchart is shown in Figure 8. The notation may be interpreted either in terms of a path being traced in the tree or in terms of the partial matches specified by the path as it is traced. The notation is as follows:

- i -- tree level indicator. The algorithm extends a path from the root by adding a branch at level i . This corresponds to adding one object-to-object match to a partial match containing $(i-1)$ object-to-object matches.
- $M^{(i)}$ -- the partial match specified by a path that has been traced i levels down the tree. $M^{(0)}$ is an empty partial match.
- $\{m_{ij}\}$ -- an ordered list of the candidate object-to-object matches at level i that need to be considered for addition to the current partial match $M^{(i)}$. Equivalently, a list of the branches to be added at the end of the path being traced. m_{ij} is the j^{th} object-to-object match in the list at level i .

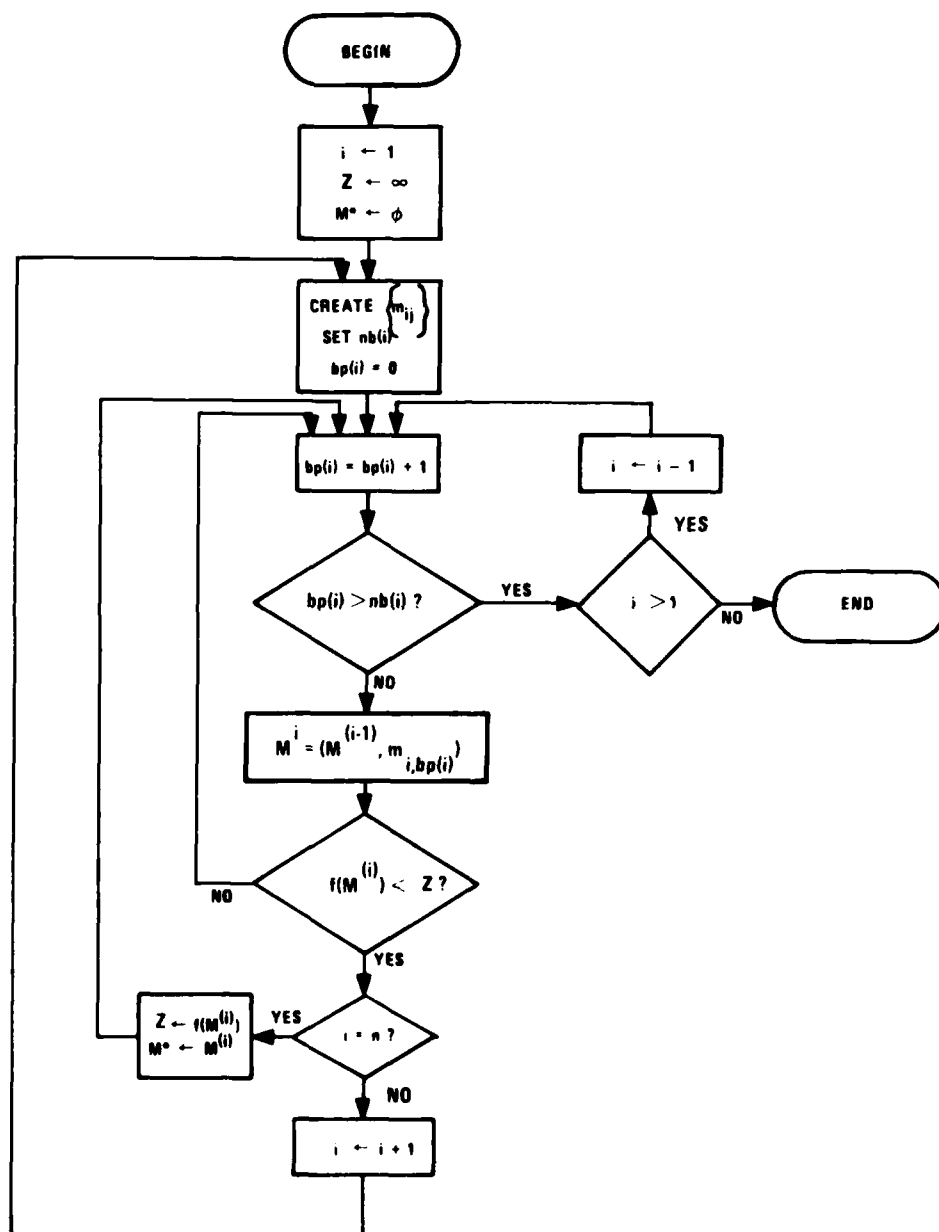


Figure 8. Branch-and-Bound Algorithm Flow Chart

$nb(i)$ -- the number of entries in $\{m_{ij}\}$, possibly zero.

$bp(i)$ -- branch pointer for level i . Only one path is traced at one time. $bp(i)$ points to one entry in $\{m_{ij}\}$ that is in the path currently being traced at the i^{th} level.

M^* -- the best n -object match found so far in the search. Initially $M^* = \phi$, the null match.

Z -- the criterion function value for M^* , initially ∞ . Z is a bound for the value of the criterion function for the best match M^{**} .

The branch-and-bound algorithm:

1. Initialize

$$\begin{aligned} i &= 1 \\ M^* &= \phi \\ Z &= \infty \end{aligned}$$

2. Enumerate possible matches for next level.

Create the ordered list $\{m_{ij}\}$.

Set $nb(i)$.

3. Preset $bp(i) = 0$.

4. Is examination of $\{m_{ij}\}$ complete?

$$bp(i) = bp(i) + 1.$$

If $bp(i) > nb(i)$, go to step 9.

5. Add to current partial match.

$$M^{(i)} = \left(M^{(i-1)}, m_{i, bp(i)} \right).$$

6. Test for situation 2.

If $f(M^{(i)}) > Z$, go to step 4.

7. Extend path if it is neither suboptimal nor a complete n-object match.

If $i=n$, then go to 8, otherwise
set $i=i+1$ and go to 2.

8. Update best match when a full match is formed.

If $f(M^{(i)}) < Z$, set $M^* = M^{(i)}$
and set $Z = f(M^{(i)})$. Go to step 4.

9. When a subtree has been completely searched, go back one level
to look for unsearched subtrees. If this is impossible because
 $i=1$, terminate.

If $i>1$, set $i=i-1$ and go to step 4.

If $i=1$, best match is M^* . Stop.

The following example of a branch-and-bound search will serve to clarify the algorithm. This example is further extended in the discussion on enumeration.

To the right of and below each node in the tree of Figure 7 is the value of a criterion function for the match specified by tracing from the root of the tree to the node. Inspection will show that the function is monotonic; i.e., going one level deeper in the tree never decreases its value.

The branch-and-bound search for the best two-object match proceeds as follows: The first time step 2 of the algorithm occurs, the tree level indicator i is 1 and the six branches from the root of the tree to nodes A, B, C, D, E, and F are enumerated in the first list. Node A is considered

first, and its criterion function is lower than the initial value of Z , which is infinity. The path from the root to node A is not terminated by either of the two situations discussed.

At step 2 with $i=2$, the four branches from node A to nodes G, H, I, and J are enumerated in a second list. The nodes are evaluated in turn. When node G is evaluated, it has $f(M^{(i)}) < Z$, so G is accepted as the best two-object match so far, and the value of Z is updated. H is even better than G; when H is evaluated, it is accepted as the best match so far. I and J are not as good. After J is evaluated, the subtree subtended by A is completely searched. This condition is detected by the test of step 4. The algorithm goes back to examine the next branch in the first list, to node B. At node B suboptimality is detected, and the subtree subtended by B is not searched. The same thing happens at node C. The last branch to be examined in the first list is D. Neither condition terminates the path to D, so a new second-level list of branches from D to Q and R is created. Nodes Q and R are evaluated. Q is a new best match. After R is evaluated, control passes back to the level 1 list. Nodes E and F are terminated because no two-object matches containing the one-object matches of those two nodes remain to be evaluated. After E and F are terminated, the algorithm stops, with the match of node Q detected as the best match.

The two most important issues in the implementation of a good branch-and-bound matching algorithm are the enumeration procedure in step 2 and the criterion function. The criterion function will be discussed in Section 4.

Discussion of the enumeration procedure is presented below. The requirements that the procedure must meet are outlined, and the procedure is presented in detail. With the aid of examples we can show how the procedure meets the requirements and what advantages it has over alternative procedures.

To guarantee that the branch-and-bound algorithm finds the best match, it is essential that the enumeration procedure be exhaustive, that is, that it be able to enumerate all allowed matches. For efficiency the enumerator should be nonrepetitive; it should enumerate no allowed match more than once. As will be seen, it is also important that the enumerator be able to prioritize, to order the lists it generates to maximize the probability that the correct match will be found early in the search. The enumerator meets all of these requirements. In addition, it accomodates many-to-one matches, and it requires only a modest amount of computer memory.

The enumeration procedure has three steps. In the first step an unedited, unprioritized list is created. It contains all matches that need to be considered in the branch-and-bound algorithm. When $i=1$, the list consists of all matches allowed by the candidate match selector. When $i > 1$, the list is copied from the list for the previous level. Copying all of the previous list would guarantee exhaustiveness, but not nonrepetitiveness. The enumeration procedure copies all of the previous list except the following:

1. The object-to-object match in the current partial match.
2. Those object-to-object matches which have already been considered at level $i-1$.

When these exceptions are made, exhaustiveness is preserved and non-repetitiveness is achieved.

In the second step the object-to-object matches in the new list that are not compatible with the current partial match are removed. This editing does not affect exhaustiveness. It is used to remove unwanted many-to-one matches.

In the third step the list is reordered so that the matches most likely to lead to the correct full match are first in the list. This prioritization affects neither exhaustiveness nor nonrepetitiveness. Enumeration is accomplished by executing steps 1 through 3 below at step 2 of the branch-and-bound algorithm. The notation is as follows:

$i, nb(i), bp(i), m_{ij}$ -- as in branch-and-bound algorithm.

$P_s(j, i), P_r(j, i)$ -- ordered pair specifying m_{ij} . $P_s(j, i)$ and $P_r(j, i)$ are pointers to the objects in the sensed and reference images in the j^{th} match m_{ij} in the list for level i .

N_s -- number of distinctive objects in sensed image.

$N_c(k)$ -- number of candidate matches in reference image for k^{th} distinctive object in sensed image.

$C(m, k)$ -- pointer to m^{th} candidate match for k^{th} object in sensed image.

Enumeration procedure:

1. Make an unedited, unprioritized list of object-to-object matches to consider for addition to the current partial match. If $i=1$, obtain a list from the candidate match selector output. If $i > 1$, obtain a list from the list for the previous level.

```

If i=1
  then
    begin
      j=0
      k=0
      while k<Ns
        k=k+1
        m=0
        while m<Nc(k)
          m=m+1
          j=j+1
          Ps(j, i)=k
          Pr(j, i)=C(m, k)
        endwhile
      endwhile
    end
  else
    begin
      j=0
      k=bp(i-1)
      while k<nb(i-1)
        k=k+1
        j=j+1
        Ps(j, i)=Ps(k, i-1)
        Pr(j, i)=Pr(k, i-1)
      endwhile
    end
  end
nb(i)=j

```

2. Remove from the list those object-to-object matches that are incompatible with the current partial match $M^{(i-1)}$. (Edit disallowed many-to-one matches.) Compact list and update $nb(i)$.
3. Prioritize by sorting the list so that the matches that appear most promising are first in the list.

An example will clarify how the enumerator achieves exhaustiveness and nonrepetitiveness. The tree in Figure 7 is generated by the enumerator when no many-to-one matches are allowed for the candidate match selection of Table 4 and no prioritization is attempted. There are $\binom{3}{1}2^1 = 6$ one-object matches, $\binom{3}{2}2^2 = 12$ two-object matches, and $\binom{3}{3}2^3 = 8$ three-object matches. All are enumerated without repetition. Consider what happens if suboptimality does not occur at node C. The list for level 1 is $\{m_{ii}\} = \{I1, I2, II3, II4, III5, III6\}$, and when the new list is enumerated $bp(1)=3$. For $bp(1)=3$, the matches I1, I2, and II3 are not copied to the new list. All the matches containing I1 and I2 have been enumerated in the subtrees of nodes A and B. II2 is being considered in the current partial match. No match needed to form the matches containing the current partial match that have not already been enumerated has been left out of the new list. Before editing, the new list is $\{II4, III5, III6\}$. II4 will be eliminated; together with II3, it would constitute a many-to-one match. The matches remaining after editing, III5 and III6, are both needed to form matches not yet enumerated. By eliminating from the new list all unnecessary matches, but keeping all the necessary ones, exhaustiveness and nonrepetitiveness are attained.

Good prioritization in the enumeration procedure causes the correct match to be found earlier in the branch-and-bound search. This means that small values of the bound Z are found sooner and, thus, more of the tree is pruned. For example, the search tree of Figure 7 could be replaced with the tree of Figure 9, if the first level list were prioritized. The number of two-object matches examined in the new search tree is 4, two fewer than in Figure 7. (Note that all possible matches are enumerated nonrepetitively in the tree of Figure 7. The gain from prioritization is modest in this small example tree. The gain is much greater in a larger tree.

The present algorithm with its enumeration procedure compares favorably with the alternatives in trading computer memory and algorithm complexity for prioritization capability. One alternative is to do prioritization of the objects to be matched with objects in the sensed image without prioritizing objects in the sensed image. This would reduce the required computer memory. However, an efficient coding of the present algorithm requires only about 400 memory words for enumerating all five-object matches for a candidate match selection retaining 10 distinct objects and 5 matches per object. The memory that could be saved would not justify the reduced prioritization capability. The other alternative is to adopt an algorithm that traces multiple paths simultaneously, extending the most promising path first. Such an algorithm would be able to prioritize more and would find the best match with fewer evaluations. It would require more memory and more complexity, and for some problems much more. The present algorithm finds the correct match in 350 to 800 evaluations, while it is estimated that a hardware implementation would be able to do 3000 evaluations per frame. Experience with the present algorithm suggests that some use of multi-path techniques might offer advantages for more difficult

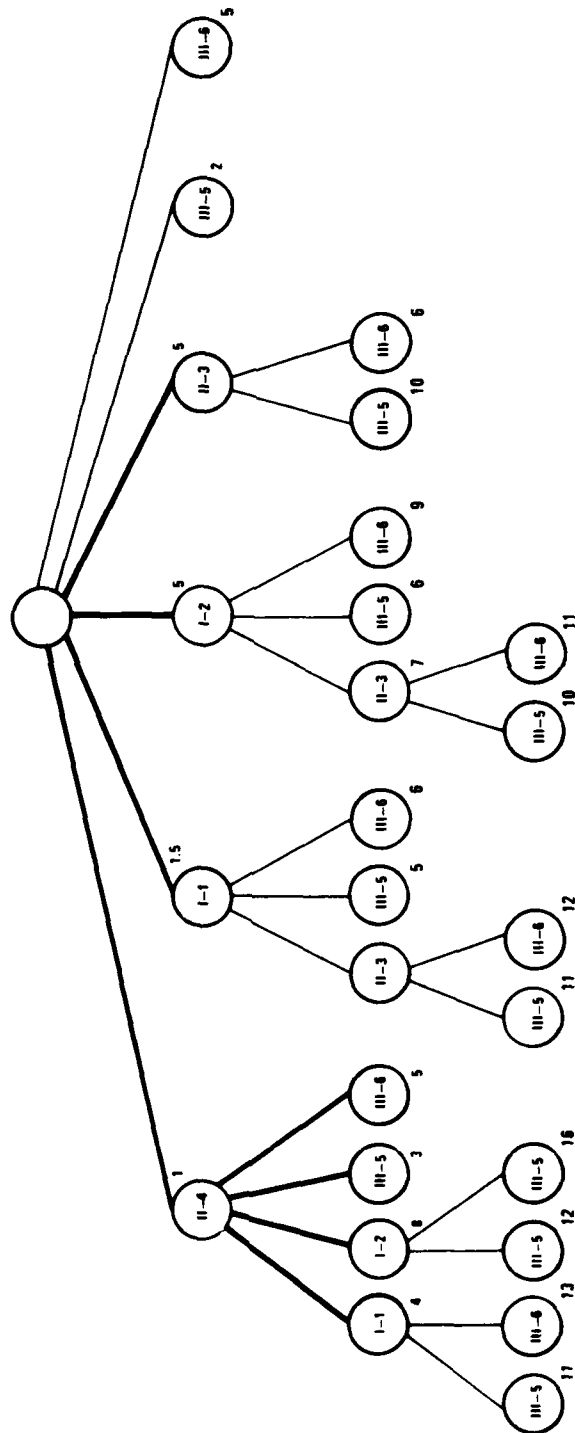


Figure 9. A Prioritized Branch-and-Bound Search Tree

problems that can be attained with acceptable amounts of computer memory. This possibility will be investigated.

The ordering required for prioritization is achieved by evaluating a heuristic function for each match in the new list and sorting the values obtained. The heuristic function is the error in predicting the position in the sensed image, given the estimate of the image-to-image transformation derived from the current partial match. It will be discussed in more detail in the next section.

Many-to-one matches do appear in correct image-to-image matches; it is an advantage to be able to handle them. Images sensed at close range tend to have distinctive objects segmented as multiple objects. At longer ranges, objects such as targets are segmented as single objects. If one image is at close range and the other at long range, there will be valid many-to-one matches. Changes in aspect also give rise to many-to-one matches. Many-to-one matches also occur with line structures, as when a gap in a long line is bridged by edge-feature extraction algorithms for one image but not the other. The enumeration procedure offers flexibility in handling many-to-one matches.

SECTION 4

CRITERION FUNCTIONS

In this subsection the criterion function will be discussed. The criterion function must satisfy two requirements:

1. It must realistically reflect the goodness of a match.
2. It must be monotonic; i. e., adding a match to a partial match must never decrease the criterion function.

Failure to meet either requirement will prevent the branch-and-bound algorithm from finding the best match. The criterion function combines information about the distinctiveness of the objects in a match and the quality of object-to-object matches with an evaluation of the configurational consistency of the match. The effectiveness of the criterion function will be discussed in the subsection on simulation results.

The criterion function f has three components, which may be written as follows:

$$\begin{aligned} f(M^{(i)}) &= w_{\text{dist}} f_{\text{dist}}(M^{(i)}) \\ &+ w_{\text{cm}} f_{\text{cm}}(M^{(i)}) \\ &+ w_{\text{conf}} f_{\text{conf}}(M^{(i)}) \end{aligned}$$

where the w 's are weights, and f_{dist} , f_{cm} , and f_{conf} are functions evaluating the distinctiveness of the objects matched by $M^{(i)}$, the quality of object-to-object matches in $M^{(i)}$, and the likelihood that the configuration of $M^{(i)}$ is correct. f is monotonic if the weights are nonnegative and the contributing functions are also monotonic.

The three functions f_{dist} , f_{cm} , and f_{conf} will be discussed in turn. To simplify discussion, let $M^{(i)} = (m_1, m_2, \dots, m_i)$, and let s_j and r_j be the objects in the sensed and reference images for object-to-object match m_j . Let (x_{s_j}, y_{s_j}) and (x_{r_j}, y_{r_j}) be the coordinates of s_j and r_j in the two images. Let t be an arbitrary object.

The Distinctiveness Component

Given two matches that are otherwise as good, the match using the more distinctive objects is preferred. The distinctiveness component of the criterion function, f_{dist} , is

$$f_{\text{dist}}(M^{(i)}) = \sum_{j=1}^i d(s_j) + d(r_j),$$

where $d(\cdot)$ is a function that measures the distinctiveness of the object it takes as an argument. f_{dist} is monotonic if $d(\cdot)$ is nonnegative, and matches using more distinctive objects are preferred by the branch-and-bound algorithm if $d(\cdot)$ increases as objects become less distinctive. Both requirements are easily met. In the present simulation, $d(t)$ is the contrast rank of the object t .

The Object Similarity Component

Evaluation of the quality of object-to-object matches is incorporated into the criterion function with f_{cm} . The function is of the form

$$f_{cm}(M^{(i)}) = \sum_{j=1}^i q(s_j, r_j).$$

f_{cm} is monotonic if $q(\cdot, \cdot)$ is nonnegative. It favors the better matches if $q(s_j, r_j)$ decreases as the quality of the match between s_j and r_j increases. In the present simulation, $q(s_j, r_j)$ is the rank of r_j in the list of candidate matches for s_j . Thus, $q(r_j, s_j)$ is determined by the candidate match selector operating on its feature base without reference to considerations of configurational consistency.

The Configuration Component

The component of the criterion function that contributes most of the power to discriminate between good and bad matches is the configuration evaluation function f_{conf} . f_{conf} has three components:

$$f_{conf}(M^{(i)}) = f_{ms}(M^{(i)}) + f_{topo}(M^{(i)}) + f_{ap}(M^{(i)}).$$

f_{ms} and f_{topo} evaluate the self-consistency of the configuration; f_{ap} evaluates the consistency of the configuration with the a priori information about the sensor positions and orientations, hereinafter referred to simply as the a priori information. Each of the three components is monotonic; therefore, f_{conf} is monotonic.

f_{ms} is a least-squares residual prediction error that is small when the positions of the objects in the sensed image may be predicted accurately with a linear transformation operating on the positions in the reference image. Let W denote a 2×3 matrix. Then

$$f_{ms}(M^{(i)}) = \min_W \sum_{j=1}^i \left\| \begin{bmatrix} x_{sj} \\ y_{sj} \end{bmatrix} - W \begin{bmatrix} x_{rj} \\ y_{rj} \\ 1 \end{bmatrix} \right\|^2.$$

An example showing the increase in f_{ms} as a match configuration becomes less self-consistent is shown in Figure 10. The optimizing W for $M^{(i)}$ will be denoted W_i^* . It is well-known from linear regression theory that

$$W_i^* = A B^{-1}$$

where

$$A = \sum_{j=1}^i \begin{bmatrix} x_{sj} \\ y_{sj} \end{bmatrix} (x_{rj} \ y_{rj} \ 1)$$

and

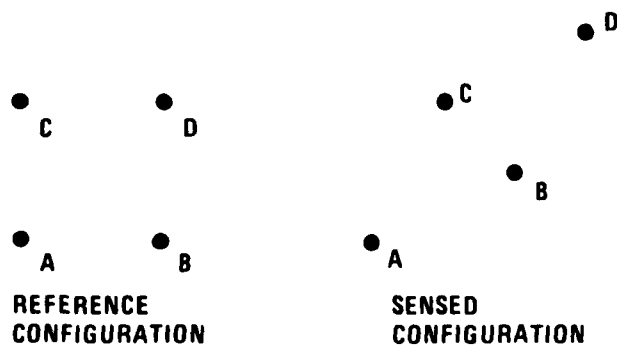
$$B = \sum_{j=1}^i (x_{rj} \ y_{rj} \ 1)^T (x_{rj} \ y_{rj} \ 1).$$

f_{ms} may be seen to be monotonic as follows: Let $M^{(i+1)} = M^{(i)} + m_{i+1}$.

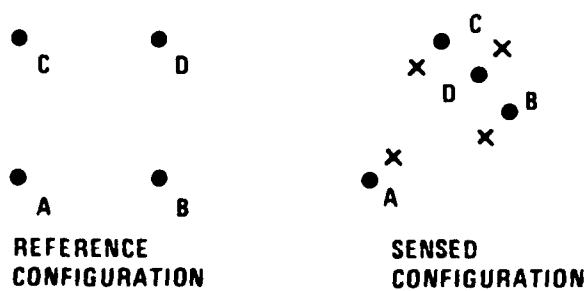
Denote

$$\vec{x}_{sj} = (x_{sj} \ y_{sj})^T$$

$$\vec{x}_{rj} = (x_{rj} \ y_{rj} \ 1)^T.$$



- a. A perfect match. Points in the sensed image are predicted exactly by a linear transformation operating on the matching points in the reference image.



- b. A bad match. The best predictions for a linear transformation are shown with x's.

Figure 10. An Example Showing How the Function f_{ms} Detects Bad Matches

Then

$$f_{ms}(M^{(i+1)}) - f_{ms}(M^{(i)}) = \left\{ \sum_{j=1}^i \left\| \vec{x}_{sj} - W_{i+1}^* \vec{x}_{rj} \right\|^2 - \sum_{j=1}^i \left\| \vec{x}_{sj} - W_i^* \vec{x}_{rj} \right\|^2 \right\} + \left\| \vec{x}_{s, i+1} - W_{i+1}^* \vec{x}_{r, i+1} \right\|^2$$

The term in the braces is nonnegative because W_i^* is optimal for $M^{(i)}$, and the second term is always nonnegative. Therefore,

$$f_{ms}(M^{(i+1)}) \geq f_{ms}(M^{(i)})$$

and f_{ms} is monotonic. In the current simulation, prioritization depends on the optimal transformations W_i^* . A heuristic function $g(\cdot)$ is evaluated for the $m_j = m_{ij}$ in $\{m_{ij}\}$ and the list is sorted. The heuristic function is:

$$g(m_j) = \left\| \vec{x}_{sj} - W_{i-1}^* \vec{x}_{rj} \right\|^2.$$

This heuristic function causes the match most consistent with the configuration of the current match to be considered next. The relationship given above for W_i^* is not valid for $i < 3$ because the matrix B is singular. For $i < 3$ the information in the match configuration is used to modify an estimate of the true matching transformation based on the a priori knowledge. For example, for $i = 1$ the single match is used to determine the translation, and the other parameters in W_i^* are taken as those estimated from the a priori knowledge.

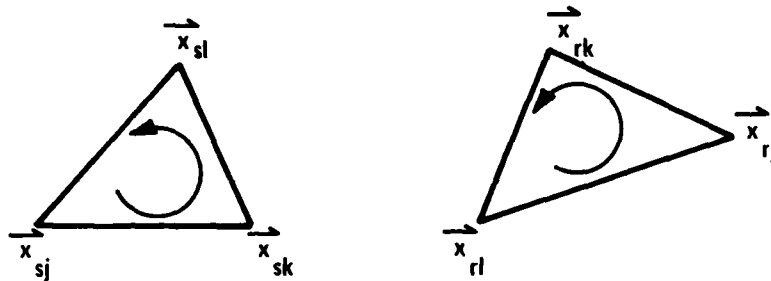
The transformation W^* corresponding to the best match may be used to determine the location in the sensed image of any point in the reference image. This gives the capability to steer toward targets that are not even segmented in the sensed image.

Correct image-to-image matches are topologically consistent regardless of sensor range, aspect, or perspective. The function f_{topo} evaluates how much a configuration must be altered to achieve topological consistency. If the inconsistency is greater than what is attributable to segmentation errors, the configuration may be rejected.

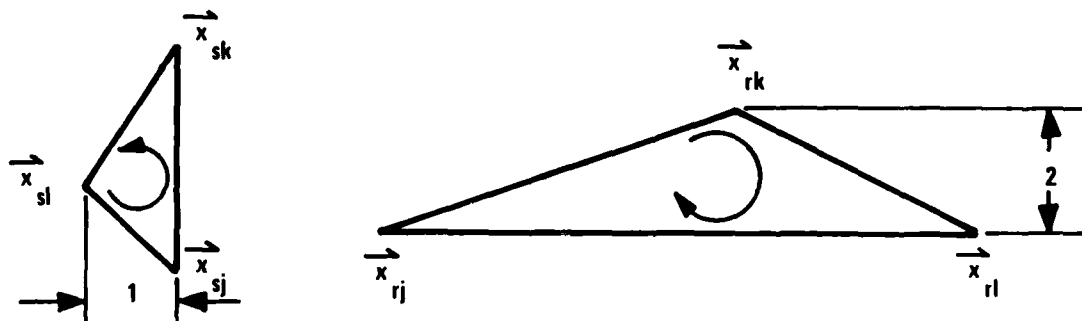
The topological consistency function $f_{\text{topo}}(M^{(i)})$ is computed as follows: Let $\Delta_{jkl}(M^{(i)})$ be a pair of triangles in the sensed and reference images defined by $\vec{x}_{sj}$, $\vec{x}_{sk}$, and $\vec{x}_{sl}$ and by $\vec{x}_{rj}$, $\vec{x}_{rk}$, and $\vec{x}_{rl}$. Two examples are shown in Figure 11. The triangle pair is consistent if a directed path from vertex j to vertex k to vertex l is either clockwise in both images or counterclockwise in both.

Three collinear points are consistent with any triangle. The function $\lambda_{jkl}(M^{(i)})$ is defined to be equal to the square of the minimum distance for which it is possible to move a point in either triangle of $\Delta_{jkl}(M^{(i)})$ and achieve topological consistency. This is the square of the shortest altitude of the two triangles. An example is shown in Figure 11b. f_{topo} combines the values of λ_{jkl} for all triangles that can be formed with $M^{(i)}$ and is defined as

$$f_{\text{topo}}(M^{(i)}) = \begin{cases} \mu & \text{if } \mu \leq \mu_{\text{crit}} \\ \infty & \text{if } \mu > \mu_{\text{crit}} \end{cases}$$



a. A topologically consistent match



b. A topologically inconsistent match
Minimum distance to consistency = 1

Figure 11. Examples of Topologically Consistent and Topologically Inconsistent Matches

where

$$\mu = \sum_{j=1}^i \max_{1 \leq l < k < j} \lambda_{jkl} (M^{(i)}).$$

The threshold M_{crit} is chosen so that an inconsistency greater than what is attributable to segmentation error always leads to rejection of $M^{(i)}$.

f_{topo} is monotonic if λ_{jkl} is nonnegative, which it is.

This method of evaluating topological consistency has the advantage that it will not reject configurations with collinear objects because of small segmentation errors.

The function $f_{ap} (M^{(i)})$ for evaluating the consistency of a match with a priori information is defined as

$$f_{ap} (M^{(i)}) = f_r (M^{(i)}) + f_p (M^{(i)}).$$

f_r is computed as the result of a ratio test:

$$f_r (M^{(i)}) = \begin{cases} 0 & \text{if } r_{crit}^{(1)} \leq r_{jk} \leq r_{crit}^{(2)} \text{ for } 1 \leq k < j \leq i \\ \infty & \text{otherwise,} \end{cases}$$

where

$$r_{jk} = \left\| \frac{\vec{x}_{sj} - \vec{x}_{sk}}{\|\vec{x}_{sj} - \vec{x}_{sk}\|} \right\|^2 / \left\| \frac{\vec{x}_{rj} - \vec{x}_{rk}}{\|\vec{x}_{rj} - \vec{x}_{rk}\|} \right\|^2.$$

f_p is computed as the result of a test of how well the a priori information predicts the differences between points in the sensed image:

$$f_p(M^{(i)}) = \begin{cases} 0 & \text{if } ||(\vec{x}_{sj} - \vec{x}_{sk}) - W(\vec{x}_{rj} - \vec{x}_{rk})||^2 < d_{crit}^2 \\ \text{for } 1 \leq k < j \leq i \\ \text{for some } W \in A \\ \infty & \text{otherwise.} \end{cases}$$

A is a set of matrices consistent with the a priori information.

The thresholds $r_{crit}^{(1)}$, $r_{crit}^{(2)}$, and d_{crit}^2 , and the set A are chosen to make the probability of rejecting an incorrect match high, while keeping the probability of rejecting a correct match low.

SECTION 5

SYSTEM SIMULATION AND RESULTS

In this section the current state of the simulation software is discussed, and results of simulation experiments are presented. The simulation is implemented in three parts. They are as follows:

- PATS segmentation algorithm
- Candidate match selector
- Branch-and-bound pattern matching algorithm

A block diagram is shown in Figure 12. The PATS segmentation algorithm has been discussed in a previous report.¹ The algorithms for the candidate match selector and the branch-and-bound pattern matcher are as discussed in other sections of this report. All the parameters of the two algorithms are inputs to the actual programs. This allows full flexibility in experimenting with the parameters of the algorithms.

The parameters input to the candidate match selector are as follows:

1. N_s , N_r , and $C_{\max}$. These are the number of objects to consider in the two images and the number of matches to retain per object as discussed earlier.

¹ P. M. Narendra and J. J. Grabau, Advanced Pattern-Matching Techniques for Autonomous Acquisition: First Quarterly Progress Report, NV&EOL Contract Number DAAK 70-79-C-0114, Honeywell Systems and Research Center, Minneapolis, Minnesota, Report Number 79SRC104, December 1979.



Figure 12. Block Diagram of Simulation Software

2. Thresholds governing the editing in step 2 of the algorithm.
3. An estimate of the ratio of areas of matching objects in the two images.

The parameters N_s , N_r , and $C_{\max}$ and the thresholds may be set large to simulate degraded candidate match selector performance.

The candidate match selector program has a display capability for interactive experimentation with the algorithm.

The parameters input to the branch-and-bound matching programs are as follows:

1. The weights w_{dist} , w_{cm} , and w_{conf} for the three main components of the criterion function.
2. Thresholds for the functions f_{topo} , f_r , and f_p .
3. An estimate of the matching transformation.
4. A specification of whether or not to prioritize.

By setting weights and thresholds appropriately, it is possible to inhibit the different features of the criterion function and determine their contributions to the power to reject incorrect matches. The effect of errors in estimating the matching transformation from the a priori information may also be determined.

Outputs of the program that are used in algorithm evaluation are as follows:

1. An image display that shows the matches evaluated and the best match in a simple color-coded format.
2. A CRT output of the values of the three components of the criterion function and the thresholding functions for which a threshold was crossed, for use with (1) above.
3. A hard-copy printout that includes the information of (2), plus a trace of the tree enumerated and statistics on the number of match evaluations at each level of the tree.

These outputs give a detailed picture of algorithm performance.

Three examples of simulation results will be presented in this section. The main purpose of the first example is to show the power of the a priori information in rejecting incorrect matches. The second example demonstrates the ability of the algorithm to find the correct match quickly after it has found a correct object-to-object match at level one of the tree. This capability exists because of the full prioritization capability of the enumeration procedure. The third example shows how the power of a test for the consistency of configurations varies with the accuracy of the a priori information.

EXAMPLE 1

The main point of this example is the power of the a priori information for rejecting bad matches. The candidate match selection for this example is the one shown in Section 2 in Figure 6 and Table 3. Algorithm parameters are as shown in Table 5. The full search tree is shown in Figure 13. The correct match was determined with 49 match evaluations. The nodes of the tree were examined in the order indicated by the numbers to the right of and below the nodes. The first 14 match evaluations are shown in Figure 14. The objects involved in a match are marked with the symbols of Table 4. The node numbers of Figure 13 are at the upper left in the photographs of Figure 14. The photographs at the top are for the sensed image; those at the bottom are for the reference image.

The algorithm starts with an incorrect match at node 1. It enumerates two-object matches containing this match at nodes 2 through 7. All of those matches can be rejected easily, using only the tests of consistency with the estimate of the matching transformation derived from the a priori information.

TABLE 5. BRANCH-AND-BOUND ALGORITHM PARAMETERS
FOR EXAMPLE 1

Weights	$W_{\text{dist}} = W_{\text{cm}} = W_{\text{conf}}$
Thresholds	$\mu_{\text{crit}} = 10 \text{ pixels}$ $r_{\text{crit}} = 70\% \text{ of true scale}$ $r_{\text{crit}}^{(2)} = 130\% \text{ of true scale}$ $d_{\text{crit}} = 50 \text{ pixels}$
Estimate of Matching Transformation from a priori information	10' roll error, translation error equal to distance between targets
Prioritization	Inhibited

When all the two-object matches are rejected, the search of the subtree subtended by node 1 is complete, and the algorithm passes to the next match at level one of the tree which is at node 8. This is a correct match. Two-object matches containing this match are enumerated at nodes 9 and 22 through 26. The first considered is correct. Three-object matches containing the correct two-object match specified at node 9 are enumerated at nodes 10 through 13. The incorrect matches of nodes 10 through 12 are easily rejected, using only the tests of consistency with a priori information. The correct three-object match is found at node 13. The four-object matches containing this match are enumerated at nodes 14 through 17. The match at node 14 is the correct four-object match.

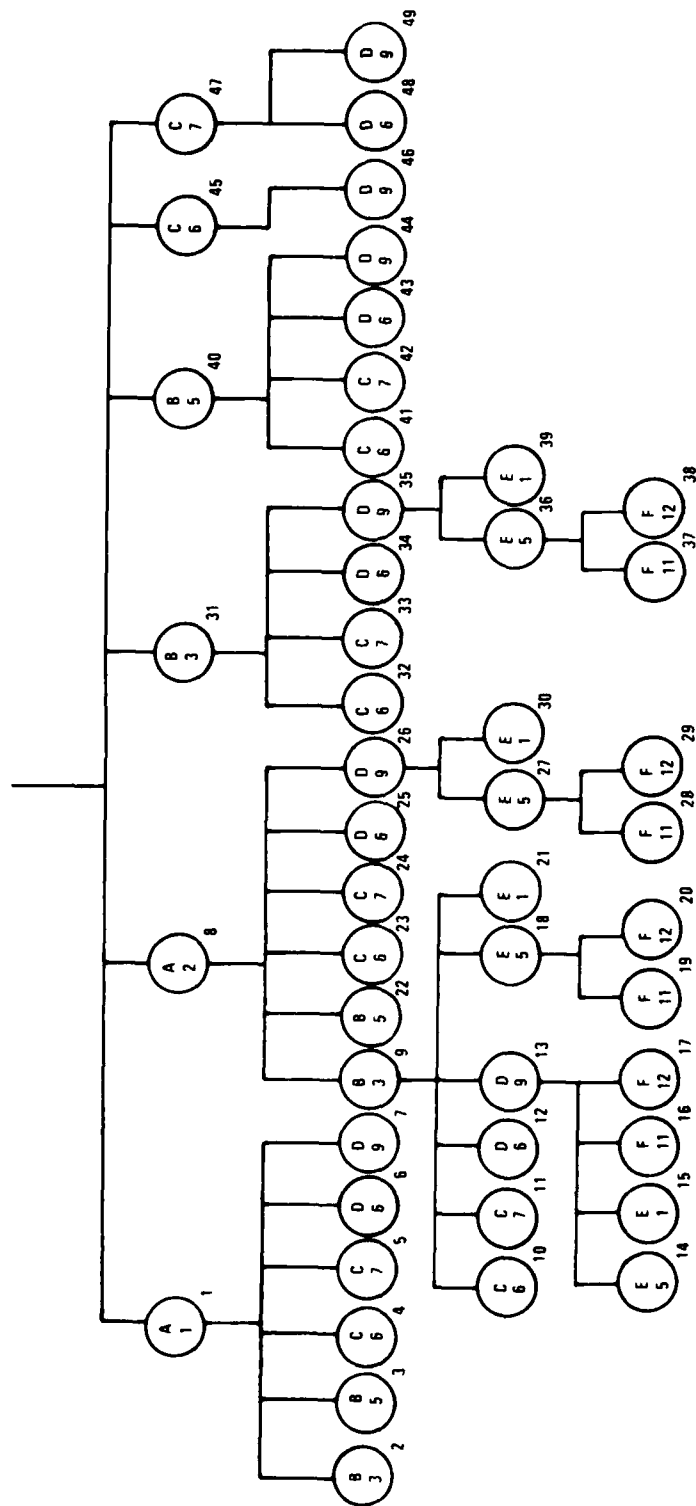
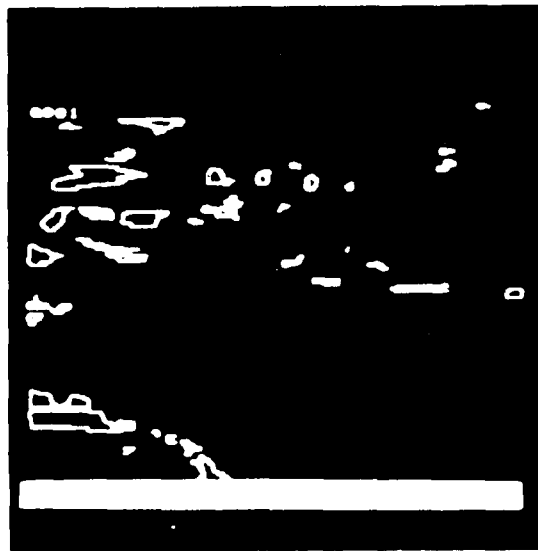
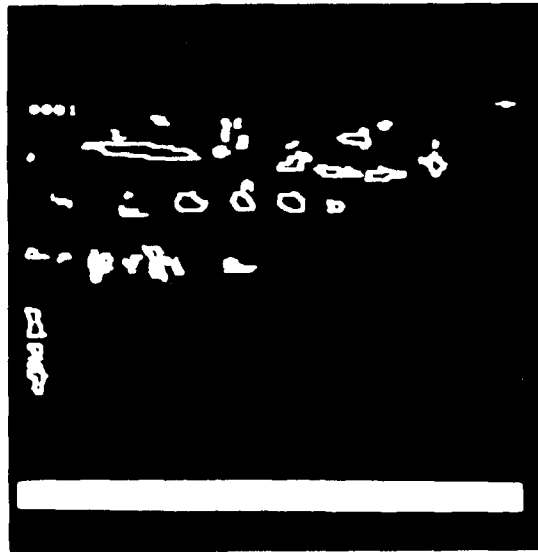
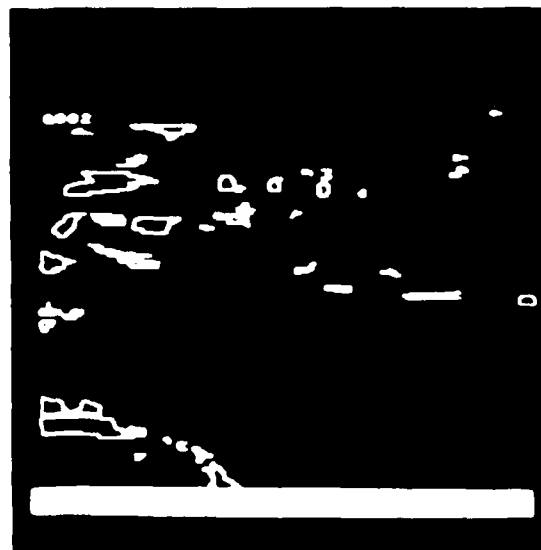
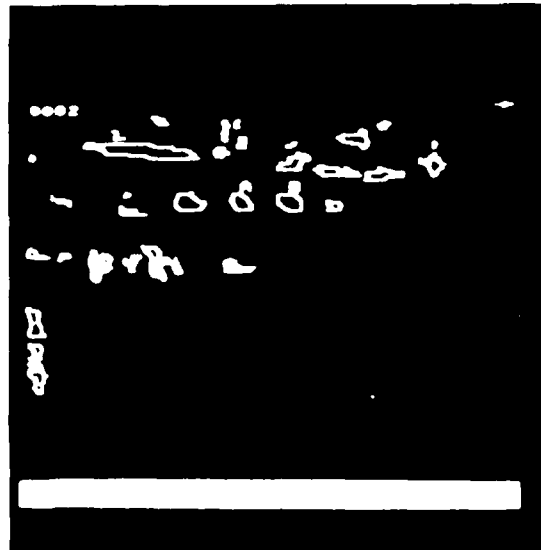


Figure 13. A Complete Search Tree for Example 1



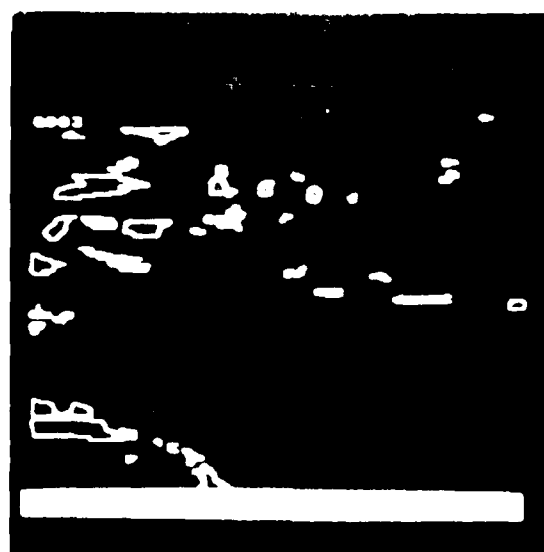
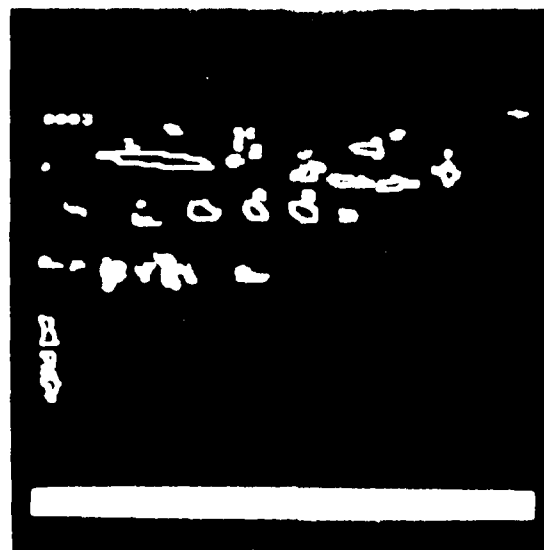
- a. Node 1. A is matched with 1. Object 1 is in the three-object group at the left of the reference image

Figure 14. The First 14 Matches Evaluated in Example 1



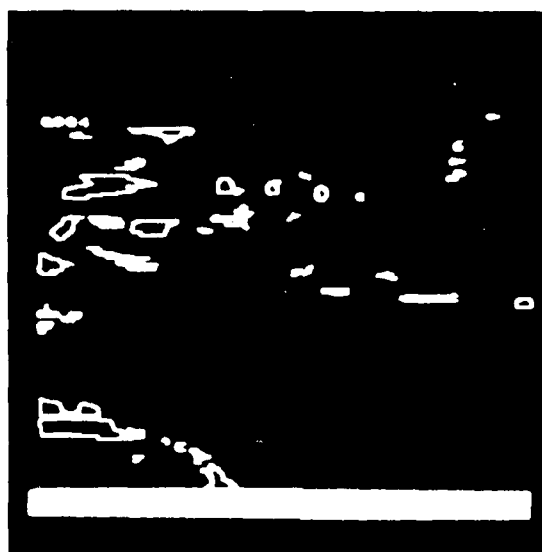
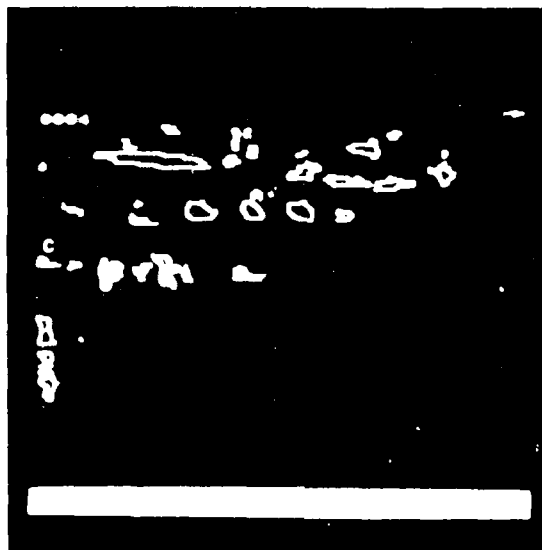
b. Node 2. A is matched with 1 and B with 3

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



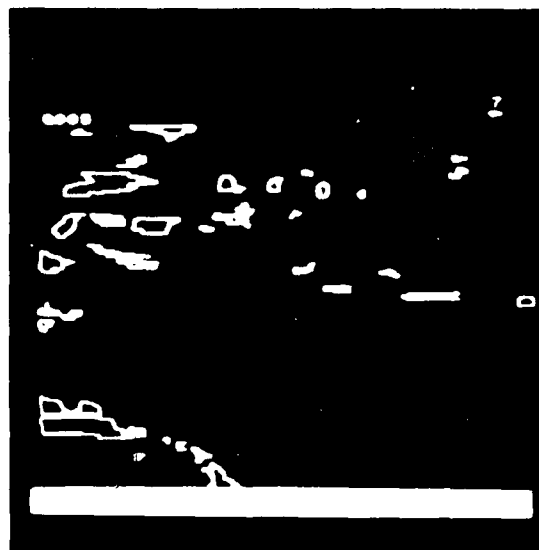
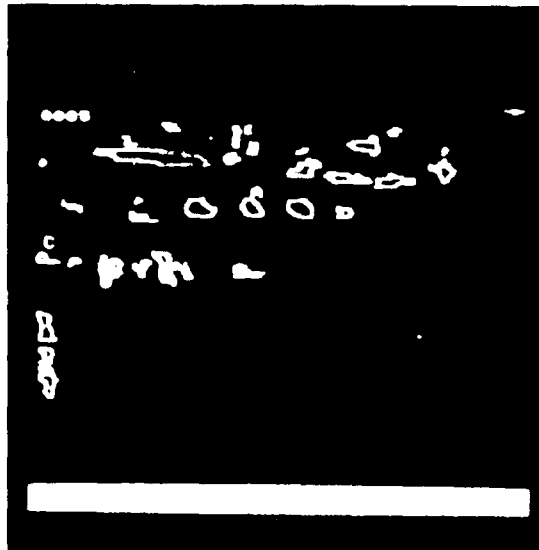
c. Non-adjacent matchsticks (continued)

Figure 14. The First 14 Matchsticks in Example 1 (continued)



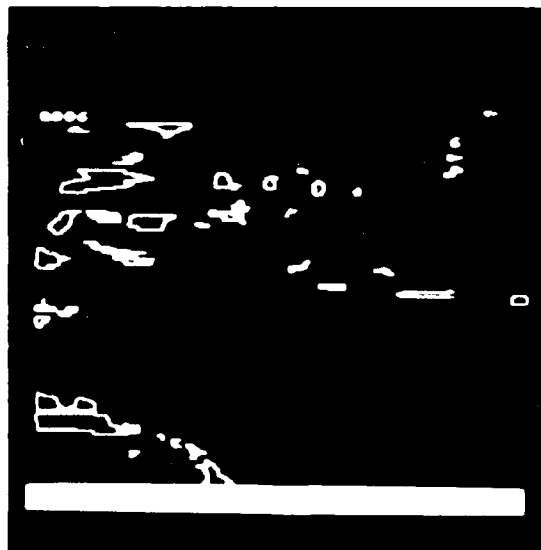
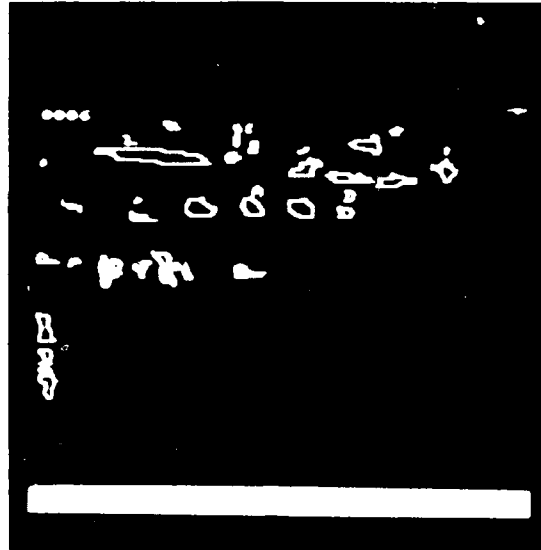
d. Node 4. A distance of 10 from the root.

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



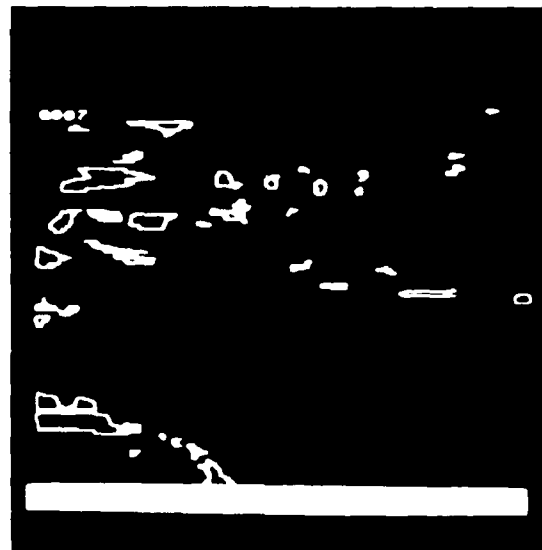
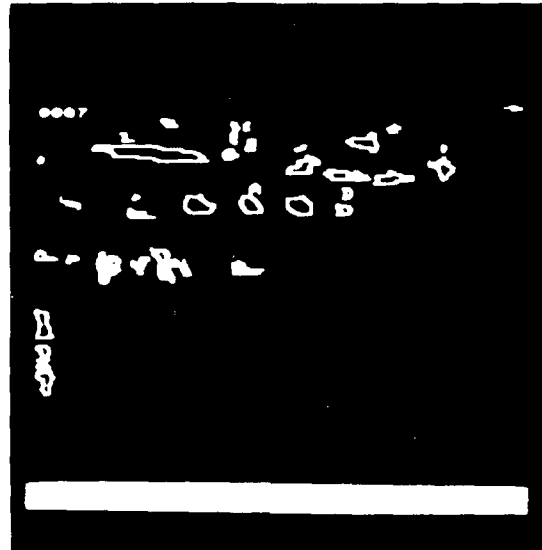
e. Node 5. A is retained with 1 and C with 7

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



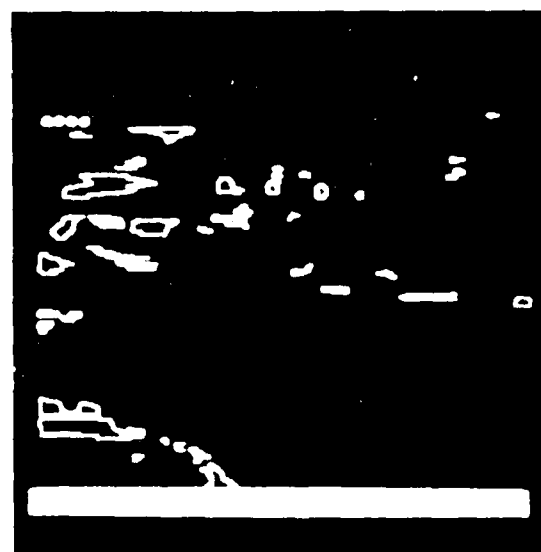
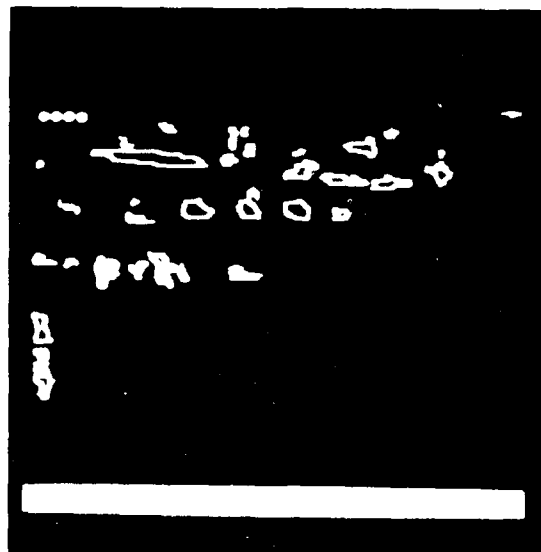
f. Node 6. A is matched with 1 and D with 6

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



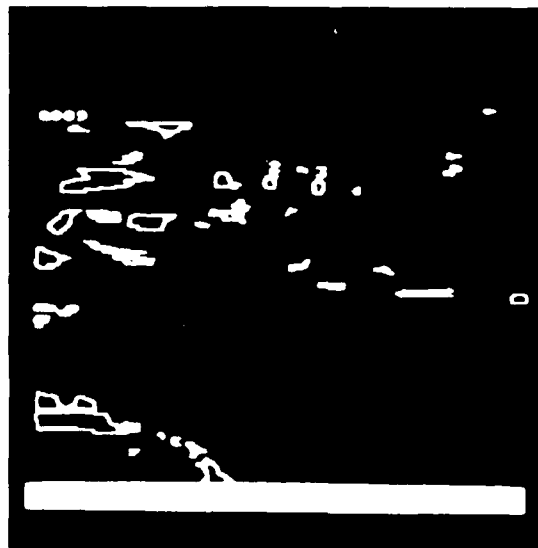
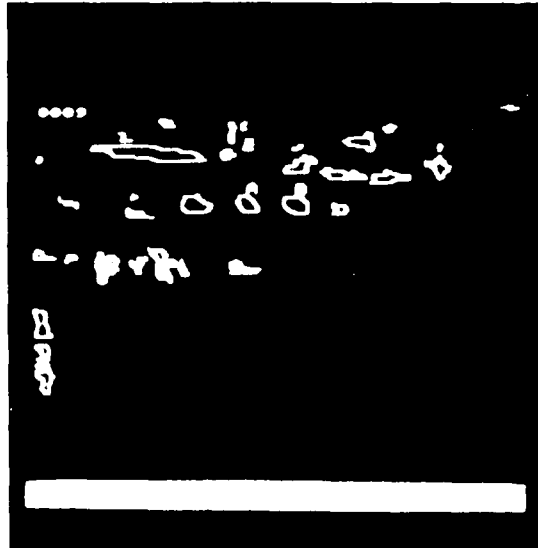
g. Node 7. A is matched with 1 and D with 9

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



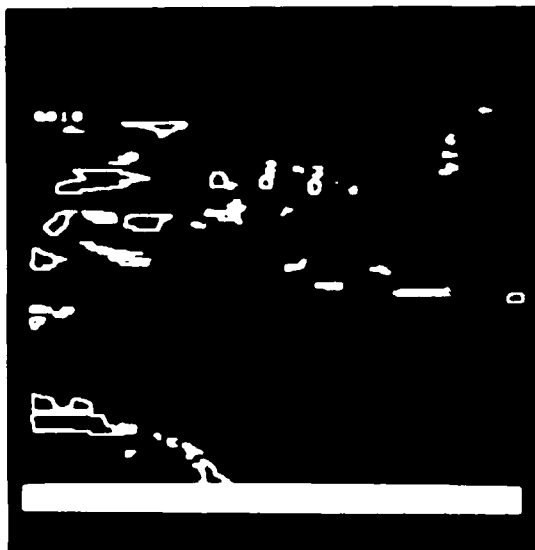
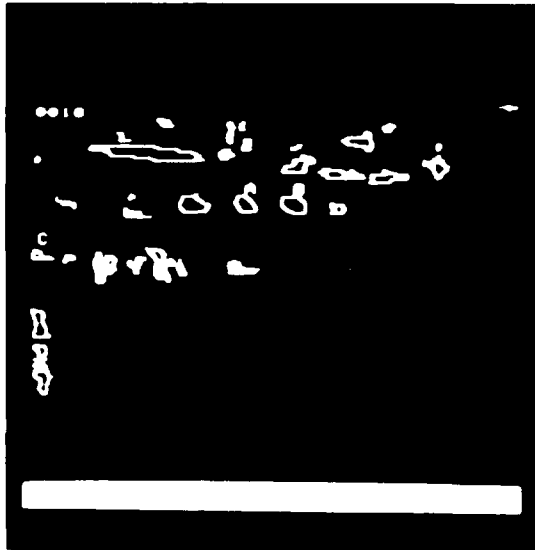
h. None of the above

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



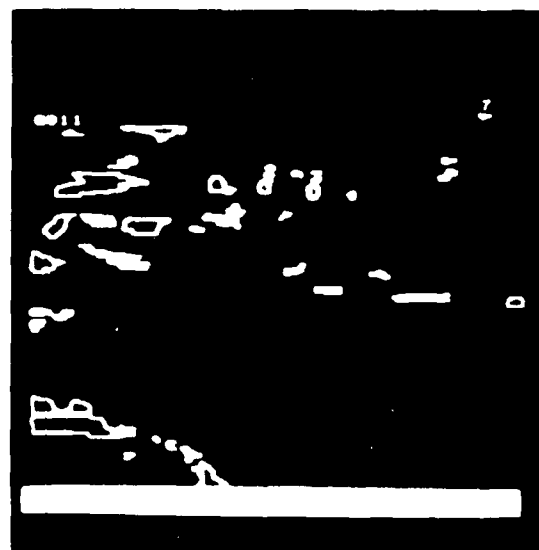
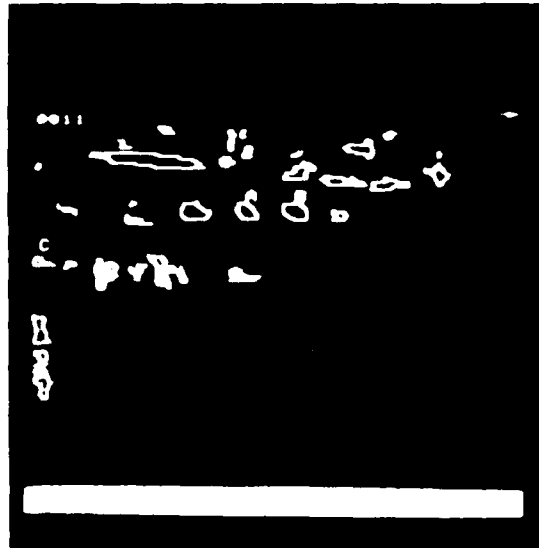
i. Node 9. A is matched with 2 and B with 3

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



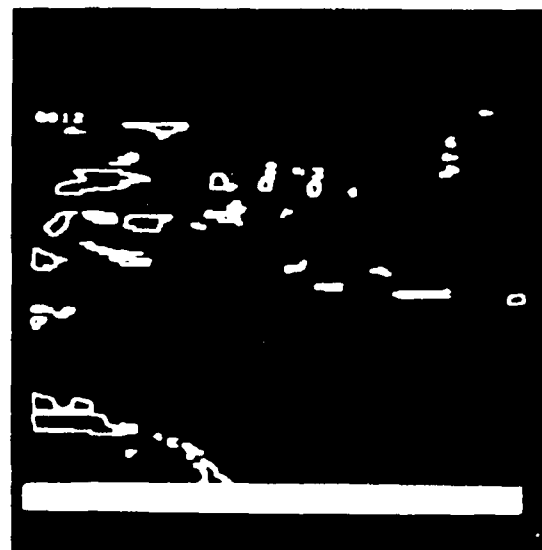
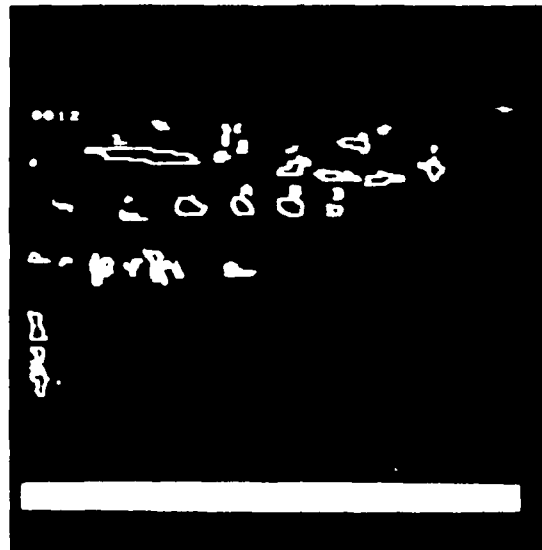
j. Node 10. A is matched with 2, B with 3, and C with 6

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



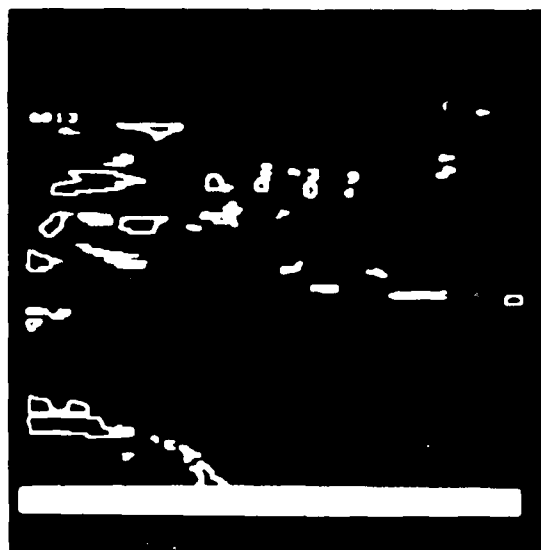
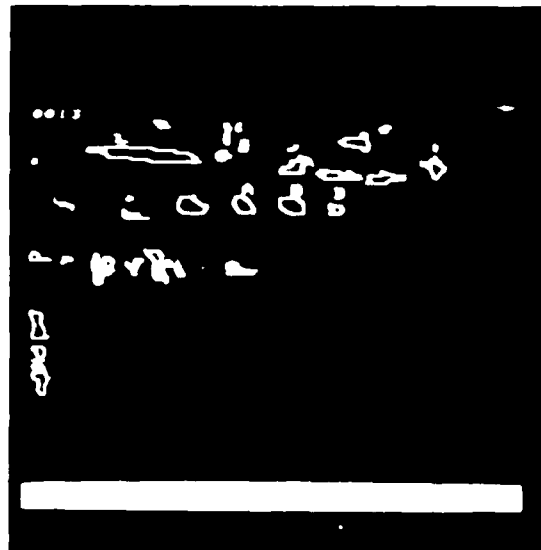
k. Node 11. A 1 in 1000, 1000, 1000, and C with 7

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



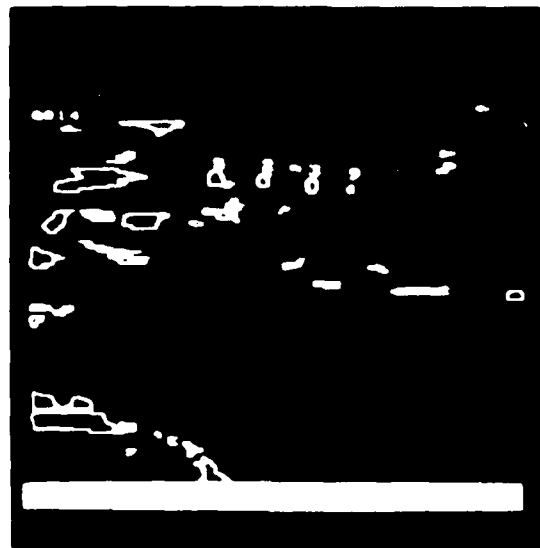
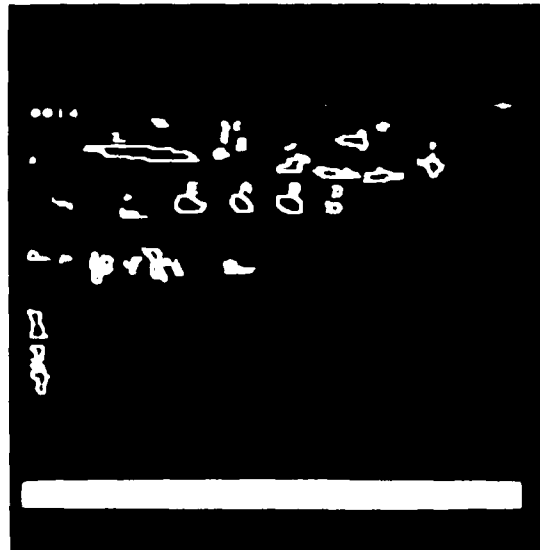
1. Node 12. A.L. matched with 7, 8, 9, and D with 6

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



m. Node 14. A match between P, F, and D with 9

Figure 14. The First 14 Matches Evaluated in Example 1 (continued)



n. Node 14. A 1-m. fence with 4, 10 with 1, 10 with 2, and 11 with 5.
This is the true solution.

Figure 14. The First 14 Matches Evaluated in Example 1 (concluded)

In the other simulation experiments it was found that, as in the example, most of the incorrect matches could be rejected at high levels in the tree by checking for consistency with the a priori knowledge. Munition roll will probably be the most important perturbation in the a priori information. Simulated errors in roll measurement of 10° had no effect on the matching process for this example.

EXAMPLE 2

This example shows that the prioritization capability makes it possible for the algorithm to find the correct match quickly. The candidate match selection for this example is shown in Table 6. The parameters of the candidate match selection algorithm were set to simulate a lower selection performance than that of example 1. The branch-and-bound parameters were as in example 1, except prioritization was enabled. The first four matches enumerated are shown in Figure 15 in the same format as for example 1. The correct four-object match was found in the minimum time--four match evaluations.

EXAMPLE 3

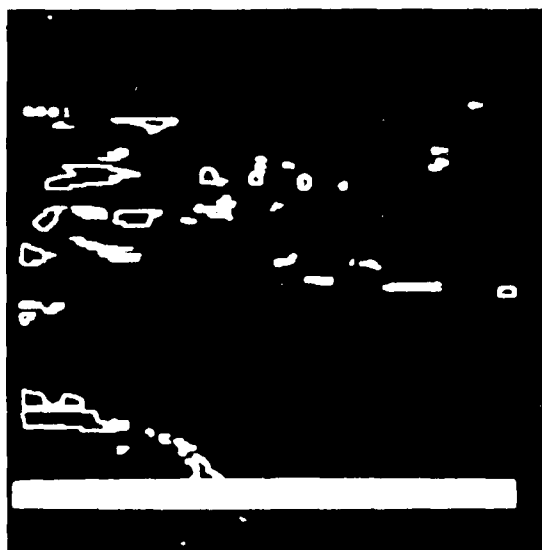
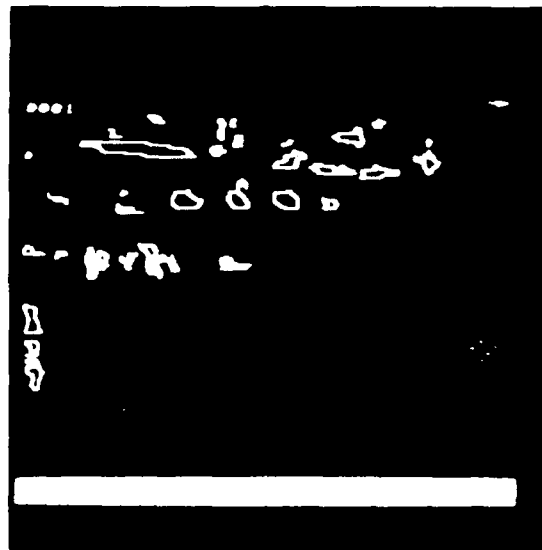
In this example data is presented to show how the power of one of the two tests of consistency with the a priori knowledge--the prediction quality test f_p --depends on the threshold d_{crit} . The candidate match selection was as in example 2. The branch-and-bound algorithm parameters were as in example 1, except d_{crit} was varied. The number of match evaluations for the different levels of the tree are shown for three different values of d_{crit} in Table 7. The test f_p operates at all levels of the tree except the first. It has no effect on the number of matches evaluated at levels 1 and 2 of the tree. Table 7 shows that there is a large reduction in the number of

TABLE 6. CANDIDATE MATCH SELECTION FOR EXAMPLE 2

Object in sensed image	Candidate matches in reference image
A	1, 2, 3, 4
B	5, 3, 2, 1
C	6, 7, 8, 9
D	6, 9
E	5, 1, 10, 3, 2
F	11, 12, 3, 13, 2
G	1, 2, 3, 4
H	3, 11, 2, 13

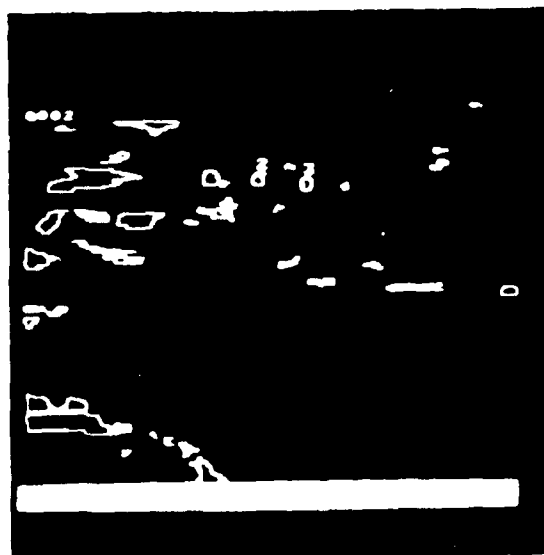
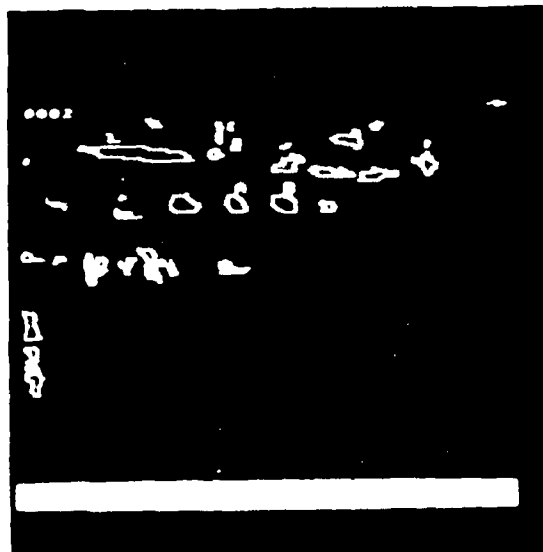
matches evaluated at levels 3 and 4 of the tree when the test is switched from $d_{crit} = \infty$ (disabled) to $d_{crit} = 100$ pixels.

The reduction occurred even though many of the incorrect matches were rejected by the other tests, and even with a value of d_{crit} as large as 100 pixels. The conclusion is that f_p is powerful, even with a high d_{crit} . This means that extreme high quality a priori information is not essential for f_p to be powerful.



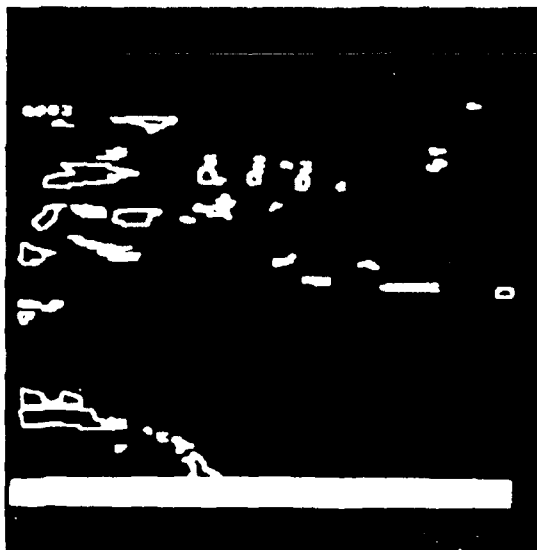
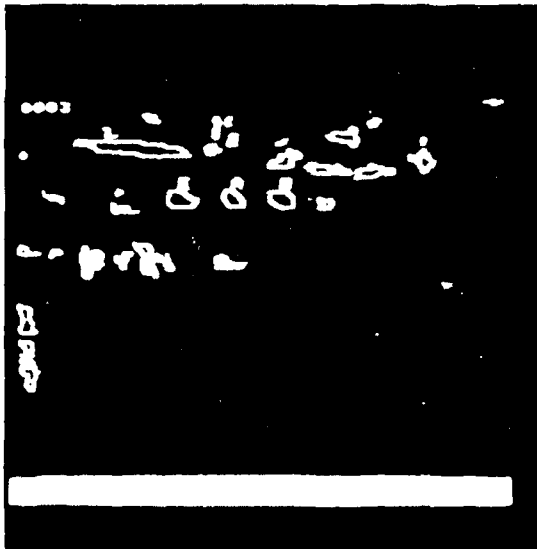
a. First match. (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21) (22) (23) (24) (25) (26) (27) (28) (29) (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) (49) (50) (51) (52) (53) (54) (55) (56) (57) (58) (59) (60) (61) (62) (63) (64) (65) (66) (67) (68) (69) (70) (71) (72) (73) (74) (75) (76) (77) (78) (79) (80) (81) (82) (83) (84) (85) (86) (87) (88) (89) (90) (91) (92) (93) (94) (95) (96) (97) (98) (99) (100)

Figure 15. The First Four Matches Evaluated in Example 2



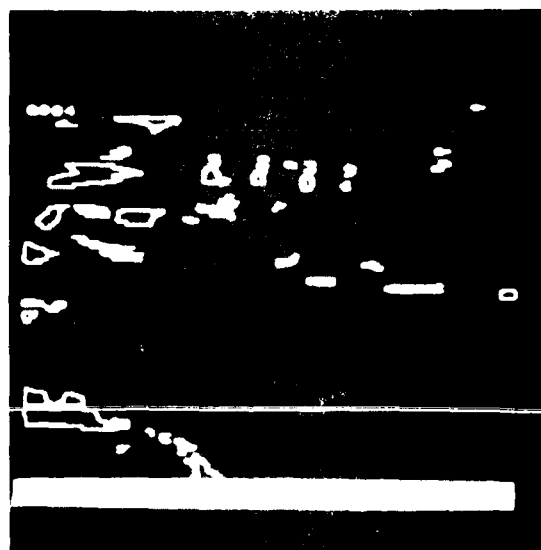
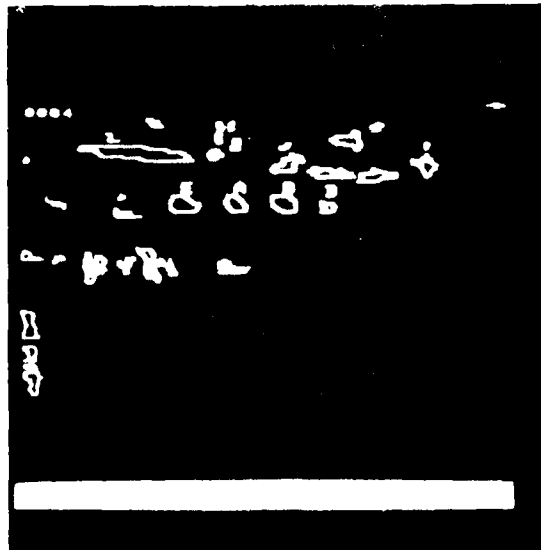
b. Second match. A is matched with 1 and B with 3

Figure 15. The First Four Matches Evaluated in Example 2 (continued)



c. Third match. A is matched with B, C, D, E, and F with 5

Figure 15. The First Four Matches Evaluated in Example 2 (continued)



d. Fourth match. A is matched with 2, B with 3, E with 5, and D with 9
This is the true match

Figure 15. The First Four Matches Evaluated in Example 2 (concluded)

TABLE 7. DATA FOR EXAMPLE 3

Level	$d_{\text{crit}} = \infty$	$d_{\text{crit}} = 100 \text{ pixels}$	$d_{\text{crit}} = 50 \text{ pixels}$
1	19	19	19
2	219	219	219
3	375	131	77
4	<u>158</u>	<u>37</u>	<u>37</u>
	771	406	342

SECTION 6

ANALYSIS OF A PRIORI INFORMATION

It has been found experimentally that the a priori information is powerful in rejecting incorrect matches. In this section a simple theoretical analysis of the power of the ratio test for two-object matches is presented. Then other work initiated toward full exploitation of the a priori information is described.

The ratio test has the advantage that it may be used at a high level in the tree--the second level. The least-squares consistency test, by a comparison, has no discriminating power until the search has reached the fourth level of the tree.

The analysis of the power of the ratio test is as follows: Assume that the coordinates of the objects are (x_{s1}, y_{s1}) and (x_{s2}, y_{s2}) in the sensed image and (x_{r1}, y_{r1}) and (x_{r2}, y_{r2}) in the reference image. Assume that all the x 's and y 's are statistically independent zero-mean Gaussian random variables normalized to have variance $\frac{1}{2}$. This assumption is the key to the simplicity of the analysis. The assumption of statistical independence between images corresponds to the situation that the match is incorrect. Define

$$Z_s = (x_{s1} - x_{s2})^2 + (y_{s1} - y_{s2})^2$$

$$Z_r = (x_{r1} - x_{r2})^2 + (y_{r1} - y_{r2})^2$$

Z_s and Z_r are independent chi-square variables with two degrees of freedom.
The ratio test is of form

$$\text{Accept match if } r_1 Z_r \leq Z_s \leq r_2 Z_r.$$

The probability of accepting is

$$\begin{aligned} \text{Pr}(\text{accept}) &= \int_0^\infty \int_{r_1 s}^{r_2 s} e^{-s} e^{-t} dt ds \\ &= \frac{(r_2 - r_1)}{(r_1 + 1)(r_2 + 1)} \end{aligned}$$

The smaller the acceptance range, the lower the probability of accepting an incorrect match. The narrowness of the range that may be specified depends on the quality of the a priori information.

A conservative analysis shows that for RPV looks and munition approaches with a depression angle of 45° , one would be able to do at least as well as

$$r_1 = 0.5 \left(\frac{\text{FOV}_r}{\text{FOV}_s} \right)^2$$

$$r_2 = 2.0 \left(\frac{\text{FOV}_r}{\text{FOV}_s} \right)^2$$

where FOV_s and FOV_r are the fields of view for the sensed and reference image. For

$$\text{FOV}_s = 3^\circ$$

$$\text{FOV}_r = 15^\circ$$

this gives

$$P_r(\text{accept}) = 5.4\%$$

The ratio test will be even more powerful for three-object matches. As a comparison, tests of topological consistency cannot reject two-object matches, and symmetry arguments show that they reject only approximately one half of all incorrect three-object matches.

It is clear that there are more powerful ways to exploit the a priori information than the simple ratio test. For example, a modification that detected 180° reversals would do twice as well. To find the best way to exploit the a priori information, a powerful program is being written to analyze how errors in estimation of the sensor orientation and positions affect the matching transformation. This program is near completion. Its important features include the following:

- It allows parameters to be specified and analyzed in their most natural coordinate system. For example, the position of an RPV with respect to the GCS and the position of a projectile with respect to its cannon may be most meaningfully analyzed in two different coordinate systems.
- It supports analysis for different matching techniques; for example, ground-plane-to-ground-plane or sensor-plane-to-sensor-plane.
- Any linear combination of 16 true parameters or 16 estimated parameters relevant to the problem may be specified as a perturbation to be analyzed.
- The matching transformation is calculated in a standard matrix form and also in a unique skew form that isolates the effect of errors in the munition roll, one of the most variable parameters.

The skew form of a transformation matrix W is specified by the six parameters θ_1 , e_1 , e_2 , θ_2 , w_{13} , and w_{23} that satisfy the equations below. In sensor-plane-to-sensor-plane matching as in the current simulation, the roll of the munition is merely a variation in the single parameter θ_1 .

The transformation $W = [A \ b]$,

$$\text{where } A = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} e_1 & 0 \\ 0 & e_2 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix}$$

and

$$b = \begin{bmatrix} w_{13} \\ w_{23} \end{bmatrix}.$$

A simple noniterative algorithm for calculating the skew form of a matrix has been implemented for use in the analysis program.

SECTION 7

MINIMAL SPANNING TREES

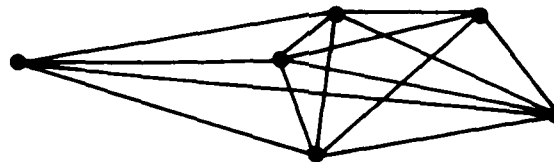
As we saw in the previous sections, the branch-and-bound algorithm considers sequentially object matches to find an optimal match subset. One way to speed up the convergence to the optimal configuration match would be to match distinctive clusters of objects rather than individual objects. The branch-and-bound algorithm can then match precisely the configurations of individual objects within the clusters. In fact, this idea corresponds to gestalt cognition phenomena encountered in human perception. A further advantage of matching clusters is that extended objects like roads, rivers, etc., are often apt to be segmented into multiple components. Matching clusters of these components between the two fields of view is, therefore, likely to be more robust.

The basic idea is to initially match distinctive clusters in the two fields of view. Several clustering algorithms exist for partitioning data points in a multi-dimensional space. However, a data structure known as minimal spanning trees offers the highest promise for extracting distinctive clusters in this application. The following is a brief discussion of minimal spanning trees and the results of initial experiments we have performed on the test images described in the previous sections.

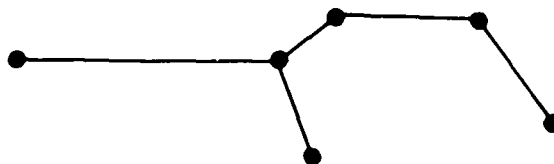
The coordinates of objects extracted from an image form a set of points. A complete graph for such a set of points has an edge linking each pair

of points as shown in Figure 16a. A minimal spanning tree consists of the subset of edges in the complete graph that minimizes the total length of its edges and still provides a path between any pair of points. The minimal spanning tree for the graph of Figure 16a is shown in Figure 16b.

Traditionally, minimal spanning trees have been used in cluster analysis as follows: A minimal spanning tree is constructed for the points specified by the data to be clustered. Then the longest edges in the minimal spanning tree are deleted. If the data really clusters, this process will break the minimal spanning tree into groups of connected points corresponding to the clusters, as shown in Figure 17.



a. Complete graph



b. Minimal spanning tree

Figure 16. A Minimal Spanning Tree for a Graph

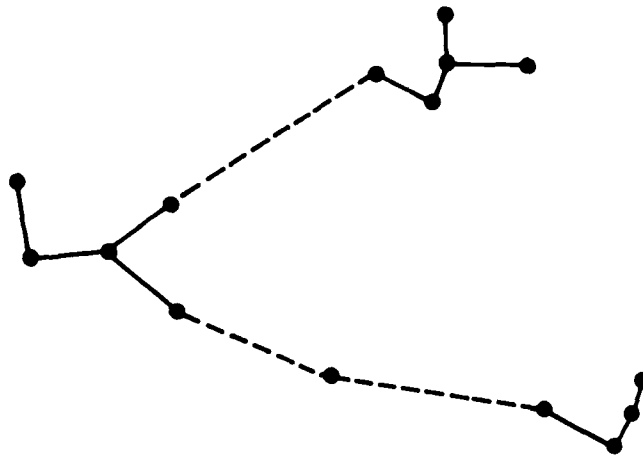
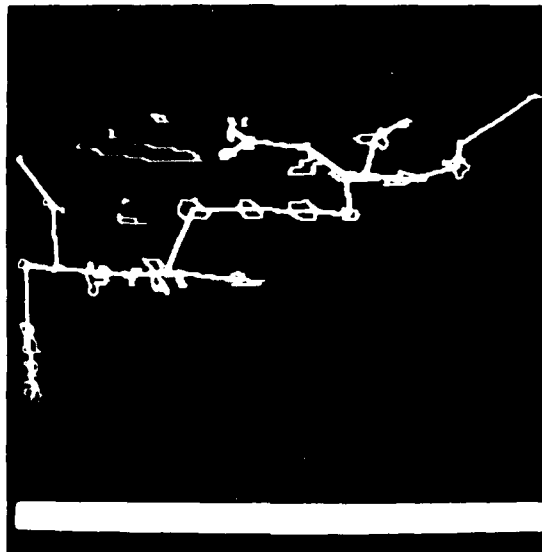


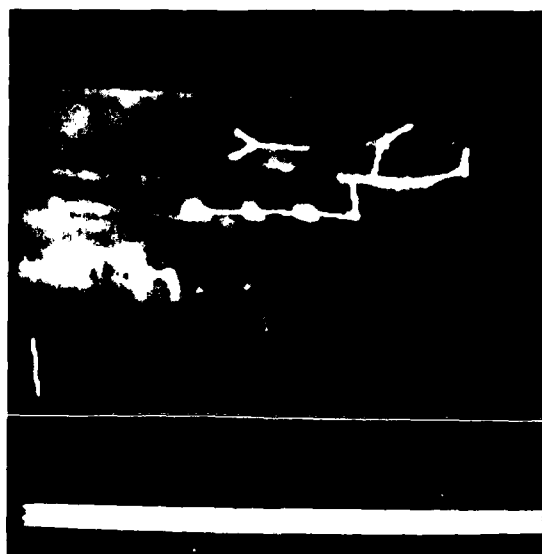
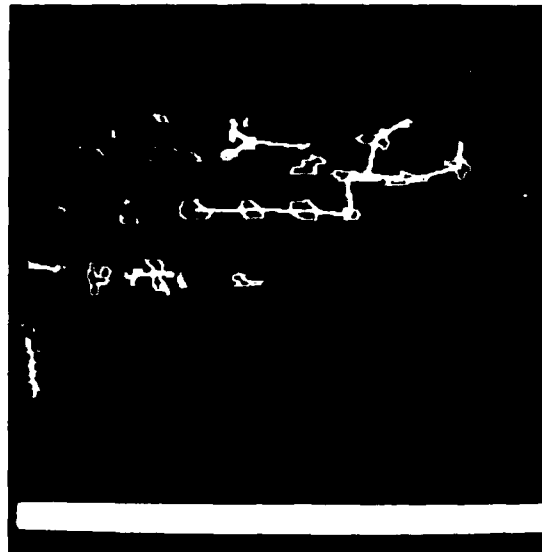
Figure 17. A Minimal Spanning Mat Clusters When it is Broken

This procedure could be used to group the distinctive objects in image pairs into clusters. The minimal spanning trees would be constructed for the distinctive objects in the two images and then broken to obtain clusters. Software was written to test this concept on real images. Examples of results for the images of Figure 3 are shown in Figure 18. The distinctive objects are indeed clustered into groups. Furthermore, for this image pair it would be easy to determine that the clusters containing the very distinctive targets should be matched.



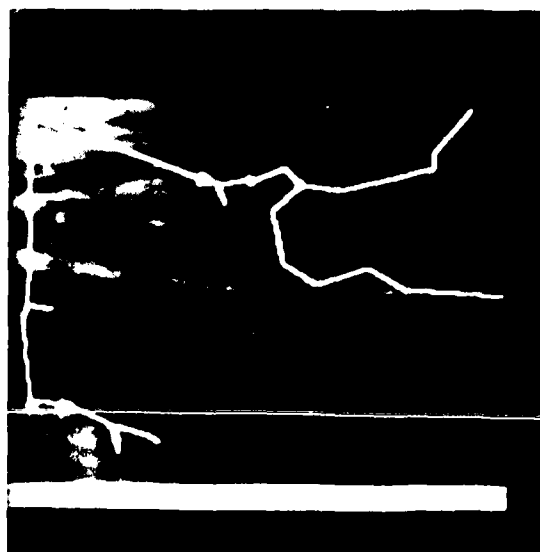
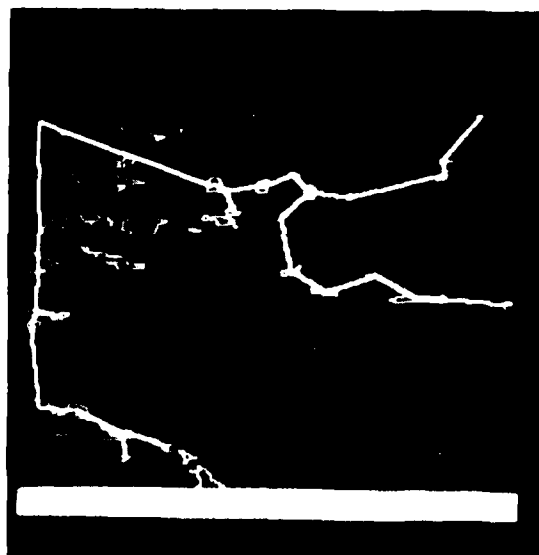
a. Minimal spanning tree for T_1 (the minimum edge set in E used in T_1)

Figure 18. Minimal Spanning Trees in Images of Figure 3



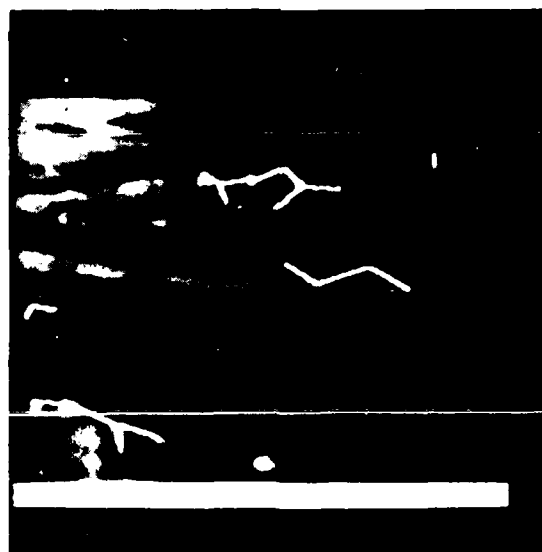
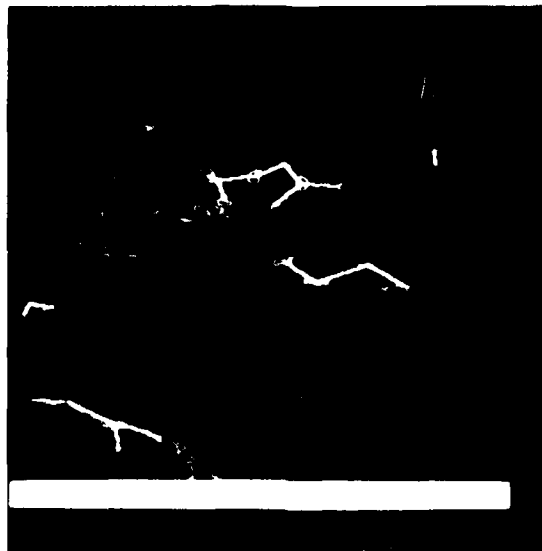
b. Broken minimal spanning tree for G₀

Figure 18. Minimal Spanning Trees in Images of Figure 3 (continued)



c. Minimal spanning tree for "Turtle" (cont.) (object in reference in (a))

Figure 18. Minimal Spanning Trees in Images of Figure 3 (continued)



d. Broken minimal spanning tree for (c)

Figure 18. Minimal Spanning Trees in Images of Figure 3 (concluded)

SECTION 8

PATTERN-MATCHING DATA BASE

As we have seen in the previous sections, the pattern-matching algorithms must successfully match images from a variety of sensor geometries and sensor types. For the simulation task, ideally, we need images of the same scene from FLIR and TV sensors, representing perspective changes from 0 to 180 deg, aspect changes of 1 to 90 deg, and a variety of illumination conditions. This section reports the status of the continuing pattern-matching data base acquisition task.

In this reporting period, we have expanded significantly the pattern-matching data base by digitizing 36 frames from the PATS training video tapes generated at AP Hill by NV&EOL. The frames are from three runs (at altitudes of 500 ft and 750 ft) at approach angles of -35° , 0° , and $+50^\circ$ over four stationary targets in a formation. The digitized frames represent ranges from 8 km to overflight. Samples from these frames are shown in Figure 19. Note that because of overcast conditions, ground clutter is not conspicuous, and the targets are in fact the most distinctive objects in the FOV.

A promising new source of pattern-matching imagery is from the March 1980 PATS flight tests at AP Hill. These tapes contain both wide and narrow fields of new imagery and also represent a wider range of background clutter conditions because of changing cloud cover and other temporal variations. These tapes are now being reviewed to select references which may provide meaningful characterization of the pattern-matching algorithm.

AD-A087 684

HONEYWELL SYSTEMS AND RESEARCH CENTER MINNEAPOLIS MN

F/O 17/7

ADVANCED PATTERN-MATCHING TECHNIQUES FOR AUTONOMOUS ACQUISITION-ETC(U)

MAR 80 P M NARENDRA J J GRABAU

DAAK70-79-C-0114

80SRC25

NL

UNCLASSIFIED

2 14 2

ALL INFORMATION CONTAINED HEREIN IS UNCLASSIFIED



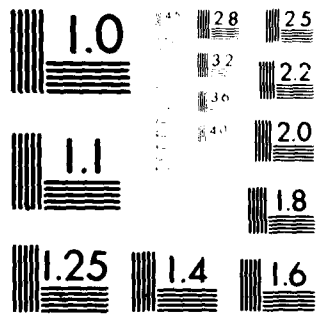
END

DATE

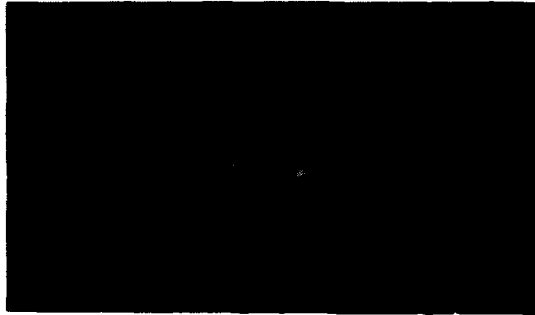
FILED

9-80

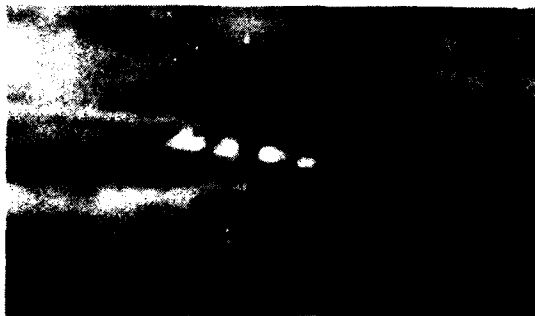
DTIC



MICROCOPY RESOLUTION TEST CHART
 NATIONAL BUREAU OF STANDARDS-1963-A

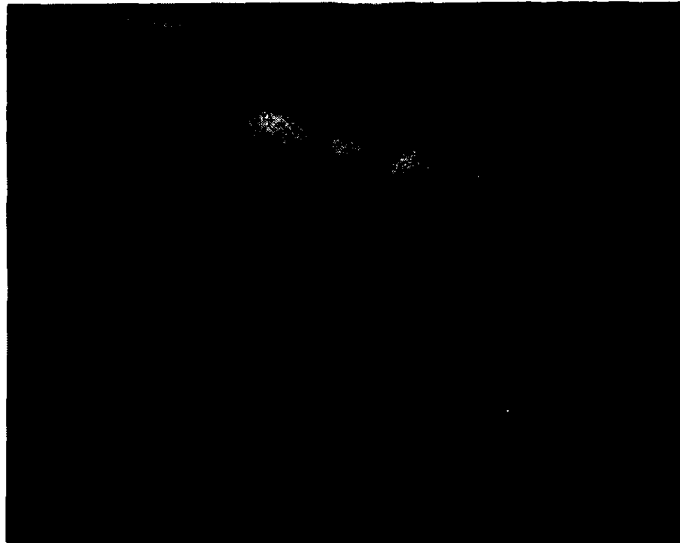


Perspective angle -35°
Range 1

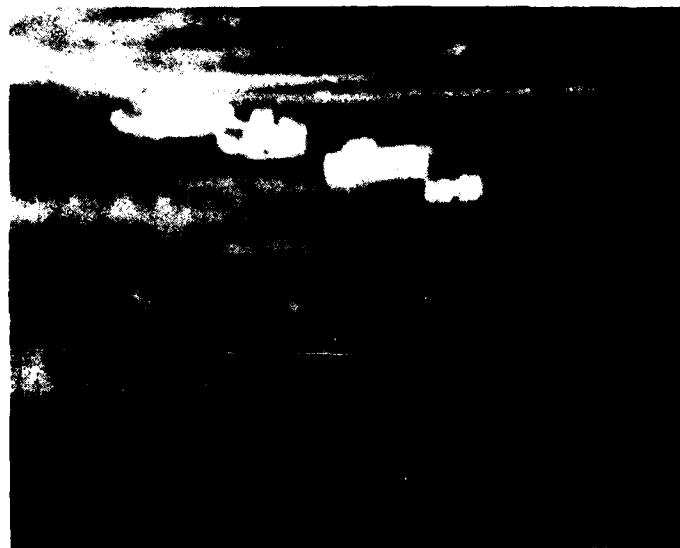


Perspective angle -35°
Range 2

Figure 19. Some FLIR Images from the AP Hill Data Base



Perspective angle -35°
Range 7



Perspective angle -35°
Range 8

Figure 19. Some FLIR Images from the AP Hill Data Base (continued)



Perspective angle 0°
Range 5



Perspective angle 0°
Range 6

Figure 19. Some FLIR Images from the AP Hill Data Base (continued)



Perspective angle 0°
Range 11



Perspective angle 0°
Range 12

Figure 19. Some FLIR Images from the AP Hill Data Base (continued)

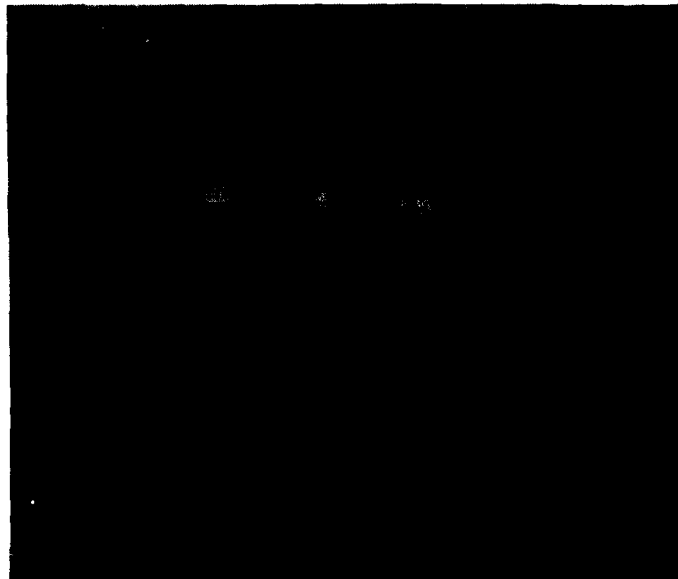


Perspective angle 50°
Range 1

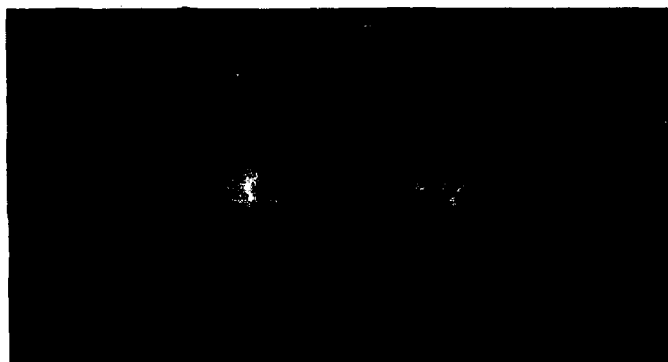


Perspective angle 50°
Range 2

Figure 19. Some FLIR Images from the AP Hill Data Base (continued)



Perspective angle 50°
Range 5



Perspective angle 50°
Range 6

Figure 19. Some FLIR Images from the AP Hill Data Base (concluded)

To acquire TV and FLIR imagery of the same scene, we are reviewing the Advanced Target Tracker video tapes (also from NV&EOL) to select appropriate sequences. Unfortunately the task is being somewhat hampered by the malfunctioning of the 875-line Westel recorder.

Table 8 summarizes the current state of the pattern-matching data base.

TABLE 8. CURRENT DATA BASE SUMMARY

SOURCE	TYPE	FRAMES	PERSPECTIVES	RANGE
PATS Training	FLIR 875 line	36	0°, -35°, +50°	0 - 8 km
LOHTADS (SWISTAK)	FLIR 525 line	6	±15°	3 - 4 km to closure
TERRAIN MODEL	Visual	16	0°, 90°	-

SECTION 9

PLANS FOR THE NEXT REPORTING PERIOD

From the results reported in the previous sections, it is apparent that although we have demonstrated successful pattern matching on FLIR examples, we still need to quantify the algorithm performance under a systematic, controlled set of sensor and scene parameters. Quantification of the performance can be divided roughly into two stages. The first fundamental issue is the probability of finding unique, distinctive objects in the two fields of view which can be used in the matching. This is a function of the field of view, the differences in the perspective angles, the angle of elevation, and the scene itself.

Second, given the presence of corresponding objects, how quickly does the branch-and-bound algorithm find the optimal corresponding matches? This is a function of not only the search procedure, but also the fidelity of the criteria used to evaluate a good match. The performance of the branch-and-bound algorithm obviously determines the computational requirements for its implementation in the munition.

Accordingly, our plans for the next reporting period are as follows:

- Quantify the probability of finding corresponding objects in the two fields of view as a function of the target signature, background signatures, sensor fields of view, perspective and angle of elevation differences, FLIR, TV sensors, etc.

- Given the corresponding objects to be matched, quantify experimentally the performance of the algorithm and the criterion function in terms of the number of subsets evaluated and the optimality of the final result for various aspect angles. Determine the sensitivity of the algorithm to segmentation noise and the absence of corresponding objects in the two fields of view.
- To provide the data for the above validation, digitize the imagery representative of the semi-autonomous acquisition scenario (especially wide and narrow fields of view) from the 875-line video tapes newly acquired from the PATS flight tests.
- Digitize frames from video tapes of the same scenes from the TV and FLIR sensors, gathered by NV&EOL for the Advanced Target Tracker Program. This will test the algorithms under different spectral characteristics.
- Explore additional techniques to make the branch-and-bound algorithm computationally more efficient. As we saw in Section 5, the optimal subset is obtained early in the search process, although the majority of the search time is spent in validating the optimality of this match. Therefore, it is feasible to trade off the guarantee of the optimality for the increased computational efficiency of the algorithm.

LIMED
-8