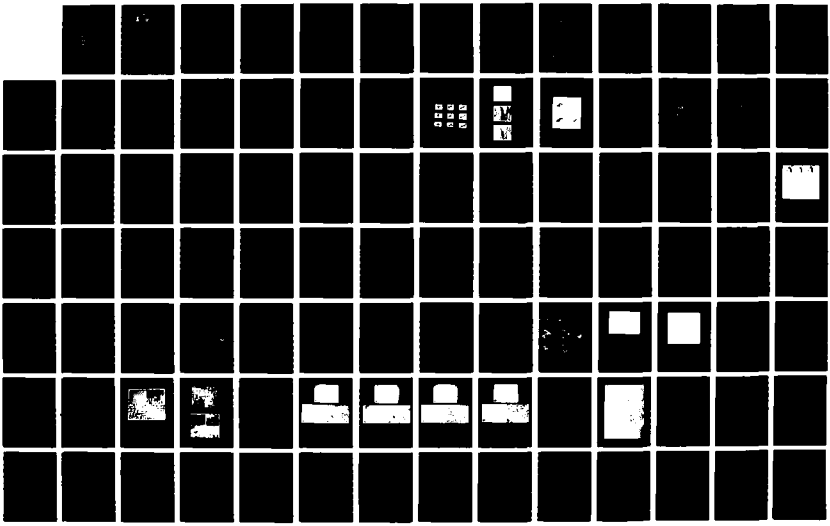
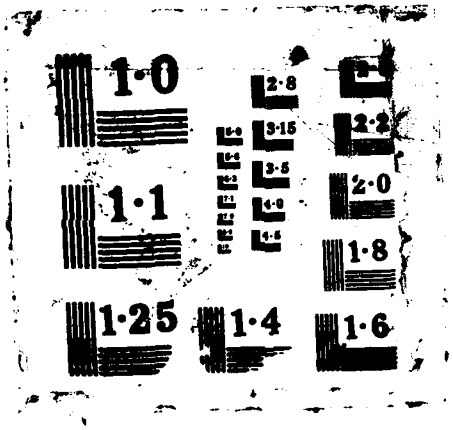
AD-A105 206 MULTI-DISCIPLINARY TECHNIQUES FOR UNDERSTANDING 1/2
TIME-VARYING SPACE-BASED..(U) CARNEGIE-MELLON UNIV
PITTSBURGH PA DEPT OF ELECTRICAL AND COM..
UNCLASSIFIED D CASASENT ET AL. 10 MAY 85 F/G 12/9 NL




1·0
2·8
3·15
3·5
4·0
4·5
2·2
2·0
1·1
1·8
1·25
1·4
1·6

AD-A185 286

DTIC FILE COPY

2

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 87-1028

6a. NAME OF PERFORMING ORGANIZATION
Carnegie-Mellon University

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
Air Force Office of Scientific Research

6c. ADDRESS (City, State and ZIP Code)
Department of Electrical & Computer Engg.
Pittsburgh, PA 15213

7b. ADDRESS (City, State and ZIP Code)
Directorate of Mathematical & Information
Sciences, Bolling AFB DC 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
AFOSR

8b. OFFICE SYMBOL (If applicable)
NM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR-79-0091

8c. ADDRESS (City, State and ZIP Code)
Bolling AFB DC 20332-6448

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A7	

11. TITLE (Include Security Classification)
Multi-Disciplinary Techniques for Understanding Time-Varying Space-Based Imagery

12. PERSONAL AUTHOR(S)
David Casasent, Arthur Sanderson and Takeo Kanade

13a. TYPE OF REPORT
~~Unclassified~~ Final

13b. TIME COVERED
FROM 5/84 TO 5/85

14. DATE OF REPORT (Yr., Mo., Day)
85/5/10

15. PAGE COUNT
1 2

16. SUPPLEMENTARY NOTATION
None

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
None

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This project is a multidisciplinary effort between 3 departments and principal investigators. It intends to combine: pattern recognition, image understanding and artificial intelligence techniques for space-based image processing as well as: optical and digital processing methods. Optical feature extraction and sub-pixel target detection and tracking results are summarized. Scene representation and modeling work using: probabilistic graph matching, multiple resolution rotation-invariant operators and texture analysis are detailed. Image understanding techniques for 3D scene interpretation discussed include 2D image-level methods (using features such as edges, lines and corners) and 3D scene-level methods. New dynamic programming, stereo image and model building results are included.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Vincent Sigillito

22b. TELEPHONE NUMBER (Include Area Code)
(202) 767- 4930

22c. OFFICE SYMBOL
NM

Table of Contents

ABSTRACT

This project is a multidisciplinary effort between 3 Departments and Principal Investigators. It intends to combine: pattern recognition, image understanding and artificial intelligence techniques for space-based image processing as well as: optical and digital processing methods. Optical feature extraction and sub-pixel target detection and tracking results are summarzied. Scene representation and modeling work using: probabilistic graph matching, multiple resolution rotation-invariant operators and texture analysis are detailed. Image understanding techniques for 3D scene interpretation discussed include 2D image-level methods (using features such as edges, lines and corners) and 3D scene-level methods. New dynamic programming, stereo image and model building results are included.

KEY WORDS

3D scene interpretation, artificial intelligence, feature extraction, hybrid processors, image understanding, multiple resolution rotation invariance, optical/digital processing, probabalistic graph matching, space-based imagery, sub-pixel targets, texture analysis, time-change imagery.

1. INTRODUCTION

1.1 OVERVIEW

This project is a multidisciplinary effort intended to combine methodologies for image analysis and interpretation, and evaluate the application of this integrated approach to problems of space-based imagery. The project has brought together research teams from within the Departments of Electrical and Computer Engineering, Computer Science, Robotics, and Biomedical Engineering of CMU.

We have chosen *time-varying space-based imagery* as the applications domain in which to evaluate our integrated approach. The two aspects of this domain are described below:

- *Space-based imagery* involves large amounts of information and incorporates both structural and textural properties of a scene. Efficient detection and representation of information in the scene are essential not only to interpretation but also to the storage and transmission of information. Scenes are predominantly two-dimensional although light and shadows affect imaging of both structures and texture, and interpretation of scenes at increasingly high optical resolution will require three-dimensional models.

- Interpretation of *time-varying data* is a primary goal of space-based image analysis and adds an additional dimension of complexity to the problem. We have chosen to look at three time-frame scenarios which require somewhat different analysis tools. High speed tracking is viewed as primarily a feature extraction problem and has been approached using optical methods. Medium and long-term time change detection must be based on a more abstract description of the scene and methods of representation and model-based interpretation must be brought to bear.

Within the context of the applications domain, we have addressed the following methodological research issues:

- Optical feature extraction and detection

- Structural and textural representation and matching

- Model-based image interpretation

- Hybrid digital/optical computer architectures

These issues are fundamental to implementation and performance of analysis tools which could imbed the inherently fast and parallel preprocessing power of optical techniques into a system which develops and tests hypotheses about scene representations and scene models.

In Chapter 1 of this report, we provide a more detailed overview of the conceptual framework of our proposed hybrid optical/digital system, define the space-based image processing problem, and discuss the importance of this work to Air Force technology and to related Air Force programs. Section 1.5 provides a summary of our research up to this year. Section 1.6 provides a summary of our current year of research, with details in Chapters 2-5.

1.2 CONCEPTUAL FRAMEWORK FOR HYBRID OPTICAL/DIGITAL IMAGE PROCESSING

In Figure 1-1, we show the general structure for our proposed hybrid optical/digital system using multiple methodologies for understanding space-based images. As shown in Figure 1-1, input images are preprocessed and then fed to parallel optical and digital channels in which multiple features are extracted. A parallel image modeling system is also shown which extracts structural descriptions of the image. These data plus image registration and target detection information obtained from an optical correlator channel are then used by an AI/IU system to modify the parallel input processing channels, to assemble and interpret a time-history track file on objects of interest in the image and to provide the necessary textural and graphic output reports.

1.3 PROBLEM DEFINITION

Advanced space-based sensor systems will provide us with high-resolution real-time multisensor data acquisition in the near future. This will totally pollute present processors unless we address how to intelligently and timely process and handle the projected data rates. NASA and others have already verified that the United States is capable of collecting more data than we can intelligently process (less than 1% of all NASA data has even been looked at [Wilson and Silverman, 1979]).

The key issue in Space-Based Image Understanding (SBIU) is not to transmit every frame of data (with 5000 x 5000 sensor elements in three bands with ten bits of data per pixel, and a 30 frame/sec rate, this is a data collection rate of over 10^{10} bits/sec). No existing technology can accommodate such a high data collection rate. Therefore, attention should be given to the algorithms required to achieve this. But first, here are several facts about SBIU problems:

1. In space-based image acquisition, we are monitoring certain areas and regions for diverse well-defined missions. We are only concerned with changes and do not need to know that nothing new has occurred in the image being looked at. When we transmit only the associated *change information*, we achieve a quite significant *bandwidth reduction*. Thus, we should process the data from space-based sensors on-board the platforms, determine image changes on-line, interpret the results and transmit only textural and graphic output reports.

2. We know rather well where the satellite is and where it is looking and we know that the scene being imaged correlates with the prior image frame or with our stored reference.

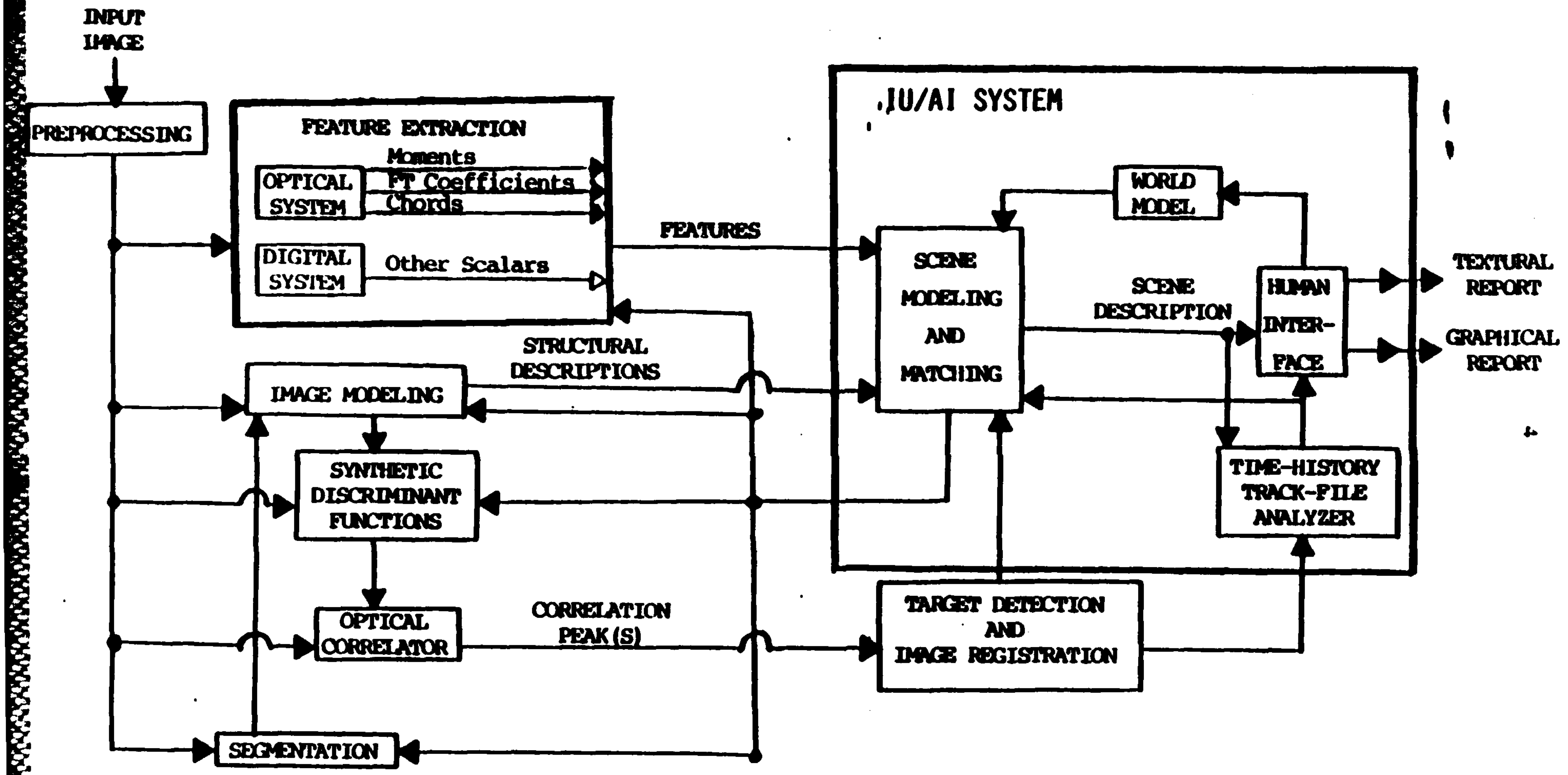


Figure 1-1 : Hybrid optical/digital multi-disciplinary (pattern recognition/image understanding and artificial intelligence) processor.

The problem is thus different from the often discussed unbounded and unsupervised target recognition problem. We can and must utilize this *a priori information* that the frame we are investigating correlates with a previous one in our processing algorithms.

3. To provide better *image registration* accuracy and to facilitate pointing of secondary sensors at given areas of interest, it is often necessary to *locate key landmarks* in the image. This is also useful in determining *geometrical corrections* needed.

4. It is also useful and necessary to register two successive image frames for *inter-frame integration* to decrease the variance of the noise and to improve the image quality. This is essential to accommodate platform variations with time and background drift. Often, *sub-pixel image registration* is necessary.

5. It is obviously essential to *subtract successive frames* since this provides the necessary change detection or time-varying target data.

6. However, in most cases, the image registration in (4) is sub-pixel and thus before performing (5), we must *interpolate* the images.

7. Once *time-history track files* of candidate objects of interest in the field-of-view of the sensors have been obtained, a multitude of discrimination analysis techniques, AI, IU, pattern recognition and human perception algorithms are necessary to classify, understand and interpret the time-change activity noted.

8. In advanced sensor systems, 3-D information on the scenes will be available from stereo satellites or other techniques. In such cases, we can fully capitalize on the available image information only by the use of advanced *3-D scene modeling and interpretation*. The key point is the extraction of scene information (3-D) from time-histories of 2-D images.

9. To detect and describe detailed changes in the 3-D structure of scenes, it is useful to first *generate 3-D scene descriptions from the 2-D images*, and then to compare the descriptions for changes. Conventional 2-D change detection approaches are not as useful for high resolution images of complex scenes since they do not take into account factors such as different viewpoints and different lighting conditions for the different images of the scene. In order to detect changes over successive images of a given scene obtained over time, it is useful to maintain a 3-D model of the scene and automatically update the model as changes occur. This requires the ability to match the model with each new view of the scene. *Matching in 3-D* is more desirable than matching in 2-D since the 3-D information is represented in a manner that is independent of viewpoint and lighting conditions.

10. The *3-D scene model* is a useful central component for many aspects of the change detection task. Not only is it useful for determining whether changes have occurred, but it also permits *model-based interpretation* of new images and serves as a central representation for accumulating 3-D scene information from various low-level experts. Our new research addresses these aspects of time-history 3-D scene information.

Items 1-6 address the high throughput signal processing aspects of SBIU, whereas items 7-10 address the advanced image understanding aspects of this problem. Table 1-1 summarized objectives which must be attained to achieve the overall goal of SBIU. In Table 1-2, techniques required to attain these objectives are listed, and Table 1-3 lists the disciplines which will contribute to the achievement of our goals. As well as image processing *per se*, we must study the importance of efficient database organization and manipulation since storage or transmission of a very large database will be required for SBIU.

To properly address understanding of time-varying space-based images, we feel that three different SBIU time-varying image processing scenarios (Table 1-4) must be separately addressed. We propose to study each of these during the course of our research. We distinguish the three cases by the change rate and the domain of analysis. In the first case (rapid time-variations), we can consider a missile launch. In this application, the objective is to track the time-history of the missile and to transmit the information that a missile has been launched (from subsequent sensors, the missile's trajectory etc. can be obtained from our system techniques and algorithms). The second case (medium time-variations) can concern monitoring of key sites such as airports, railroads and harbors and know areas of anticipated concentrations of troops or armor. In this case, troop or armor movement and air, land and sea activity can be obtained from time-varying image data. This second scenario is typical of a case in which extensive AI and IU techniques are appropriate (i.e., the use of information on the locations of hangers, runways, railroad tracks, terminals, switching yards, harbor channels, docks, piers, etc.). This also requires the locations and registration of these items in sequential image frames. The third case (slow time-variations) addresses urban development and agricultural or land use activity (as in Landsat and ERTS case-studies).

Table 1-1: Objectives of Space-Based Image Processing

- Detection of image changes
- Use of *a priori* knowledge
- Location of key landmarks
- Time-history track file acquisition
- Interpretation of time-history data
- 3-D scene interpretation
- Efficient storage and retrieval of information from database

The three scenarios noted in Table 1-4 constitute our definition of the SBIU problem. All cases require the techniques and disciplines noted in Tables 1-2 and 1-3. The first case (rapid time-

Table 1-2: Image Processing Techniques Required for SBIU

- Image enhancement and preprocessing
    - Image registration (sub-pixel) for frame integration
    - Image subtraction for time-history extraction
    - Image interpolation for image subtraction
- Image segmentation
- Feature extraction
- Image modeling
- 3-D scene modeling and interpretation
- Hierarchical database design

Table 1-3: Disciplines Required to Achieve Real-Time Space-Based Image Processing

- Pattern recognition
- Image understanding
- Human perception
- Artificial Intelligence
- Optical Processing
- Digital Processing

Table 1-4: Time-Change Scenarios

TIME CHANGE	EXAMPLES	DOMAIN OF ANALYSIS
Rapid	Missile Launch	Image Pixels
Medium	Railroad, Airport, Harbor, Troops, Armor	Scene Structure
Slow	Agricultural, Land-use, Urban Development	Statistical Image Modeling

variations) requires primarily sub-pixel image registration, frame integration, frame interpolation, and image differencing. The second case requires techniques involving image interpretation, 3-D scene modeling, 3-D matching and comparison, plus knowledge-based geometric reasoning. The third case needs more statistical techniques and statistical image models, more so than do the others. All cases require object and scene modeling, image preprocessing and enhancement plus segmentation, feature extraction and classification. Figure 1-1 depicts these aspects and the interactive multi-disciplinary feedback required to solve these SBIU problems.

1.4 BENEFIT TO AIR FORCE TECHNOLOGY

With our three scenario problem definition (Table 1-4), we now consider the myriad of Air Force programs and technology that can benefit from our proposed research. First, we note that our research is directed toward the development of new algorithms and their realization in a hybrid optical/digital architecture. However, devices and architectures being developed in related Air Force programs in VHSIC and VLSI, systolic array processors, Josephson junction devices, etc. can also be used for implementation of these algorithms. Our work will thus provide problem definition and direction regarding algorithms for such parallel processor architectures and technology programs. Large data storage requirements and studies of what constitutes a valid database are also integral parts of this program. Similar Air Force efforts toward data storage and database acquisition are thus of direct concern to this program. The Air Force programs in: intelligent sensors, intelligent task automation, automated manufacturing, image understanding, human perception and visual psychophysics will directly benefit from the inter-disciplinary nature of our research. The large Air Force effort in optical data processing will directly benefit since real-time spatial light modulators and holographic optical elements will be needed for implementation of our algorithms in real-time. The Air Force programs in missile guidance require a new set of algorithms and attention to the database requirements and performance measures used and thus they will likewise benefit extensively from this program. Darpa/AF programs such as HALO and HICAMP will clearly benefit from our chosen time-varying SBIU tasks.

The monitoring of changes and developments at cultural sites, such as urban areas and military bases, is a very useful application of space-based sensors. The techniques we develop will aid in detecting and describing both large-scale and detailed changes. Furthermore, the techniques dealing with 3-D matching and comparison, and knowledge-based geometric reasoning will enhance Air Force programs in sensing and robotics.

1.5 SUMMARY OF RESEARCH DONE IN YEAR ONE

In our first year of research, we focused on the development and evaluation of methods which yield representations of structural and textural information in an image, and relate these representations to object and surface contour properties of the scene. The techniques studied included *Probabilistic Graph Matching*, *Multiple Resolution Structural Basis Functions*, and *Textural Surface Models*. The structural basis function and texture models were found to be particularly well suited to parallel or optical processor implementation. Two digital processing facilities for use in this program were also assembled: the RAPIDbus architecture, and an Optical Data Processing, Digital Processing and Simulation Facility.

We also achieved a major effort on the extraction of time-varying sub-pixel targets in noise. This time-change scenario concerns applications such as the detection of missile launches or aircraft in flight. In the first year, we successfully demonstrated the conceptual ability to detect and track sub-pixel targets.

In the 3D change detection task, we achieved results in two aspects: the low-level problem of analyzing images and the high-level problem of representing, constructing, and updating the scene model. We developed techniques for extracting building structures from high resolution aerial images of urban scenes, including lines not originally found but predicted by the model. Image lines were classified as building boundaries or other lines which arise from texture and shadow boundaries. We also experimented with efficient methods of searching a line image in order to form junctions which can then be used for stereo matching.

At the higher level of processing, we developed techniques for representing, constructing, and updating the scene model, using task-specific knowledge.

1.6 RESEARCH PROGRESS IN YEAR TWO

1.6.1 Optical Feature Extraction and Sub-pixel Tracking

The optical feature extraction phase of this project has been terminated except for a small synthetic discriminant function (SDF) effort we still report upon (for aircraft) in our 1985-86 report. This was necessary because of the reduction in ECE funding for year 3 to one-third of our prior year 2 level. Our year 2 progress and the proposed tasks for year 3 work included in the new task list are all addressed herein and terminated (except for the one SDF effort noted above). Our final report on these is contained in chapter 2, appendix A1 and the appendices of our proposal referenced in chapter 2.

This effort included attention to moment, chord and other optically-generated feature spaces. Architectures for each of these methods were devised and initial results were obtained. These showed : the ability to optically implement various feature extractors: the architecture for a hybrid optical/digital moment processor, successful initial tests of this architecture on a ship image data base and a robotic pipe part data base: new results on the accuracy of distortion parameter estimation with this processor, an advanced correlation SDF synthesis method and most successful initial test results of it on ATR vehicles.

Our time change detection work has achieved various significant results and demonstrations of the ability to detect sub-pixel target: rearrangement of our software to insure proper statistical characteristics of the generated scenes; the development of new single differencing methods that prove promising for clutter suppression; the initial formulation of general space/time filtering for target enhancement and background suppression; the investigation of detector limitation effects.

Our investigations have revealed a potential nonzero mean problem in the correlated noise images with high correlation coefficients. This problem is overcome by appropriate modifications to our software. Our software is also rearranged to provide a more unified control of the various parameters characterizing the synthetic image. We have observed that while the exponential sub-pixel shift estimator performs better than the parabolic estimator for the synthetic data, the reverse is true for LFM signals. This indicates the need to consider both sub-pixel estimators in the future. Our efforts have also pointed towards more sophisticated space/time processing methods for better clutter suppression.

1.6.2 Algorithms for Hybrid Digital/Optical Representation and Matching

This phase of the project has focussed on the development and evaluation of methods which yield representations of structural and textural information in an image, and may be used for matching images to scene models. The principal results achieved in this research include:

- *Probabilistic Graph Matching* - Attributed graph structures are used as models of structural and statistical information in the image. Matching of these graph structures using probabilistic similarity methods poses a number of interesting problems in the mathematical formalism, in the computational matching algorithms, and in the application of these methods to real images. We have investigated methods of subgraph decomposition which permit branch-and-bound search of the matching tree and provide efficient pruning of the possible matches.

- *Multiple Resolution Rotation-Invariant Operators* - The MRI (Multiresolution Rotation Invariant) operator and the MRD (Multiresolution Difference) transform have been introduced to extract structural and textural features of images for use in matching and interpretation phases of analysis. The MRI is a complex operator derived from derivative expansions of Gaussian kernels and will have magnitude of response independent of feature orientation and phase angle of response which provides information about orientation. The spatial and frequency domain properties of these operators have been studied and an approximate MRI operator which uses difference of shifted Gaussian kernels has been derived and shown to be computationally efficient due to the scaling and shift properties of the Gaussian kernel. The MRI operators have been applied to aerial images of objects and textures.

- *Texture Analysis* - The MRI operators described above have been used to characterize and classify textures from aerial images. This set of multiresolution operators permits classification of texture independent of the size and orientation of the texture pattern itself. The statistical distribution of the magnitude responses is analyzed across the set of operators for regions of the image. Correlation with the corresponding magnitude range and the corresponding phase distribution provides information on the relative scale and the relative orientation. Experiments on textures from aerial images and textures from simple patterns have been carried out and compared to previous texture energy operators.

The algorithms studied in this section reflect the interdisciplinary nature of the project. The MRI operators and associated texture measures are particularly well-suited to parallel or optical processor implementation. They will be implemented and evaluated on the array processor with *RAPIDbus* host. Our formulation of the *recursive model-matching algorithms* is also intended for implementation on this type of architecture with extensions which may integrate symbolic and numerical processing. The interactive use of parallel and optical preprocessing with hypothesis formation and adaptive search strategies will be natural continuation of the work completed.

1.6.3 Image Understanding Techniques for 3D Scene Interpretation

The problem of detecting three-dimensional changes in a complex urban scene is a very difficult one, particularly since any information extracted from the complex images is highly incomplete and contains many errors. Therefore, we have thus far concentrated mainly on the problems of extracting information from such images and accumulating the information in a 3D scene model.

In this report, we describe results in two aspects of these problems: low level image analysis and high-level model maintenance. The goal of low-level image analysis is to determine a set of 3-dimensional line segments in the scene which correspond to building boundaries. The first step in such a process is to map the two-dimensional image into a 3-dimensional scene. One method of doing this is to perform stereo matching on a pair of images and use triangulation to determine the third dimension.

We have developed a stereo algorithm using the technique of dynamic programming. The stereo matching problem, *i.e.* obtaining a correspondence between right and left images, can be cast as a search problem. When a pair of stereo images is rectified, pairs of corresponding points can be searched for within the same scanlines. We call this search *intra-scanline* search. This intra-scanline search can be treated as the problem of finding a matching path on a 2D search plane whose axes are the right and left scanlines. Vertically connected edges in the images provide consistency constraints across the 2D search planes. *Inter-scanline* search in a 3D search space, which is a stack of the 2D search planes, is needed to utilize this constraint.

Our stereo matching algorithm uses edge-delimited intervals as elements to be matched, and employs the above-mentioned two searches: inter-scanline search for possible correspondences of connected edges in the right and left images, and intra-scanline search for correspondences of edge-delimited intervals on each scanline pair. Dynamic programming is used for both searches which proceed simultaneously at two levels: The former supplies the consistency constraints to the latter, while the latter supplies the matching score to the former. An interval-based similarity metric is used to compute the score.

In order to pursue the problem of high-level model maintenance independent of the current state of the low-level image analysis research, we have chosen to investigate model building using rangefinder data, which is already three dimensional. Specifically, we have developed techniques for extracting detailed, complete descriptions of polyhedral objects from light-stripe rangefinder data. The descriptions are in the form of 3D faces, edges, vertices, and their topology and geometry. A range image is first segmented into edge points. A line drawing is then obtained by fitting linear segments to the points in the image, and refining the segments to eliminate gaps. Faces are then generated from the line drawing. Interestingly, although the final description is in 3D, most of the processing is done in the 2D image space. This work will be applied towards the goal of obtaining a full symbolic description of a scene from range data obtained from multiple viewpoints. Our 3D model building and updating results are detailed in Chapter 5.

2. GENERAL 3-D FEATURE EXTRACTORS AND CORRELATORS

This ECE project phase has been terminated because of the significant reduction in funds for 1985-1986. Because of the lack of future support possibility, contingency funds available from other sources were not spent to continue this project.

The intent of this task was to employ feature extraction and correlation techniques to locate, track and identify large targets in 3-D in severe clutter. These output track files on candidate targets would then be processed by the IU/AI portion of the system.

2.1 TARGET GENERATION

Moving targets and aircraft imagery were emphasized. For such a scenario, we proposed a quite novel image generation software package for aircraft imagery. This routine (Figure 2-1) consists of 3 stages. The final output is a 2-D image of the aircraft as seen from any user specified orientation angle φ and for any object-centered rotations θ_x, θ_y and θ_z and at any scale and resolution. The aircraft data base consists of Soviet and U.S. military aircraft as well as commercial aircraft. Figure 2-2 shows typical images of several of these aircraft at different orientations. A most attractive aspect of this routine is the efficiency of Step 3. Specifically, our initial calculations indicate that the required matrix transform operations can be computed (for all target vertices, to determine the 2-D projections of the image to be seen) within 150μsec using a quite modest array processor. This has significant importance for PR since one can now realistically assume that any necessary reference image (for correlation or feature extraction purposes) can be computed on-line. We also began initial efforts to modify this algorithm to enable range images to be processed (with pixel values proportional to the range of that portion of the target). This satisfies our promised research on the proposed Task 2 item for 1985-1986.

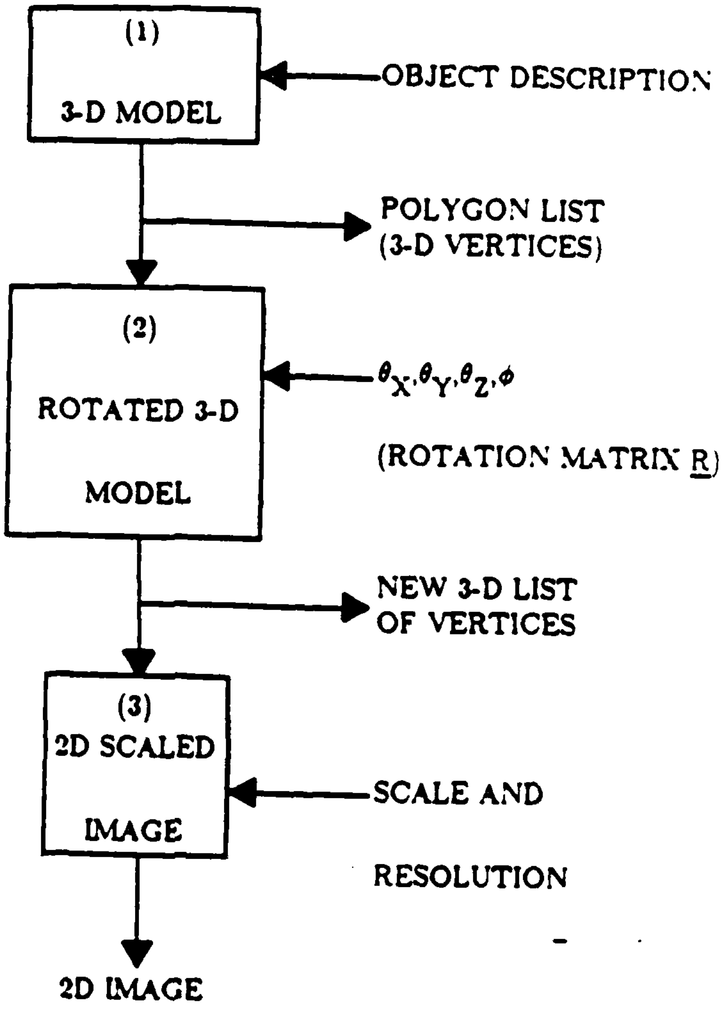


Figure 2-1: 3-D Model-Based Object Recognition

2.2 TARGET DETECTION

For moving target acquisition of such objects, we simply generate two image frames with a standard background and with an aircraft target from our routine in Figure 2-1 inserted in each, with a displacement of the target between two successive frames (Figures 2-3a and 2-3b). A simple differencing of these two frames results in extraction of the target (Figure 2-3c). More advanced (higher level, etc.) image differencing methods are required for other cases. A time-sequence of such output imagery provides information for a target track file and for input to an optical Kalman filter for state estimation and trajectory estimation. At this point, a high-resolution adjunct sensor can be activated to track the object. Alternatively, a laser radar providing range data can be activated. Figure 2-4 shows typical range images of the aircraft generated on our routine in Figure 2-1.

2.3 MOMENT FEATURE SPACE

The moments of an object can be optically computed [Casasent et al, 1982]. In Appendix A of our proposal, we fully detailed our proposed moment processor for aircraft classification. This hierarchical system employs two first-level estimators (one for aspect and one for the object class) and a second-level Bayesian classifier (requiring a nonlinear iterative technique to achieve class and information parameter estimation). In recent work, this algorithm has been fully encoded, but due to lack of funds, we were unable to test it on our aircraft image data base. Initial tests showed that it provides quite accurate object orientation estimates. For sufficiently separated classes, it was found to have surprisingly good noise immunity. The major attraction of this system is its theoretical basis. We have now showed that moment features are jointly Gaussian random variables for input plane translation, scale and rotation distortions. A Bayesian classifier is possible and optimal (however, each object class and object aspect view must now be treated as a different image class). The first-level estimators achieve a reduction in the aspect views and classes that the second-level system must search. The class estimator in this first-level of the processor uses unique organized hierarchical tree structure methods for synthesis of the tree. The node selection and discrimination function per node in the tree are selected automatically using a two-level Fisher classifier (following the first-level multi-class Fisher classifier, employed to achieve organized hierarchical structure for the tree). The resultant node structure is thus not ad hoc, as is generally done. [Casasent and Cheatham, 1985] detail the most recent and our expected performance of this algorithm (Appendix A).

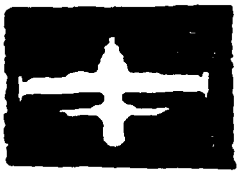
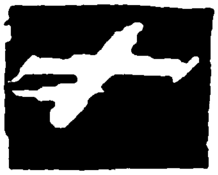
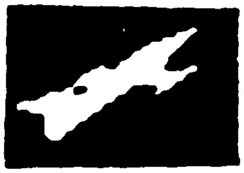
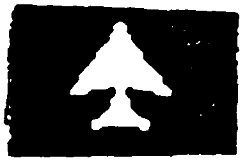
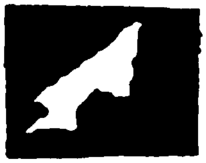
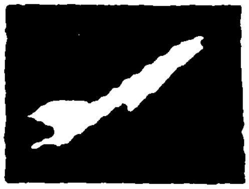
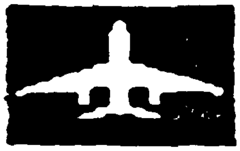
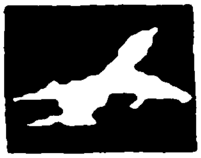
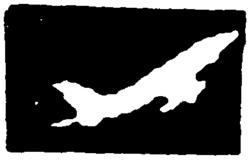
ORIENTATION / AIRCRAFT	$\phi=30°$ ELEVATION $\theta_X=0°$ $\theta_Y=0°$ $\theta_z=0°$	$\phi=30°$ ELEVATION $\theta_X=0°$ $\theta_Y=30°$ $\theta_z=0°$	$\phi=30°$ ELEVATION $\theta_X=0°$ $\theta_Y=60°$ $\theta_z=0°$
F104 (USA)			
MIG (USSR)			
B737 (COMMERCIAL)			

Figure 2-2: TYPICAL 2D IMAGE OUTPUTS FROM THE SYSTEM OF FIGURE 2-1

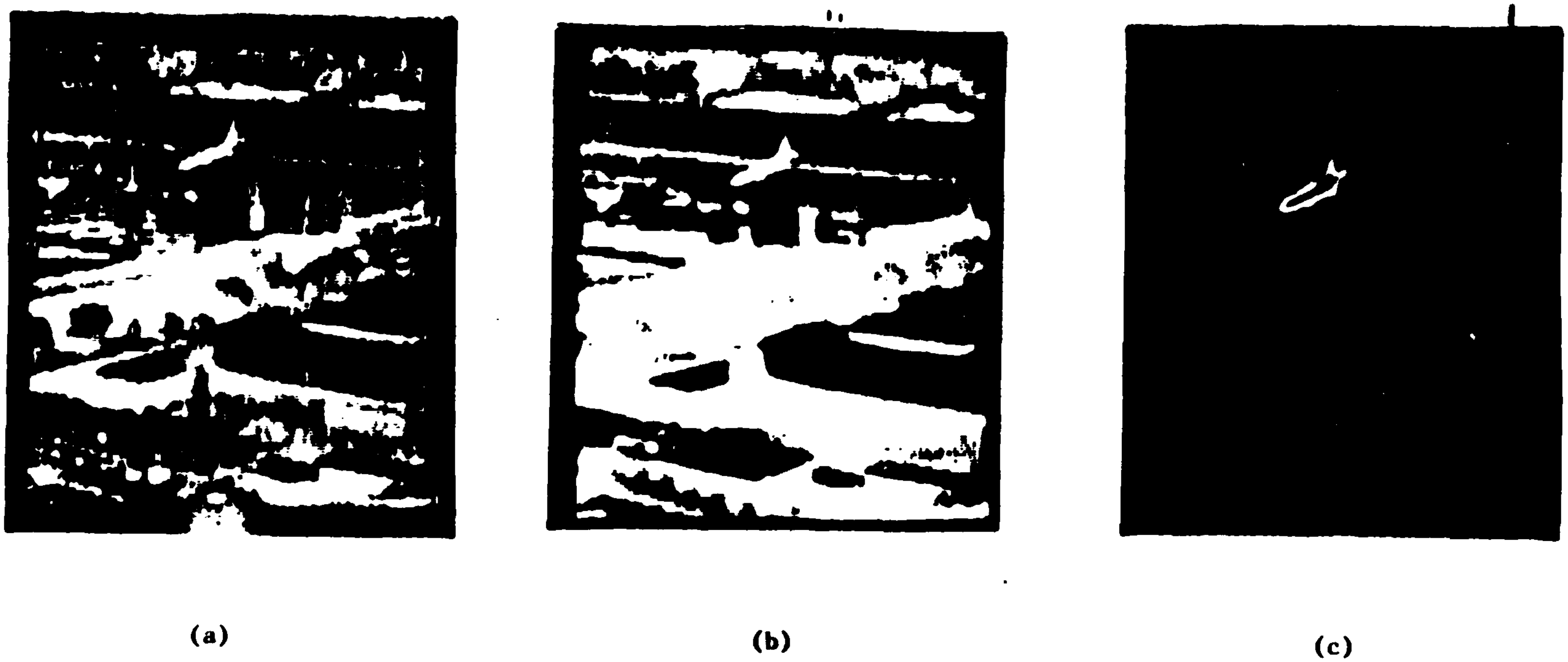


Figure 2-3:
Frame 1 (a) and Frame 2 (b) with a target object shifted by several pixels between frames, and the resultant output difference frame (c)

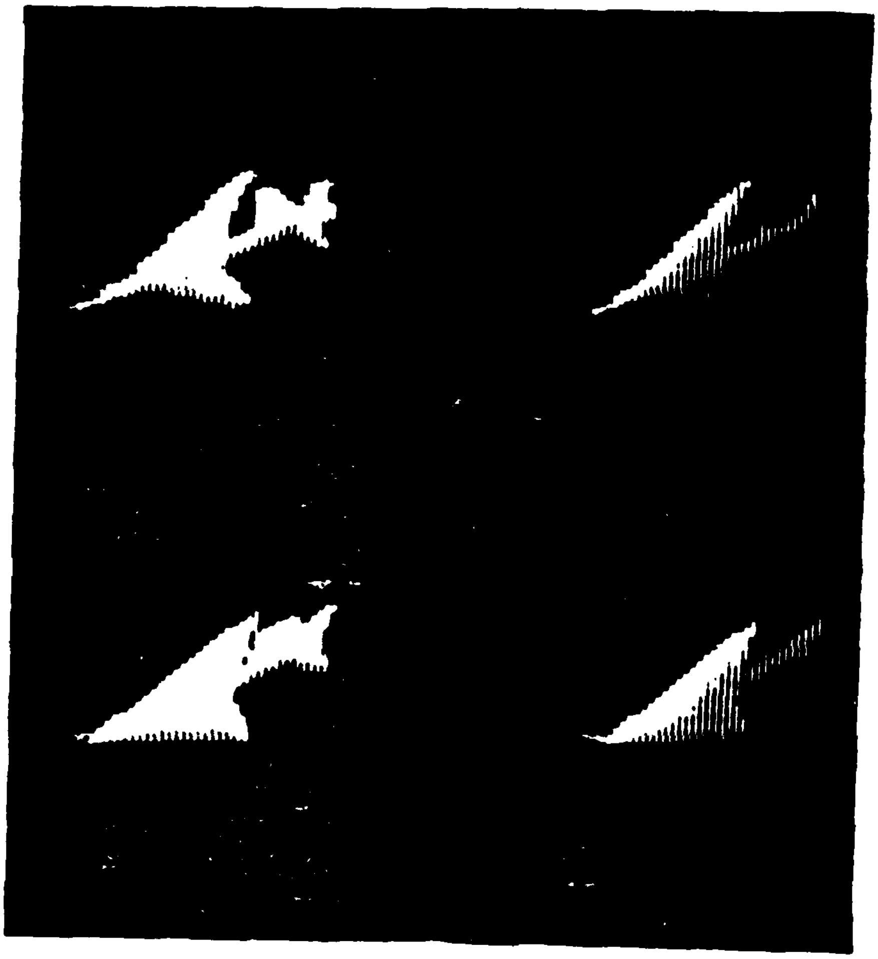


Figure 2-4: Examples of synthetically generated range images

2.4 CHORD FEATURE EXTRACTION

Our work on this feature extractor was summarized in Appendix B of our proposal. This technique appears most attractive for range imagery with reliable gray scale, and in this case it appears preferable to other realization schemes. No further work on this powerful optical technique to generate these distortion-invariant features in parallel has been performed.

2.5 SDF CORRELATORS

This novel class of correlator that promises distortion-invariant object identification was fully summarized in Appendix C of our proposal. It is thus not documented further here. Initial tests of this technique on aircraft will be included in our 1985-1986 research as our proposed Task 7 research for 1985-1986.

2.6 HISTOGRAM PROCESSING

Only initial work on this proposed (1985-1986) task item was advanced. We developed routines to compute and display histograms. We then generated range imagery of selected aircraft and investigated their histograms and their potential use in distortion-invariant object classification. Figure 2-5 shows the histogram of an F15 with in-plane rotations about the z axis by 0°. 30°, 60° and 90°. As seen, all histograms are essentially identical. This verifies that histograms are invariant to in-plane rotations. Figure 2-6 shows the histograms for the same object in Figure 2-5 scaled in range by a factor α. Comparing Figures 2-5 and 2-6, we note that the shapes are the same for both patterns, but that one pattern axis is shifted. This verifies the invariance of the shape of the histogram distribution with object scale and the ability to determine object scale or target range from such data. Figure 2-7 shows histogram plots for similar object rotations for an MIG. The numbers 1-3 denote different portions of the aircraft (wings, fuselage, tail assembly). A comparison of Figure 2-7 with Figures 2-5 and 2-6 shows that the shape of the histogram can provide aircraft discrimination. This concludes our report for 1985-1986 on our proposed Task 1 and Task 6 research.

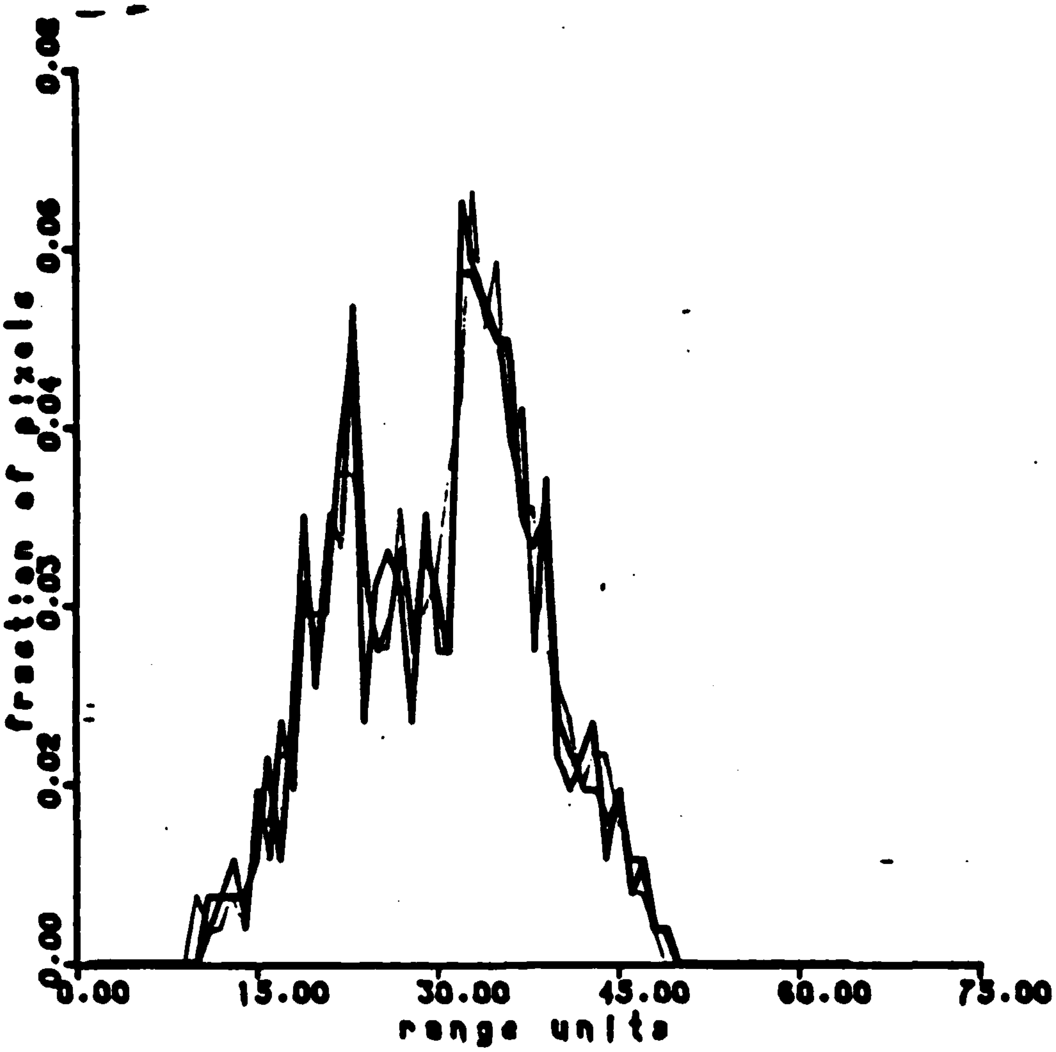


Figure 2-5: Range histogram of an F15 (64 x 64 pixels). The reference image is at an elevation of 45°

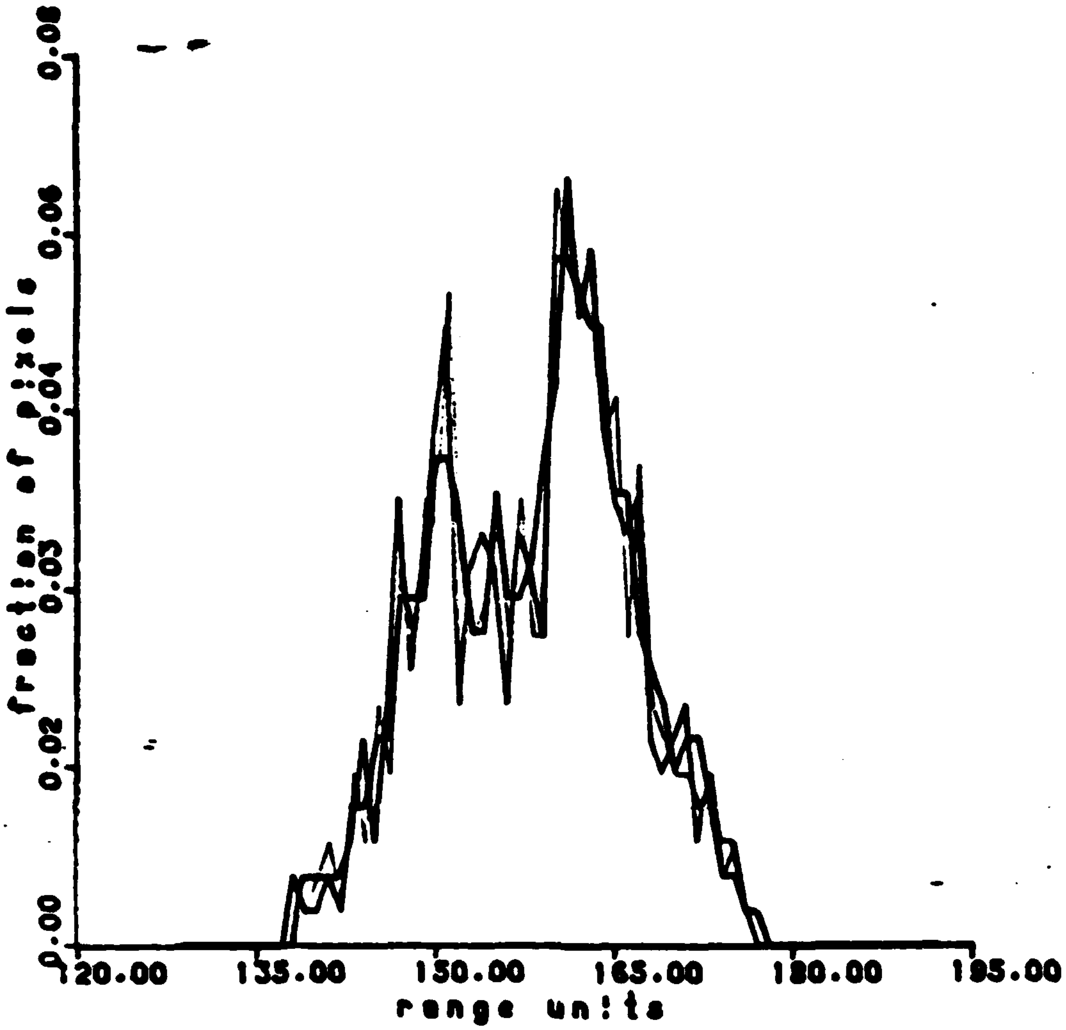


Figure 2-6: Range histogram of an F15 displaced in z-range by 128 pixels. Size 64 x 64

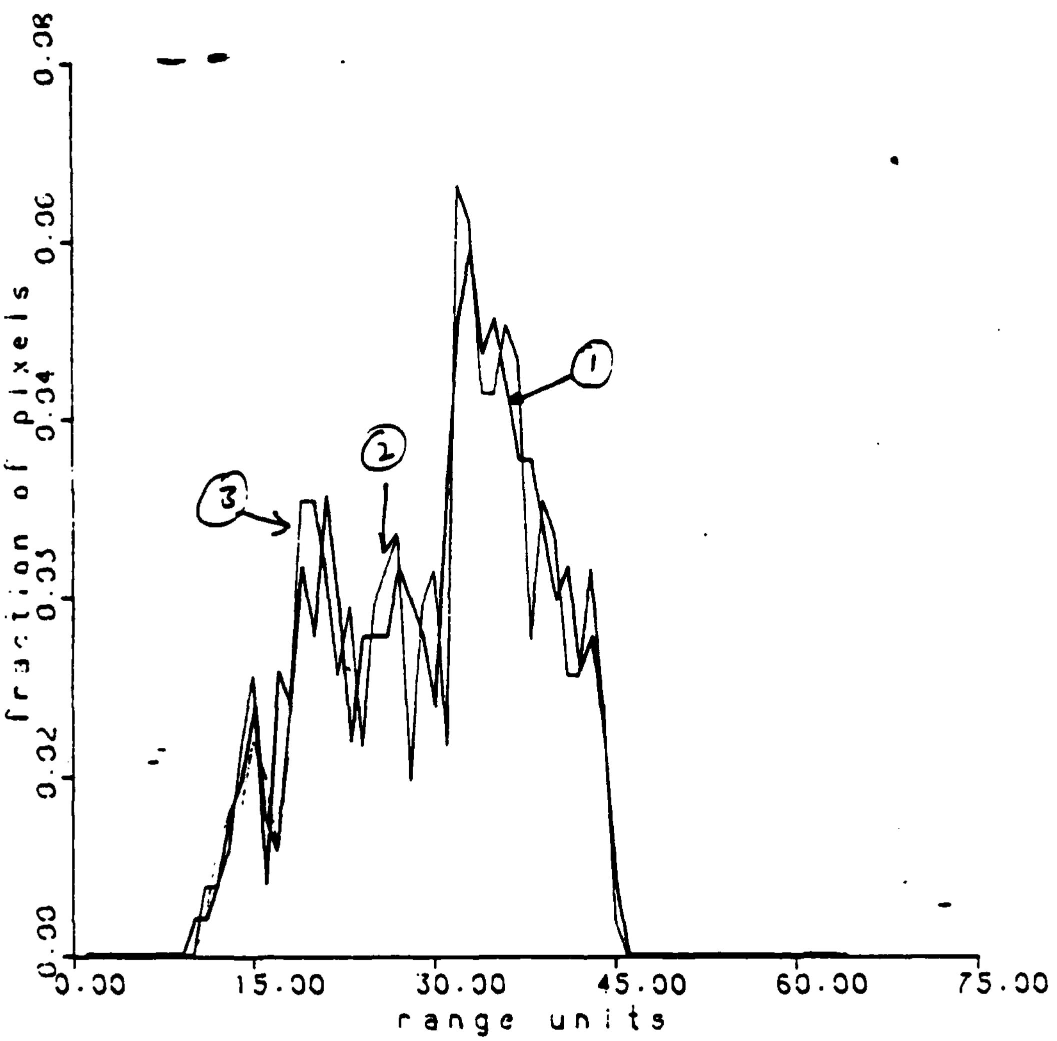


Figure 2-7: Range histogram of a MIG (64 x 64 pixels) corresponding to the same orientation as in Figure 2-5

REFERENCES

[Casasent et al. 1982].
D. Casasent, R.L. Cheatham and D. Fetterly, "Optical System to Compute Intensity Moments : Design", Applied Optics, *21*, 3292, September 1982.

[Casasent and Cheatham, 1985].
D. Casasent and R.L. Cheatham, "Hierarchical Feature-Based Object Identification", OSA Topical Meeting on Machine Vision, Incline Village, NV, March 1985.

3. SUB-PIXEL TARGET DETECTION AND TRACKING

In our year 2 effort, we investigated several issues concerned with the detection and tracking of sub-pixel targets. These issues include improved database generation, selection of optimal sub-pixel location estimators and interpolators and quantification of detector limitations. In this chapter, we present the various results obtained in these efforts.

3.1 IMAGE GENERATION

Our year 1 report fully outlined the software needed for the generation of synthetic imagery being used for simulation. The staring sensor image I(x,y) consists of three separate images : a sub-pixel (of extent less than 1 pixel in the detected image) target with a constant value, Gaussian correlated noise (CN) image with prescribed mean, variance and correlation coefficients to simulate the clutter background and uncorrelated white Gaussian noise (UCN) image of zero mean and prescribed variance to simulate the instrumentation noise. These can be generated as below.

Let g(x,y) denote a NxN array of white, Gaussian random numbers of zero mean and unit variance. Such an array can be obtained from the IMSL software package [IMSL1982]. Then a zero mean CN image f(x,y) with variance σ^2 and correlation coefficients ρ_x and ρ_y can be obtained by the following 2-D digital infinite impulse Response (IIR) filtering.

$$f(x,y) = -\rho_x \rho_y f(x-1,y-1) + \rho_x f(x-1,y) + \rho_y f(x,y-1) + \sigma(1-\rho_x^2-\rho_y^2+\rho_x^2\rho_y^2)^{1/2} g(x,y) \tag{3.1}$$

These CN images are used to simulate clouds at various heights. Different cloud levels are characterized by different means, variances and correlation coefficients. Because of the small time interval between successive image frames, we assume that the CN images are coherent between successive frames[Rauch 1981]. This property is easily accomplished in our software by maintaining the seed value for the random number generator (RNG) to be the same. On the other hand the UCN image is used to model instrumentation noise and it is independent from frame to frame. UCN can be easily generated by multiplying the $g(x,y)$ of eq.(3.1) (obtained with different seeds) by proper constants. The sub-pixel point target is modeled as zero outside a constant square region (whose dimensions are less than 1 detector image pixel) in the high resolution image.

The CN, UCN and target images are combined to yield a high resolution (532x532 pixel) image. Overlapping blocks of approximate size 8x8 are then combined with the help of a blur function to yield the detector image $d(x,y)$ of size 64x64 pixels. The blur function is constant in the interior of a 8x8 region and has Gaussian tails at the boarders. Sub-pixel motion of the target can now be easily simulated by moving the target by integer pixels in the high resolution imagery.

While the above procedure of generating a staring sensor image by combining CN, UCN and sub-pixel targets at high resolution and blurring them seems satisfactory, we observed that the detected images had a significant non zero mean. To detect the source of this discrepancy, we conducted an investigation of the statistical parameters yielded by the IMSL programs.

The mean of the random array g(x,y) of size NxN is obtained as

$$\hat{\mu}_g = \frac{1}{N^2} \sum_{x=1}^{N} \sum_{y=1}^{N} g(x,y) . \qquad (3.2)$$

It can be easily shown [Papoulis] that this mean estimator is unbiased and has a standard deviation of (σ/N) where σ^2 is the variance of the noise $g(x,y)$. For the images of interest, $\sigma = 1$ and N $\approx$ 500 yielding an expected standard deviation of 0.002 in the estimated mean. In Table 3-1, we show the estimated means and variances as well as the theoretical standard deviation in this estimated mean as a function of the image size N. It can be seen from this table that the estimated means are well within (one σ) their expected statistical fluctuations. Thus the RNG being used seems satisfactory.

For a 512x512 UCN image $g(x,y)$, the estimated mean $\hat{\mu}_g$ is of the order 0.002. When this UCN image is input to the 2D digital IIR filter of eq.(3.1), we can show that the resulting CN image f(x,y) has following estimated mean.

$$\hat{\mu}_f = \sigma \frac{[(1+\rho_x)(1+\rho_y)]^{1/2}}{[(1-\rho_x)(1-\rho_y)]^{1/2}} \hat{\mu}_g \qquad (3.3)$$

In our simulation $\sigma = 1$ and $\rho_x = \rho_y = 0.95$. Thus, $\hat{\mu}_f$ is about 40 times as large as $\hat{\mu}_g$. Thus a variation of almost 0.08 can be seen in the CN image mean as a result of variation of 0.002 in the UCN

SIZE N	Mean Estimate $\hat{\mu}$	St.Dev$\{\hat{\mu}\}$	Variance Estimate $\hat{\sigma}^2$
10	0.00218	(1/10)	0.9488
50	-0.01724	(1/50)	1.0018
64	-0.00181	(1/64)	0.9843
128	0.00548	(1/128)	0.9947
256	0.00252	(1/256)	0.9961
512	0.00055	(1/512)	1.0025

Table 3-1: Measured estimated statistical parameters for UCN data with $\mu = 0$, $\sigma = 1$

mean. This amplification factor increases as ρ_x and ρ_y values approaches 1. As an example, $\hat{\mu}_f$ is about 200 times as large as $\hat{\mu}_g$ for $\rho_x = \rho_y = 0.99$. This problem is illustrated in Table 3-2 where we show the estimated means of a CN image for various $\rho_x = \rho_y = \rho$ values.

$\rho_x = \rho_y = \rho$	Estimated Mean $\hat{\mu}_f$
0.0	-0.0004
0.5	-0.0014
0.75	-0.0034
0.90	-0.0099
0.95	-0.0217
0.97	-0.0386
0.99	-0.0683

Table 3-2: Measured mean estimates for a CN image of size 532x532

To overcome this problem of non-zero mean amplification due to digital IIR filtering, we forced the data arrays to be of zero mean at all points in the processing. This is accomplished by estimating the mean values $\hat{\mu}$ at various stages and then simply subtracting them from the data. This process resulted in a mean value of -3.3×10^{-7} (close to computer precision) for the CN image whereas it was -2.18×10^{-2} before this processing. This important check is now incorporated into our image generation software.

Once the high resolution (532x532) image containing CN, UCN and sub-pixel target is obtained, it is converted to a sensor image (64x64) using a blur function $b(x,y)$. This function $b(x,y)$ has a constant value 1 in the center and decreases monotonically in a Gaussian error function manner towards the edges. Such a blur function model accounts for finite aperture effects in many imaging systems[Hall].

For simplicity of analysis, we use a rectangular blur function model instead of the correct Gaussian function, i.e. we assume $b(x,y)$ to be 1 inside a square region of dimensions PxP and zero outside. Then the detector image $d(x,y)$ is obtained from the high resolution image $f(x,y)$ as below.

$$d(x,y) = \frac{1}{P^2}\sum_{i=1}^{P}\sum_{j=1}^{P} f(i+P(x-1), j+P(y-1)) \qquad (3.4)$$

Since the operation in eq.(3.4) is linear, $d(x,y)$ is also Gaussian and can be characterized using only first and second order moments. Since $f(x,y)$ is of zero mean, so is $d(x,y)$. It is instructive to analytically derive the second order statistics of $d(x,y)$.

$$\mathrm{E}\{d(x,y)\cdot d(x+\Delta x, y+\Delta y)\}$$

$$= \frac{1}{P^4}\sum_{i=1}^{P}\sum_{j=1}^{P}\sum_{k=1}^{P}\sum_{l=1}^{P} \mathrm{E}\{f(i+P(x-1), j+P(y-1))\cdot f(k+P(x+\Delta x-1), l+P(y+\Delta y-1))\}$$

$$= \frac{1}{P^4}\sum_{i=1}^{P}\sum_{j=1}^{P}\sum_{k=1}^{P}\sum_{l=1}^{P} [\sigma^2\cdot \rho_x^{|P\Delta x+k-i|}\cdot \rho_y^{|P\Delta y+l-j|}]$$

$$= \frac{\sigma^2}{P^4}\{\sum_{i=1}^{P}\sum_{k=1}^{P}\rho_x^{|P\Delta x+k-i|}\}\cdot\{\sum_{j=1}^{P}\sum_{l=1}^{P}\rho_y^{|P\Delta y+l-j|}\} \qquad (3.5)$$

To determine the variance of the detector image, we use $\Delta x = 0 = \Delta y$ in eq.(3.5) along with the fact the terms inside the double sums depend only on the difference in the indices to obtain the following.

$$\mathrm{Var}\{d(x,y)\} = \frac{\sigma^2}{P^4}\{P\cdot\sum_{k=-P}^{P}(1-\frac{|k|}{P})\rho_x^{|k|}\}\cdot\{P\cdot\sum_{k=-P}^{P}(1-\frac{|k|}{P})\rho_y^{|k|}\}$$

$$= \frac{\sigma^2}{P^2}\{\frac{P(1-\rho_x^2) - 2\rho_x(1-\rho_x^P)}{(1-\rho_x)^2}\}\cdot\{\frac{P(1-\rho_y^2) - 2\rho_y(1-\rho_y^P)}{(1-\rho_y)^2}\} \qquad (3.6)$$

This expression is used in Table 3-3 to show how the variance of the detector image changes as a function of detector size P and original CN image correlation coefficients. We see from this table that the variance decreases as the blur function size increases and as the original CN image becomes uncorrelated. We see that for $\rho = 0.95$ with $P = 8$, the detector image has a variance of 0.771 instead of one. This discrepancy is taken into account in evaluating the performance of our various algorithms for sub-pixel target detection and tracking. These analytical results were compared with calculated/measured estimates of variance of synthetic images and very good agreement (only 2%) error was observed.

Correlation		Detector Size P		
Coefficient ρ	2	4	8	12
0.0	0.250	0.063	0.016	0.007
0.5	0.563	0.266	0.098	0.049
0.75	0.766	0.525	0.289	0.178
0.90	0.903	0.776	0.598	0.471
0.95	0.951	0.882	0.771	0.679
0.97	0.970	0.927	0.855	0.791
0.99	0.990	0.975	0.949	0.924

Table 3-3: Theoretical detector image variance as a function of detector size P and CN image correlation coefficient $\rho = \rho_x = \rho_y$

Finally, analytical results are derived for the correlation coefficients ρ_x' and ρ_y' of the detector image $d(x,y)$. This is achieved by using $(\Delta x = 1, \Delta y = 0)$ and $(\Delta x = 0, \Delta y = 1)$ separately in eq.(3.5). After tedious, but straight forward algebra, we obtain

$$\rho_x' = \frac{\rho_x(1 - \rho_x^P)^2}{P(1 - \rho_x^2) - 2\rho_x(1 - \rho_x^P)}$$

and

$$\rho_y' = \frac{\rho_y(1 - \rho_y^P)^2}{P(1 - \rho_y^2) - 2\rho_y(1 - \rho_y^P)} \tag{3.7}$$

The analytical relations in eq.(3.7) are used in Table 3-4 to show how the correlation coefficients of the detector image $d(x,y)$ are affected by ρ and P. This clearly shows that the increasing P or decreasing the ρ value of original CN image leads to decrease in the detector image ρ values.

While the above theoretical analysis was carried out with the assumption of rectangular blur functions, experimental results indicate no significant differences in the estimates for Gaussian blur

	Blur Size P			
$\rho_x = \rho_y = \rho$	2	4	8	12
0.80	0.720	0.563	0.358	0.245
0.85	0.786	0.653	0.458	0.334
0.90	0.855	0.755	0.590	0.469
0.95	0.926	0.870	0.766	0.676
0.97	0.955	0.920	0.852	0.789
0.99	0.985	0.973	0.948	0.923

Table 3-4: Theoretical correlation coefficients in the detector image as a function of blur size P and original CN correlation coefficient ρ

functions. The various observations noted in this section are incorporated in our software to provided a unified framework for image synthesis.

3.2 SUB-PIXEL SHIFT ESTIMATION

An important aspect of our image sequence processing is the estimation of sub-pixel shift in the background CN images between successive frames. This shift is then used along with all interpolators to produce two aligned images. These two properly aligned images are then subtracted from each other to enhance the target and suppress the background. In our year 1 report, we investigated the use of 4 sub-pixel estimators, namely (i) gradient-based estimator, (ii) exponential model estimator, (iii) parabolic model estimator, (iv) Least Mean Squared (LMS) estimator. At that time, we showed through simulation that the exponential model based sub-pixel estimator performs best as this model matches precisely with the correlation function of the CN data. In this section, we present our result on the use of the two non-parametric methods (parabolic and exponential) on a more general data sequence.

Because of the ease with which we can control its bandwidth, duration and time bandwidth product, we have chosen a linear frequency modulation (LFM) signal for our investigation. The pulse compression ratio (PCR) of this LFM sequence is defined as the ratio of the uncompressed pulse width to the compressed pulse width, or the product of the pulse spectral bandwidth B and the uncompressed pulse width T. Thus, PCR is equal to the time bandwidth product. The sub-pixel delay estimates obtained for 3 different PCR values and 3 different sequence lengths are shown in Table 3-5.

PCR	Sequence Length	Estimator	Estimated Delay 0.1	0.2	0.4
15	300	Parabolic	0.071	0.142	0.284
		Exponential	0.062	0.132	0.306
	900	Parabolic	0.088	0.175	0.352
		Exponential	0.068	0.145	0.342
	1500	Parabolic	0.092	0.184	0.369
		Exponential	0.069	0.148	0.350
22.5	300	Parabolic	0.091	0.181	0.364
		Exponential	0.068	0.147	0.349
	900	Parabolic	0.096	0.192	0.386
		Exponential	0.069	0.150	0.358
	1500	Parabolic	0.097	0.194	0.391
		Exponential	0.070	0.151	0.359
37.5	300	Parabolic	0.094	0.188	0.381
		Exponential	0.067	0.146	0.356
	900	Parabolic	0.096	0.193	0.391
		Exponential	0.067	0.147	0.357
	1500	Parabolic	0.096	0.193	0.393
		Exponential	0.067	0.147	0.358

Table 3-5: Sub-pixel delay estimates for the LFM signal

One can see from Table 3-5 that increasing sequence length improves the estimation accuracy in general and increasing the PCR also improves the estimation accuracy. In general, the parabolic estimator seems to outperform the exponential estimator. The exponential estimator seems to perform better for large sub-pixel delays, short sequences and low PCRs. As will be seen in the next section, use of LFM signals enables us to observe the effect of estimator inaccuracies on the process performance without worrying about the interpolators. This is because, once the sub-pixel shift is estimated, it can be used in the analytical expression for LFM signal to obtain an ideally interpolated signal. With this analytically interpolated image, we observed background suppression of almost 50 dB (far better than observed with the synthetic images).

The estimated sub-pixel shifts for the synthetic images are shown in Table 3-6. We see from this table that the exponential estimator outperforms the parabolic one in all cases. This is because the synthetic images being generated have exponential correlation functions. Since such correlation structure can not always be guaranteed, it is decided to pursue both estimators in future.

	Estimated Shifts	
Exact Shift	Parabolic	Exponential
(0.25,-0.25)	(0.184,-0.174)	(0.227,-0.237)
(0.25, 0.25)	(0.186, 0.174)	(0.232, 0.233)
(-0.25,-0.25)	(-0.184,-0.173)	(-0.219,-0.238)
(-0.25, 0.25)	(-0.181, 0.171)	(-0.234, 0.239)

Table 3-6: Sub-pixel shift estimates for the synthetic CN imagery

3.3 INTERPOLATOR SELECTION

After the sub-pixel shift between two successive frames is estimated, we have to interpolate one of the two image frames to align it with other. We will denote the two detector image frames by $d_1(x,y)$ and $d_2(x,y)$ and we denote the interpolated image 1 by $\hat{d}_1(x,y)$. Then the performance of the interpolation is estimated by the following measure known as the Background Suppression Ratio (BSR)

$$BSR = 10 \cdot \log \frac{\mathrm{Var}\{d_i(x,y)\}}{\mathrm{Var}\{d_2(x,y) - \hat{d}_1(x,y)\}} \tag{3.8}$$

This BSR measure is useful in evaluating the performance of the estimators and interpolators separately.

The objective of the interpolators is to produce $\hat{d}_1(x,y)$ which is a shifted version of $d(x,y)$, namely

$$\hat{d}_1(x,y) = d_1(x+\Delta x, y+\Delta y) \tag{3.9}$$

where Δx and Δy denote the shifts in x and y directions. We consider several interpolator schemes to be discussed below.

The 2-D linear interpolator estimates the value $d_i(x+\Delta x, y+\Delta y)$ from its 4 nearest neighbors as below.

$$\hat{d}_1(x,y) = (1-\Delta x)(1-\Delta y)d_i(x,y) - \Delta x(1-\Delta y)d_i(x,y+1) +$$

$$\Delta y(1-\Delta x)d_i(x+1,y) - \Delta x\Delta y d_i(x+1,y+1) \tag{3.10}$$

where $\Delta x \geq 0$ and $\Delta y \geq 0$. The correctness of eq (3.10) can be easily seen by using $\Delta x = 0 = \Delta y$ which yields $\hat{d}_i(x,y) = d_i(x,y)$.

The 2-D quadratic interpolator uses a 3×3 array of values in $d_1(x,y)$ as below to yield the estimate $\hat{d}_1(x,y)$.

$$\hat{d}_1(x,y) = [Y][A(x,y)][X]$$

where

$$[X] = [0.5\Delta x(\Delta x - 1) \quad (1 - \Delta x^2) \quad 0.5\Delta x(\Delta x + 1)]^T$$

$$[Y] = [0.5\Delta y(\Delta y - 1) \quad (1 - \Delta y^2) \quad 0.5\Delta y(\Delta y + 1)]$$

and

$$[A(x,y)] = \begin{bmatrix} d_1(x-1,y-1) & d_1(x-1,y) & d_1(x-1,y+1) \\ d_1(x,y-1) & d_1(x,y) & d_1(x,y+1) \\ d_1(x+1,y-1) & d_1(x+1,y) & d_1(x+1,y+1) \end{bmatrix} \qquad (3.11)$$

The cubic spline interpolators are discussed thoroughly elsewhere [Hou and Andrew] and it is probably sufficient to point out the fact these are based on local piecewise polynomial fit to the available data. We carry out this cubic spline interpolation by using IMSL software. One can easily see that the computational complexity of the interpolators increases as we go from linear interpolation to quadratic method to cubic spline based method.

We can analytically predict the BSR to be observed. The numerator of eq.(3.8), namely, Var $\{d_2(x,y)\}$ is given by σ^2 whereas the denominator of eq.(3.8) is as below. We make the assumption of perfect interpolation. Then,

$$\begin{aligned} \text{Var}\{d_2(x,y) - \hat{d}_1(x,y)\} &= 2\sigma^2 - 2 \cdot \text{Cov}\{d_2(x,y), \hat{d}_1(x,y)\} \\ &= 2\sigma^2 - 2 \cdot \text{E}\{d(x,y) \cdot d(x+\Delta x, y+\Delta y)\} \end{aligned} \qquad (3.12)$$

For the detector image $d(x,y)$, the required covariance can be shown to be given as

$$\mathrm{Cov}\,\{d(x,y),d(x+\Delta x,y+\Delta y)\}$$

$$=\sigma^2\,\frac{(P+\Delta x)(1-\rho_x^2)-2\rho_x^{(1-\Delta x)}+\rho_x^{(P+1)}(\rho_x^{\Delta x}+\rho_x^{-\Delta x})}{P(1-\rho_x^2)-2\rho_x(1-\rho_x^P)}$$

$$\bullet\,\frac{(P+\Delta y)(1-\rho_y^2)-2\rho_y^{(1-\Delta y)}+\rho_y^{(P+1)}(\rho_y^{\Delta y}+\rho_y^{-\Delta y})}{P(1-\rho_y^2)-2\rho_y(1-\rho_y^P)}\,. \tag{3.13}$$

The analytically derived cross covariances for $\sigma = 1$ (equivalent to detector image correlation coefficients) are shown in Table 3-7 for $P = 8$ and $\rho_x = \rho_y = 0.95$. Note from this table that the correlation coefficient of the covariance value (not ρ) changes from a maximum of 1.0 (when the two images are perfectly aligned) to a minimum of 0.784 (when one image is shifted by 0.5 pixels in each direction with respect to the other).

Vertical-	Horizontal Sub-pixel shift								
	-0.500	-0.375	-0.250	-0.125	0.000	0.125	0.250	0.375	0.500
-0.500	0.850	0.879	0.902	0.917	0.922	0.917	0.898	0.866	0.817
-0.375	0.879	0.909	0.932	0.948	0.953	0.948	0.929	0.895	0.844
-0.250	0.902	0.932	0.956	0.972	0.978	0.972	0.953	0.918	0.866
-0.125	0.917	0.948	0.972	0.988	0.994	0.988	0.968	0.933	0.880
0.000	0.922	0.953	0.978	0.994	1.000	0.994	0.974	0.939	0.886
0.125	0.917	0.948	0.972	0.988	0.994	0.988	0.968	0.933	0.880
0.250	0.898	0.929	0.953	0.968	0.974	0.968	0.949	0.914	0.863
0.375	0.866	0.895	0.918	0.933	0.939	0.933	0.914	0.881	0.831
0.500	0.817	0.844	0.866	0.880	0.886	0.880	0.863	0.831	0.784

Table 3-7: Correlation coefficient of Cov (not ρ) comparison between two detector images with different sub-pixel shifts

For a sub-pixel shift of (0.25,-0.25), Table 3-7 indicates that the two image frames have a correlation coefficient of 0.953 yielding a variance in the difference image of 0.069 according to eq.(3.12). The experimentally observed variance of the difference image is 0.061 agreeing well with our theoretical results. Then the limit on BSR achievable seems to be more fundamental than the simple interpolation problems. Based on this, the simple 2-D linear interpolator seems to be our best choice as it needs the minimum computational complexity.

3.4 DETECTOR LIMITATIONS

Our previous simulations do not take into account the fact that the correlation plane detector needed for sub-pixel shift estimation suffer from nonidealities such as limited dynamic range (DR), detector noise and detector area. Vijaya Kumar et.al. [Kumar] have previously investigated the effects of finite detector area on parabolic sub-pixel shift estimators and have shown that these introduce small biases in the estimated shifts. These sub-pixel shift estimators use the central 5x5 region of the correlation plane and the acceptable DR limitations on the detector (about 30-50 dB) seem to pose no problems in accurately detecting these correlation values. This is because of the large correlation coefficients of the CN part of the image.

To observe the effect of detector noise on sub-pixel shift estimators, we added uniformly distributed random numbers to the central 5x5 region correlation plane values. The variance of this uniformly distributed numbers is chosen such that signal to noise ratios (SNRs) of 20, 30, 40 and 50 dB are obtained in the detector plane. The sub-pixel shift estimates for various SNRs are shown in Table 3-8.

		Estimated Shifts	
Correct Shift	SNR(dB)	Parabolic	Exponential
(0.25,-0.25)	20	(0.183,-0.173)	(0.226,-0.235)
	30	(0.184,-0.173)	(0.226,-0.236)
	40	(0.184,-0.174)	(0.227,-0.237)
	50	(0.184,-0.174)	(0.227,-0.237)
(0.25, 0.25)	20	(0.186, 0.172)	(0.231, 0.233)
	30	(0.186, 0.171)	(0.232, 0.233)
	40	(0.186, 0.171)	(0.232, 0.233)
	50	(0.186, 0.171)	(0.232, 0.233)
(-0.25,-0.25)	20	(-0.183,-0.171)	(-0.222,-0.236)
	30	(-0.182,-0.173)	(-0.220,-0.238)
	40	(-0.182,-0.173)	(-0.219,-0.238)
	50	(-0.182,-0.173)	(-0.219,-0.238)

Table 3-8: Sub-pixel shift estimates for various detector plane SNRs

Comparing Tables 3-6 and 3-8, we see that SNRs higher than 40 dB have very little effect on the estimated sub-pixel shifts. Thus SNRs of 40 dB are required in the correlation plane.

3.5 DOUBLE DIFFERENCING

The approach to background suppression discussed so far has been to estimate the sub-pixel shift followed by a first difference operation. In this section, we present some preliminary results indicating the role of double differencing for target detection and tracking.

Let $d_1(x,y)$, $d_2(x,y)$ and $d_3(x,y)$ represent three successive image frames in which the target and the background are moving at different velocities. The single difference image is given by

$$\hat{d}(x,y) = |d_1(x,y) - d_2(x,y)| \tag{3.14}$$

whereas the double differencing yields

$$\bar{d}(x,y) = |0.5d_1(x,y) - d_2(x,y) + 0.5d_3(x,y)| \tag{3.15}$$

In Fig. 3-1, we show the results obtained by these processes on three images. The top three images in the figure are three detector images d_1, d_2 and d_3. The CN background in d_2 is shifted by (0.25,0.25) with respect to the background in d_1, and the CN background in d_3 is shifted by (0.25,0.25) with respect to the background in d_2. The target in all three detector images is of size equal to one detector plane pixel and is of uniform intensity of 2, while the variance of CN background is approximately one. The target is moving at a constant velocity with a horizontal shift of 3 detector pixels and a vertical shift of 2 detector pixels between each adjacent frames. We see from the two single difference images in the second row of this figure that the background is not suppressed completely. On the other hand, double differencing result shown in the last row of this figure displays a clear track of the target movement. One should keep in mind that the images in Fig. 3-1 are thresholded optimally and thus may not convey the complicated nature of the processing.

The results presented in this section are only preliminary. Some year 3 effort will be devoted to analyzing the capabilities of this double differencing. The relevant issues include the BSR achievable, the quantization effects, the resulting frequency responses and the need or lack of need for interpolation. Fruitful research in this direction is anticipated for year 3.

3.6 SPACE/TIME FILTERING

The single differencing and double differencing approaches discussed earlier represent two special cases of a more general philosophy of target detection and tracking known as "space/time filtering". To understand this, we consider the various image frames available as samples of a 3-D function $f(x,y,t)$. The sampling intervals Δx and Δy denote the spatial sampling according to the detector size. This 3-D function can be modeled as

$$f(x,y,t) = s(x,y,t) + CN(x,y,t) - UCN(x,y,t) \tag{3.16}$$

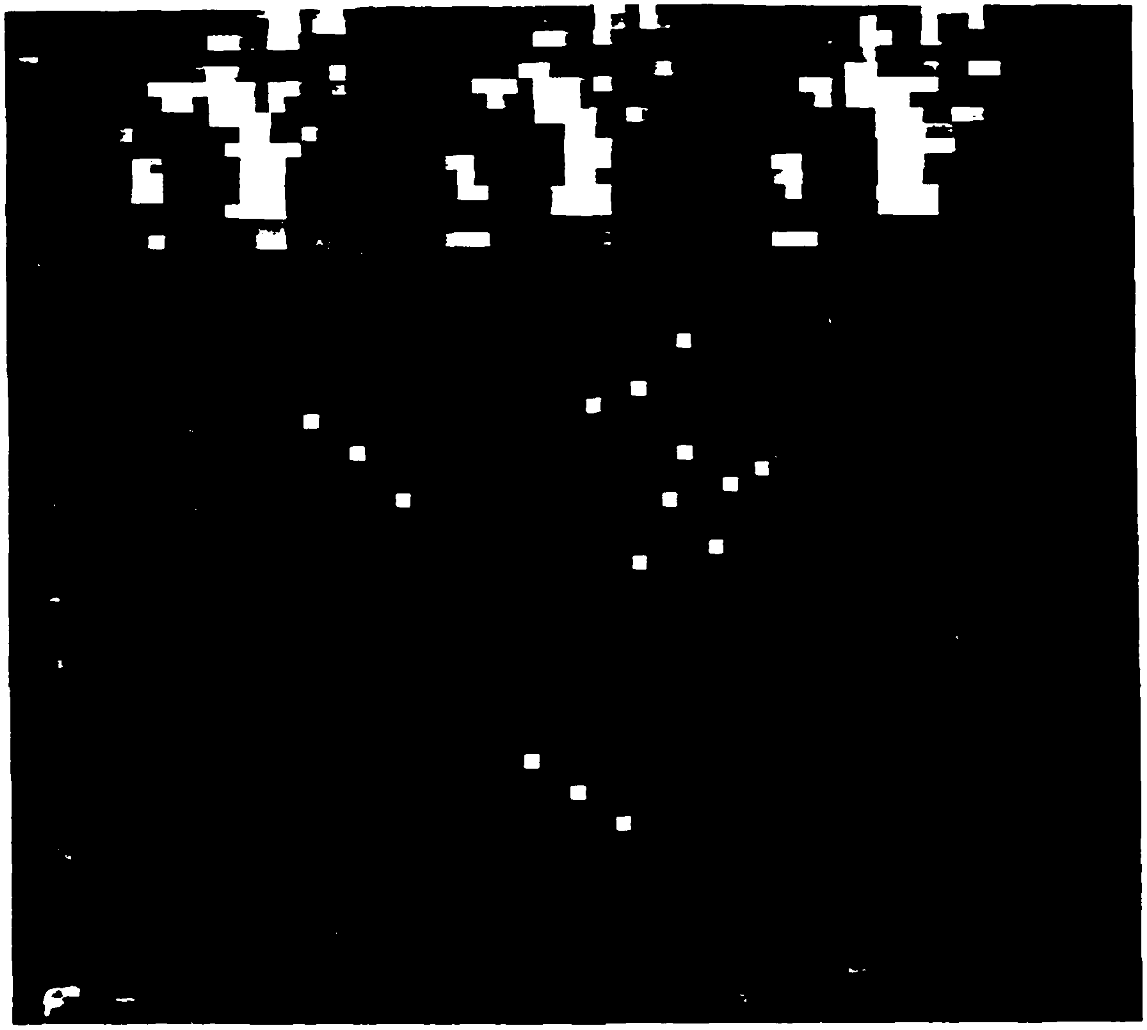


Figure 3-1: Single difference and double difference images of an image sequence

where $s(x,y,t)$ denotes the sub-pixel target, $CN(x,y,t)$ denotes the correlated noise and $UCN(x,y,t)$ denotes the uncorrelated noise. The goal of the space/time filtering is to process the sampled 3-D function to enhance the target $s(x,y,t)$ while suppressing the remaining terms in eq.(3.16).

The target $s(x,y,t)$ can be modeled as a thin straight line in the 3-D space with the dimensions of this line in x and y axis being sub-pixel in nature. The $CN(x,y,t)$ varies slowly in x and y and shows linear shift in t. On the other hand, the uncorrelated noise $UCN(x,y,t)$ is completely random and is characterized by high frequencies. By observing the 3-D spectra of the three components in eq.(3.16), we plan to derive an optimal space/time filtering scheme for the 3-D sequence $f(x,y,t)$. Issues to be resolved in this connection are sampling effects, 'optimal' filters, computation complexities and computationally efficient (sub-optimal) filters.

3.7 FUTURE WORK

Our year 3 effort will focus on better understanding of the general techniques presented here. In addition to this, we will improve our image generation software to incorporate multi-region image generation. We will also explore the optical interpolation methods. Other advanced sub-pixel shift estimators such as maximum-likelihood and maximum a posteriori will be considered.

REFERENCES

[IMSL 1982]
International Mathematics and Statistics Library User Manual, Version 9. IMSL Inc., 7500 Bellaire Blvd., TX 77036, 1982.

[Rauch 1981]
H.E. Rauch, W.I. Futterman and D.B. Kemmer, "Background suppression and tracking with a staring mosaic sensor", Optical Engineering, Vol.21, No.1, 103-110, 1981.

[Papoulis]
A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, New York, 1965.

[Hall]
E.L. Hall, Computer Image Processing and Recognition, Academic Press, New York, 1979.

[Hou and Andrews]
H.S. Hou and H.C. Andrews, "Cubic splines for image interpolation and digital filtering", IEEE Trans. ASSP, Vol.26, No.6, 508-517, 1978.

[Kumar]
B.V.K. Vijaya Kumar, D. Casasent and A. Goutzoulis, "Fine delay estimation with time integrating correlation", Applied Optics, Vol.21, 3855-63, 1982.

4. MODEL-BASED ALGORITHMS FOR HYBRID DIGITAL/OPTICAL PROCESSING

4.1 SUMMARY

The objective of this research effort is to develop algorithms for representation and interpretation of space-based images which are well-suited to hybrid digital/optical implementation.

In the first phase of this program we have developed a multiresolution rotation- invariant (MRI) operator which may be used to extract structural features as well as characterize textures using statistical measures. Experiments in texture classification have shown that the MRI operator is a useful representation of texture properties and provides classification independent of rotation and scale. Probabilistic graph matching was used to demonstrate matching between attributed graph representation of structural image elements. The operators we have described are well-suited for optical implementation, and the matching of representations derived from these operators is suited for implementation on a hybrid digital/optical system. Evaluation of these algorithms and their hybrid system implementation will be carried out through simulation on the RAPIDbus II system. Further refinements of the high-speed RAPIDbus architecture would support a hybrid digital/ optical interface when available.

These approaches may be integrated into a recognition framework based on recursive model matching in which composite MRI kernels are generated adaptively based on hypothesis formation in a model-based setting. Recursive model matching is intended to explore the capabilities of a highly interactive hybrid digital/optical system which utilizes digital hardware to generate hypotheses in a knowledge- based environment and uses optical hardware to explore and validate hypotheses using convolution-based adaptive feature extraction mechanisms.

4.2 RAPIDBUS ARCHITECTURE

One important aspect of the integrated image analysis system sought by this project is the hardware and programming environment. Contemporary environments were not designed to couple a high bandwidth electro-optic processor with digital processors doing numeric and symbolic calculation. The RapidBus II prototype, being developed for this project, provides both a near-term execution environment, and a longer term opportunity to develop new architectural concepts oriented toward the needs of an integrated image analysis system.

The RapidBus II prototype is currently in the assembly and testing stage. Design documentation is being completed using our enhanced SCALD II CAD system. Many components and subassemblies have arrived and are waiting for integration and test. Over the coming few months we anticipate testing to progress through a two, six, and finally twelve processor stage. Through the donation of a PCB design system from IBM, our CAD system is being enhanced to carry the design through multi-layer PCB film.

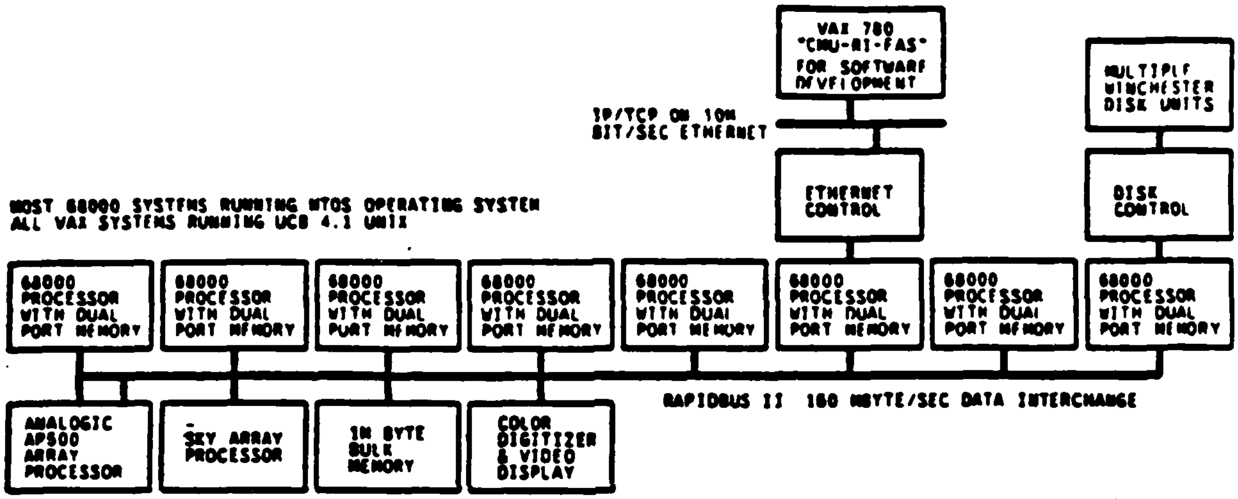

VAX 780
"CMU-RI-FAS"
FOR SOFTWARE DEVELOPMENT
MULTIPLE WINCHESTER DISK UNITS
IP/TCP ON 10M BIT/SEC ETHERNET
ETHERNET CONTROL
DISK CONTROL
MOST 68000 SYSTEMS RUNNING MTOS OPERATING SYSTEM
ALL VAX SYSTEMS RUNNING UCB 4.1 UNIX
68000 PROCESSOR WITH DUAL PORT MEMORY
68000 PROCESSOR WITH DUAL PORT MEMORY
68000 PROCESSOR WITH DUAL PORT MEMORY
68000 PROCESSOR WITH DUAL PORT MEMORY
68000 PROCESSOR WITH DUAL PORT MEMORY
68000 PROCESSOR WITH DUAL PORT MEMORY
68000 PROCESSOR WITH DUAL PORT MEMORY
68000 PROCESSOR WITH DUAL PORT MEMORY
RAPIDBUS II 160 MBYTE/SEC DATA INTERCHANGE
ANALOGIC AP500 ARRAY PROCESSOR
SKY ARRAY PROCESSOR
1M BYTE BULK MEMORY
COLOR DIGITIZER & VIDEO DISPLAY

Figure 4-1: RapidBus II prototype multiprocessor configuration.

Software development is proceeding in parallel with hardware development using both a multiple processor Versabus system, and a stand-alone RapidBus II node (IBM CS-9000). The host development system is functional for C and assembly code. The target operating system is operating on a single node, and will soon be expanded to a dual-processor Versabus system. An outside group is doing parallel development of a multiple-processor Franz-Lisp system in return for a duplicate RapidBus II.

Design concepts for an advanced machine are emerging from the RapidBus II effort. Under the system name *RapidGraph*, a new high speed interchange, multi-programmed processor, and object support node have been developed. The interchange provides message passing at multi-gigabyte / sec rates using a small number of bipolar VLSI devices. The processor is designed to handle digital computation using an "object-flow" model to integrate both symbolic and numeric computation. The object support node simplifies the design of large, highly parallel software systems through the encapsulation of objects at the memory rather than through the processor instruction stream.

4.3 Probabilistic Graph Matching

In a variety of image processing problems, the data contains stereotyped subpatterns which are well-described by symbolic representations. Such symbolic representations include graph, grammar, and automata models. While these models are very useful when subpatterns are highly invariant to image variability, symbolic search and manipulation techniques become very complex when symbol correspondence becomes uncertain. Symbolic representations may be enhanced in two respects which increase their applicability to real data. First, stochastic structures may be used to associate outcome probabilities with structural relations of the model. Second, attributed structures offer a rich class of models where subpatterns or symbols have associated features or attribute values. Such attributed structural models pose many difficult methodological issues for implementation. In this study we have addressed problems of the dichotomy between symbolic and statistical information and its effect on the choice of symbol primitives, issues of structural observability, structural matching, assumptions of component independence, and identification of structural transformations. These issues will be discussed in papers and reports now in preparation.

An attributed random graph model consists of a 4-tuple $R = (V,\alpha,E,\beta)$ where:

1. the *random vertex set* $V = \{ V , i = 1,\ldots,n\}$, where each V is a random variable called the random vertex.

2. the *random edge set* $E = \{ E , i = 1,\ldots,n, j = 1,\ldots,n\}$ in V X V where each E is a random variable called the random edge.

3. the *random vertex attribute set* $\alpha = \{\alpha_i, i = 1,\ldots,n\}$ where each α is a random variable with possible outcomes {a}.

4. the *random edge attribute set* $\beta = \{\beta_i , i = 1,\ldots,n, j = 1,\ldots,n\}$ where each β is a random variable with possible outcomes {b}.

5. Each outcome of $R = (V,\alpha,E,\beta)$ is an attributed graph $H = (v,a,e,b)$ with probability P(H) = Prob $\{V = v,\alpha = a,E = e,\beta = b\}$ such that

   - $P(H) = 0$ for all $H.\varepsilon.\Gamma$,

   - $\Sigma_\Gamma P(H) = 1$, where Γ is the range of R.

   P(H) is the *probability distribution* of R.

The attributed random graph model defined above provides a basis for the definition of likelihood functions over the observed outcomes from the class of graphs and the attribute set A. The likelihood of an observed outcome may be used as a basis for the matching and recognition of patterns in the image. In this application the structural elements of the image are associated with vertex and edge symbols of the model, and both structural relations and quantitative properties of the elements are retained in the model. Image components such as vertices, edges, regions, or intensity peaks may be used as structural elements. In the resulting probabilistic model, each element has some outcome probability, some observation probability, and some probability density of attribute values.

A simple example of a graph representation derived from a gray level image of a polyhedron is shown in figure 4-2. A line drawing of the original image is shown in figure 4-2a. The graph structure extracted from a single observation is shown in figure 4-2b where graph vertices have been attached to structural corner elements of the original image and graph edges have been attached to edge elements of the original image. An ensemble of observations such as that in figure 4-2b is used to derive a probabilistic graph model such as that shown in figure 4-2c. In figure 4-2c, the probability distribution of positions of the vertices vare indicated by circles. The probability distribution of vertex angle attributes is indicated by $p(\theta)$.

Probabilistic graph matching may be used for matching of images and recognition of objects in images using likelihood criteria as a basis for search correspondence trees. The likelihood of any observed graph, subgraph, or structural element may be computed and used for estimation or decision making. Such problems incorporate three phases: (1) correspondence matching of graph elements, (2) rigid graph pose estimation, and (3) likelihood calculation. The probabilistic graph model uses pose independent likelihoods to hypothesize correspondence, then estimate pose. The use of attributes to guide correspondence matching, and the use of observation probabilities to structure the search results in simplified and reliable algorithms.

We have applied the probabilistic graph matching approach to two types of image representation. Graphs which are derived from edge, corner, and junction components of gray level images are useful for description and matching of objects. In this case, attributes include lengths, angles, and positions of elements. The resulting graph models have been used to classify objects, inspect objects, and determine orientation of objects in scenes where edge information is a reliable clue. An example of matching likelihoods between a model graph and various distorted observation graphs including partial views is shown in figure 4-3. The likelihood is a measure of the correspondence between the two structures in each case. A similar approach may be used to track movement of the model object by matching successive views and computing the pose changes between views. Such an example is shown in figure 4-4 for the same object used in the previous examples.

Figure 4-2: Probabilistic graph derived from an ensemble of gray-level images.

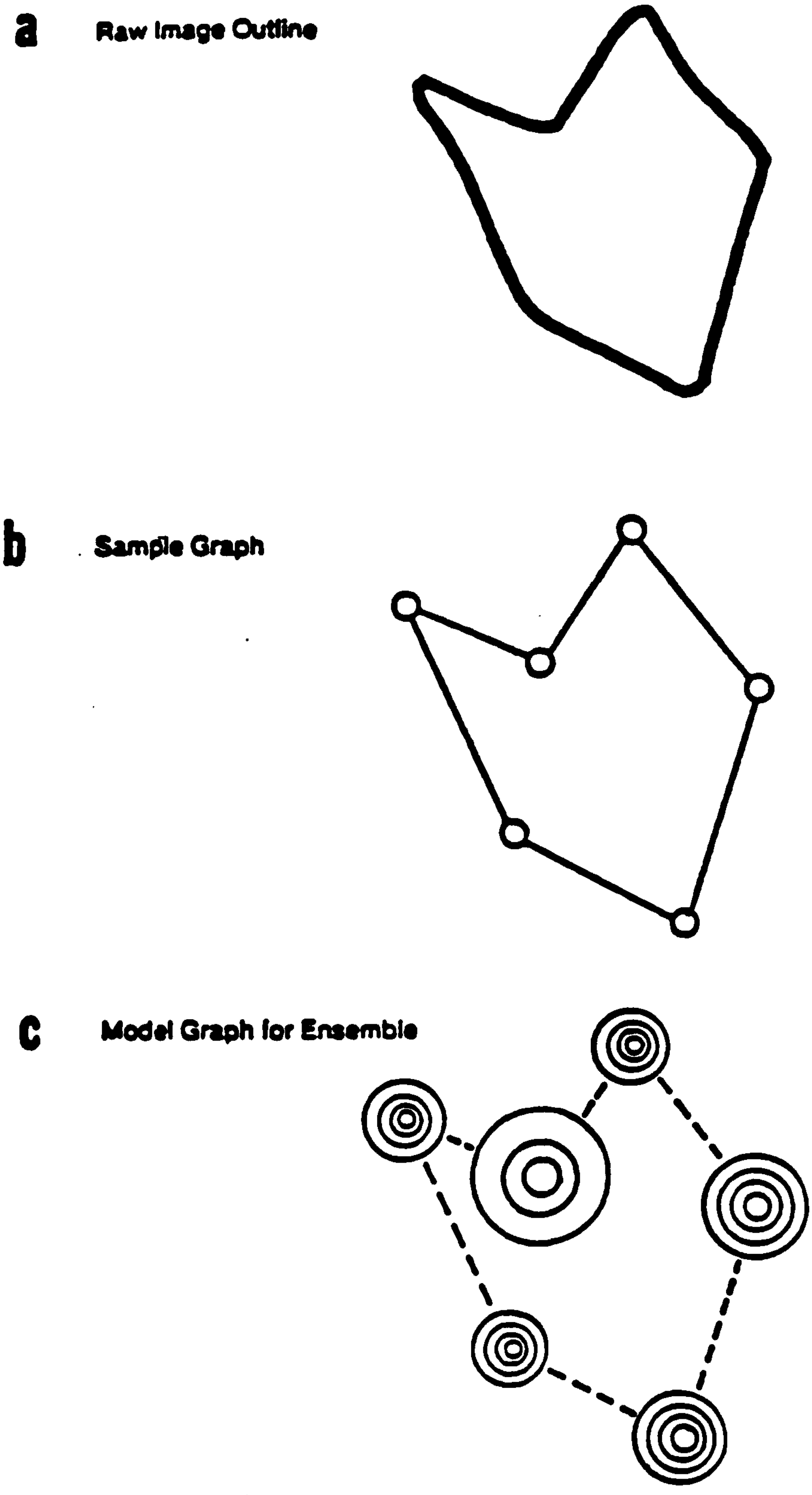
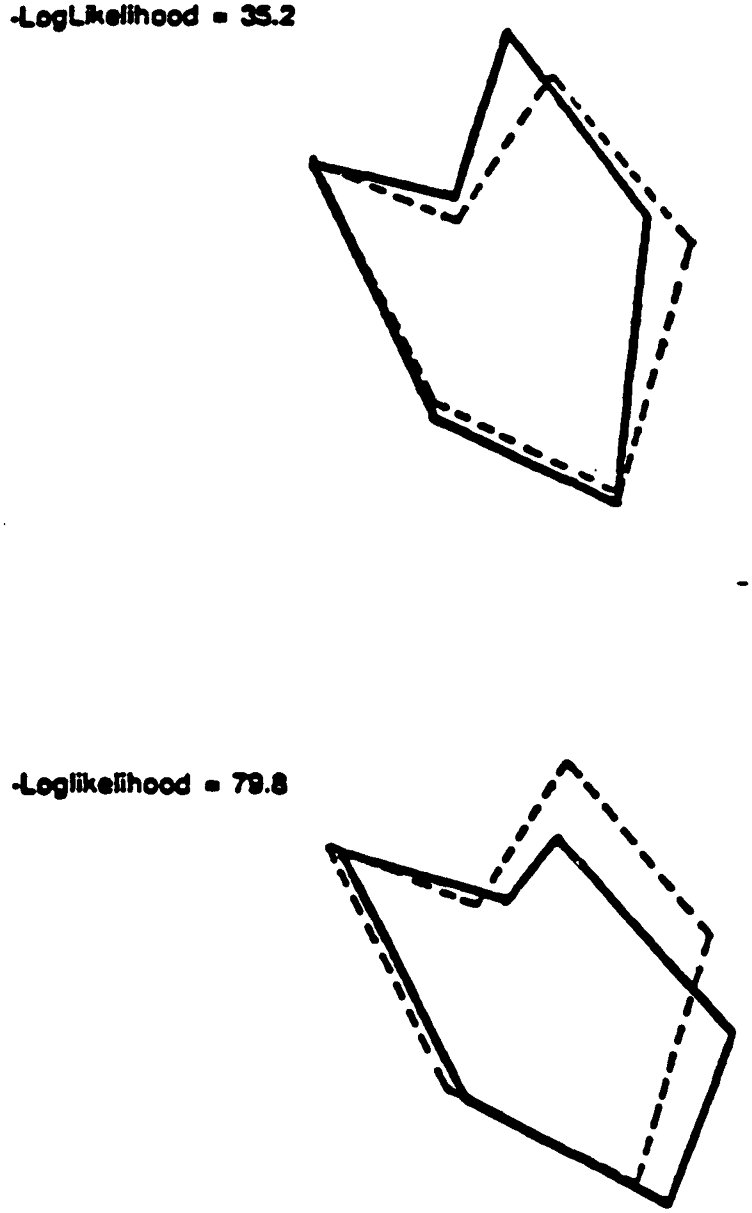


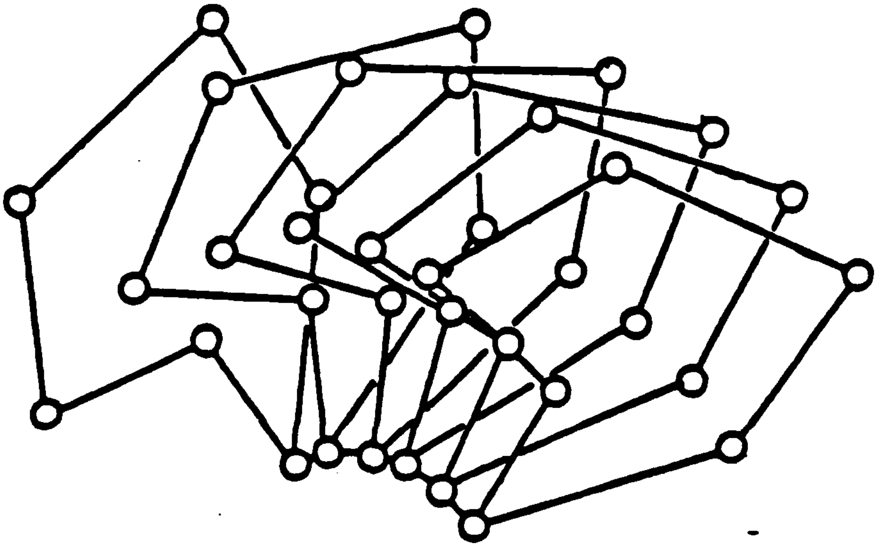
Figure 4-3: Matching graphs between a model and distorted observations with associated likelihoods.



-LogLikelihood = 35.2

-Loglikelihood = 79.8

Figure 4-4: Tracking movement of an object using successive likelihood matches with a probabilistic graph model.



As a second example, we have used probabilistic graph models for matching of multiresolution tree structures derived from gray-level images. The derivation of such a multiresolution tree using the difference of low-pass transform is described in the next section. In this case, the matching algorithm is formulated as a hierarchical tree search using likelihoods to guide correspondence matching at each level of the tree. Results of this approach are described in [Crowley and Sanderson 84].

Probabilistic graph models provide a general approach to many practical matching and recognition problems where a training set of images of a priori knowledge of the probability structure is assumed. In this way, object model-based information may be incorporated using a priori probability structures. Many previous approaches to matching and pose estimation may be considered subsets of the probabilistic matching approach in which edge or region attributes of images are related by heuristic similarity measures rather than likelihoods.

4.4 MRI Operators for Shape Representation

Description of gray-scale shape in images is complex because shapes are often defined by some combination of region information and edge information. From the point of view of image processing, region information is often contained in the lower spatial frequency components of the image, while edge information is contained in the higher spatial frequency components. A complete description is difficult to achieve therefore from extraction of structural elements at one resolution level. A number of techniques have been proposed which transform the two-dimensional gray-level image to a three-space representation in (x,y,k) space, where k is the parameter of the resolution space. Such a representation has the advantage that peak structures in the (x,y,k) space shift uniformly along the k axis under scale transformations, and therefore objects of different size may be recognized in a representation with the same structural relationships. In this section we describe an extension of the multiple resolution tree which incorporates structural basis functions at each level in order to provide a more complete description at each level and include orientation specific structural components at each resolution level.

Our preliminary work on representation and probabilistic matching of multiple resolution structures has been carried out using a reversible transform called the "difference of low-pass" or DOLP transform developed by Crowley [Crowley 81, Crowley and Stern 84, Crowley and Parker 84]. The DOLP transform expands a single gray-level image f(x,y) into a set of bandpass images b(x,y,k), where k is the index of the multiple resolution tree. Each bandpass image is obtained by convolution of the original image with an appropriate bandpass impulse response function h(x,y,k). In the implementation of this transform by Crowley [Crowley 81], the bandpass impulse responses are constructed out of difference of Gaussian kernels. The most efficient implementation utilizes properties of the Gaussian kernel function which permits resampling and cascaded convolution with expansion and reduces the sequential computation of this transform from O(N..) multiplies and additions to O(N).

The DOLP transform set itself is not a very efficient representation of the image since it requires expanded storage space in a normal digital representation. However, it is possible to extract a symbolic representation of important information from the DOLP transform using peaks of the transform arrays as key structural elements of the image. In [Crowley and Sanderson 84], we introduced two levels of symbolic representation. The first level is composed of symbols derived directly from the DOLP images based on local positive maxima or negative minima in one, two, or three dimensions of the DOLP space representation. The second level of symbols utilize the connectivity among peaks and ridges to form peak paths and ridge paths. These symbol structures are defined in detail in [Crowley and Sanderson 84].

The advantage of the multiresolution representation techniques is the ability to describe both high resolution and low resolution structural image features in the same representation. The disadvantages of the bandpass filter approach are the difficulty in describing complex shapes, particularly those involving oriented components and the current demands of the computation to compute such extensive filtering operations. In order to enrich the capabilities of the multiple resolution transform, we have introduced a set of basis functions at each resolution level which includes oriented two-dimensional basis functions. This set of structural basis functions provides a much more complete description of the image at each level, at the expense of redundant information and increased computational load. In the context of this project we propose to explore the implementation of such techniques using parallel and optical processors. In this context the basis function tree provides a richer source of information for matching and interpretation which may be searched interactively rather than exhaustively computed.

Multiresolution representations provide a basis for searching an image database with respect to size independent features. Our experience with the DOLP transform described in section 4.2.4 has suggested that configurations of DOLP peaks are in fact good descriptors of image structures, but that as the resolution is increased such configurations become exceedingly complex. In addition, lighting conditions, shadows, and background variations may cause significant distortions of the DOLP representation which are difficult to interpret due to the lack of more specific orientation and structure information in the DOLP transform itself. The transform itself accentuates symmetrical contrasting regions and is useful for locating regions of interest, but may not be very efficient in describing complex structures.

We have introduced the MRI (multiresolution rotation invariant) operator and the MRD (multiresolution difference) transform in order to derive a more efficient representation of complex structure as well as texture. The MRI operator of order n and resolution k is defined by:

$$\bar{h}_k^{(n)}(r,\sigma) = h_k^{(n)}(r)\, e^{jn\phi}, \qquad (1)$$

$$h_k^{(n)}(r) = \frac{r^n}{2\pi k\sigma^2}\, e^{-r^2/2k\sigma^2} \qquad (2)$$

where (r,φ) are polar coordinates of the operator space. The MRI operator is rotation invariant in the sense that the magnitude of the response is independent of the orientation of a directional component of the input image.

The significance of the MRI operators may be seen by examining their projection p(x) along any single radial axis. These operators of order n have the following interpretation:

- $n = 0$: Point Detector
- $n = 1$: Edge Detector
- $n = 2$: Line Detector
- $n>2$: Higher order ripple detectors

Each of the complex operators defined in this manner will have magnitude of response related to the magnitude of that feature, and angle of the response related to the orientation of that feature. In addition, the detector masks for different orders are orthogonal, and therefore energy is distributed independently among the features. The Fourier spectra of these MRI operators show that $n = 0$ corresponds to a low-pass filter, while operators of increasingly higher order n correspond to band-pass filters of decreasing bandwidth.

The MRD transform is defined by:

$$\bar{d}_k^{(n)} = \bar{h}_k^{(n)} - \bar{h}_{k+1}^{(n)}, \quad k=1, \ldots, N \tag{6}$$

$$= [-\frac{r^n}{2\pi\sigma^2}] \; [\; \frac{1}{k} \, e^{-r^2/2k\sigma^2} - \frac{1}{k+1} \, e^{-r^2/2(k+1)\sigma^2}] \, e^{jn\phi} . \tag{7}$$

Based on this definition the MRD of order 0 is just DOLP transform. The MRD terms of higher order also provide a reversible decomposition of the gradient image of order n. That is, the MRD transform of order n of an image is sufficient to reconstruct the nth order gradient of the same image, but not necessarily the original image itself.

The MRI operators provide a basis for multiresolution decomposition of an image. The resulting multiresolution representation may then be interpreted as the response of a set of orthogonal feature detectors and searched for significant response regions which will characterize the structure in the image. Unlike peaks in the DOLP transform space which do not carry orientation information, magnitude peaks in the MRI space may be associated with the angle response to provide important structural clues. In the proposed research program we will implement and evaluate the MRI operators as tools for the representation and detection of structural features in aerial images of airports and harbors.

The interpretation of the MRI operators in the Fourier spectrum may be related to the performance of the texture energy measures reviewed above. The texture energy measures provide statistical information about the sampled local two-dimensional spectrum of the image at some resolution level. Statistical summary information from the MRI operator space includes distribution estimates of orientation from multiple operators as well as magnitude information. In the proposed research we will implement and evaluate the MRI operators as tools for the description and segmentation of textured regions in aerial images of airports and harbors.

A number of extensions to the MRI operators and MRD transform have been investigated. The *shifted- Gaussian MRI* operator seems to provide significant computational advantages for digital implementation although it is an approximation to the rotational invariance property of the MRI itself.

The aerial imagery being examined is often obtained in a *multispectral* format, and spectral contrast is often a useful clue to structural and textural features. Extensions of the MRI operators in which the phase space is mapped into a spectral domain seems to be a feasible extension of the concepts to provide representation of color features. Initially, application of the SG-MRI operator with contrasting color coordinates for the component Gaussians will be examined.

The SG-MRI operator also has a direct extension to the detection of *temporal shift* of structural features. By associating Gaussian components with different time frames, magnitude and orientation of time shifts between frames can be obtained. Initially, we propose to develop these concepts in a space-time frequency domain framework and study the tuning of operators to various types of feature shifts.

The operators and extensions described above are all well-suited to optical implementation, but for our studies this functionality will be simulated using the RAPIDbus II architecture. This implementation provides the basis to examine the use of these operators in a system where interactive search and adaptive tuning of the masks is a significant property of the algorithms.

The recursive model matching strategy described in the section 3.6 requires interactive tuning of correlation masks. The masks used in those studies will be derived from the set described here and their extensions. A great many possible *composite MRI operators* could be defined based on the primary set, and these could play a useful role in the recursive strategy.

Figures 4-5-4-8 show the effects of changing one parameter at a time on the mask shapes of the MRI operators of the real and imaginary planes and their corresponding magnitude and phase planes.

Fig. 4-5 demonstrates the effect of changing the size of the mask. It is important that the mask size chosen be large enough so that the values at the edges of the mask are near zero. Failure to do this will mean that the mask is not symmetric and the property of rotational invariance will no longer be valid. As can be seen, changing the size of the mask does nothing to the generated kernel itself; it only affects the extent of the kernel that will be included in the mask.

Fig. 4-6 shows the effect of varying the order of the masks created while keeping the remaining parameters the same. The value of n is equal to half the number of zero-crossings encountered in either the real or the imaginary planes as a contour at a fixed radius $R > 0$ is followed for 2π radians. $n = 0$ is a low-pass filter, $n = 1$ is an edge detector, $n = 2$ is a line detector, $n = 3,4,5,...$ are higher order ripple detectors. Note that the radius of the maximum magnitude of the mask pair increases linearly as n increases. This holds true only when $\sigma * k$ is a constant.

Figs. 4-7 & 4-8 show the effects of varying σ and k respectively. These two parameters always appear together in the equation above. Together, they form the term σˋk which can be considered to be the "variance" of the gaussian filter. Increasing either will make the spread of the gaussian larger. The resolution parameter, k can be regarded as "fine resolution" since increasing by a small amount will change the spread of the gaussian far less than will the same change in σ, the "coarse resolution". Therefore, in the two figures we see that there is very little difference in the plots in which k was increased but we see a much larger difference in the plots in which σ was increased.

The following is an example of the results obtained by applying different masks to a single image.

4.5 Texture Classification Using MRI Operators

Texture occurs in images due to either irregular surface topography or to nonuniform surface reflectance. There have been a number of approaches to the modeling of texture in images [Haralick 70, Laws 79, Harwood et al 83]. Most of these rely on the modeling of local correlation properties of the gray-level image using either direct statistical measures or using the response to specific masks. In particular, [Laws 79] described a set of texture energy measures in terms of the response to linear 3 x 3 or 5 x 5 masks. These masks are chosen to reflect combinations of center-weighting, edge detection, and spot detection templates. The distribution of the outputs of these masks averaged across a textured region was shown to be useful for the discrimination of texture types. Harwood [Harwood et al 83] extended this idea and studied the use of rank correlation statistics as a basis for discrimination.

Texture models such as those described above summarize descriptions of the variations in image intensity, but do not relate image properties to either surface topography or surface reflectance. An alternative model of image texture has been proposed using fractal geometries to model image texture and relate image texture to surface topography. Fractal geometry was introduced by [Mandelbrot 77, Mandelbrot 82] to describe certain classes of irregular edges or surfaces including coastlines and mountain profiles. More recently, [Pentland 83a, Pentland 83b] proposed the use of the fractal dimension to characterize images of natural scenes and perform texture segmentation.

The fractal dimension D is the dimension of a measurement space expressed relative to the topological dimension E. If the parameter H = D - E is used to characterize the roughness of the observed texture, then H = 0 corresponds to a flat plane, while H = 1 corresponds to an array of spikes covering the plane. In terms of H, the cumulative distribution function of the fractal Brownian function B(t) is:

$$F(n) = Pr([B(t+\Delta t) - B(t)|/|\Delta t| < n).$$

Figure 4-5: Masks Generated By Varying the Parameter size
$n = 2, \sigma = 3, k = 1$

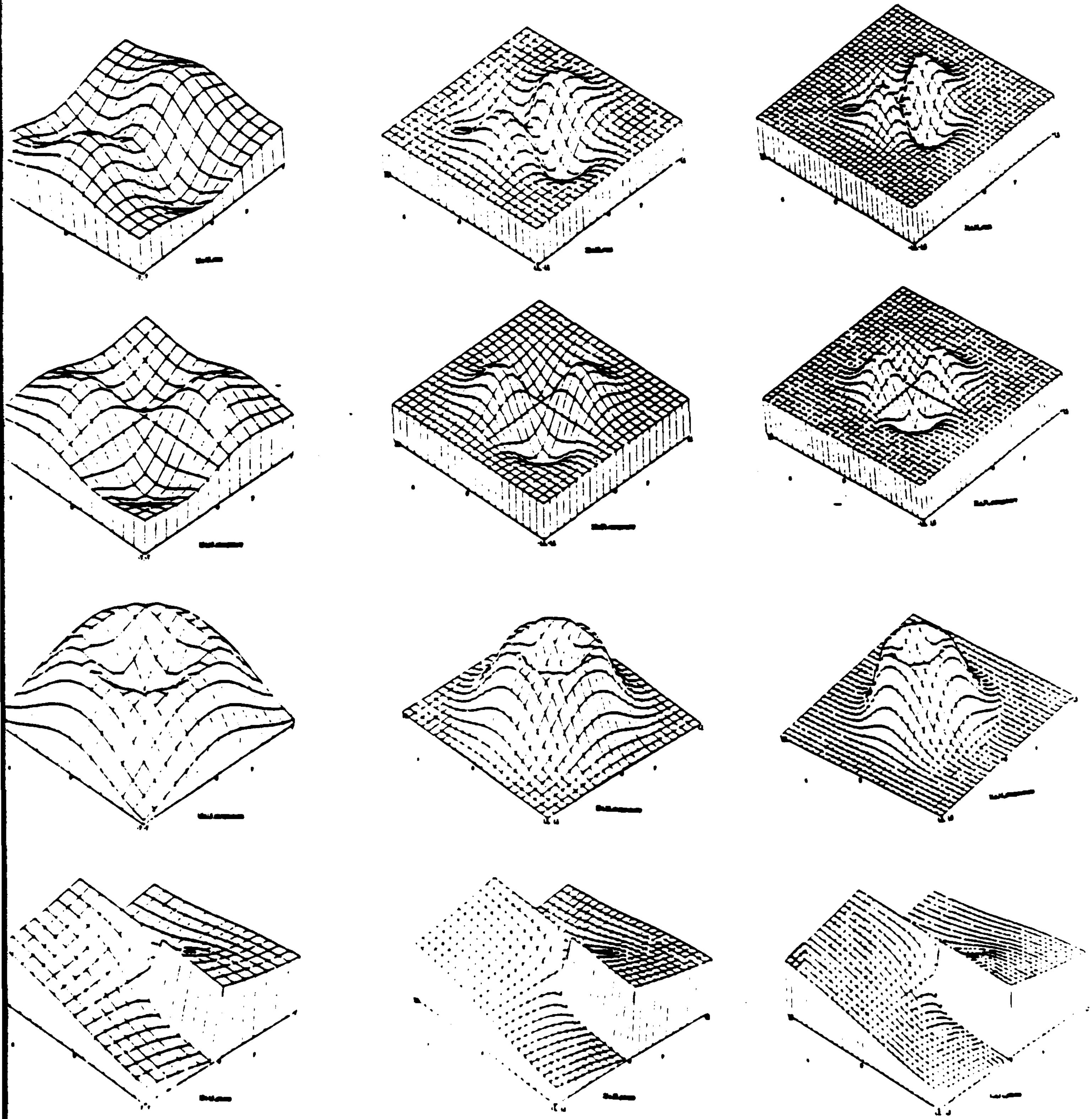


Figure 4-6: Masks Generated By Varying the Parameter n
size = 31x31, σ = 4, k = 1

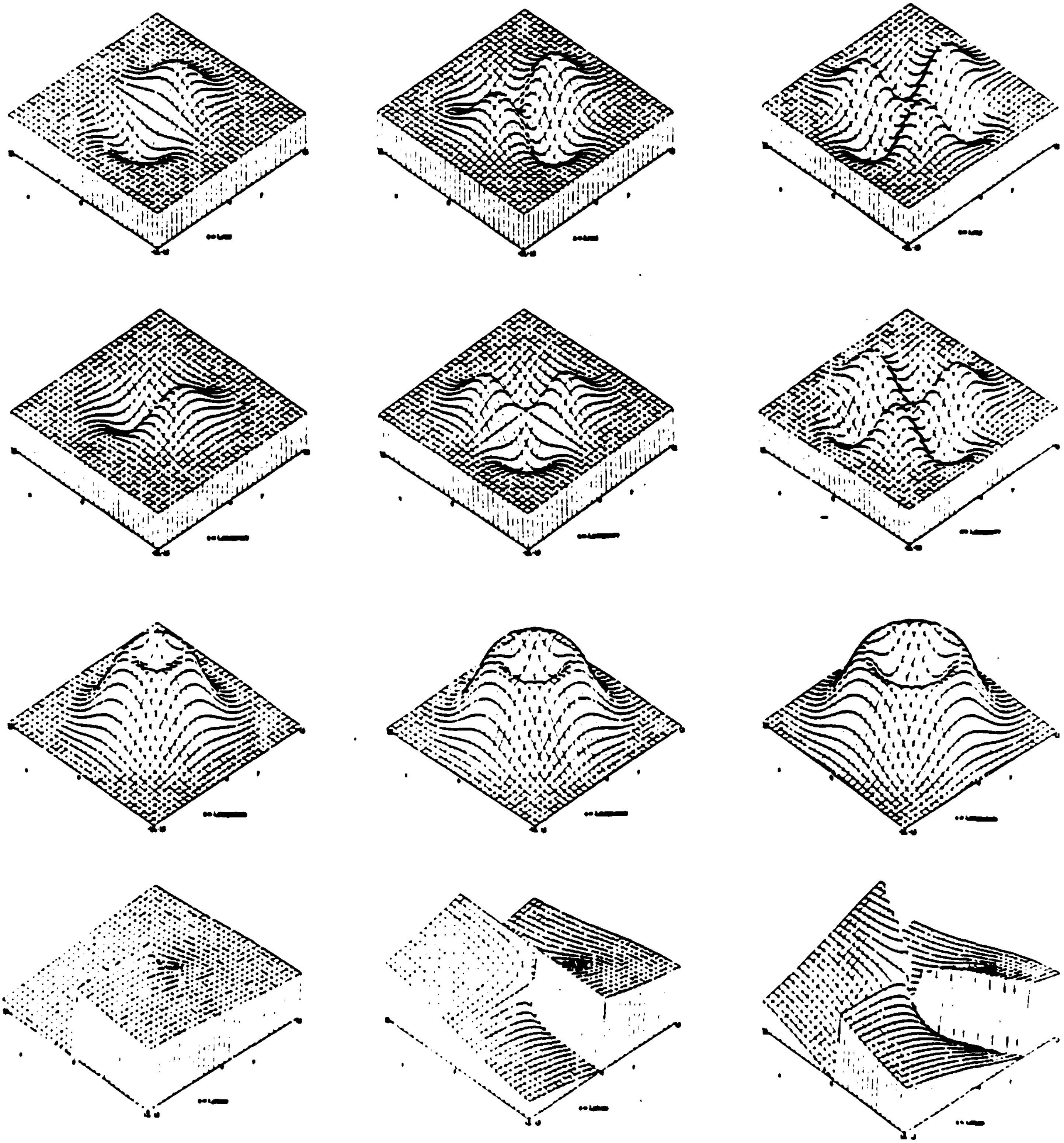


Figure 4-7: Masks Generated By Varying the Parameter σ
size = 31x31, n = 2, k = 1

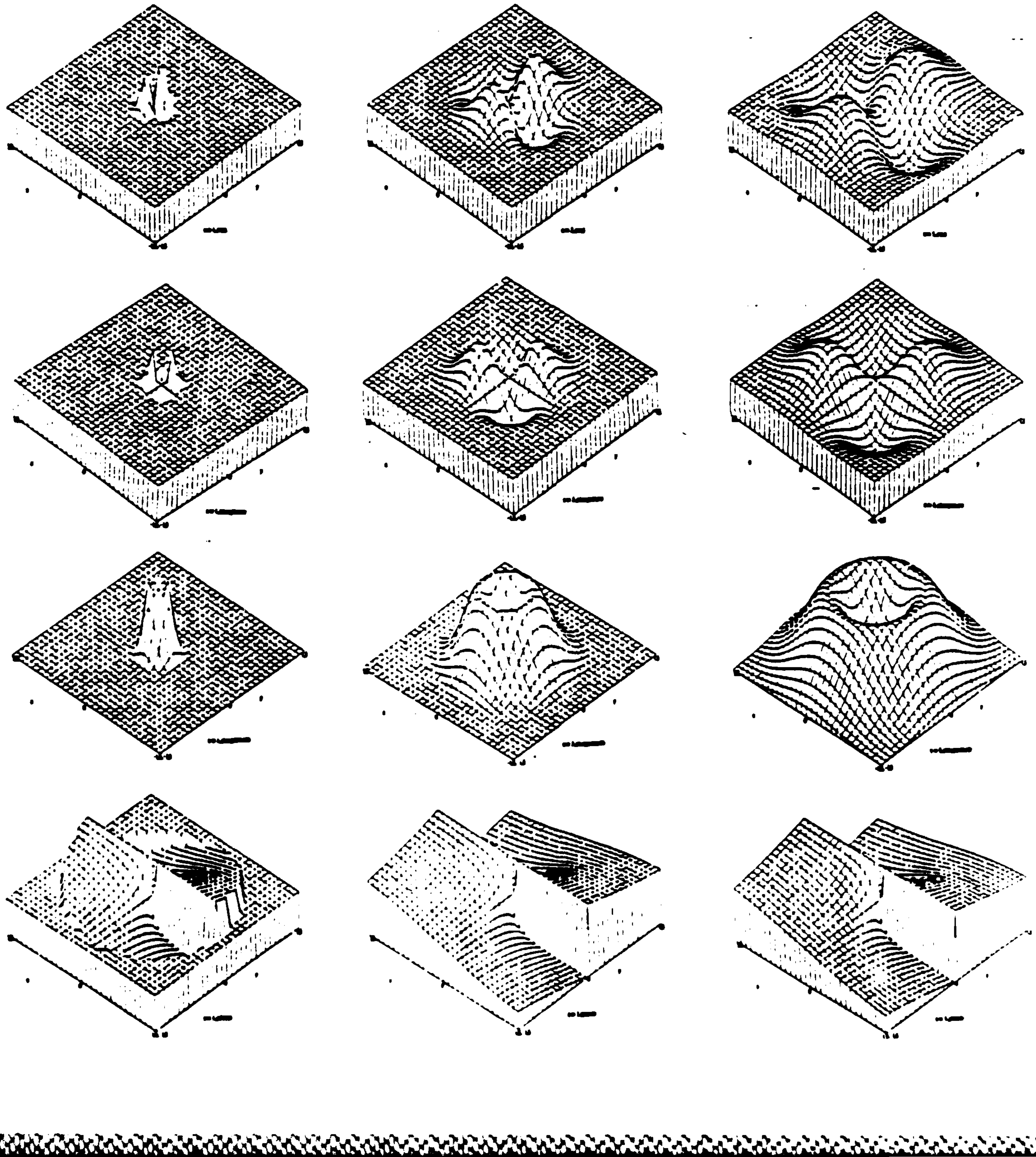
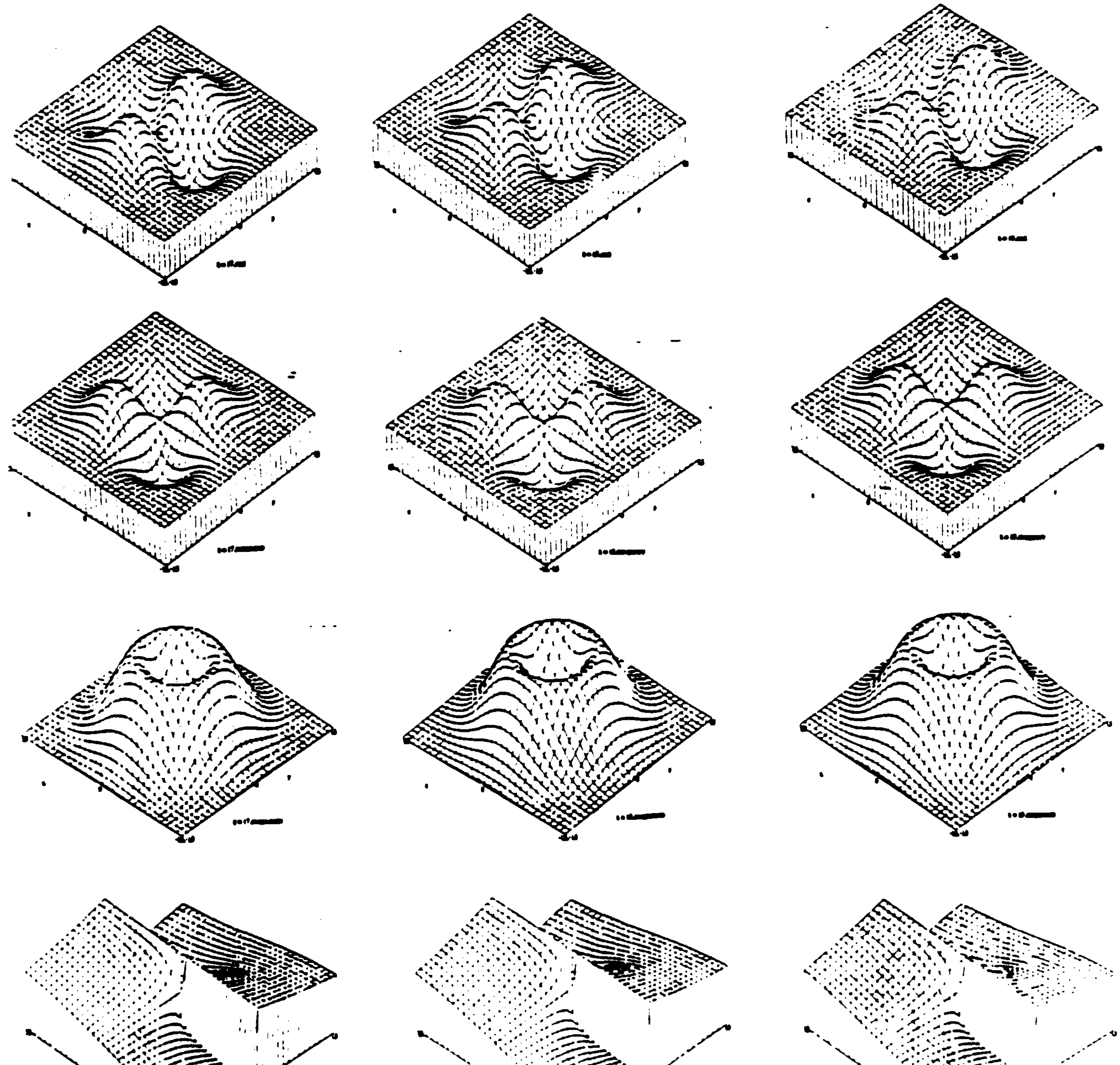


Figure 4-8: Masks Generated By Varying the Parameter <u>k</u>
size = 31x31, n = 2, σ = 1



where if $H = 1/2$ and if $F(n)$ is a zero-mean, unit variance Gaussian, the $B(t)$ defines a classical Brownian motion process. The fractal dimension in this sense is a compact parametric description of a homogeneous, isotropic random process which has advantages since it is invariant to scale. Pentland [Pentland 83a, Pentland 83b] related the image texture model to a natural surface model and showed that if the surface topology is fractal then the image intensity is also fractal if the surface obeys Lambertian assumptions. This result for fractal models is suggestive of more powerful results which might be achieved by relating image texture to more general random models of surface topography.

Studies of random surface topography suggest four principal contributions to the resulting textured image:

- Local edges of the surface elements
- Shading due to surface gradient and reflectance
- Shadows due to disparities between light incidence and viewing angles
- Local edges of one surface element occluding another.

These mechanisms are associated with surface topography and not with reflectance changes due, for example, to surface markings. The observed image texture varies in predictable ways with angle of view and lighting directions, and we would like to identify image texture measures which provide consistent measures of such changes.

Work in this area has centered around the use of texture energy measures as described in [Laws 79]. Both real and simulated images have been used to demonstrate the efficacy of this technique in distinguishing different 2-dimensional textures. Extension of this method to 3-dimensional texture analysis is currently being studied. Such analysis will aid in understanding the relationship between surface contours and image texture.

Simple one- and two-dimensional masks form the basis for texture energy measures. The distribution of the outputs of the different masks averaged over a textured region is useful in texture discrimination. These masks are chosen to reflect combinations of level, edge, and spot templates. The one-dimensional vector masks are weighted towards the center and all are either symmetric or antisymmetric and all but one are zero-sum Five length vectors are generated by convolving two three-length vectors. One-dimensional vector masks can be run both horizontally and vertically across an image. The two-dimensional masks are formed by convolving a horizontal vector with a vertical vector of the same length Figure 4-3 shows a number of the masks that are used in texture energy measurements.

Eight of the 3x3 masks and all one-dimensional vector masks except for the l3 and l5 level vectors are zero-sum. Convolved over a region with uniform pixel intensity (i.e.--no texture), a zero-sum mask will produce an output that is identically zero. "Texture Energy" refers to non-zero values resulting from convolutions with zero-sum masks. A Textured region is first histogram equalized to ensure that every region starts with the same average intensity. The various masks are convolved over the region separately, creating a number of "texture planes", one for each mask. The average pixel intensity for each resulting plane is then taken as a texture energy measure and collectively they form a feature vector that can be used in texture classification.

Fourier analysis of the various kernels reveal that these texture energy masks serve as bandpass filters in the frequency domain. Alone, the 3-length vector masks l3, s3 and e3 correspond to low-pass, high-pass and band-pass filters respectively. The 5-length vector masks also operate as filters, though each mask peaks in a narrower frequency range. The two-dimensional masks work as bandpass filters in the 2-d frequency plane. Each of the nine 3x3 masks is found to peak in a predictable manner in each quadrant of this plane as shown in figure 4-10. Each 5x5 and 7x7 mask also has a unique peak in the 2-d frequency plane. The set of all masks of size NxN covers the entire frequency plane.

```
                                  ripple r5 = [ 1 -4  6 -4  1]
                                  wave   w5 = [-1  2  0 -2  1]
spot  s3 = [-1  2 -1]             spot   s5 = [-1  0  2  0 -1]
edge  e3 = [-1  0  1]             edge   e5 = [-1 -2  0  2  1]
level l3 = [ 1  2  1]             level  l5 = [ 1  4  6  4  1]

   3-length vectors                      5-length vectors
```

```
 1   2   1           -1   0   1           -1   2  -1
 2   4   2           -2   0   2           -2   4  -2
 1   2   1           -1   0   1           -1   2  -1

   L3L3                L3E3                  L3S3


-1  -2  -1            1   0  -1            1  -2   1
 0   0   0            0   0   0            0   0   0
 1   2   1           -1   0   1           -1  -2  -1

   E3L3                E3E3                  E3S3


-1  -2  -1            1   0  -1            1  -2   1
 2   4   2           -2   0   2           -2   4  -2
-1  -2  -1            1   0  -1            1  -2   1

   S3L3                S3E3                  S3S3


-1  -4  -6  -4  -1            1  -4   6  -4   1
-2  -2 -12  -8  -2           -4  16 -24  16  -4
 0   0   0   0   0            6 -24  36 -24   6
 2   8  12   8   2           -4  16 -24  16  -4
 1   4   6   4   1            1  -4   6  -4   1

       E5L5                        R5R5
```

Figure 4-9: Texture Energy Masks

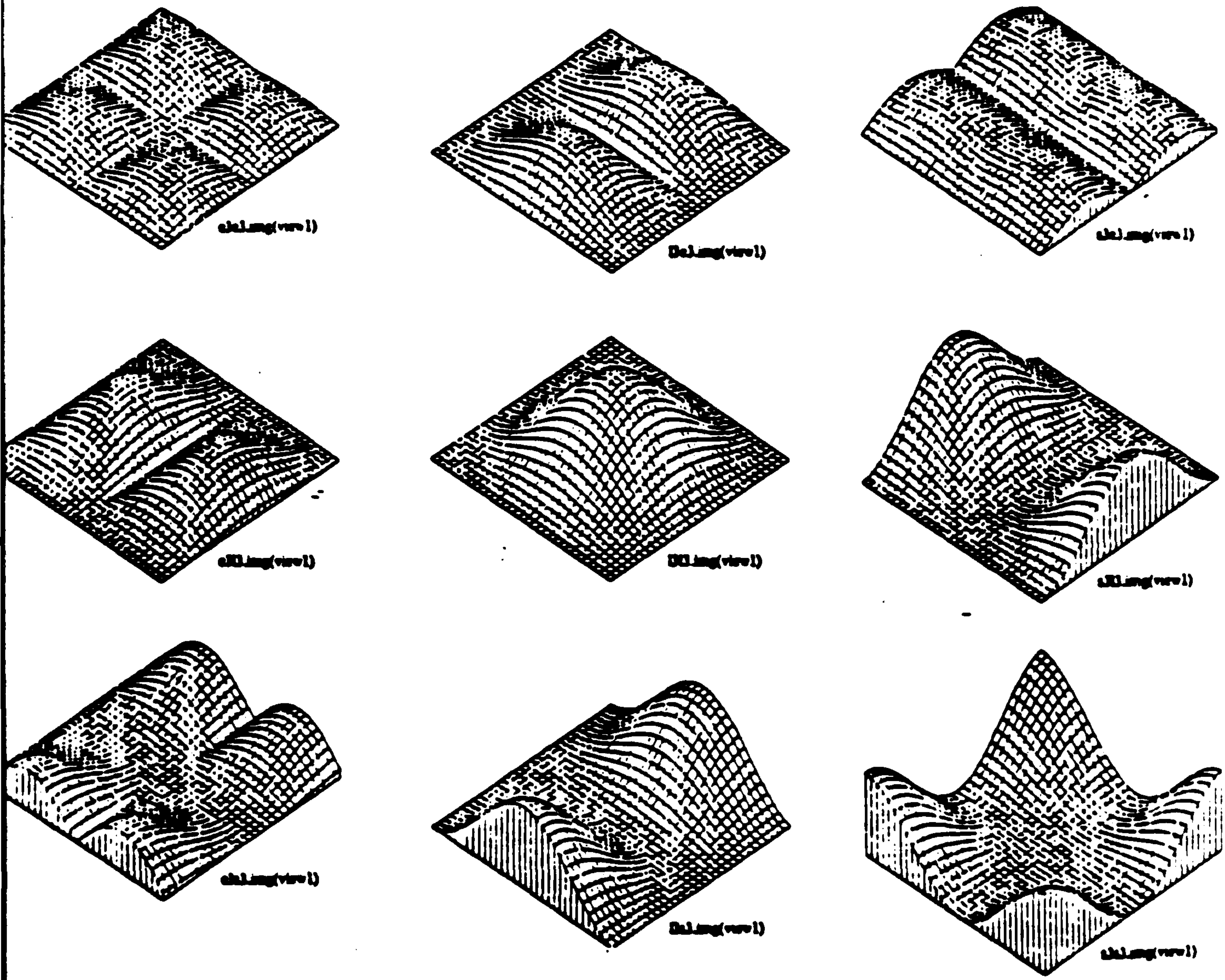
An image with a number of simulated textures (4-11) was generated and texture energy measures were determined for each of those textures. The first row of textures is column-oriented. The first block alternates dark and light columns; the second block alternates two dark and two light columns; the third block alternates four dark and four light columns; and the fourth block of the first row repeats the sequence of (...light,medium,dark, medium..) columns. The second row of textures is row oriented. It is identical to the first row except that the textures have been rotated 90 degrees. The first block of the third row has a checkerboard pattern. Each square of the checkerboard is one pixel in size. The second block of third row has alternating light and dark diagonal lines that are two pixels wide. Initially, all texture patterns have the same average intensity. Random noise was introduced into the image before processing. The texture measures for this simulated image are tabulated below in Table 4-1. Because the row- and column-oriented textures are obviously linear, 3- and 5-length linear vector masks were used in addition to the 3x3 masks. The values obtained indicate that the different textures do indeed result in unique texture energy measures and that these measures could be used to distinguish between textures.

The aerial image shown in figure 4-12 was used in applying texture energy masks to a real image. Four textured sections containing dirt, grass, and two different sections of water were taken from the original image and each was histogram equalized. The 3x3 masks were then used to generate the textural planes and from them the texture energy measures were obtained. Two of the sections were images of water, one section was a grassy field and the other an uneven area of dirt & vegetation. The results of this texture analysis are tabulated in Table 4-2. The two sections of water had, as expected, very similar energy measures. The measures for the section with the grassy field and the section with the dirt both differ significantly from the water-containing sections but differ from each other (by > 10) only in the e3e3, e3l3, and s3l3 masks. However, using just these three masks, they can be separated.

Results here show that texture energy measures can be applied to aerial images which have sizable textured regions. Information gained from macroscopic texture analysis can aid in understanding changes in land usage and local scene analysis.

The primary objective of texture classification is to identify textured regions within the image. If it is known where the texture boundaries are, classification of the different regions can be done with relative ease. In such a segmented image, the task is one of determining a number of texture measures for the large areas and then classifying those areas into one of several known classes. However, when no *a priori* information is known about the texture boundaries, the task becomes harder. Texture measures must be calculated for *each pixel* and the pixels then classified. The

Figure 4-10: Fourier Domain of Texture Energy Masks



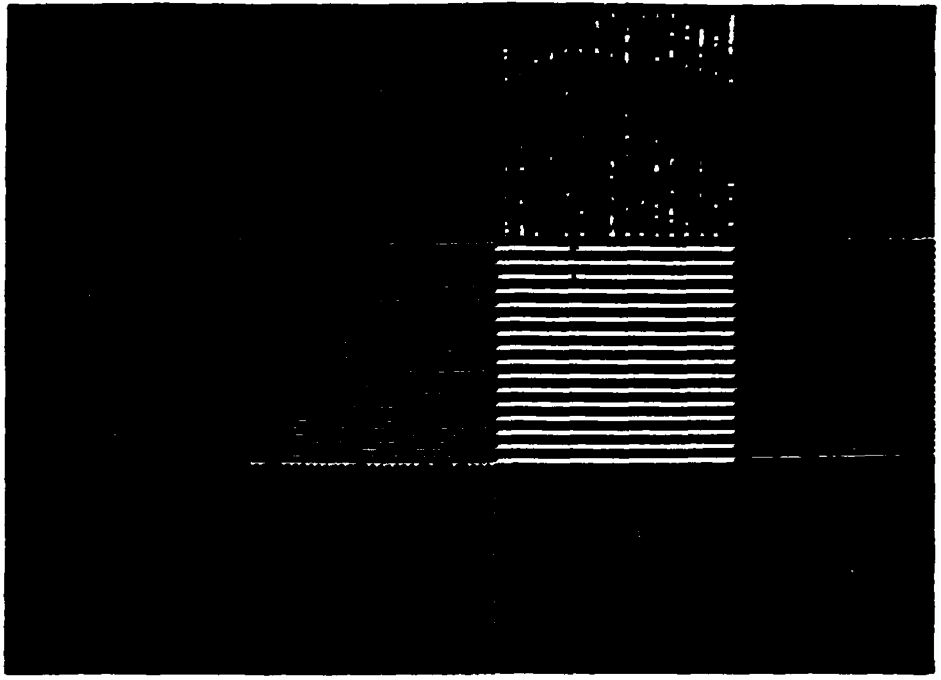


Figure 4-11: Simulated Texture Image

mask	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)
orig.	128	128	128	128	128	128	128	128	128	128
e3e3	6	7	6	6	7	9	7	7	7	126
e3l3	16	16	17	17	21	127	73	71	17	125
e3s3	9	8	8	8	10	11	9	8	10	126
l3e3	10	126	68	66	11	13	12	12	11	126
l3s3	126	126	70	69	16	17	16	15	16	127
s3l3	30	30	30	29	126	127	79	76	31	126
s3s3	15	15	14	14	17	17	15	15	126	126
s3col	9	9	9	9	127	95	53	52	127	96
s3row	127	96	50	51	6	6	6	6	126	96
s5col	9	10	10	9	10	127	96	69	11	127
s5row	9	127	94	67	8	9	9	8	8	127
e3col	5	5	5	5	6	96	50	50	5	94
e3row	4	96	50	49	5	5	5	5	4	95
e5col	12	11	11	12	14	128	112	68	11	125
e5row	12	127	111	67	13	14	14	13	13	125
r5col	32	31	31	31	127	126	108	77	127	126
r5row	125	127	111	72	19	20	20	19	125	127
w5col	13	12	12	12	13	126	95	69	13	126
w5row	8	127	96	65	8	9	9	8	8	126

Table 4-1: Texture Measures for the Simulated Image

challenge here is to develop an algorithm that will accurately classify pixels without incurring a great computational cost. This involves finding the smallest possible feature vector that will offer

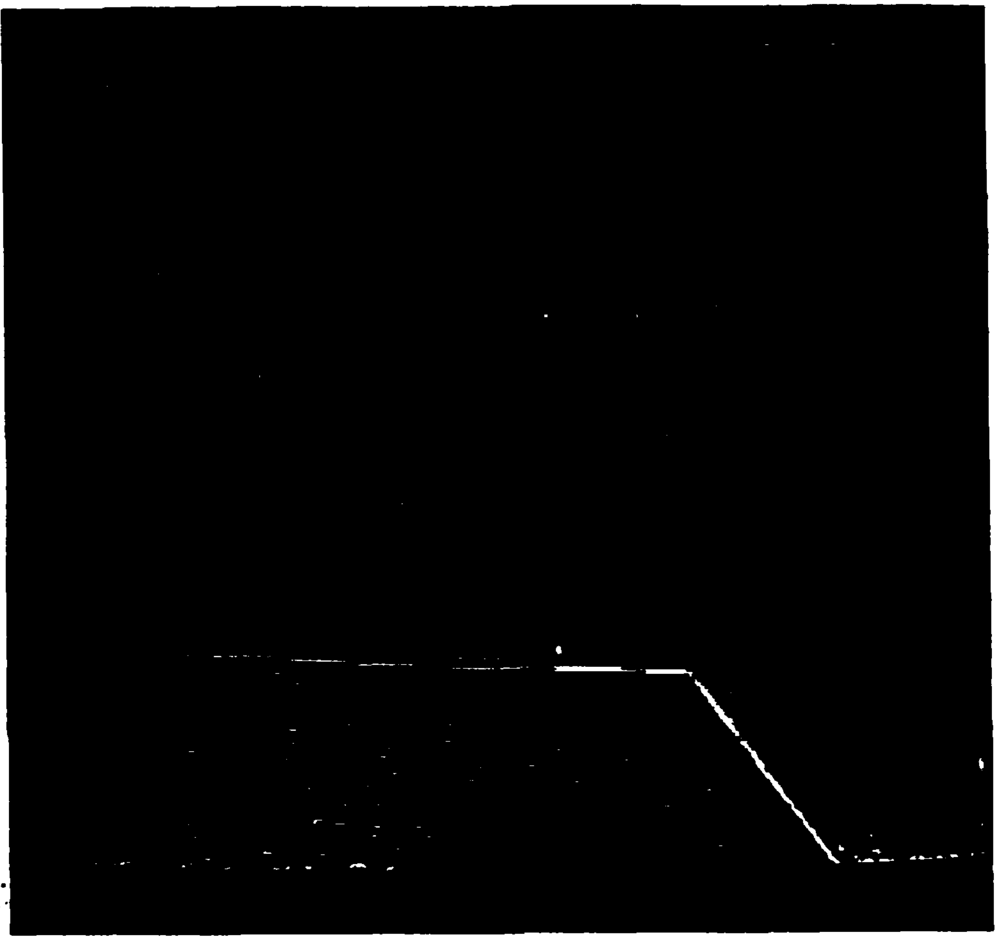


Figure 4-12: Aerial Image used for Analysis with Texture Energy Masks

mask	dirt	field	water1	water2
orig.	128	130	128	128
e3e3	23	43	84	84
e313	90	118	184	187
e3s3	33	43	84	84
13e3	102	102	141	135
1313	123	129	125	123
13s3	102	102	141	135
s3e3	33	43	84	84
s313	90	118	184	187
s3s3	33	43	84	84

Table 4-2: Texture Energy Measures for Real Images

reasonably good inter-class separation. As described below, the MRI operator was found to be very effective in classifying textures pixel by pixel.

Training

The [Brodatz 68] texture album contains photographs of many textures. Four stucturally similar "cellular" textures, aluminum wire, cotton canvas, raffia and oriental straw cloth were chosen for this work. These textures were chosen because of their similarity in cell size and cell shape. They may be considered to form a "worst case" set in the sense that they are very similar to each other and therefore may be hard to tell apart.

Fig 4-13 shows the composite image that was used as the training image. 128x128 samples each of the four textures were used for this purpose. Each sample was individually histogram equalized to eliminate first order differences. This ensures that differences in brightness between samples *after* processing with the MRI operator are caused by the convolution operation and are not due to differences in the initial brightnesses of the samples. Training involved the use of 28 different MRI operators and each time the following process was carried out:

1. convolution of the composite image with the operator in the frequency domain and conversion of the result back into a complex image in the spatial domain.

2. conversion of the complex image into a normalized 8-bit integer image giving the gray-level magnitude for each pixel.

3. an 11x11 average smoothing of the 8-bit integer image

4. calculating average gray-level intensity and standard deviation measures for 100x100 internal regions of each of the four samples. The internal regions were used rather than the whole 128x128 section to avoid including edge pixels in the average and standard deviation measures.

The results of performing this set of steps with each operator is summarized in Table 4-3.

This was followed by determining which masks gave the greatest between-class separations. To do this, the "inter-class ratio" was determined for each pair of classes for every mask. Briefly stated, the inter-class ratio is the difference in averages between two classes divided by the sum of their standard deviations at a particular mask. For mask m and any two classes i and j, the inter-class ratio R^m_{ij} is defined as:

$$R^m_{ij} = \frac{|Avg(m,i) - Avg(m,j)|}{[Std(m,i) + Std(m,j)]}$$

The inter-class ratios calculated from the entries in Table 4-3 are given in Table 4-4.

MRI MASK	straw cloth		raffia		cotton can		alu wire	
	avg	std	avg	std	avg	std	avg	std
n0s1f.m	80.8	7.2	99.8	17.6	78.5	12.2	90.2	14.2
n0s2f.m	40.7	10.5	74.2	21.4	53.3	23.6	64.8	16.1
n0s3f.m	33.6	15.8	60.8	25.6	58.8	30.4	58.9	27.3
n0s4f.m	38.1	21.2	61.4	30.2	74.5	39.6	71.7	39.3
n0s5f.m	42.4	24.2	61.1	31.5	86.4	46.4	83.8	46.8
n0s6f.m	46.0	26.3	60.2	32.3	95.9	51.9	94.1	52.8
n0s7f.m	46.0	26.6	55.6	31.3	97.6	53.2	96.9	54.5
n1s1f.m	103.6	7.1	82.4	9.6	96.9	7.4	81.7	10.8
n1s2f.m	60.1	4.4	87.9	11.3	46.9	6.7	76.8	9.7
n1s3f.m	39.1	4.3	83.1	18.1	27.5	8.5	59.4	8.4
n1s4f.m	32.6	6.0	75.8	25.8	29.8	12.2	35.1	7.3
n1s5f.m	29.4	7.1	67.4	27.3	33.9	16.4	24.7	9.2
n1s6f.m	28.0	8.2	63.8	27.6	41.4	21.0	28.8	12.2
n1s7f.m	27.7	9.8	61.9	27.8	51.4	25.9	36.7	15.0
n2s1f.m	95.3	5.6	58.1	7.7	93.1	7.6	67.0	9.7
n2s2f.m	82.0	5.1	81.5	10.2	69.9	7.2	77.4	10.3
n2s3f.m	44.7	3.3	81.1	11.0	27.2	6.5	71.4	8.7
n2s4f.m	36.7	4.9	83.8	19.3	24.3	7.5	53.6	7.7
n2s5f.m	32.8	6.9	80.5	25.7	24.2	7.9	27.2	5.3
n2s6f.m	32.0	8.6	77.6	27.9	25.0	9.4	16.4	5.7
n2s7f.m	29.6	10.0	74.3	29.1	27.0	11.8	16.2	8.5
n3s1f.m	72.0	4.3	43.5	6.0	76.7	6.7	57.5	8.9
n3s2f.m	101.8	6.0	75.2	10.1	85.0	6.6	73.6	11.3
n3s3f.m	50.7	3.7	82.6	11.5	33.5	6.0	73.3	9.9
n3s4f.m	40.8	3.6	87.7	13.6	25.2	7.3	68.2	9.3
n3s5f.m	33.1	5.0	81.4	20.2	23.4	6.8	38.5	6.0
n3s6f.m	33.6	6.1	78.4	24.5	22.9	7.2	19.1	4.5
n3s7f.m	33.7	7.2	75.7	26.5	23.4	8.5	15.7	6.2

Table 4-3: average & std dev for each class for each MRI operator

MASK	CLASS PAIR					
	cloth	cloth	cloth	raffia	raffia	cotcan
	raffia	cotcan	aluwir	cotcan	aluwir	aluwir
n0s1f.m	0.766	0.119	0.439	0.715	0.302	0.443
n0s2f.m	1.050	0.370	0.906	0.464	0.251	0.290
n0s3f.m	0.657	0.545	0.587	0.036	0.036	0.002
n0s4f.m	0.453	0.599	0.555	0.188	0.148	0.035
n0s5f.m	0.336	0.623	0.583	0.325	0.290	0.028
n0s6f.m	0.242	0.638	0.608	0.424	0.398	0.017
n0s7f.m	0.166	0.647	0.628	0.497	0.481	0.006
n1s1f.m	1.269	0.462	1.223	0.853	0.034	0.835
n1s2f.m	1.771	1.189	1.184	2.278	0.529	1.823
n1s3f.m	1.964	0.906	1.598	2.090	0.394	1.888
n1s4f.m	1.358	0.154	0.188	1.211	1.230	0.272
n1s5f.m	1.105	0.191	0.288	0.767	1.170	0.359
n1s6f.m	1.000	0.459	0.039	0.461	0.879	0.380
n1s7f.m	0.910	0.664	0.363	0.196	0.589	0.359
n2s1f.m	*2.797	0.167	1.850	2.288	0.511	1.509
n2s2f.m	0.033	0.984	0.299	0.667	0.200	0.429
n2s3f.m	2.545	*1.786	*2.225	*3.080	0.492	*2.908
n2s4f.m	1.946	1.000	1.341	2.220	1.119	1.928
n2s5f.m	1.463	0.581	0.459	1.676	1.719	0.227
n2s6f.m	1.249	0.389	1.091	1.410	1.821	0.570
n2s7f.m	1.143	0.119	0.724	1.156	1.545	0.532
n3s1f.m	2.767	0.427	1.098	2.614	0.940	1.231
n3s2f.m	1.652	1.333	1.630	0.587	0.075	0.637
n3s3f.m	2.099	1.773	1.662	2.806	0.435	2.503
n3s4f.m	2.727	1.431	2.124	2.990	0.852	2.590
n3s5f.m	1.917	0.822	0.491	2.148	1.637	1.180
n3s6f.m	1.464	0.805	1.368	1.751	*2.045	0.325
n3s7f.m	1.246	0.656	1.343	1.494	1.835	0.524

Table 4-4: Inter-class Ratios. Those marked with a '*' are the maximum for that pair of classes

For classification, it is important to use masks that give the greatest inter-class separation. It is also important to limit the number of masks used in the interest of minimizing the amount of computation required. Therefore, for classification purposes, it is best to choose that minimum set of MRI masks that will give us maximum separation between any pair of classes. From Table 4-4 it is evident that one of the masks: n2s1, n2s3 or n3s6 will give the maximum inter-class separation between any pair of the four textures. Therefore, only these three masks need to be used for classification of textures known to belong to this set of four.

Classification Results

A second composite image similar to the first was created. Different portions of the same photographs used in the training image were used to make this second image. Each of the three masks was convolved with the composite image, creating three feature planes. The gray-level values provided by these planes at each pixel served as a feature vector of length three which was then used to classify that pixel. A minimum distance classifier, using the averages found for each class and for each mask from the training image, was employed to perform the classification. Results show that 95 % of pixels in the interior regions of the different sections can be classified accurately while in the entire composite image 88 % of the pixels are correctly classified. Fig. 4-14 shows the composite image used for classification and 4-15 shows the resulting pixel-by-pixel segmentation.

Tables 4-5 & 4-6 give a detailed evaluation of how the classifier worked including numbers and percentages of correctly classified and mis-classified pixels in both internal regions and in the composite image.

Classified As	Belonging to Class			
	straw cloth	raffia	cot canvas	alu. wire
straw cloth	9429(94.3)	138(1.4)	237(2.4)	0(0.0)
raffia	1(0.0)	8902(89.0)	0(0.0)	0(0.0)
cot canvas	444(4.4)	0(0.0)	9763(97.6)	0(0.0)
alu. wire	126(1.3)	960(9.6)	0(0.0)	10000(100.0)

Table 4-5: Classification Accuracy of 100 x 100 Interior regions
Overall accuracy 95 %

Classified As	Belonging to Class			
	straw cloth	raffia	cot canvas	alu. wire
straw cloth	14570(88.9)	208(1.3)	2444(14.9)	789(4.8)
raffia	464(2.8)	14203(86.7)	6(0.0)	1267(7.7)
cot canvas	1130(6.9)	0(0.0)	13931(85.0)	0(0.0)
alu. wire	220(1.3)	1373(12.0)	3(0.0)	14328(87.5)

Table 4-6: Classification Accuracy of Entire Composite Image
Overall accuracy 87%

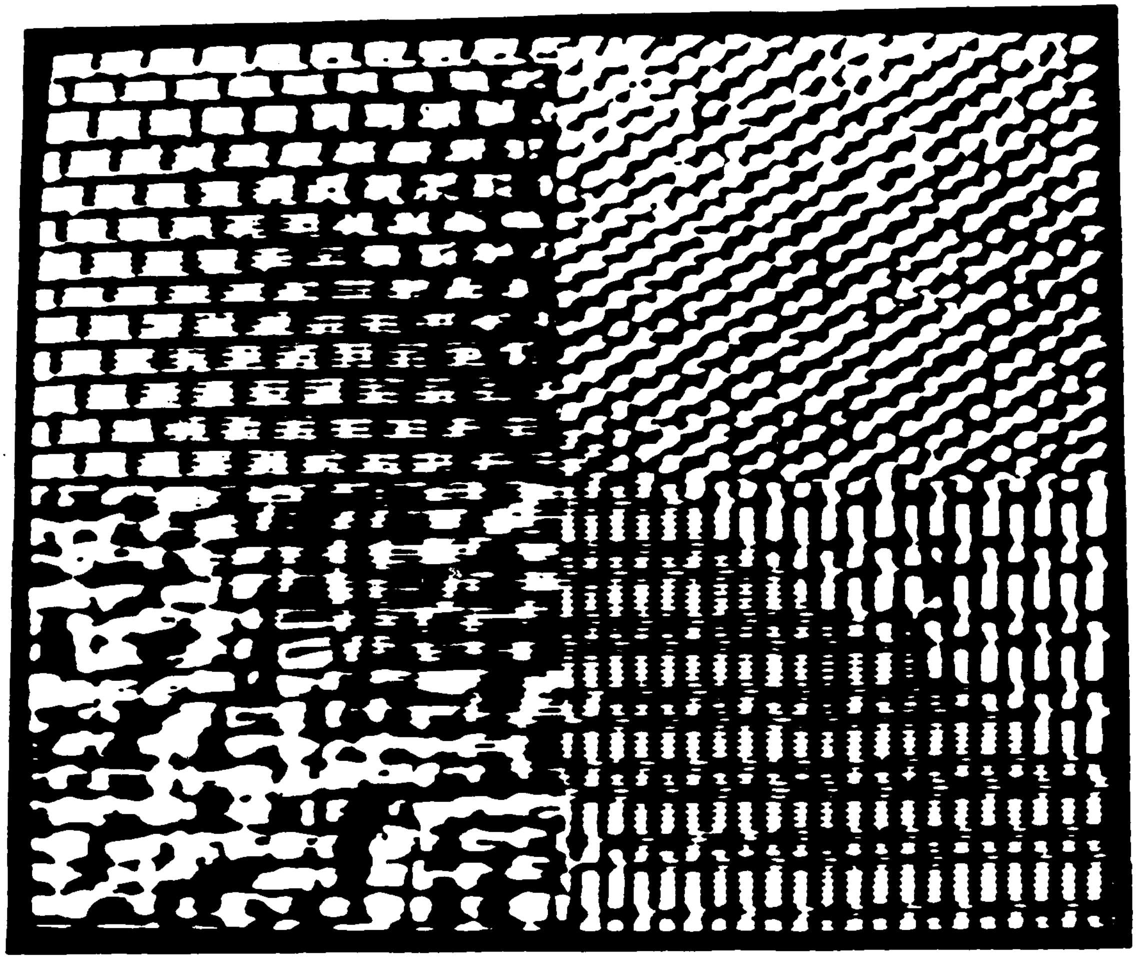


Figure 4-13: Texture samples used for training
top left: straw cloth, top right: raffia
bot left: cotton canvas, bot right: aluminum wire

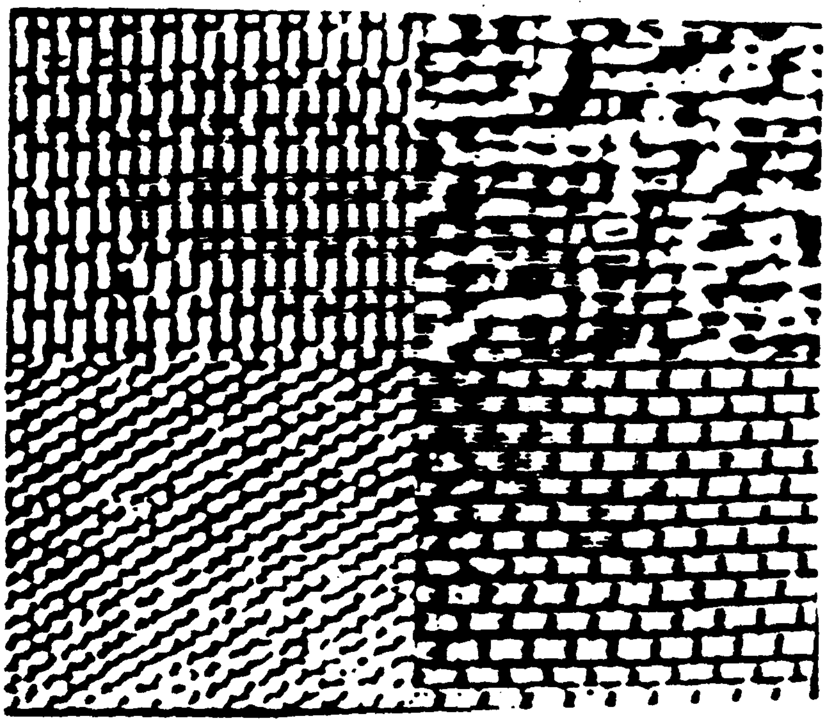


Figure 4-14: Composite image used for pixel-by-pixel classification

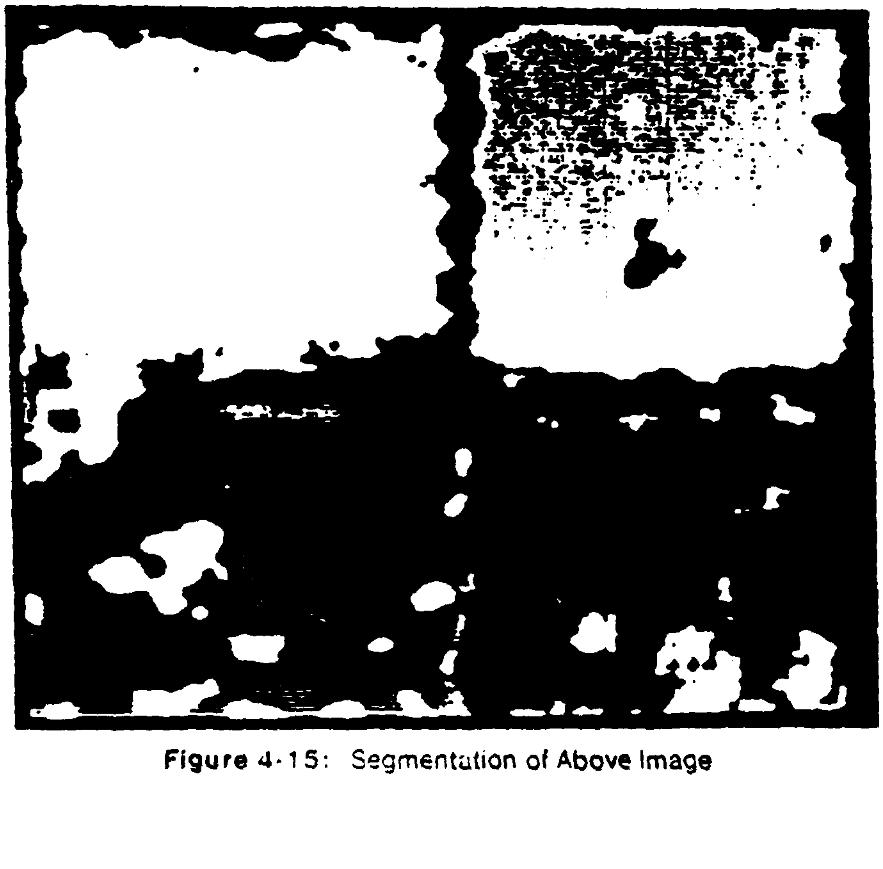


Figure 4-15: Segmentation of Above Image

4.6 Structural Analysis Using MRI Operators

The MRI operator is also useful for the early processing of structural images. Both the order and standard deviation of the operator can be tuned to different image features. A single correlation between image and complex MRI kernel determines both the strength of the feature and it's orientation (for higher than zero order operators). The following four figures illustrate different kernels applied to the same input image of an aircraft on a runway. Notice for the first order operators, a standard deviation of six effectively traces most of the aircraft outline. Once the aircraft is located operators with a lower standard deviation help to locate features such as the engines. Region operators, with a second order kernel, help to locate the center line of the airframe and engine cowlings respectively.

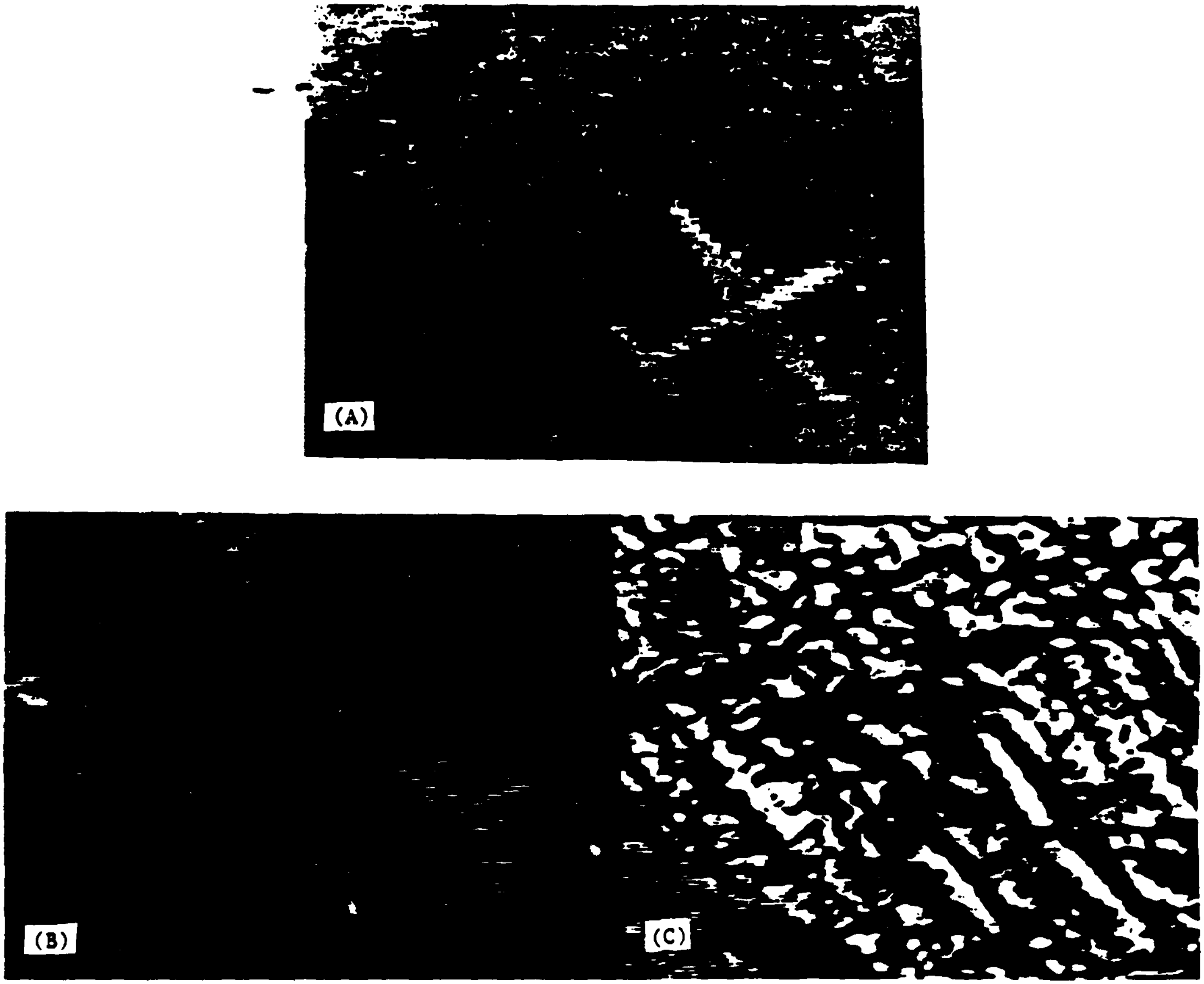


Figure 4-16: Result of applying the MRI edge detecting operator [N = 1] with a standard deviation of two pixels. [A] is the original image, [B] is the magnitude of the transform, and [C] the transform phase.

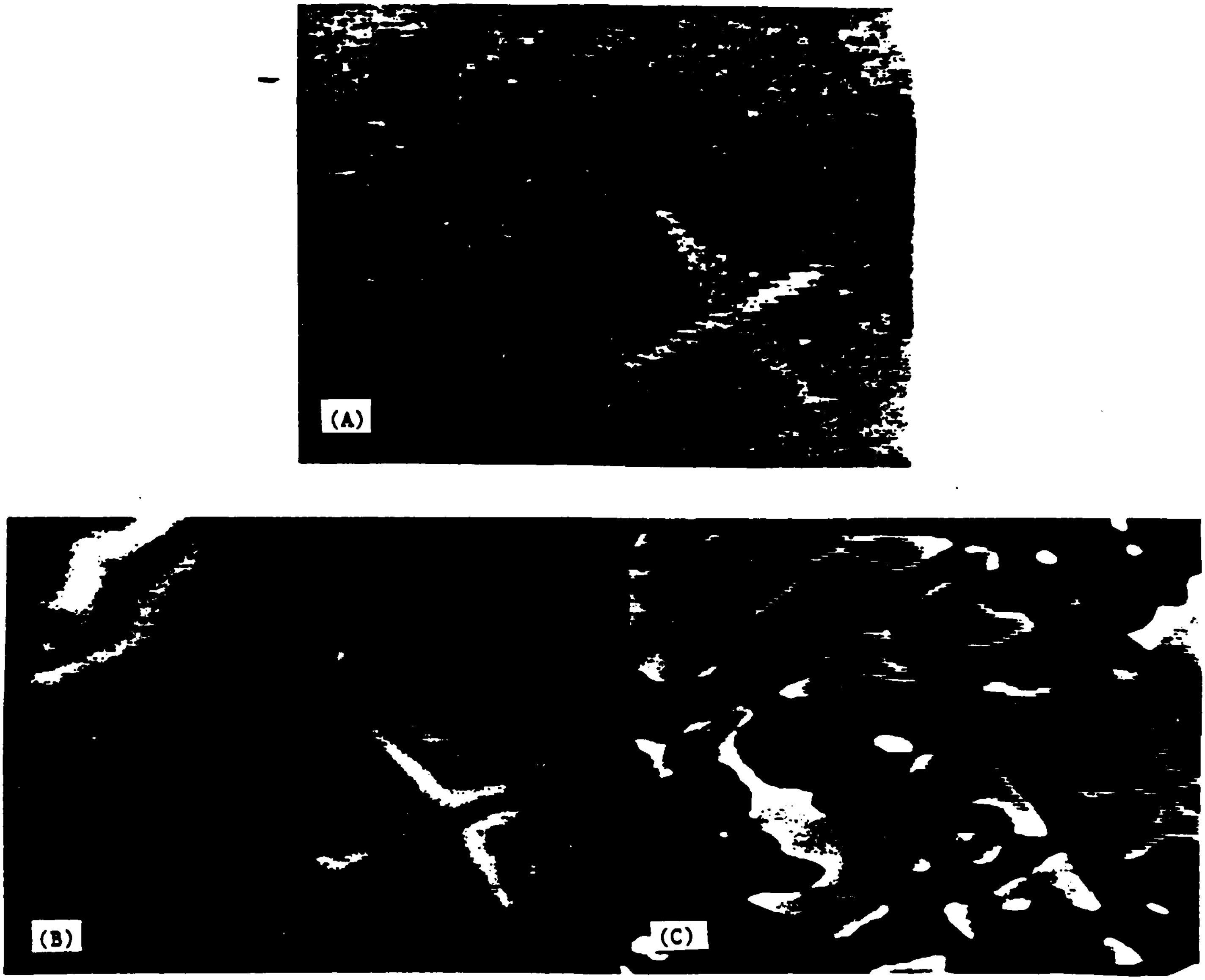


Figure 4-17: Result of applying the MRI edge detecting operator [N = 1] with a standard deviation of six pixels. [A] is the original image, [B] is the magnitude of the transform, and [C] the transform phase.

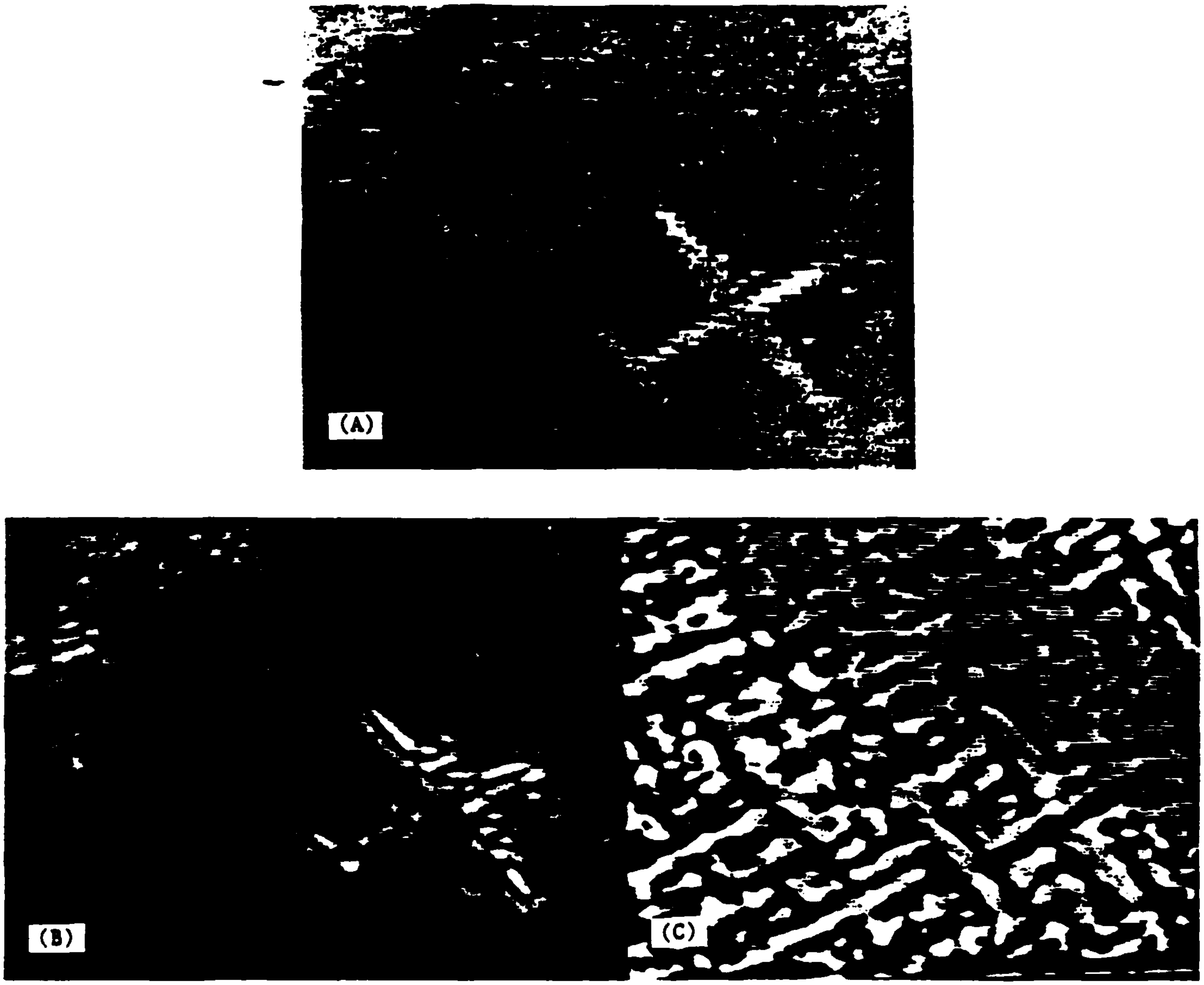


Figure 4-18: Result of applying the MRI region detecting operator [N = 2] with a standard deviation of three pixels. [A] is the original image, [B] is the magnitude of the transform, and [C] the transform phase.

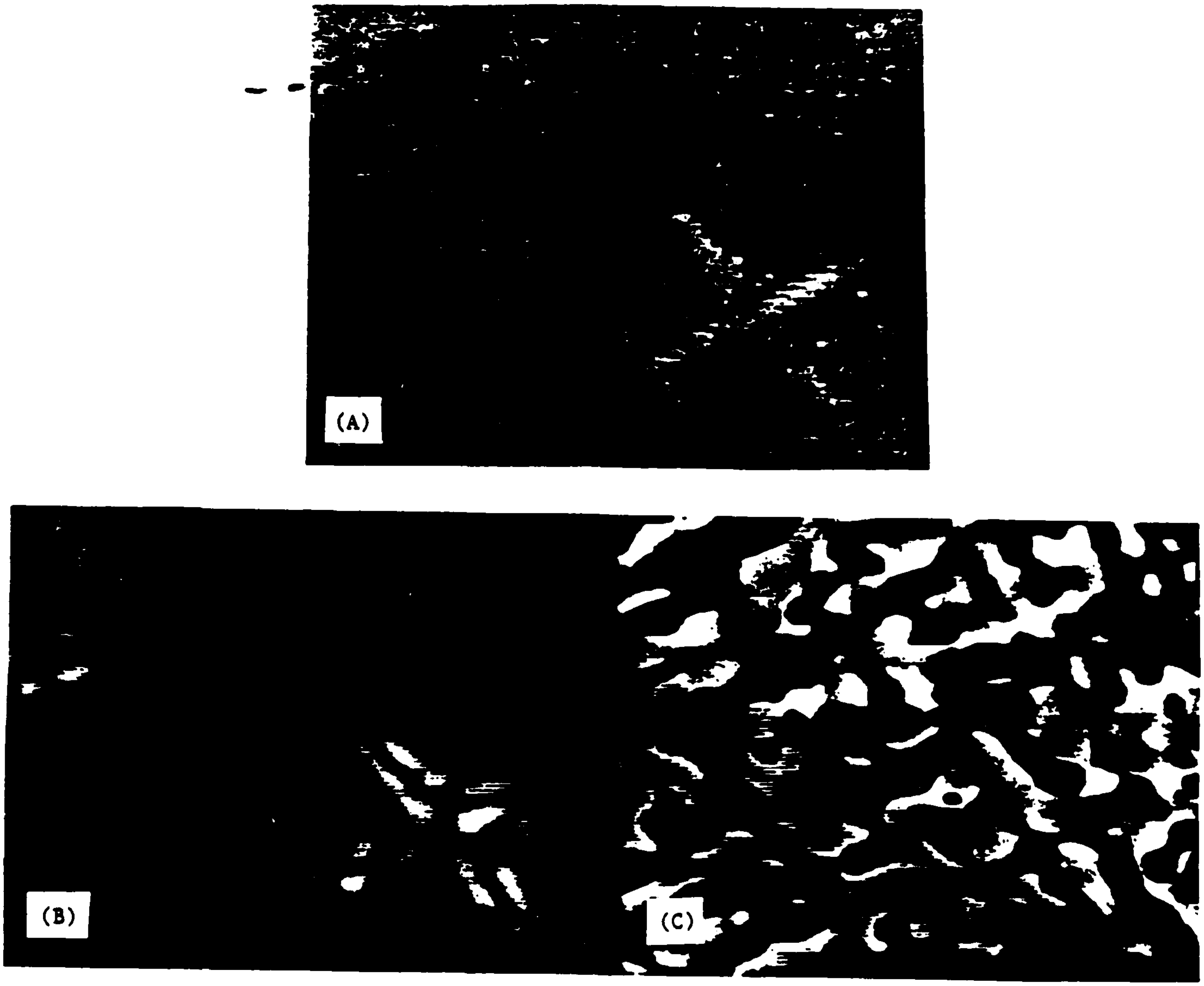


Figure 4-19: Result of applying the MRI region detecting operator [N = 2] with a standard deviation of six pixels. [A] is the original image, [B] is the magnitude of the transform, and [C] the transform phase.

4.7 Recursive Model Matching Algorithms

Recursive model matching allows a broad range of feature extractors to be interactively applied to an incoming image under guidance from a library of hierarchically structured models. At each step in the recursive analysis of a frame, an ensemble of hypotheses are active suggesting objects and textures present in the image. Evaluation of each model based hypothesis will suggest additional, specific features which might be extracted from the image in order to lend support or disprove the hypothesis. As additional features which support a particular hypothesis become known to the evaluation process, more narrowly defined models may be invoked as new hypotheses. Ultimately each high level hypothesis must be resolved down to either highly probable terminal models or determinations of an unfounded hypothesis.

The strengths of both electro-optic and digital multiprocessor technology are symbiotically paired by the recursive model matching structure. Processors such as a real-time, optical correlator [Casasent 78] allow a reference kernel function to be applied in parallel to an entire grey level image, limited only by the rate at which data can be digitally scanned in and out of the device. Decision intensive steps, in which the detected features are evaluated in the context of specific object models, map well onto a digital multiple instruction stream, multiple data stream processor. Associated work in computer architecture (RAPIDBUS & RAPIDGRAPH) is providing a basis for the simulation and implementation, respectively, of such a tightly coupled system.

4.7.1 Feature Space

Evaluation of actual high altitude or space based imagery, such as that shown in figure 4-20, underscores the wealth of different kinds of features which need to be identified and cataloged in order to match existing model data. Through the integration of data from diverse features, ambiguity caused by lighting, partial object occlusion, and sampling noise can frequently be resolved. Several different classes of feature extractors have been identified as being potentially useful:

- The multiresolution, rotationally invariant operator, described earlier in this report, assists in the detection of points, edges, and lines within a single brightness or spectral plane. Evaluation of an image by the MRI operator yields a list of candidate features described by an x,y location and magnitude for $n = 0$ at each of k resolution levels. For higher order operators ($n = 1$, $n = 2$), a direction is assigned to each feature point.

- Texture energy measures provide an useful means of identifying, characterizing, and segmenting "background" regions. A set of N kernel functions are used as a basis set, characterizing a sample texture in the N dimensional feature space by the "nearest" reference texture point. Boundaries of a particular texture region can be detected through gradients in the texture energy measure assigned to local windows.

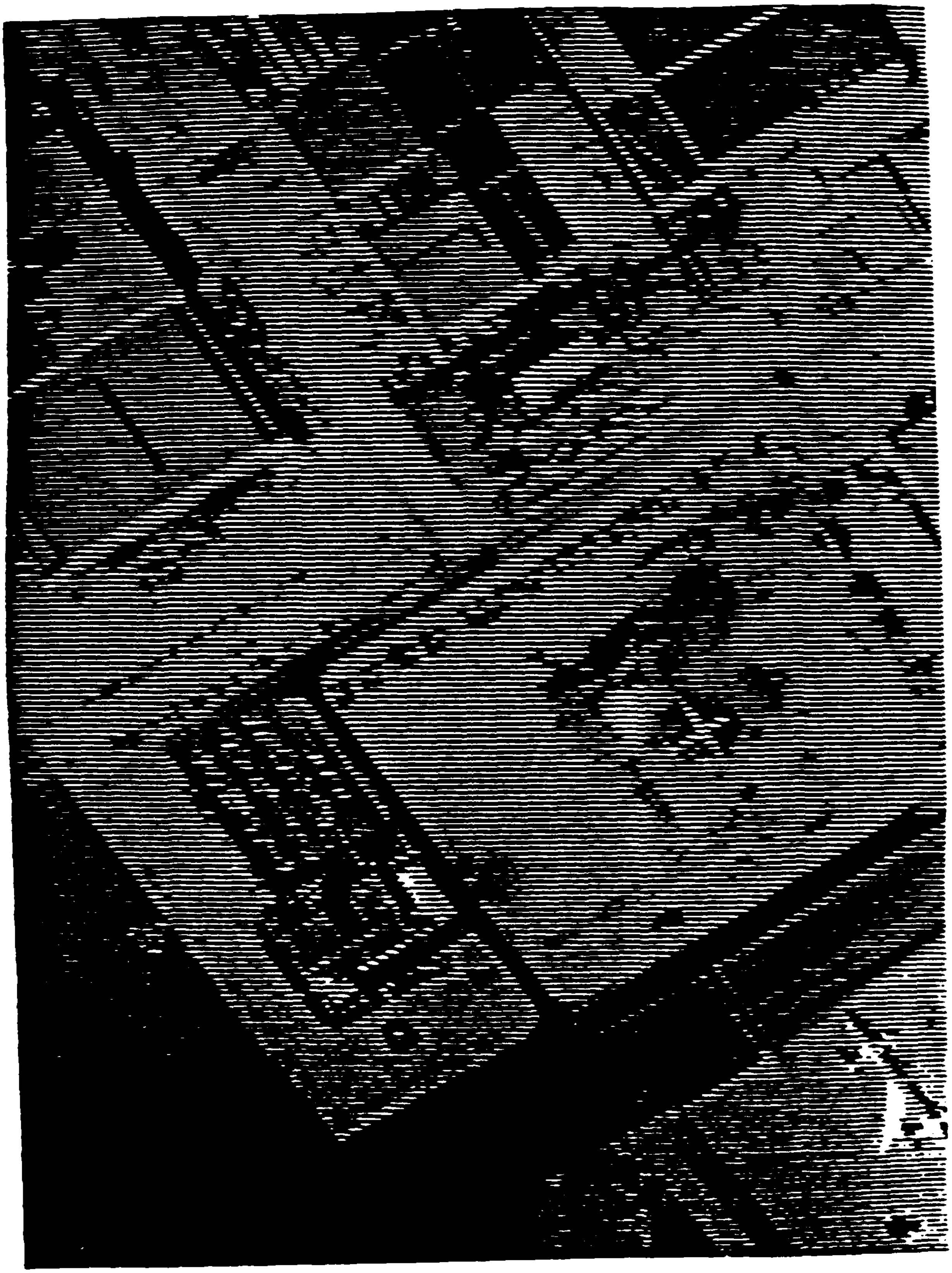


Figure 4-20: A multitude of different features and feature occurrences may be present in high altitude imagery such as this airfield used in our study.

- Boundaries not readily evident as a change in brightness are often visible as a change in the reflected spectrum. Adding an additional dimension, spectral frequency, often sampled by discrete channels such as R(x,y), G(x,Y), and B(x,y), can provide information on the magnitude and direction of spectral "edges" or "regions". As models generate increasingly detailed questions about the image, reporting criteria can be narrowed to inquiry about shifts from a specific spectral reference, or in specific image regions.

- Temporal changes between frames, or motion features, can be extracted using a variety of techniques. We are interested in exploring the extension of gaussian operators to the temporal dimension in analogy to spectral and temporal operators. Temporal information assists in both object/background separation and object identification[1].

- Macro-operators, suggested by detailed object models, may combine several of the above possibilities. Questions about the angle at which a wing meets the aircraft body may be resolved by an operator tuned to a particular angle described in terms of the spectral/textural properties of the aircraft and background. As very narrow hypotheses are formed, features searches may be required which could not practically be anticipated a priori.

The computational expense of many interesting operators, and the number of possible parameter combinations suggests a feature extraction task ideally suited to an electro-optic processor. Yet even assuming for a moment that each feature possibility could be extracted instantaneously, any digital representation of the complete feature space would result in tremendous organizational and storage difficulties in an effort to make the features available in usable form. When the number of possible operators, combinations of parameters, and substantial kernel sizes are taken into account, the scan-in/scan-out limitations of foreseeable devices makes a priori computation of all potentially interesting features unappealing. Recursive evaluation of the feature space provides an alternative to a priori pruning of the feature space.

4.7.1.1 Recursive, Goal Driven Image Exploration

Recursive, goal driven image exploration allows a developing description of the visual environment, described in terms of an object and texture data base, to select the particular regions of the feature space which are evaluated in an interactive fashion. Depending on the diversity of objects in the image, and the amount of prior knowledge coming from other frames, several hundred cycles may be required to converge on an adequate frame analysis. Each cycle consists of a feature extraction operation pipelined into a digital analysis by a multitude of processes, each investigating a particular hypothesis somewhere in the image.

[1] Object motion might lend support to the hypothesis that a moving object was a vehicle in preference to a building. The velocity, taken in context, provides further descriptive characterization of the object.

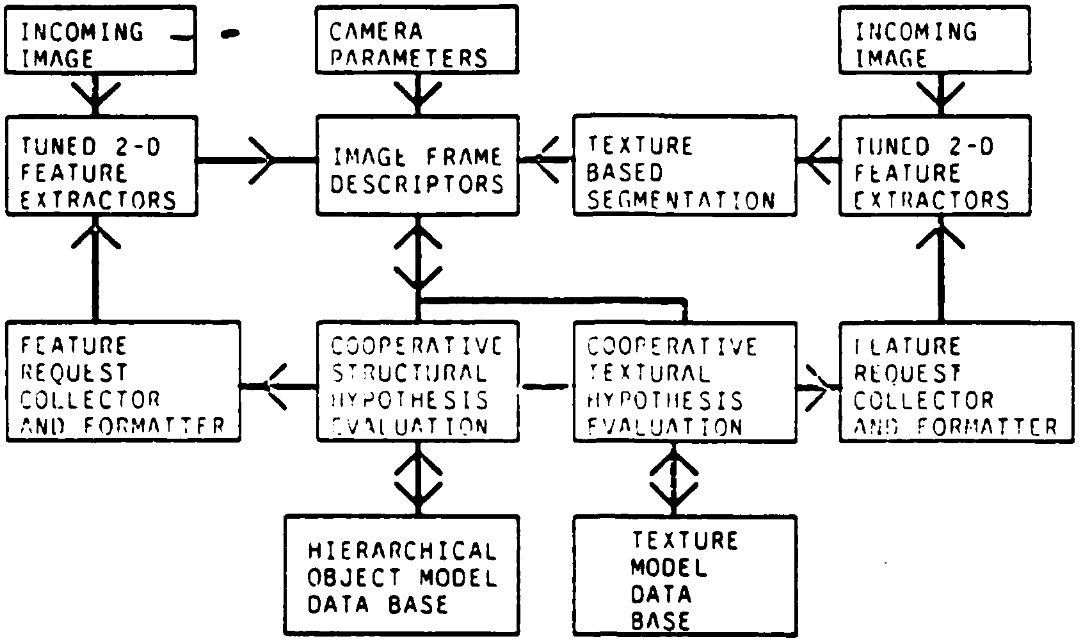

INCOMING IMAGE
CAMERA PARAMETERS
INCOMING IMAGE
TUNED 2-D FEATURE EXTRACTORS
IMAGE FRAME DESCRIPTORS
TEXTURE BASED SEGMENTATION
TUNED 2-D FEATURE EXTRACTORS
FEATURE REQUEST COLLECTOR AND FORMATTER
COOPERATIVE STRUCTURAL HYPOTHESIS EVALUATION
COOPERATIVE TEXTURAL HYPOTHESIS EVALUATION
FEATURE REQUEST COLLECTOR AND FORMATTER
HIERARCHICAL OBJECT MODEL DATA BASE
TEXTURE MODEL DATA BASE

Figure 4-21: Recursive image analysis system using texture and structure.

The proposed iterative structure is shown in figure 4-21 such that structure and texture analysis are unfolded into separate iterative loops sharing the same image frame descriptors (center of diagram). In an actual implementation, these might be folded together into common hardware and software mechanisms.

In the laboratory, incoming images are received already sampled in space (x,y), spectral frequency (λ), and time (t). Additional camera parameters describing camera altitude, angle, cloud cover, latitude, and longitude are supplied to the internal image frame descriptors.

The proposed electro-optic feature extractors, simulated in the laboratory by array processors, provide the only access the system has to image pixels. Early boot processes are used to initiate extraction of simple edge and region information in order to initiate the formation of image hypotheses. Texture analysis has an additional segmentation step in which the extracted texture features are clustered into proposed regions, again iteratively.

Both structural and textural information is summarized within image frame descriptors. These active processes maintain internal feature representations in order to service questions from cooperative hypothesis evaluation processes which may be indexed by one of numerous regional or feature space keys.

The dynamic pool of hypothesis evaluation processes are built upon specific models pulled from a hierarchical object and texture data base. Valid scheduling requests inside the pool include activation of processes for the same model at different locations within the image (high level), spawning of subprocesses investigating specific possibilities down the object model hierarchy (lower level), or termination of hypotheses which cannot be supported relative to competing hypotheses for the same structure. All communication between processes investigating hypotheses occurs through the image frame descriptor processes or the process scheduler.

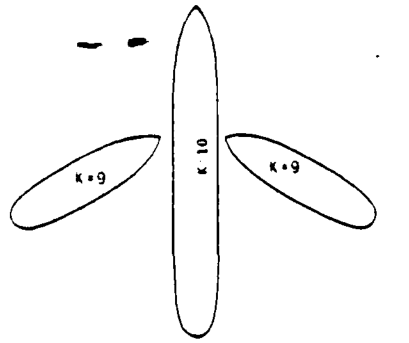
Questions which arise in the course of trying to support or deny a hypothesis are collected, condensed, and format prior to triggering feature evaluation. Requests from numerous evaluation processes must be condensed into a serial stream of feature requests such that the expectation value for feature evaluation is, perhaps suboptimally, minimized. The proper feature operator and parameters must then be prepared prior to the queuing of an evaluation request.

Two important functions are intentionally not shown in the diagrams. Data base information must somehow be acquired by the system, either through structured learning or direct data entry. Image analysis reports must be generated to provide system output based on the image frame descriptors. The reporting system may include filters to forward very limited kinds of data. Central research issues tied to recursive matching can be explored without these functions, directing limited manpower resources to tasks where basic research issues can be addressed.

4.7.1.2 An Example

The recursive matching structure can be illustrated by a simple example describing the detection and identification of a parked aircraft, as illustrated in figure 4-22 through 4-25. Initial feature extractors, such as low frequency MRI operators will help to locate candidate regions of interest. Numerous high level object hypotheses may arise from these operators. Although this example describes a structural analysis, similar operations might be used to describe a texture region.

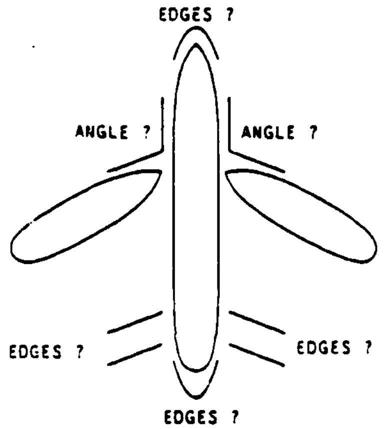
Shown in figure 4-22, high level analysis, perhaps corresponding to MRI levels k = 9, and k = 10 respectively identify three candidate regions that might describe an aircraft, truck, building or other object. Each high level hypotheses will in turn activate a process built around the appropriate high level object model. Object model evaluation by each process will result in numerous questions which help to support or deny the hypothesis. Requests by one evaluation process can be expected to provide clues to other hypothesis evaluation processes since each kernel function is run on the entire image.



HYPOTHESES: AIRCRAFT,
TRUCK, BUIDLING, ETC.

Figure 4-22: Initial region operators locate a structure suggesting an aircraft, truck, or building.

Either by direct request, or through the request of another process, additional detail describing the object will become available, perhaps describing the outline of the nose, tail, and horizontal stabilizers. No one feature results in an absolute identification, each merely adds or subtracts support for a given hypothesis. A narrow feature request by this process might pin down a spectral angle defined by the proposed wings and aircraft body. The resulting estimate, shown in figure 4-24 may lend enough substance to aircraft subclasses two and three that additional processes are activated, exploring these hypotheses.

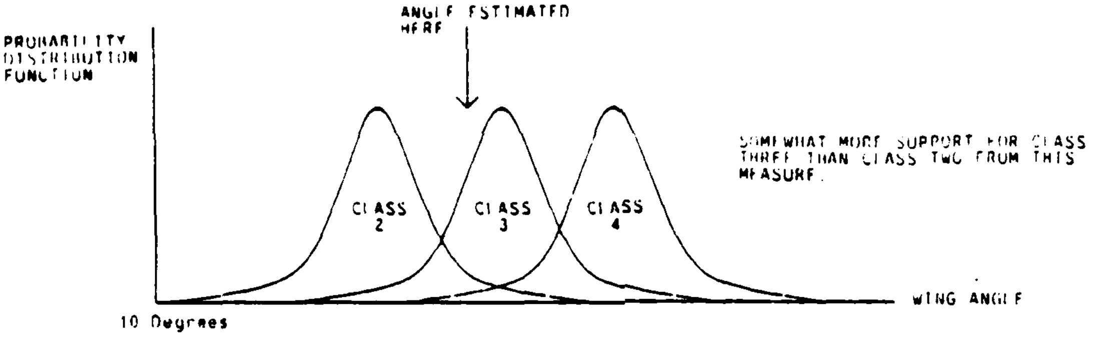


INVESTIGATION OF AIRCRAFT HYPOTHESIS
FIRES HYPOTHESES OF CLASS 2 OR 3.
(SOME FORM OF MEDIUM SIZE TRANSPORT)

Figure 4-23: Following the aircraft hypothesis, one process examines the hypothesized wing angle, nose, tail, and horizontal stabilizer structure.

As intermediate hypotheses are posted along with relative certainty of identification, processes based on incorrect hypotheses and high level hypotheses which have been replaced by low level hypotheses should deactivate, freeing resources for active pathways.

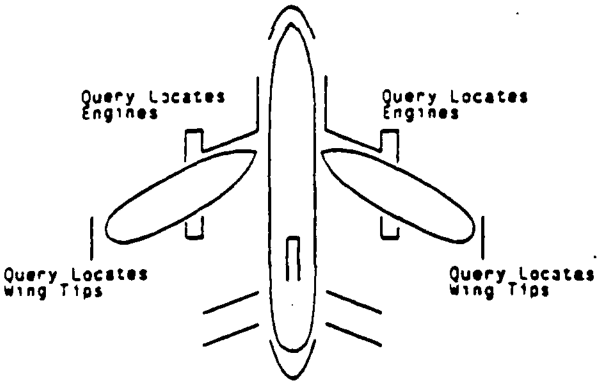
[illegible] incoming features by processes exploring aircraft classes two and three may in turn



SOMEWHAT MORE SUPPORT FOR CLASS THREE THAN CLASS TWO FROM THIS MEASURE.

Figure 4-24: For instance, the estimated wing angle lends support to aircraft class 3.

result in the class two process deactivating, and the class three process spawning processes exploring the possibility of a DC-8 or L-1011 aircraft based on engine cowling location and wing features. Shown in figure 4-25, this hierarchical processing would continue until reporting terminals were reached.



HYPOTHESES FIRED FOR DC-8 OR L-1011 AIRCRAFT.

Figure 4-25: Within the process investigating the class three hypothesis, search for the engine and wing length estimates triggers investigation of the possibility that the aircraft may be a DC-10 or L-1011

REFERENCES

[Brodatz 68] P. Brodatz.
Textures: A Photographic Album for Artists and Designers.
Reinhold, New York, 1968.

[Casasent 78] D Casasent.
Optical Data Processing for Engineers.
Electro-Optical Systems Design :26-36. February, 1978.

[Crowley 81] J.L. Crowley.
A representation for visual information.
PhD thesis. Carnegie-Mellon University, Nov., 1981.

[Crowley and Parker 84]
J.L. Crowley and A.C. Parker.
A representation of shape based on peaks and ridges in the difference of low pass transform.
IEEE Trans. on PAMI , March, 1984.

[Crowley and Sanderson 84]
J. L. Crowley and A. C. Sanderson.
Multiple Resolution and Probabilistic Matching of 2-D Grey-Scale Shape.
In *Proceedings 2nd IEEE Computer Society Workshop on Computer Vision, Representation, and Control*, pages 95-105. IEEE, May, 1984.

[Crowley and Stern 84]
J.L. Crowley and R.M. Stern.
Fast computation of the difference of low-pass transform.
IEEE Trans on PAMI , March, 1984.

[Haralick 70] R.M. Haralick.
Statistical and structural approaches to texture.
Proc. IEEE 67:786-804, 1970.

[Harwood et al 83]
D. Harwood, M. Subbarao, and L.S. Davis.
Texture Classification by local rank correlation.
Technical Report TR-1314, University of Maryland Computer Science, August, 1983.

[Laws 79] K.I. Laws.
Texture Energy Measures.
In *Proc. Image Understanding Workshop*, pages pp 47-51. 1979.

[Mandelbrot 77] B.B. Mandelbrot.
Fractals, Form, Chance, and Dimension.
W.H. Freeman and Co., San Francisco, 1977.

[Mandelbrot 82] B.B. Mandelbrot.
The Fractal Geometry of Nature.
W.H. Freeman and Co., San Francisco, 1982.

[Pentland 83a] A. Pentland.
Fractal-based description of natural scenes.
Proc. IEEE CVPR .pp 201-209. July, 1983.

[Pentland 83b] A. Pentland.
Fractal Textures.
Proc. IJCAI 1983 :pp. 973-981, 1983.
Karlsruhe, Germany.

5. IMAGE UNDERSTANDING TECHNIQUES FOR 3D SCENE INTERPRETATION

5.1 INTRODUCTION

In this chapter. we present results in two aspects of the 3D change detection task: the low-level problem of analyzing images. and the high-level problem of representing, constructing and updating the 3D scene model. For the low-level processing we describe a new method of computing the stereo correspondences which can be used to determine the 3D positions of points from a pair of aerial images. For the high-level processing, we describe methods of representing and constructing scene models from multiple views. using rangefinder data. The use of rangefinder data allows us to decouple the high-level processing problem from the low-level problem. for more efficient research into the high-level problems.

5.2 STEREO BY TWO-LEVEL DYNAMIC PROGRAMMING

5.2.1 Introduction -

Stereo is a useful method of obtaining depth information. The key problem in stereo is a search problem which finds the correspondence points between the left and right images. so that, given the camera model (ie., the relationship between the right and left cameras of the stereo pair), the depth can be computed by triangulation. In edge based stereo techniques, edges in the images are used as the elements whose correspondences to be found [Grimson and Marr 79, Baker and Binford 81, Baker 82, Barnard and Fischler 82]. Even though a general problem of finding correspondences between images involves the search within the whole image, the knowledge of the camera model simplifies this image-to-image correspondence problem into a set of scanline-to-scanline correspondence problems. That is, once a pair of stereo images is rectified so that the epipolar lines are horizontal scanlines, a pair of corresponding edges in the right and left images should be searched for only within the same horizontal scanlines. We call this search *intra-scanline* search. This intra-scanline search can be treated as the problem of finding a matching path on a two-dimensional (2D) search plane whose vertical and horizontal axes are the right and left scanlines. A dynamic programming technique can handle this search efficiently [Baker and Binford 81, Baker 82].

However, if there is an edge extending across scanlines, the correspondences in one scanline have strong dependency on the correspondences in the neighboring scanlines, because if two points are on a vertically connected edge in the left image, their corresponding points should, most likely, lie on a vertically connected edge in the right image. The intra-scanline search alone does not take into account this mutual dependency between scanlines. Therefore, another search is necessary which tries to find the consistency among the scanlines, which we call *inter scanline* search.

By considering both intra- and inter-scanline searches, the correspondence problem in stereo can be cast as that of finding in a three-dimensional (3D) search space an optimal matching surface that most satisfies the intra-scanline matches and inter-scanline consistency. Here, a matching surface is defined by stacking 2D matching paths, where the 2D matching paths are found in a 2D search plane whose axes are left image column position and right-image column position, and the stacking is done in the direction of the row (scanline) number of the images. The cost of the matching surface is defined as the sum of the costs of the intra-scanline matches on the 2D search planes, while vertically connected edges provide the consistency constraint across the 2D search planes and thus penalize those intra-scanline matches which are not consistent across the scanlines. Our stereo matching uses dynamic programming for performing both the intra-scanline and the inter-scanline searches, and both searches proceed simultaneously. This method reduces the computation to a feasible amount.

5.2.2 Use of Inter-scanline Constraints

As mentioned above, for a pair of rectified stereo images, matching edges within the same scanline (ie., the intra-scanline search) should be sufficient in principle. However, in practice, there is much ambiguity in finding correspondences solely by the intra-scanline search. To resolve the ambiguity, we can exploit the consistency constraints that vertically connected edges across the scanlines provide . Suppose a point on a connected edge u in the right image matches with a point on a connected edge v in the left image on scanline t. Then, other points on these edges should also match on other scanlines. If edges u and v do not match on scanline t, they should not match on other scanlines, either. We call this property inter-scanline consistency constraint. Thus, our problem is to search for a set of matching paths which gives the optimal correspondence of edges within scanlines under the inter-scanline consistency constraint.

A few methods have been used to combine the inter-scanline search with the intra-scanline search. Henderson [Henderson, et al. 79] sequentially processed each pair of scanlines and used the result of one scanline to guide the search in the next scanline. However, this method suffers in that the errors made in the earlier scanlines significantly affect the total results.

Baker [Baker 82] first processed each pair of scanlines independently. After all the intra scanline matching was done, he used a cooperative process to detect and correct the matching results which violate the consistency constraints. Since this method, however, does not use the inter scanline constraints directly in the search, the result from the cooperative process is not guaranteed to be optimal. Baker suggested the necessity of a search which finds an optimal result satisfying the consistency constraints in a 3D search space, but a feasible method was left as an open problem.

A straightforward way to achieve a matching which satisfies the inter-scanline constraints is to consider all matchings between connected edges in the right and left images. However since the typical number of connected edges is a few to several hundred in each image, this brute force method is usually infeasible.

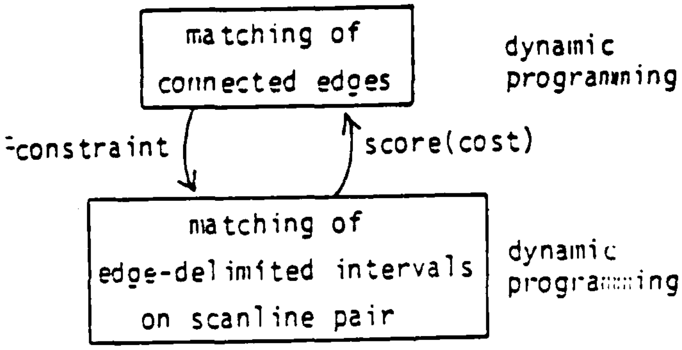


Figure 5-1: Two searches involved in stereo matching

We propose to use dynamic programming, which is used for the intra-scanline search, also for the inter-scanline search. These two searches are combined as shown in figure 5-1. One is for the correspondence of all connected edges in the right and left images, and the other is for the correspondence of edges (actually, intervals delimited by edges) on right and left scanlines under the constraint given by the former. The scheme to use dynamic programming in two levels was first employed in the recognition of connected spoked words [Sakoe 79]. They used one search for the possible segmentation at word boundaries and the other for the time-warping word matching under the constraint given by the former. In connected word recognition, however, the pattern to be processed is a single 1D vector. In our case, a connected edge crosses over multiple scanlines (ie., 1D vectors). This means that we need a 3D search space which is a stack of 2D search planes for intra-scanline matching.

Dynamic programming [Aho, Hopcroft and Ullman 74] solves an N-stage decision process as N single-stage processes. This reduces the computational complexity to the logarithm of the original combinatorial one. In order to apply dynamic programming, however, the original decision process must satisfy the following two requirements. First, the decision stages must be ordered so that all the stages whose results are needed at a given stage have been processed before then. Second, the decision process should be *Markovian*: that is, at any stage the behavior of the process depends solely on the current state and does not depend on the previous history. It is not obvious whether these properties exist in the problem of finding correspondences between connected edges in stereo images, but we clarify them in the following sections.

5.2.3 Correspondence Search Using Dynamic Programming

5.2.3.1 Intra-scanline search on 2D plane

The problem of obtaining correspondences between edges on the right and left epipolar scanlines can be solved as a path finding problem on a 2D plane. Figure 5-2 illustrates this 2D search plane. The vertical lines show the positions of edges on the left scanline and the horizontal ones show those on the right scanline. We refer to the intersections of those lines as nodes. Nodes in this plane correspond to the stages in dynamic programming where a decision should be made to select an optimal path to that node. In the intra-scanline search, the stages must be ordered as follows: *When we examine the correspondence of two edges, one on the right and one on the left scanline, the edges which are on the left of these edges on each scanline must already be processed.* For this purpose, we give indices for edges in left-to-right order on each scanline: $[0:M]$ on the right and $[0:N]$ on the left. Both ends of a scanline are also treated as edges for convenience. It is obvious that the condition above is satisfied if we process the nodes with smaller indices first. Legal paths which must be considered are sequences of straight line segments from node $(0,0)$ at the upper left corner to node (M,N) at the lower right corner on a 2D array $[0:M,0:N]$. They must go from the upper left to the lower right corners monotonically due to the above-mentioned condition on ordering. This is equivalent to the non-reversal constraint in edge correspondence: that is, the order of matched edges has to be preserved in the right and left scanlines. This constraint excludes from analysis thin objects such as wires and poles which may result in positional reversals in the image. A path has a vertex at node $\mathbf{m}=(m,n)$ when right edge m and left edge n are matched.

The cost of a path is defined as follows. Let $D(\mathbf{m},\mathbf{k})$ be the minimal cost of the partial path from node $\mathbf{k}$ to node $\mathbf{m}$. We denote $D(\mathbf{m},\mathbf{k})$ as $D(\mathbf{m})$ when $\mathbf{k}$ is $(0,0)$. $D(\mathbf{m})$ is the cost of the optimal path to node $\mathbf{m}$ from the origin $(0,0)$. The cost of a path is the sum of those of its primitive paths. A primitive path is a partial path which contains no vertices and it is represented by a straight line segment as

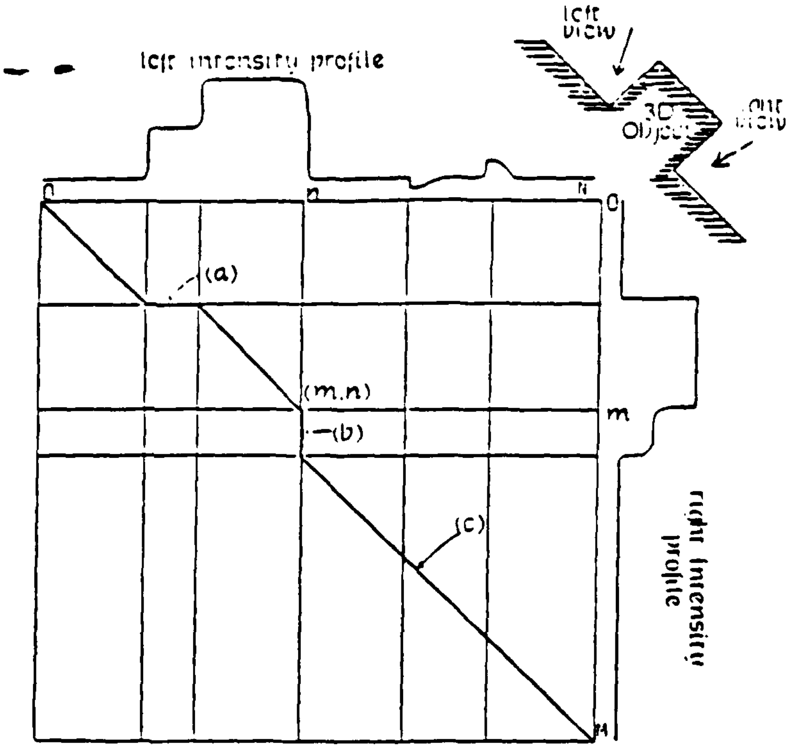


Figure 5-2: 2D search plane for intra-scanline search.
Intensity profiles are shown along each axis.
The horizontal axis corresponds to the left scanline and the vertical one corresponds to the right scanline. Vertical and horizontal lines are the edge positions and path selection is done at their intersections.

shown in figure 5-2. It should be noted that a primitive path actually corresponds to matching the intervals delimited by edges at the start and end nodes rather than edges themselves. Let $d(\mathbf{m},\mathbf{k})$ be the cost of the primitive path from node $\mathbf{k}$ to node $\mathbf{m}$. Obviously, $d(\mathbf{m},\mathbf{k}) \geq D(\mathbf{m},\mathbf{k})$ and on an optimal path $d(\mathbf{m},\mathbf{k}) \equiv D(\mathbf{m},\mathbf{k})$.

Now, $D(\mathbf{m},\mathbf{k})$ can be defined recursively as:

$$D(\mathbf{m},\mathbf{k}) = \min_{\{\mathbf{i}\}} \{d(\mathbf{m},\mathbf{m}-\mathbf{i}) + D(\mathbf{m}-\mathbf{i},\mathbf{k})\}$$

$$D(\mathbf{k},\mathbf{k}) = 0 \qquad (5.1)$$

$$\text{where} \quad \mathbf{m}=(m,n), \quad \mathbf{k}=(k,l), \quad \mathbf{i}=(i,j),$$

$$0 \leq i \leq m-k, \quad 0 \leq j \leq n-l, \quad i+j \neq 0.$$

Vector $\mathbf{i}$ represents a primitive path arriving to node $\mathbf{m}$. When $i=0$, the primitive path is horizontal, as shown at (a) in figure 5-2. It corresponds to the case in which a visible part in the left image is occluded in the right image. When $j=0$ the primitive path is vertical, as shown at (b). When $i>1$

and/or $j > 1$, the primitive path skips or ignores beyond $i - 1$ and/or $j - 1$ edges on the right and/or left scanlines as shown at (c) in the figure. Such a path corresponds to the case where some edges have no corresponding ones on the other scanline because of noise.

The path with cost $D(M, O)$ gives the optimal correspondence between a pair of scanlines.

5.2.4 Inter-scanline search in 3D space

The problem of obtaining a correspondence between edges under the inter-scanline consistency constraints can be viewed as the problem of finding a set of paths in a 3D space which is a stack of 2D planes for intra-scanline search. Figure 5-3 illustrates this 3D space. The side faces of this space correspond to the right and left images of a stereo pair. The cost of a set of paths is defined as the sum of the costs of the individual paths in the set. We want to obtain an optimal (ie., the minimal cost) set of paths satisfying the inter-scanline constraints. A pair of connected edges in the right and left images make a set of 2D nodes in the 3D space when they share scanline pairs. We refer to this set of 2D nodes as a single 3D node. The optimal path on a 2D plane is obtained by iterating the selection of an optimal path at each 2D node. Similarly, the optimal set of paths in a 3D space is obtained by iterating the selection of an optimal set of paths at each 3D node. Connected edges, 3D nodes, and sets of paths between 3D nodes are illustrated in figure 5-3.

As described in section 5.2.2, the decision stages must be ordered in dynamic programming. In the intra-scanline search, their ordering was straightforward; it was done by ordering edges from left to right on each scanline. A similar consideration must be given to the inter-scanline search in 3D space where the decision stages are the 3D nodes. A 3D node is actually a set of 2D nodes, and the cost at a 3D node is computed based on the cost obtained by the intra-scanline search on each 2D search plane. This leads to the following condition: *When we examine the correspondence of two connected edges, one in the right and one in the left image, the connected edges which are on the left of these connected edges in each image must already be processed.*

A connected edge u_1 is said to be on the left of u_2, if all the edges in u_1 on the scanlines which u_1 and u_2 share are on the left of those in u_2. The "left-of" relationship is transitive; if there is a connected edge u_3 and u_1 is on the left of u_3 and u_3 is on the left of u_2, then u_1 is on the left of u_2 (if u_1 and u_2 share any scanlines). The order of two connected edges which do not satisfy both of these relations may be arbitrarily specified. We assign an ordering index from left to right for every connected edge in an image. This ordering is possible without contradiction when a connected edge never crosses a scanline more than once and when two connected edges never intersect each other. Our edge-linking process which will be explained in section 4 is devised so that it does not produce such cases.

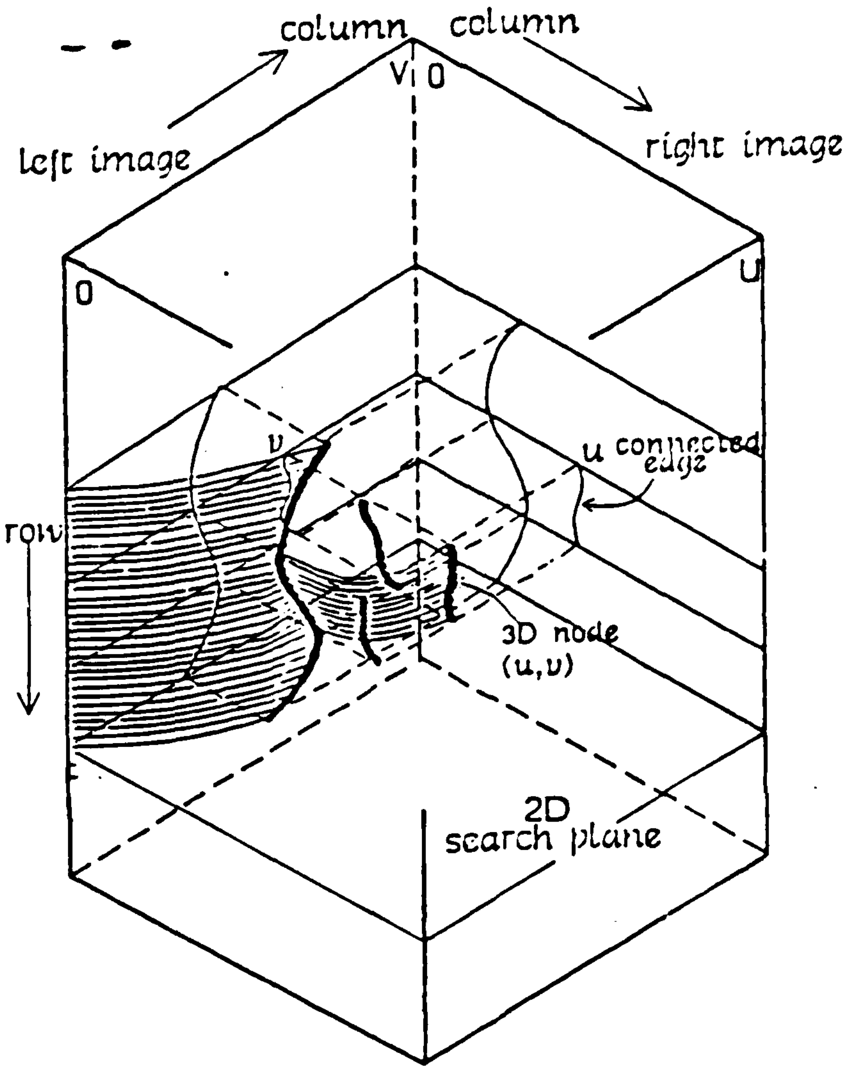


Figure 5-3: 3D search space for intra- and inter-scanline search. *This may be viewed as a rectangular solid seen from above. The side faces correspond to the right and left stereo images. Connected edges in each image form sets of intersections (nodes) in this space. Each set is called a 3D node. Selection of a set of paths is done at every 3D node.*

Now we will present how the cost of a 3D path is defined. Suppose we assign indices $[0:U]$ to connected edges in the right image, and $[0:V]$ in the left. The left and right ends of an image are treated as connected edges for convenience: the left ends are assigned index 0's. Let $\boldsymbol{u}=(u,v)$ be a 3D node made by a connected edge u in the right image and a connected edge v in the left image. Let $C(\boldsymbol{u})$ be the cost of the optimal set of paths which reach to the 3D node $\boldsymbol{u}$. The cost $C(\boldsymbol{u})$ is computed as follows:

$$C(\boldsymbol{u}) = \min_{\{i(t)\}} \sum_{t=s(\boldsymbol{u})}^{e(\boldsymbol{u})} \{D(I(\boldsymbol{u}:t), I(\boldsymbol{u}-\boldsymbol{i}(t):t):t) + C(\boldsymbol{u}-\boldsymbol{i}(t):t)\} \tag{5.2}$$

$$C(\boldsymbol{O}) = 0, \quad i.e., \quad C(0:t)=0 \quad \text{for all} \quad t$$

$$\textit{where} \quad \boldsymbol{u}=(u,v), \quad \boldsymbol{i}(t)=(i(t),j(t)), \quad 0 \le i(t) \le u, \quad 0 \le j(t) \le v, \quad i(t)+j(t) \neq 0.$$

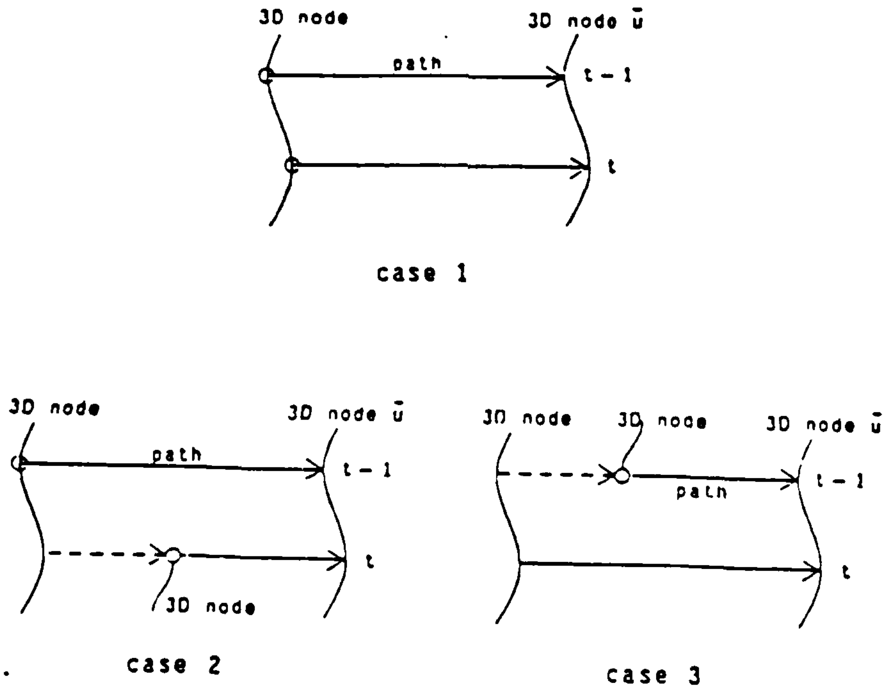
Here, $C(\boldsymbol{u}:t)$ is the cost of the path on scanline t in the optimal set; that is, $C(\boldsymbol{u})=\sum_{t=s(\boldsymbol{u})}^{e(\boldsymbol{u})} C(\boldsymbol{u}:t)$, and $D(\boldsymbol{m},\boldsymbol{k}:t)$ is the cost of the optimal 3D primitive path from node $\boldsymbol{k}$ to node $\boldsymbol{m}$ on the 2D plane for scanline t. A 3D primitive path is a partial path between two 3D nodes on a 2D search plane and it has no vertices at the nodes belonging to a 3D node. So a 3D primitive path is a chain of 2D primitive paths and an intra-scanline search is necessary to obtain the optimal 3D primitive path on a 2D plane between two given 3D nodes. The function $I(\boldsymbol{u}:t)$ gives the index of a 2D node belonging to the 3D node $\boldsymbol{u}$ on the 2D plane for scanline t. The numbers $s(\boldsymbol{u})$ and $e(\boldsymbol{u})$ specify respectively the starting and ending scanlines between which the 3D node $\boldsymbol{u}$ exists. The cost $C(\boldsymbol{u})$ is minimized on the function $\boldsymbol{i}(t)$. A 3D node $\boldsymbol{u}-\boldsymbol{i}(t)$ gives the start node of the 3D primitive path on scanline t. The inter-scanline constraint is represented by $\boldsymbol{i}(t)$. For example, if $\boldsymbol{i}(t)$ is independent of $\boldsymbol{i}(t-1)$, there are no constraints between scanlines and the search represented by equation ((5.2)) becomes equivalent to a set of intra-scanline searches which are performed independently on each scanline. Intuitively, $\boldsymbol{i}(t)$ must be equal to $\boldsymbol{i}(t-1)$ in order to keep the consistency constraint.

The iteration starts at $\boldsymbol{u}=(0,0)$ and computes $C(\boldsymbol{u})$ for each 3D node $\boldsymbol{u}$ in ascending order of $\boldsymbol{u}$. At each 3D node the $\boldsymbol{i}(t)$'s which give the minimum are recorded. The sequence of 2D primitive paths which forms the 3D primitive path is also recorded on each scanline. The set of paths which gives $C(\boldsymbol{U})$ at the 3D node $\boldsymbol{U}=(U,V)$ (which is the 3D node formed by the right ends of stereo images) is obtained as the optimal set.

It should be noted that when there are no connected edges except for the right and left sides of the images. the algorithm ((5.2)) works as a set of intra-scanline searches repeated on each scanline independently. In this sense. the 3D algorithm completely contains the 2D one.

5.2.4.1 Consistency constraints in inter-scanline

Using the term 3D node defined in the previous section, we can describe the inter-scanline consistency constraints as follows: *For any 3D node. either all corresponding 2D nodes are the vertices on the set of paths in the 3D search space. or none of them are the vertices on the set of paths.* We need to represent this constraint as the relation between $i(t)$ and $i(t-1)$ in equation (5 2). To do this. let us consider the example in figure 5 4 Suppose we are trying to obtain a set of 3D primitive paths which reach to node u. In order to satisfy the consistency constraints above, all the starting points of these paths should be the same 3D node; that is $i(t)=i(t-1)$. The cases when the starting point is a different 3D node are shown as case 2 and case 3 in the figure. In case 2, a new 3D node appears at scanline t and the starting point changes to the new one. Of course, it is possible that the starting point does not change to the new 3D node. This will happen if the cost of the paths having vertices on the 3D node is higher than the cost of the paths not having vertices on it. In case 3, the 3D node $u-i(t-1)$ disappears on scanline t and the starting point is forced to move elsewhere.



case 1

case 2

case 3

Figure 5-4: Three cases for consistency constraint.

Let us denote the 3D node $u-i(t)$, from which the 3D primitive path starts and reaches to the 3D node u on scanline t, by $frm(u;t)$. Then the following rules should be satisfied in each case.

$$
\begin{aligned}
&case1: \quad frm(u;t) = frm(u;t-1)\\
&case2: \quad frm(frm(u;t);t) = frm(u;t-1)\\
&case3: \quad frm(u;t) = frm(frm(u;t-1);t-1)
\end{aligned}
\tag{5.3}
$$

The rules in case 2 and case 3 require that the decision at 3D node u depend on decisions at preceding 3D nodes. Unfortunately, a decision system with such a property is not *Markovian* as described in section 5.2.2, and therefore there is no guarantee of obtaining an optimal solution by using dynamic programming. This means if we search for a solution using dynamic programming with those rules, the result might be poorer than that of the 2D algorithm.

In order to assure optimality in dynamic programming, we modify the rules in ((5.3)) as follows.

$$
\begin{aligned}
&case1: \quad frm(u;t) = frm(u;t-1)\\
&case2: \quad frm(u;t) \geq frm(u;t-1)\\
&case3: \quad frm(u;t) \leq frm(u;t-1)
\end{aligned}
\tag{5.4}
$$

The new rule for case 2 requires that the new 3D node on scanline t be on the right of the 3D node that is the starting point on scanline $t-1$. For case 3, the new starting node on scanline t should be on the left of that on scanline $t-1$. It should be noted that though the new rules are always satisfied when the rules in equation ((5.3)) are satisfied, the converse is not true. Thus, under the new rules the consistency constraint might not be satisfied at all places. In other words the [illegible] represented by the rules in equation ((5.4)) are weaker than those of equation (5.3) [illegible] we can expect to obtain an optimal solution in dynamic programming, we can expect [illegible] the 3D search algorithm than by the 2D search algorithm.

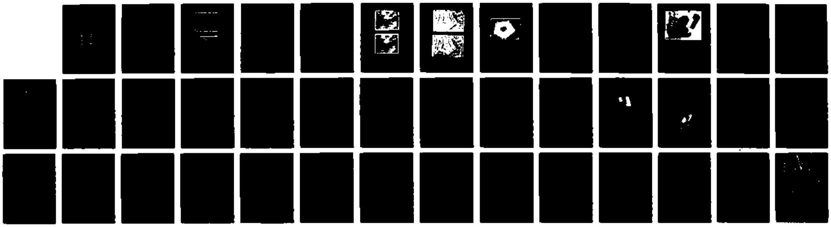
5.2.5 Experiments

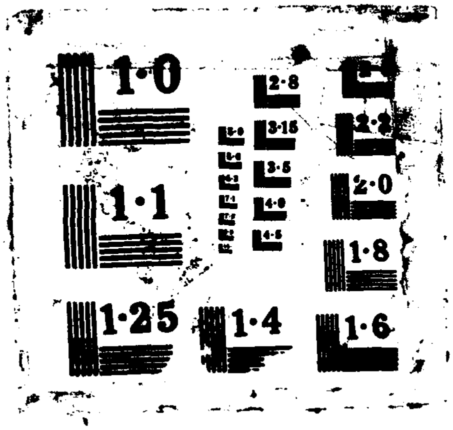
Implementation of the stereo algorithm which has been presented [illegible] edges and linking them, and a definition of similarity measures [illegible] details of the method of detecting edges [illegible]

The computation of cost in our search [illegible] search plane. We define the cost of [illegible]

AD-A185 286 MULTI-DISCIPLINARY TECHNIQUES FOR UNDERSTANDING TIME-VARYING SPACE-BASED..(U) CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF ELECTRICAL AND COM.. 2/2

UNCLASSIFIED D CASASENT ET AL. 10 MAY 85 F/G 12/9 NL




1·0
2·8
3·15
3·5
4·0
4·5
2·2
2·0
1·1
1·8
1·25
1·4
1·6

by edges in the right and left images on the same scanline. If we let $a_1 \ldots a_k$ and $b_1 \ldots b_l$ be the intensity values of the pixels which comprise the two intervals, then the mean and variance of all pixels in the two intervals are computed as:

$$m = \frac{1}{2}\left(\frac{1}{k}\sum_{i=1}^{k} a_i + \frac{1}{l}\sum_{j=1}^{l} b_j\right)$$
$$\sigma^2 = \frac{1}{2}\left(\frac{1}{k}\sum_{i=1}^{k} (a_i - m)^2 + \frac{1}{l}\sum_{j=1}^{l} (b_j - m)^2\right) \tag{5.5}$$

In the definition above, both intervals give the same contribution to the mean m and variance σ^2 even when their lengths are different. The cost of the primitive path which matches these intervals is defined as follows:

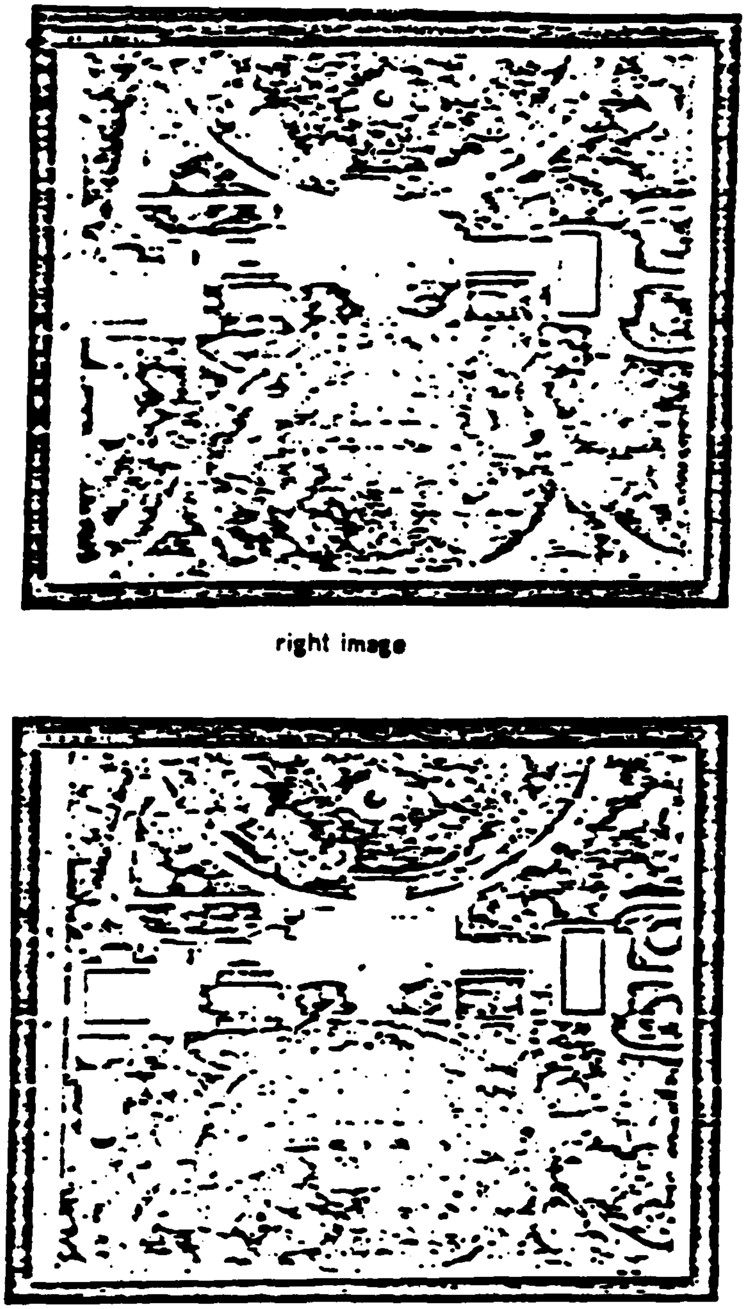
$$C_p = \sigma^2 \sqrt{k^2 + l^2} \tag{5.6}$$

We have applied our stereo algorithm to images from various domains including synthesized images, urban aerial images, and block scenes. Only the results of urban aerial images are presented here.

The stereo pairs used here are aerial photographs of the Washington, D.C. area. The first stereo pair is "white house" and the second one is "pentagon". They have been rectified using the camera models which was computed by Gennery's program [Gennery 79] using manually selected point pairs.

Figures 5-5, 5-6, and 5-7 show the original stereo pair, edges and connected edges, for the "white house" scene, respectively. The image size is 388x388 pixels and the intensity resolution is 8 bits. This example is an interesting and difficult one because it includes both buildings and highly textured trees. Many connected edges are obtained around the building while few are obtained in the textural part. The disparity maps obtained by the 2D and 3D search algorithms are shown in figure 5-8. Since the maps are registered in the right image coordinates, the disparity values for pixels on the right wall of the central building, which is visible in the right image but occluded in the left, are undetermined. Considerable improvements can be observed at the boundaries of buildings. In the textural part, the two algorithms provide approximately the same results.

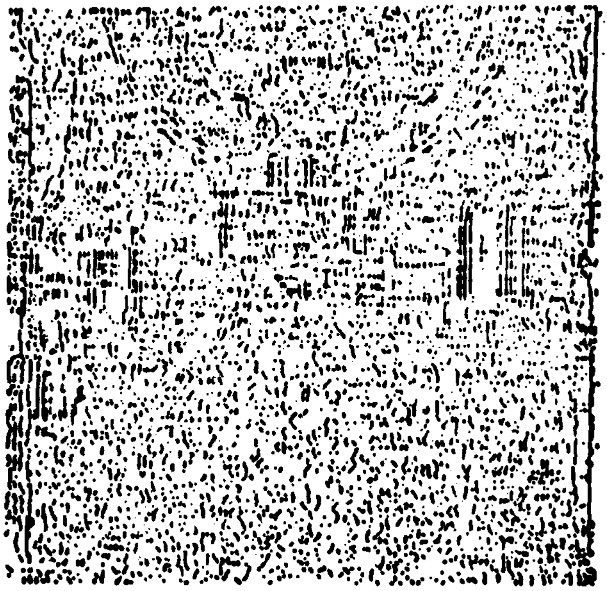
We counted the number of positions where the consistency constraint, described in section 5.2.4.1 is not satisfied. It is 436 in the 2D search and 32 in the 3D search. These numbers quantitatively show a significant improvement achieved by the 3D search algorithm. The reason the inconsistency is not completely removed in the 3D case is that we used "weaker" rules for the constraint as described earlier.



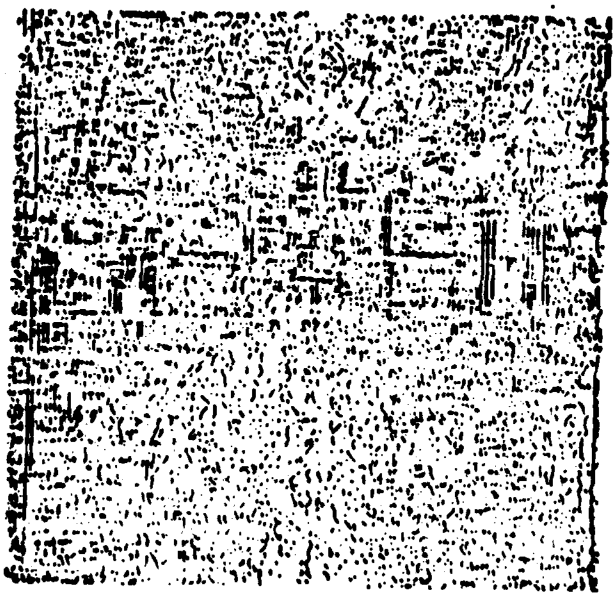
right image

left image

Figure 5-5: The "white house" stereo pair of urban aerial images.

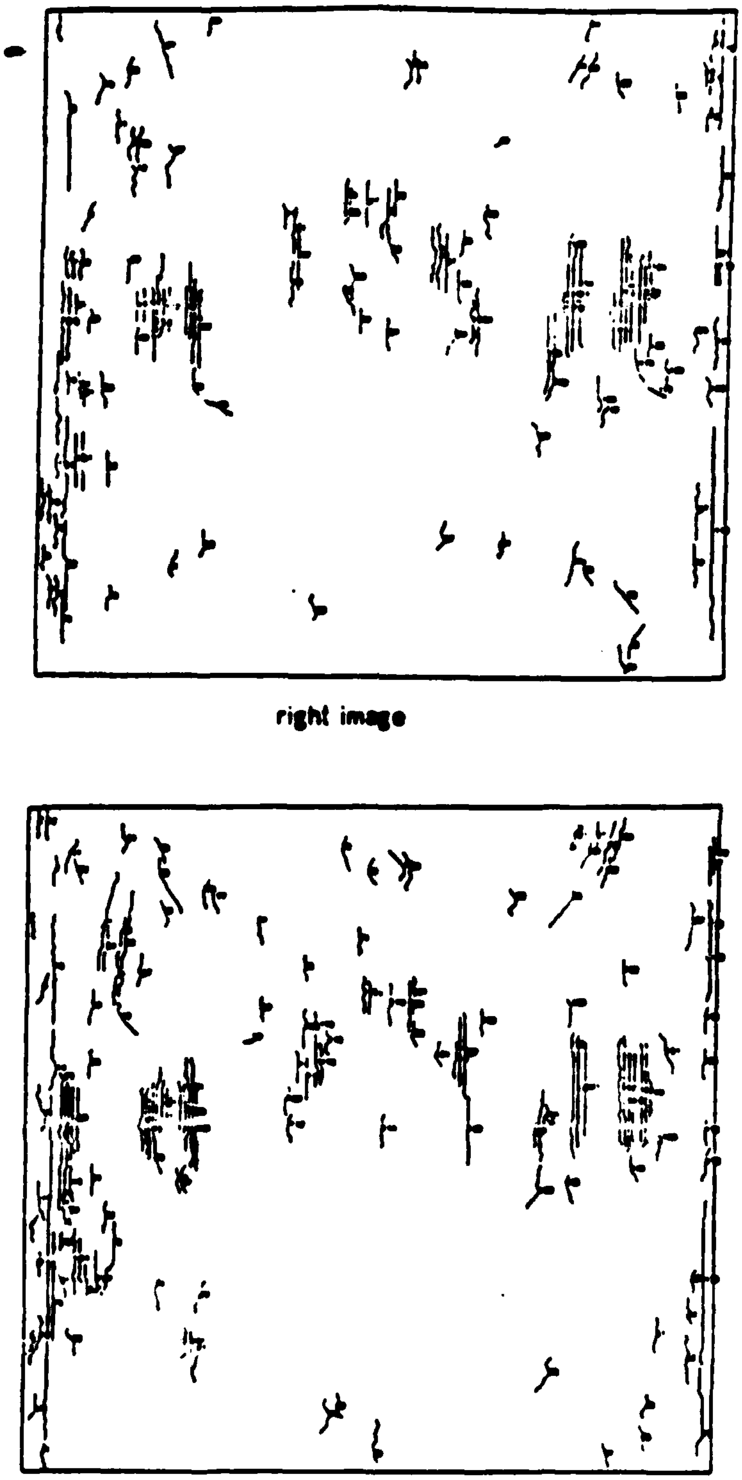


right image



left image

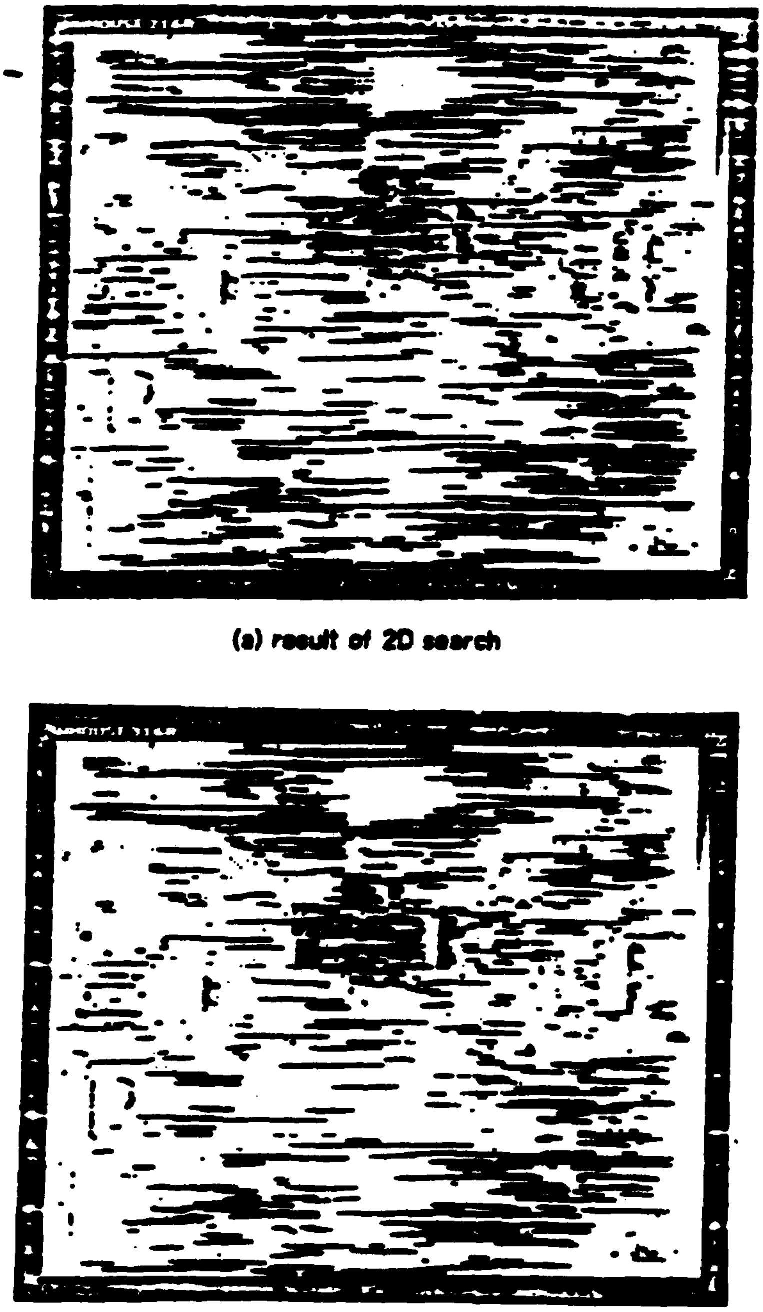
Figure 5-6: Edges extracted from the images in figure 5-5.



right image

left image

Figure 5-7: Connected edges obtained from figure 5-6.

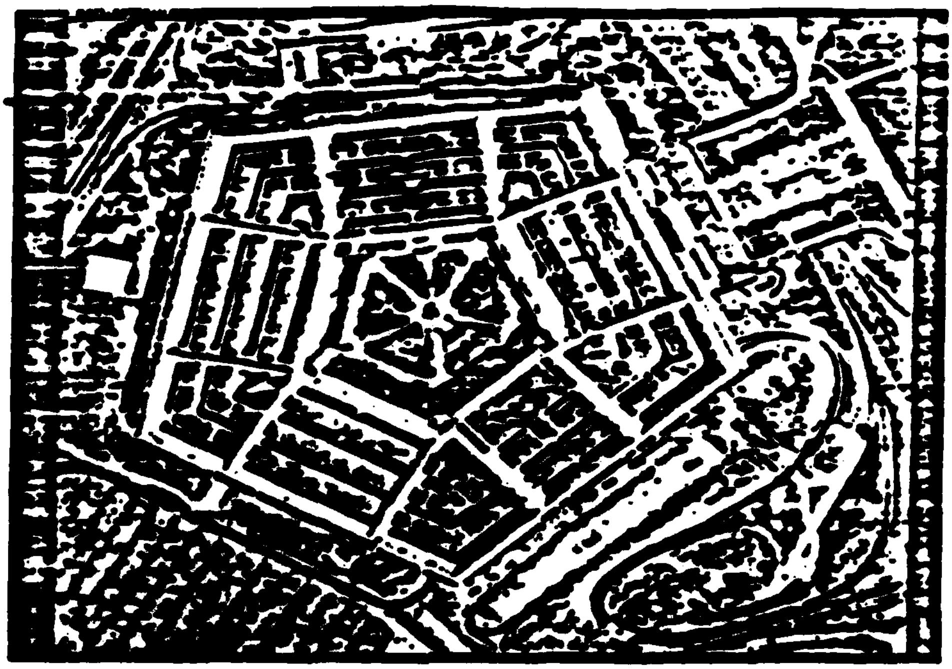


(a) result of 2D search

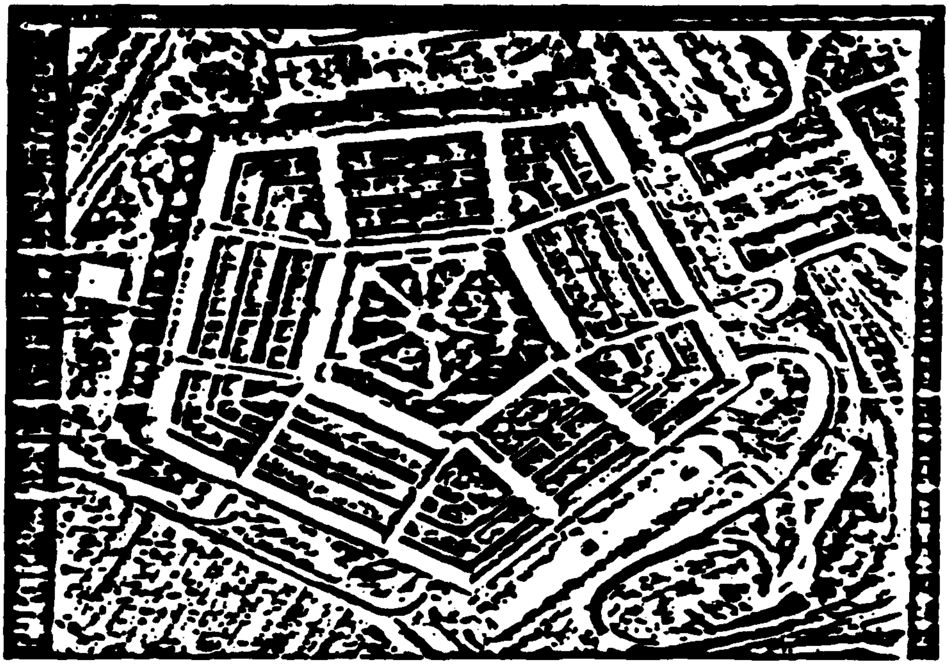
(b) result of 3D search

Figure 5-8: Disparity map obtained for the "white house" stereo pair (figure 5-5).

Both are registered in the right image coordinates.



right image



left image

Figure 5-9: Th: "pentagcn" stereo pair of urban aerial images.

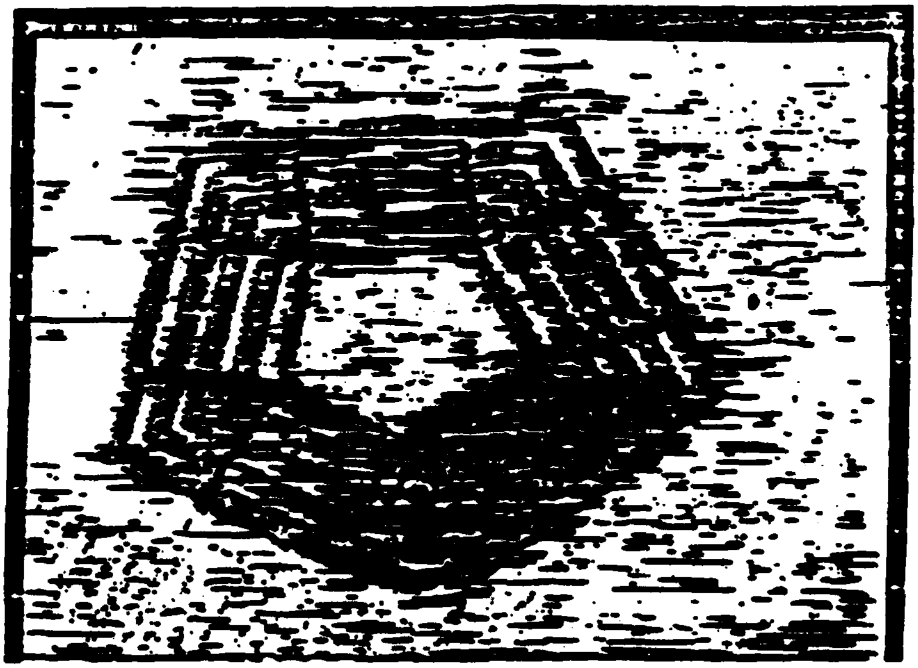


Figure 5-10: Disparity map obtained for the "pontagon" stereo pair (figure 5-9).
*This is registered in the left image coordinates.
Notice that the detailed structures of the building roof
and the bridge over the highway (upper left corner) have been recovered.*

5.3 GENERATING DETAILED SCENE DESCRIPTIONS FROM RANGE IMAGES

5.3.1 Introduction

An important problem for robotics vision is that of generating a 3D description of an unknown scene from range data. The range data themselves, a set of 3D surface points, are often not useful for tasks such as model-based recognition and localization, model-based inspection and verification, and change detection.

The result of our research is a method to extract a compact, symbolic, three-dimensional description of polyhedral objects in a scene. Importantly, the descriptions are quite complete, that is, most of the visible faces, edges, and vertices are represented. Most previous attempts at range data analysis did not result in such complete descriptions [Agin 72, Duda, Nitzan, and Barrett 79, Oshima and Shirai 79, Smith and Kanade 84, Tomita and Kanade 84]. (An exception is the work of Sugihara [Sugihara 79].)

5.3.2 Approach

The overall goal of this research is to obtain a full symbolic description of a scene from range data obtained from multiple views. In our approach, each view is processed in sequence, and the 3D information obtained from each view is used to incrementally construct a model of the scene environment.

The main steps followed by the overall system are the following. A description of the scene, in terms of faces, edges, and vertices, is obtained from each view. Descriptions from separate views are then matched to obtain corresponding elements and to obtain the global coordinate transformation. This permits the separate descriptions to be merged, resulting in a more complete overall description of the scene. The matching and merging algorithms are described elsewhere [Herman 85]. Here, we will explain how the initial descriptions are obtained.

Two general approaches for segmenting range images are edge/line extraction [Smith and Kanade 84, Tomita and Kanade 84, Sugihara 79] and region extraction [Faugeras and Hebert 83, Duda, Nitzan, and Barrett 79, Oshima and Shirai 79]. Our method is based primarily on edge and line extraction because we are attempting to obtain complete, detailed descriptions of the faces, edges, and vertices in the scene. Furthermore, our matching algorithm assumes such complete descriptions. Such descriptions are more difficult to obtain when region segmentation methods are primarily used.

Our method involves the following steps: (1) acquire the range images using a light-stripe rangefinder. (2) find edge points in the image. (3) fit linear segments to the edge points using the Hough transform. (4) connect the segments by extending, shortening, or shifting them. (5) convert the lines and junctions in the image to 3D edges and vertices. (6) generate faces from the edges in the scene.

It is interesting to note that although we are working with 3D data, most of the steps in the algorithm are performed in the 2D image space. This is because algorithms for 2D are often simpler and more efficient, in both space and time, than those for 3D. One example is finding lines with the Hough transform. The 3D version of the algorithm is much more expensive and complicated than the 2D version.

5.3.3 Range Data Acquisition

The range data we use are obtained with the White Scanner light-stripe rangefinder. The illuminator is a laser which projects a vertical plane of light into the scene. The intersection of the plane of light with an object surface results in a light stripe, which is imaged by a camera lying to the left of the illuminator. The further a surface point on the stripe is from the illuminator, the further to the left it will be seen in the camera image. The rangefinder determines the position of the stripe at each camera scan line, and triangulation is used to obtain the 3D coordinates at these positions. The result is represented as a column vector. When the illuminator is swept across the field of view, we obtain a sequence of such column vectors, one for each stripe. The sequence of columns forms a range image (actually a set of images, one each for a binary mask and for x, y, and z values).

Fig. 5-11a shows the mask image for a polyhedral object. Each column in the image corresponds to a column of light. The rows in the image correspond to camera scan lines. This is called a "mixed registration" [Smith and Kanade 84]. The geometry in this image cannot be treated as in a camera image, since it is formed differently. However, the outline of the objects in this image are very nearly the same as would be seen if the eye were placed at the illuminator. The object as seen from the camera is reconstructed in Fig. 5-11b.

5.3.4 Threo-dimensional Edge Detection

This section describes how points in the range image that arise from real scene edges are found. We consider three kinds of edge points (occluding, convex, and concave. Occluding edge points are located where there is a discontinuity in depth (i.e., the difference in z values between adjacent pixels exceeds a threshold) or where there is a boundary between data and no data regions.

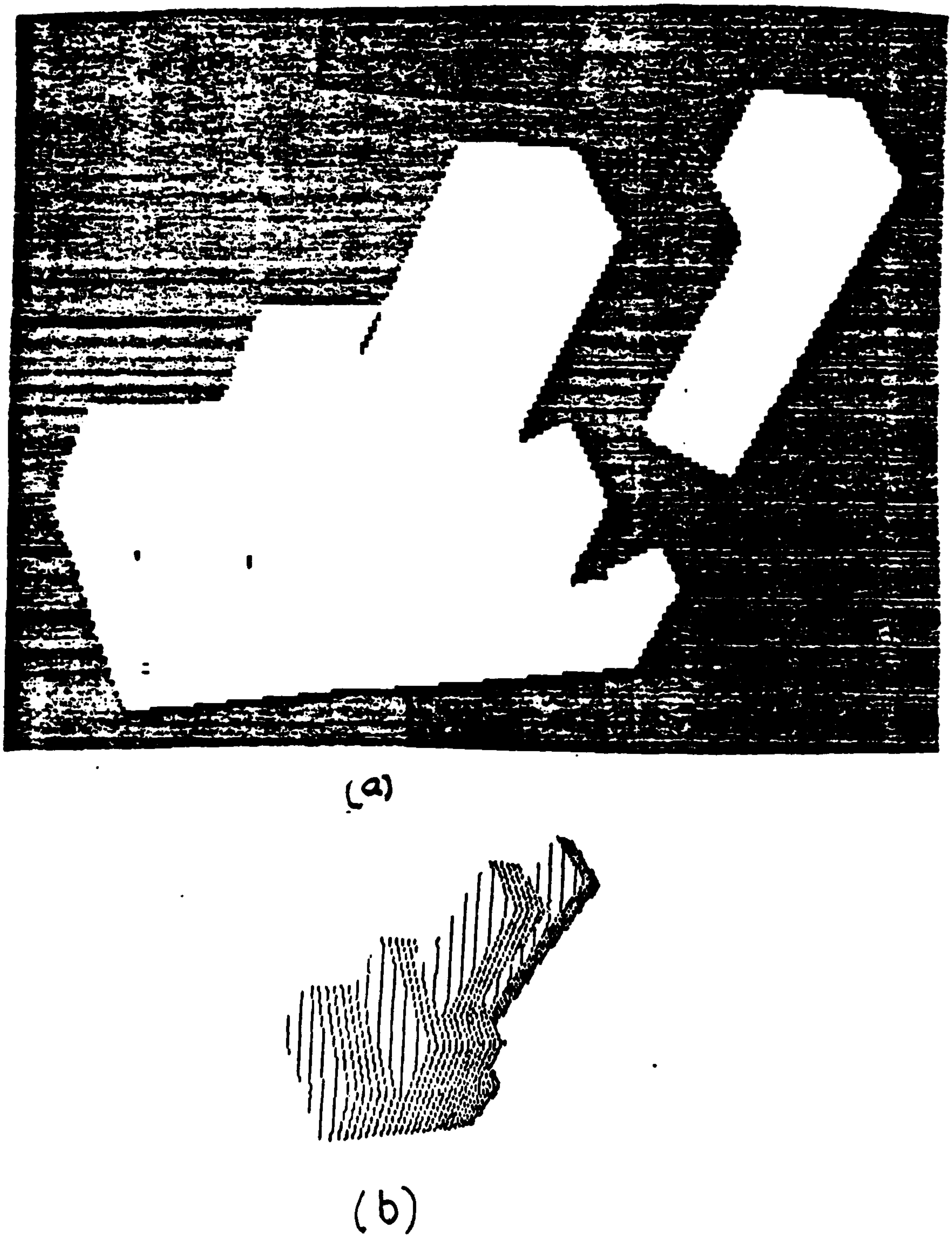


Figure 5-11: (a) Mask image in mixed registration. (b) Camera-reconstructed image.

Convex and concave edge points are found by examining, in turn, each column in the range image, and calculating the 3D curvature at each point. If the curvature is a local maximum or minimum and exceeds a threshold, the point is a concave or convex edge point, depending on whether the curvature is positive or negative. The results of this process are shown in Fig. 5-12 for two range images, where convex points are signified by " + ", concave points by "-", and occluding points by "•". These are the mixed registration images.

Notice that many occluding points have concave or convex points very near them. We believe that this is inherently due to the thickness of the light stripes [Yoshida 84]. Fig. 5-13a shows a vertical light stripe lying on a face with diagonal boundaries, as seen from the camera. As described above, at each scan line, the rangefinder chooses a point (which is probably near the center of the stripe thickness) to represent the position of the stripe. Since the stripe's appearance is beveled near the face boundaries, the center of the stripe is shifted. Since points on a stripe that are further to the left in the camera image are assumed to arise from scene points further from the illuminator, and vice versa, the measured light stripe in Fig. 5-13a results in a slight concavity near the top of the stripe, and a slight convexity near the bottom. In Fig. 5-13b, the results are just the opposite, with a convexity near the top of the stripe and a concavity near the bottom. In Fig. 5-12, this phenomenon occurs primarily on faces that are highly oblique with respect to the illuminator, since the stripes appear thicker when viewed from the camera.

5.3.5 Fitting Linear Segments

Once the edge points have been found, we want to fit linear segments to them. The Hough transform [Duda and Hart 72] is used here. However, the straight-forward method of choosing all cells in the Hough accumulator whose values exceed a threshold was not successful because clusters tend to cover several cells and they overlap, resulting in several extracted lines for each cluster. To get around this problem, as soon as a line is extracted, the effects on the accumulator of all the edge points corresponding to the line are eliminated. The algorithm we use is the following.

1. Transform each point in the edge image to a sinusoidal curve in the r-θ accumulator space.

2. Choose the accumulator cell (r, θ) with the largest value. If the value is less than a threshold, exit.

3. Find linear clusters of points in the edge image that represent line segments along the line (r, θ). This is done by searching for points in the edge image within some thickness t of the line (r, θ), and determining which of these points cluster together. Each resulting line segment is defined by it's 2D end points, its 3D end points, and its 3D line parameters.

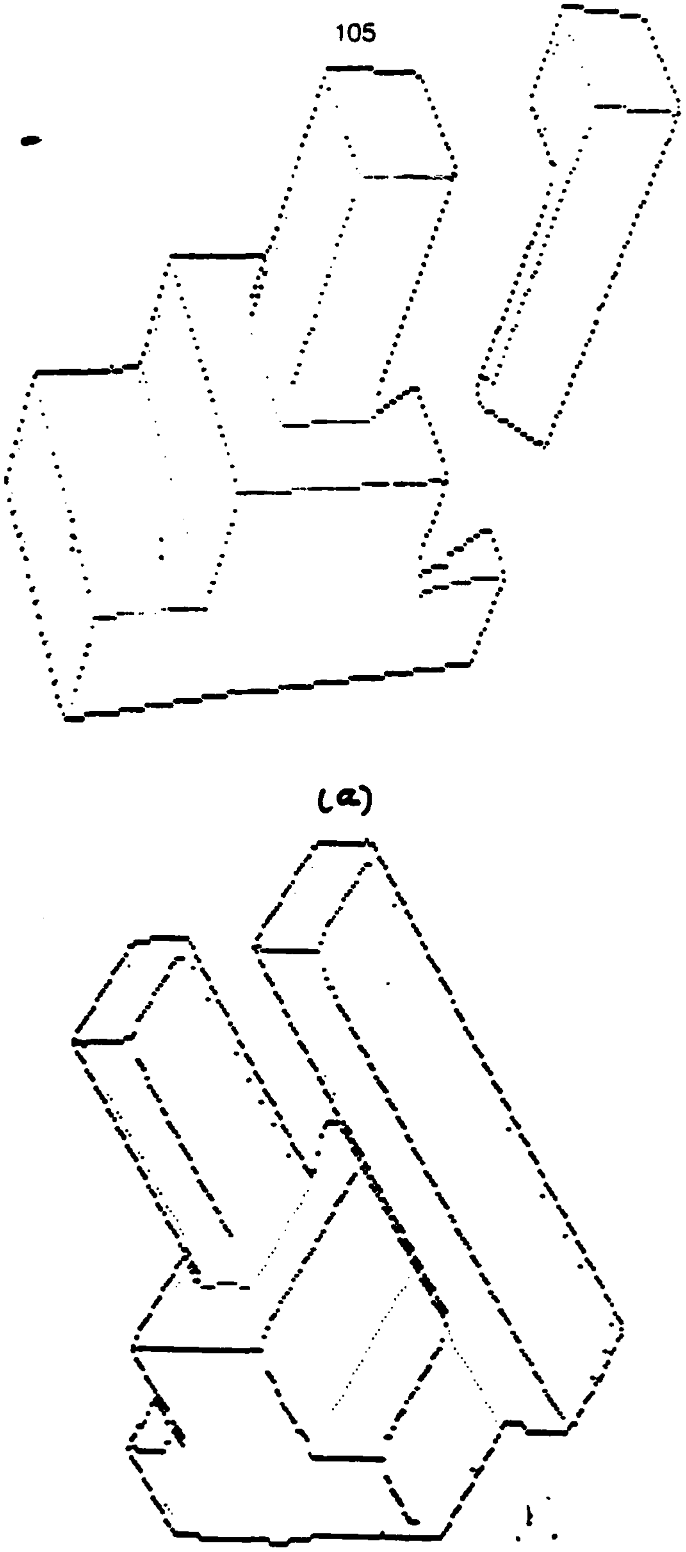


Figure 5-12: Edge points registered with range images. Convex points are " + ", concave points are " - " and occluding points are " • ". (a) Same image as Fig. 5-11a. (b) Edge points of another range image.

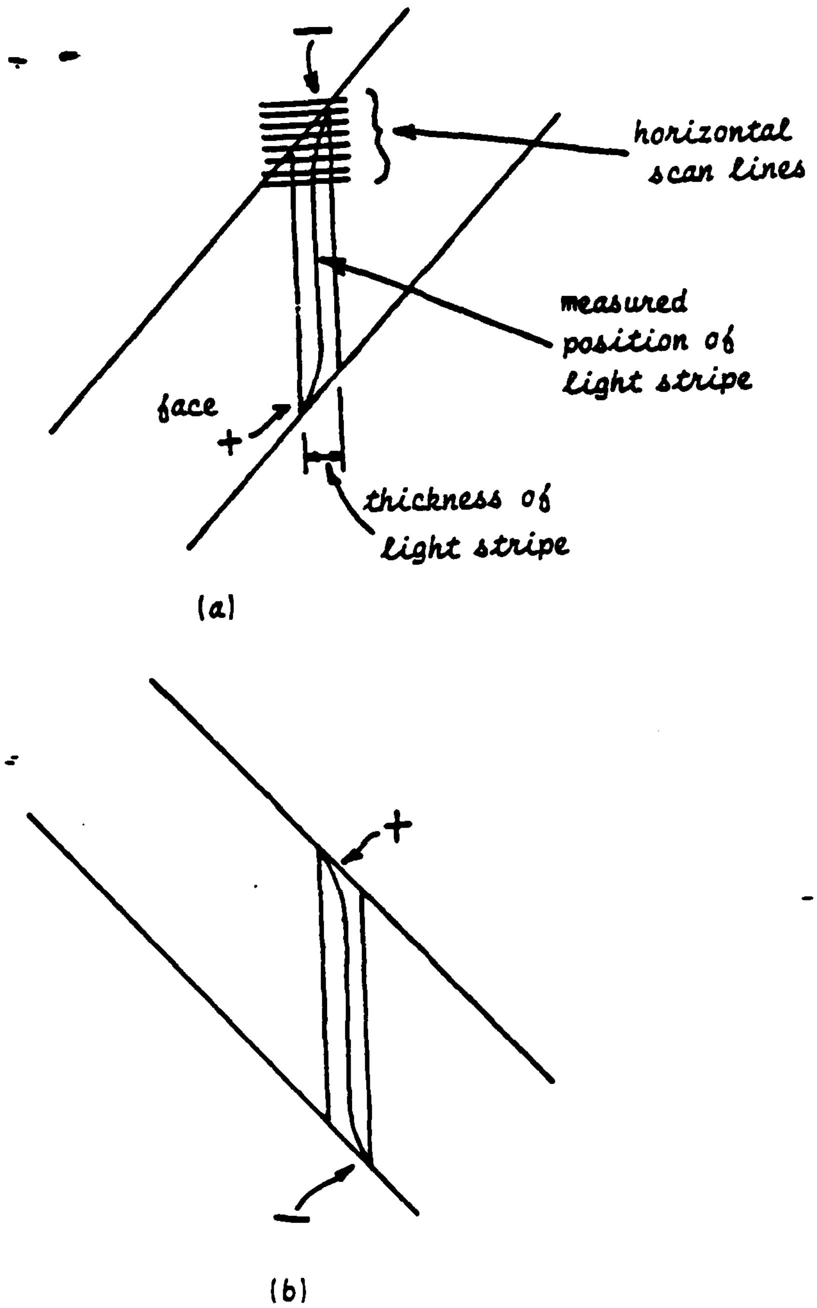


Figure 5-13: Camera viewpoint of vertical stripes lying on faces with diagonal boundaries. The measured stripe position results in a concavity or convexity near its top and bottom.

4. Eliminate the effect on the accumulator of the points lying on the line (r, θ). An efficient way to accomplish this is to decrement each accumulator cell lying on the sinusoidal curve corresponding to each point.

5. Go back to step 2.

Each class of edge points (i.e., convex, concave, and occluding) is treated separately and independently. In this way, the resulting line segments can be given the same class labels. Also, the direction of the occluding arrow for each occluding line is determined as a unit vector in the image plane. (The occluding surface is on the right side of the arrow.) This is done by comparing the average z values of points on either side of the line segment that are very near the segment. The results of the line fitting are shown in Fig. 5-14, where occluding lines are represented by solid lines, concave lines by dashed lines, and convex lines by dot-dash lines.

As explained earlier, some convex or concave edge points may lie near occluding edge points. This may result in convex or concave line segments near occluding line segments. These segments should be eliminated since they do not correspond to real scene features; they are an artifact of the range finding process. Fig. 5-14 actually shows the result after such segments have been deleted.

5.3.6 Connect Lines and Form Junctions

Although the basic line segments forming the edges of the object have now been extracted, as shown in Fig. 5-14, there are still many gaps and inaccuracies near the junctions of the object. Our next step is therefore to fill in these gaps and form junctions where necessary. This is done in three steps. First, segments that are close and almost collinear are connected. Second, (extended) segments that intersect and are close are connected. Third, if a segment has a dangling (i.e., unconnected) end point, an attempt is made to connect it with other segments as in the second step, but using a larger threshold than in this step.

In the first step, if two segments are almost collinear and have close end points (e.g., segments L1 and L2 in Fig. 5-14), a junction is formed at the point midway between these end points to connect the two segments.

In the second step, intersecting pairs of (extended) line segments are connected if the intersection point lies within a given threshold distance of the end points of the segments. We consider five cases here. In case 1, the intersection point lies outside the two segments (Fig. 5-15a). Both are extended and a junction is formed. In case 2, the intersection point lies inside both segments (Fig. 5-15b). Both are shortened and a junction is formed. In case 3, the intersection point lies outside one segment but

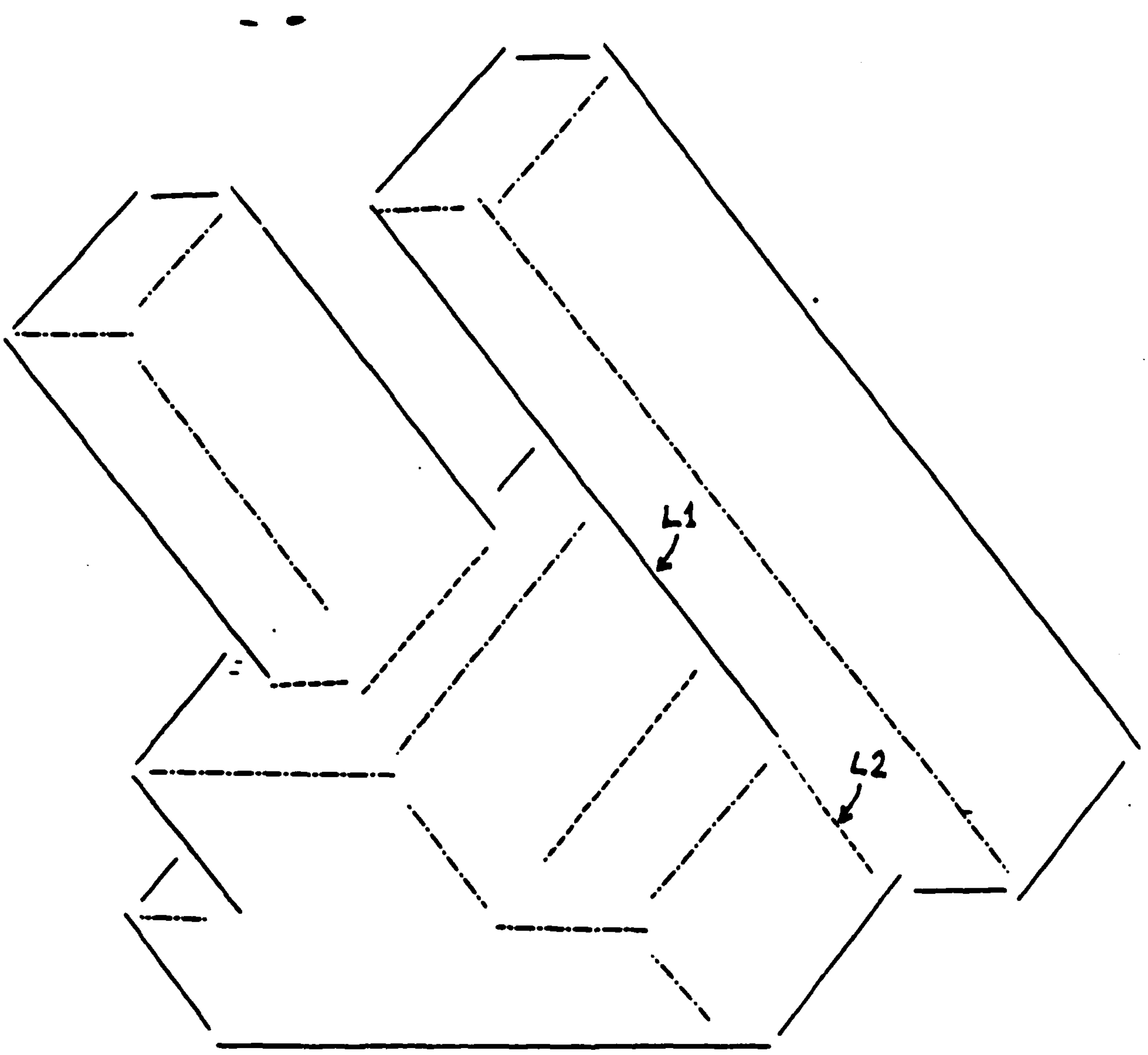


Figure 5-14: Result of fitting line segments to the edge image in Fig. 5-12b.Solid lines are occluding, dashed lines are concave, and dot-dash lines are convex.

inside the other (Fig. 5-15c). The former is extended, the latter is shortened, and a junction is formed. In case 4, the intersection point lies inside both segments (Fig. 5-15d), but is beyond the threshold distance from each end point of one of the segments. The other segment is therefore shortened, but a junction is not formed connecting the two segments. Case 5 is the same as case 4, except the intersection point lies outside one segment but inside the other (Fig. 5-15e).

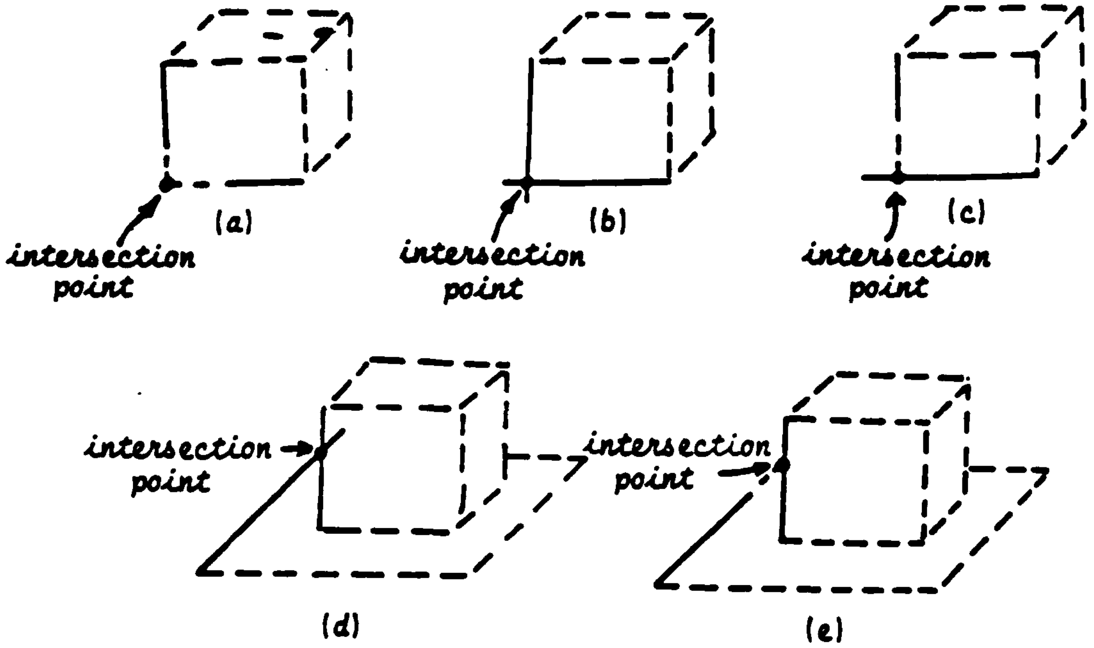


Figure 5-15: Connecting intersecting pairs of (extended) line segments.

The thresholds used in the first two steps just described are conservative and are only meant to connect segments that are quite close to each other. If liberal thresholds were used, connections would be established between segments that should not be connected. The result after these first two steps is shown in Fig. 5-16. Note that the two end points P1 and P2 in the figure seem to be "dangling." In the third step, therefore, a top-down type of process is initiated. We assume that a dangling end of a segment should probably be extended or shortened by a larger amount than the previously specified thresholds. Intersections between such an (extended) segment and other segments are obtained, and the same tests and procedures as described in the second step are performed, except that larger thresholds are used.

As a result of this process, all gaps are eliminated. However, lines that should form a single junction often do not intersect at a single point, resulting in separate junctions. To merge such junctions, a rectangular window is placed at each junction point in the image, and all junctions within the window are replaced by a new junction defined by the average position of all the junctions. The result of this step is shown in Fig. 5-17. At this point, partially occluded segments are labeled as such. These are found by checking how many segments form each junction. If a junction is formed by only one segment, the segment is marked as incomplete. In Fig. 5-17, segments L1 through L5 are incomplete.

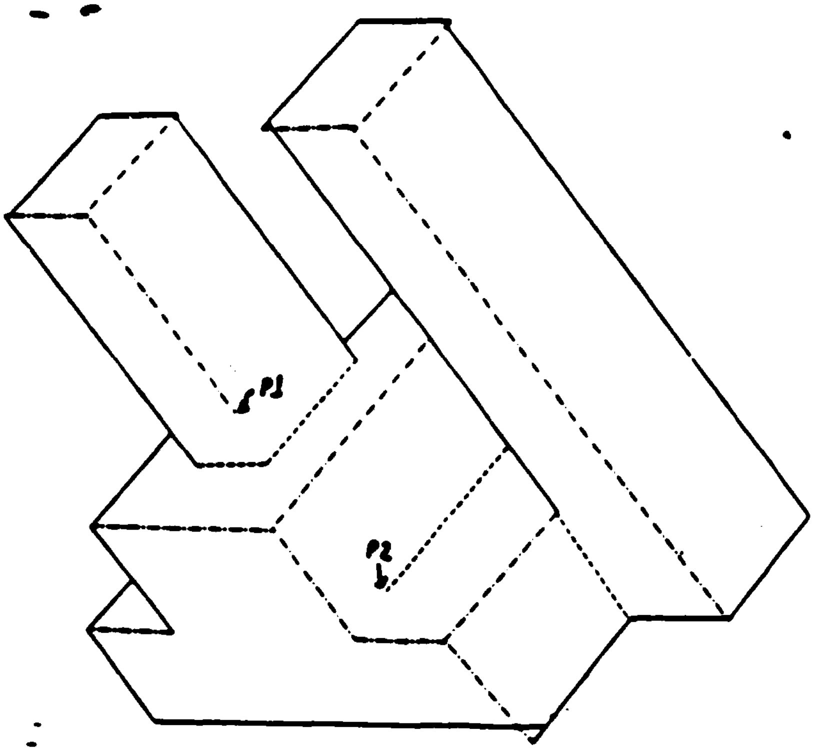


Figure 5-16: Result of connecting segments in Fig. 5-14 with conservative threshold.

5.3.7 Convert to 3D

Thus far, all of the processing has been in 2D, in the mixed registration image. In the next step, all the junctions and segments in the image are converted into 3D vertices and edges. Afterward, the 3D faces in the scene will be obtained.

The obvious method for getting the 3D position of some point in the image is to merely extract its x,y,z coordinates from the x,y,z range images. The problem with this method is that it can result in a large depth error even if there is a small error in the 2D position of the point. To see why, consider Fig. 5-18. Suppose that the position of a junction determined by the methods described above is at point *a*, but the true position of the junction is at point *b*. If we obtain the 3D coordinates of point *a* by extracting them from the x,y,z range images, we would really be extracting the coordinates of a point that lies significantly inside face A, resulting in a large error in the z coordinate. The more oblique face A is, the greater the error.

To overcome this problem, we use the known 3-space positions of the lines that were initially

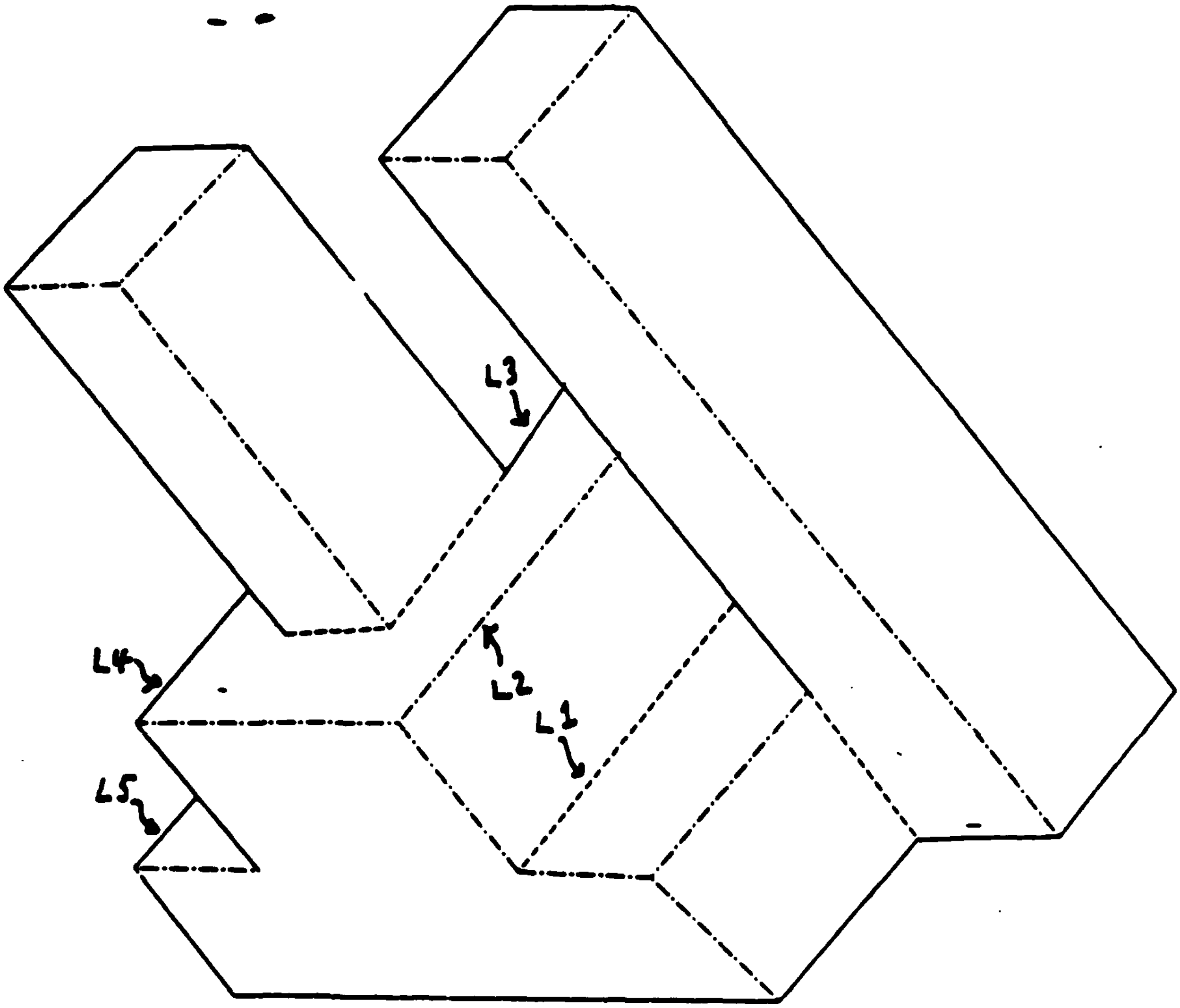


Figure 5-17: Result of extending dangling ends and merging junctions in Fig. 5-16.

extracted from the edge image (Fig. 5-14). To obtain the 3D position of a point in the image known to lie on a given line in 3 space, we calculate the intersection of the line with the plane of illumination corresponding to the column C in which the point lies (see Fig. 5-19).

Each junction in Fig. 5-17 has a pointer to a list of all the initially extracted segments (Fig. 5-14) that ultimately led to the junction. Fig. 5-20 provides an example of how the 3D coordinates of a junction J are determined. Suppose J was initially obtained by averaging the 3 intersection points of the segments L1, L2, and L3. To get its 3D position, J is first projected, in 2D, onto each of its segments,

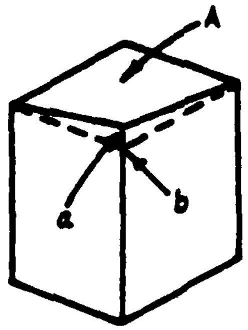


Figure 5-18: Locating the 3D position of a vertex directly from range data sometimes results in a large depth error

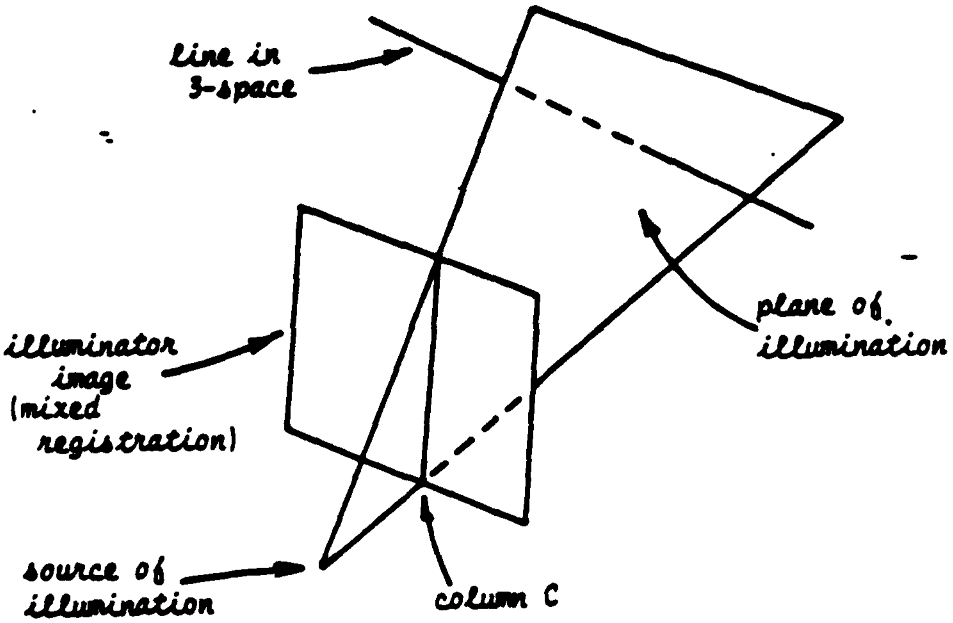


Figure 5-19: Calculating the intersection of a line in 3-space with the plane of illumination

resulting in the projected points *p1*, *p2*, and *p3*. Assuming that each of these points lies on its respective line in 3-space, their 3D positions are obtained as described above. The 3D position of the vertex corresponding to J is then obtained by averaging the 3D positions of *p1*, *p2*, and *p3*.

After each junction has been converted into a 3D vertex, it is simple to obtain the 3D line parameters of the edges connecting these vertices.

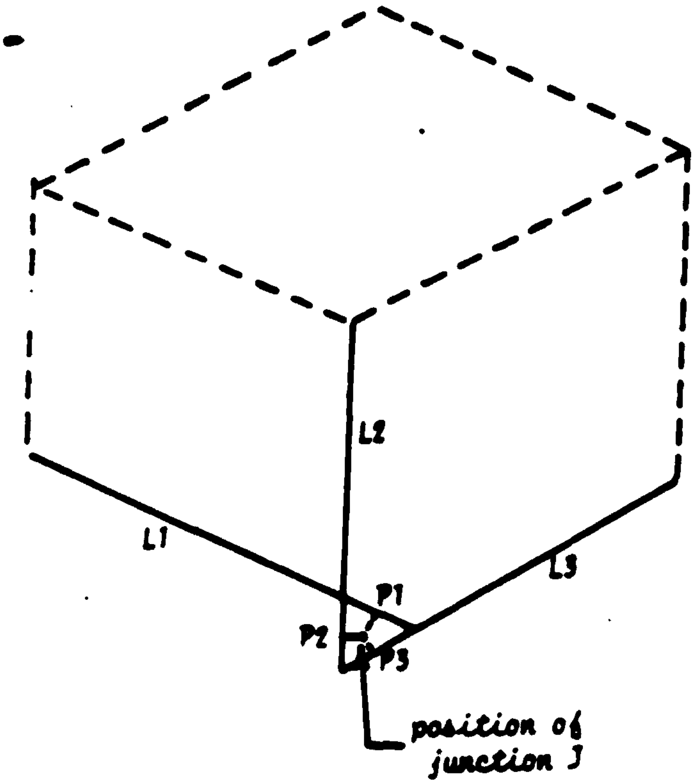


Figure 5-20: Determining the 3D coordinates of junction J

5.3.8 Generate 3D Faces

The next step is to extract the 3D faces in the scene. Since a face may be defined by the edges that bound it, faces are found by following (or traversing) their chains of edges. The edge traversal is arbitrarily chosen to be clockwise and is two-dimensional, that is, it occurs in the plane of the image.

Because an occluding edge belongs to one visible face, such edges must be traversed exactly once, and only in the direction of the occluding arrow Because concave and convex edges belong to two visible faces, such edges must be traversed twice, once in each direction. Fig 5-21 shows how the faces of an object may be recovered by traversing the edges. We distinguish two classes of faces, those with all their edges visible (e.g., face A in Fig. 5-21) and those with partially or totally occluded edges (e.g., face B in Fig. 5-21). Faces in the first class are found by a complete traversal of their edges. The traversal can therefore begin with any edge on the face. Since faces in the second class are found by a partial traversal of their edges, we must make certain that the edge with which a traversal begins will permit all visible edges to be included.

A face traversal occurs as follows. First an edge, called the "seed" edge, is chosen along with a

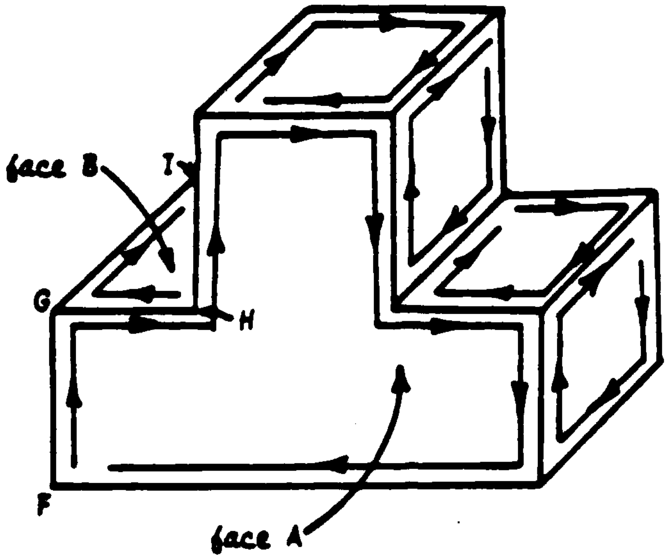


Figure 5-21: Recovering the faces by traversing their edges.

traversal direction along that edge. The method of choosing the edge and direction will be described shortly. To obtain the next edge in the traversal, a simple test is used to determine the next most clockwise edge in the traversal direction. In Fig. 5-21, for example, if the edge FG is the seed edge and the traversal direction is from F to G, then the next most clockwise edge is GH, and its traversal direction is from G to H. Successive edges are chosen in this manner until one of the following conditions is met: (1) an edge which has previously been traversed in the same direction is reached (e.g., a seed edge will be reached again for a totally visible face), (2) an incomplete (i.e., partially occluded) edge is reached (e.g., edge GI in Fig. 5-21), (3) an occluding edge whose occluding arrow is opposite to the traversal direction is reached. When one of these conditions is met, the traversal is terminated and all the traversed edges are assumed to belong to a single face.

Seed edges are chosen so that all faces with some occluded edges are processed before faces with no occluded edges. The algorithm proceeds as follows. First, incomplete occluding edges whose occluding arrows point away from the incomplete portion of the edge (e.g., edge AB in Fig. 5-22a) are found. Each of these edges is used as a seed, and the traversal direction is that of the occluding arrow. In Fig. 5-22a, the clockwise traversal results in the edges AB, BC, and CD, which are used to form face F. Next, each incomplete convex and concave edge (e.g., edges AB and EF in Fig. 5-22b) is used as a seed, and the traversal direction is from the incomplete to the complete portion of the edge.

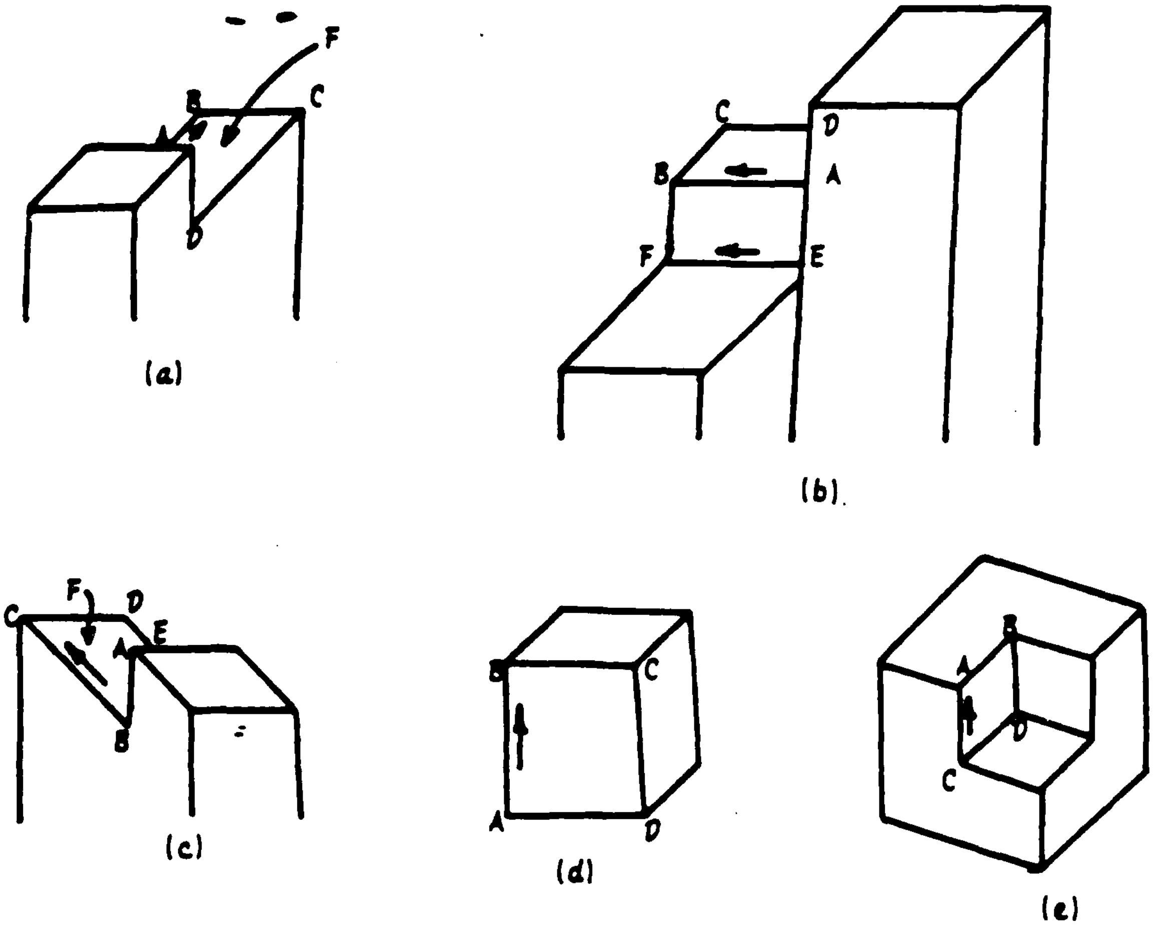


Figure 5-22: Finding seed edges to use in traversing faces.

Then, for each occluding edge in the scene (e.g., edge AB in Fig. 5-22c), find the next most clockwise edge as if traversing the occluding edge in the direction opposite to its occluding arrow. If this next edge is concave or convex (e.g., edge BC in Fig. 5-22c), it is used as a seed, and the traversal direction is the same direction used to find the edge.

At this point, the algorithm processes faces with no occluded edges. First, each complete occluding edge that has not yet been traversed (e.g., AB in Fig. 5-22d) is used as a seed, with the traversal direction the same as that of the occluding arrow. Finally, each complete concave or convex edge that has not been traversed in both directions (e.g., CA in Fig. 5-22e) is used as a seed, traversing in the direction(s) not yet traversed.

Each set of traversed edges is used to form a single face, and the 3D positions of the vertices connecting these edges are used to obtain the plane equation of the face. Although our current techniques will find inner edge chains (e.g., those that bound a hole in a face) as well as outer ones. the two sets of chains will not be associated together as belonging to a single face. Fig. 5-23 shows a perspective view of the final 3D description generated from the line drawing of Fig. 5-17.

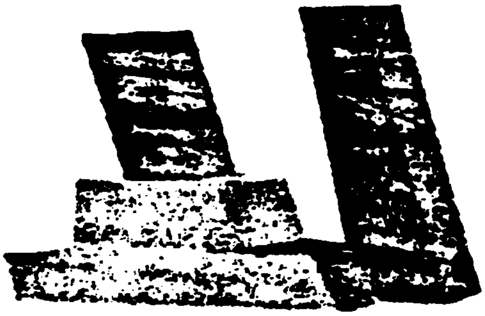


Figure 5-23: Perspective view of final 3D description generated from Fig. 5-17.

5.3.9 Multiple Views

The processing result for the edge image in Fig. 5-12a is shown in Fig. 5-24. The final 3D reconstruction is shown in Fig. 5-25. The two sets of range images discussed in this paper are two views of the same object. The next step in the processing will involve matching the two 3D models and merging them so as to generate a more complete model. The matching algorithm matches vertices in the two descriptions, and propagates constraints through the edges and faces. This is one reason why it has been important to recover almost all vertices, edges, and faces in the scene.

5.4 SUMMARY

This chapter has presented results in both low-level and high-level aspects of the 3D change detection task. For low-level processing, a new method of determining stereo correspondences which are used in the computation of depth for a pair of aerial images was described. For high-level processing, we have described our methods of representation and construction scene models from multiple views. We have bypassed the low-level problems by using rangefinder data as our input for the high-level processing.

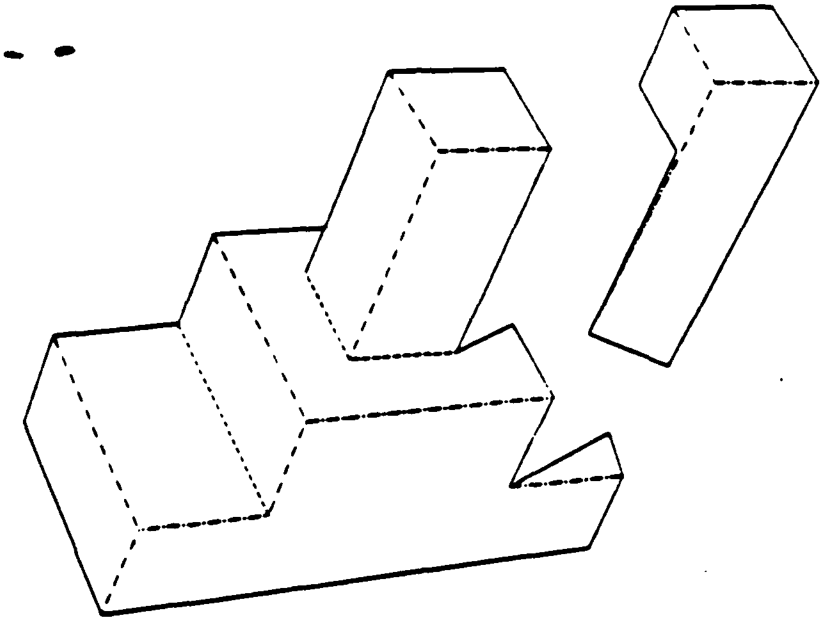


Figure 5-24: Final line drawing obtained from the edge image in Fig. 5-12a.

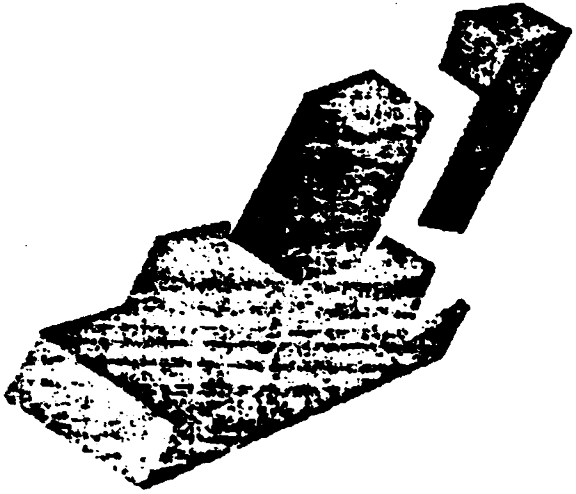


Figure 5-25: Perspective view of final 3D description generated from Fig. 5-24.

REFERENCES

[Agin 72] Agin, G. J.
Representation and Description of Curved Objects.
PhD thesis, Stanford University, October, 1972.

[Aho, Hopcroft and Ullman 74]
A. V. Aho, J. E. Hopcroft, and J. D. Ullman.
The Design and Analysis of Computer Algorithms.
Addison-Wesley, Reading, MA, 1974.

[Baker 82] Baker, H. H.
Depth from Edge and Intensity Based Stereo.
Technical Report AIM-347, Stanford Artificial Intelligence Laboratory, 1982.

[Baker and Binford 81]
Baker, H.H. and Binford, T.O.
Depth from edge and intensity based stereo.
In *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 631-636. Aug., 1981.

[Barnard and Fischler 82]
Barnard, S.T. and Fischler, M.A.
Computational Stereo.
Computing Surveys 14(4):553-572, Dec., 1982.

[Duda and Hart 72]
Duda, R.O. and Hart, P. E.
Use of the Hough Transformation to detect Lines and Curves in Pictures.
Communications of the ACM 15(1):11-15, January, 1972.

[Duda, Nitzan, and Barrett 79]
Duda, R.O., Nitzan, D., and Barrett, P.
Use of Range and Reflectance Data to Find Planar Surface Regions.
IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1(3):259-271, July, 1979.

[Faugeras and Hebert 83]
Faugeras, O.D. and Hebert, M.
A 3-D Recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces.
In *Proc. Eighth International Joint Conference on Artificial Intelligence*, pages 996-1002. August, 1983.

[Gennery 79] Gennery, D.
Stereo-camera calibration.
In *Proceedings of Image Understanding Workshop*, pages 101-107. DARPA, Nov., 1979.

[Grimson and Marr 79]
Grimson, W.E.L. and Marr, D.
A computer implementation of a theory of human stereo vision.
In *Proceedings of Image Understanding Workshop*, pages 41-47. DARPA, Apr., 1979.

[Henderson, et al. 79]
Henderson, R.L., Miller, W.J., and Grosch, C.B.
Automatic stereo reconstruction of man-made targets.
SPIE 186(6):240-248, 1979.

[Herman 85]
Herman, M.
Matching Three-Dimensional Symbolic Descriptions Obtained from Multiple Views of a Scene.
In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. June, 1985.
to appear.

[Medioni and Nevatia 83]
Medioni, G.G. and Nevatia, R.
Segment-based Stereo Matching.
In *Proceedings of Image Understanding Workshop*, pages 128-136. DARPA, June, 1983.

[Ohta and Kanade 83]
Ohta, Y. and Kanade, T.
Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming.
Technical Report CMU-CS-83-162, Carnegie-Mellon University, Computer Science Department, 1983.

[Oshima and Shirai 79]
Oshima, M. and Shirai, Y.
A Scene Description Method Using Three-dimensional Information.
Pattern Recognition 11:9-17, 1979.

[Sakoe 79]
Sakoe, H.
Two-level DP-Matching - A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition.
IEEE Trans. ASSP 27(6):588-595, 1979.

[Smith and Kanade 84]
Smith, D.R. and Kanade, T.
Autonomous Scene Description with Range Imagery.
In *Proc. Image Understanding Workshop*, pages 282-290. October, 1984.

[Sugihara 79]
Sugihara, K.
Range-Data Analysis Guided by a Junction Dictionary.
Artificial Intelligence 12 :41-69, 1979.

[Tomita and Kanade 84]
Tomita, F. and Kanade, T.
A 3D Vision System: Generating and Matching Shape Descriptions in Range Images.
The First Conference on Artificial Intelligence Applications . December, 1984.

[Yoshida 84] Yoshida, K.
1984
Personal communication.

6. PUBLICATIONS, PRESENTATIONS, AND STAFF SUPPORTED

6.1 STAFF SUPPORTED

Electrical and Computer Engineering --

D. Casasent (Professor), Principal Investigator

B.V.K. Vijaya Kumar (Assistant Professor), Associate Principal Investigator

Yeou-Lin Lin (Graduate Student)

R. Krishnapuram (Graduate Student)

The Robotics Institute --

Artnur C. Sanderson (Professor), Principal Investigator

John Willis (Graduate Student)

Nanda Alapati (Graduate Student)

Computer Science --

Takeo Kanade (Professor), Principal Investigator

Martin Herman (Research Associate)

Peter Highnam (Graduate Student)

Ellen Walker (Graduate Student)

6.2 PUBLICATIONS

Electrical and Computer Engineering -- (from start of contract)

1. B.V.K. Vijaya Kumar and C. Carroll, "Loss of Optimality in Cross Correlators", JOSA-A, Vol. 1, 1984, pp. 392-397.

2. D. Casasent and V. Sharma, "Feature Extractors for Distortion-Invariant Robot Vision", Optical Engineering, Vol. 23, September/October [1984, pp. 492-498.

3. B.V.K. Vijaya Kumar, "Lower Bound for the Suboptimality of Cross-Correlators", Applied Optics, Vol. 23, July 1984, pp. 2048-2049.

4. D. Casasent, A. Goutzoulis and B.V.K. Vijaya Kumar, "Time-Integrating Acousto-Optic Correlator: Error Source Modeling", Applied Optics, Vol. 23, September 1984, pp. 3230-3237.

5. R.L. Cheatham and D. Casasent, "Hierarchical Fisher and Moment-Based Pattern Recognition", Proc. SPIE, Vol. 504, August 1984, pp. 19-26.

6. D. Casasent and R.L. Cheatham, "Hierarchical Feature-Based Object Identification", OSA Topical Meeting on Machine Vision, March 1985.

7. D. Casasent, "A Recent Review of Holography in Coherent Optical Pattern Recognition", Proc. SPIE; Vol. 532, January 1985.

8. D. "Hybrid Optical/Digital Image Pattern Recognition: A Review", Proc. SPIE, Vol. 528, January 1985.

9. W.T. Chang and D. Casasent, "Chord Distributions in Pattern Recognition: Distortion-Invariance and Parameter Estimation", Proc. SPIE, Vol. 521, November 1984, pp. 2 6.

10. W.T. Chang, D. Casasent and D. Fetterly, "SDF Control of Correlation Plane Structure for 3-D Object Representation and Recognition", Proc. SPIE, Vol. 507, August 1984, pp. 9-18.

11. D. Casasent and R.L. Cheatham, "Image Segmentation and Real-Image Tests for an Optical Moment-Based Feature Extractor", Optics Communications, 51, September 1984, pp. 227-230.

12. D. Casasent, "Coherent Optical Pattern Recognition: A Review", Optical Engineering, 24, Special Issue, January 1985, pp. 26-32.

13. D. Casasent and V. Sharma, "Feature Extractors for Distortion-Invariant Robot Vision", Optical Engineering, 23, September October 1984 pp. 492-498.

The Robotics Institute --

1. J.L. Crowley and A.C. Sanderson [illegible] resolution representation and probabalistic matching of 2-D grey-scale shape [illegible] IEEE Comp Soc Workshop on Computer Vision, Representation, and Control [illegible] pp 95-105.

2. J.C. Willis. "RAPIDbus:-Design of an Extensible Multiprocessor Structure." Master's thesis. Carnegie-Mellon University. May, 1984.

3. J.C. Willis. A.C. Sanderson. N.K. Alapati. "Rapidbus: Design of an Extensible Multiprocessor Structure," Technical report 84-13. Carnegie-Mellon Robotics Institute.

Computer Science --

1. M. Herman and T. Kanade. "The 3D MOSAIC Scene Understanding System: Incremental Reconstruction of 3D Scenes from Complex Images". Carnegie-Mellon University Computer Science Department Technical Report CMU-CS-84-102. February 1984.

2. M. Herman "Representation and Incremental Construction of a Three-Dimensional Scene Model." Carnegie-Mellon University Computer Science Department Technical Report CMU-CS-85-103, January 1985.

3. Y. Ohta and T. Kanade, "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol. PAMI-7:2, 1985, pp. 139-154.**

6.3 CONFERENCE PRESENTATIONS AND SEMINARS

Electrical and Computer Engineering --

1. D. Casasent, "Fourier Transform Feature-Space Studies". Presented at the SPIE Conference, November 1983, Cambridge, Massachusetts.

2. D. Casasent, "Synthetic Discriminant Functions", Presented at DARPA, February 1984.

3. D. Casasent, "Robotics Applications of Optical Data Processing", Presented at Polytechnic Institute of New York, February 1984.

4. D. Casasent, "Optical Information Recognition", Presented at the Air Force Office of Scientific Research, May 1984.

5. D. Casasent, "Parallel Coherent Optical Processor Architectures and Algorithms for ATR". Presented at the Workshop on Algorithm Guided Parallel Architectures for Automatic Target Recognition, Leesburg, Virginia, July 1984.

6. D. Casasent, "Hierarchical Fisher and Moment-Based Pattern Recognition", Presented at the SPIE Conference in San Diego, California, August 1984.

7. D. Casasent, "SDF Control of Correlation Plane Structure for 3-D Object Representation and Recognition", Presented at the SPIE Conference in San Diego, California, August 1984.

8. D. Casasent, "Research in the Center for Excellence in Optical Data Processing", Presented at Carnegie-Mellon University ECE Sophomore Seminar, October 1984.

9. D Casasent. "Advanced Multi-Class Distortion-Invariant Pattern Recognition". Presented at the University of Pittsburgh. Center for Multivariate Analysis. Pittsburgh. PA. October 1984.

10. D Casasent. "Optical Information Processing". Presented at George Mason University. Washington. D.C.. October 1984.

11. D. Casasent. "Chord Distributions in Pattern Recognition". Presented at the SPIE Conference. Cabrkige. Massachusetts. November 1984.

12. D Casasent. "Hybrid Optical/Digital Image Pattern Recognition: A Review". Presented at the SPIE Conference. Los Angeles. CA. January 1985.

13. D Casasent. "A Recent Review of Holography in Coherent Optical Pattern Recognition", Presented at the SPIE Conference. Los Angeles. CA. January 1985.

14. D Casasent. "Optical Pattern Recognition and Optical Processing". Presented at Fairchild Weston. Long Island. New York. January 1985.

15. D. Casasent. "Hierarchical Feature-Based Object Identification". Presented at OSA Topical Meeting on Machine Vision. Lake Tahoe. NV. March 1985.

The Robotics Institute --

1. J.C. Willis. A.C. Sanderson. "Segmented Crossbar Switching: Design for a Hybrid Message Passing Structure," in review for ICCD '85.

Computer Science --

1. M. Herman. "Representation and Incremental Construction of a Three-Dimensional Scene Model," presented at the Workshop on Sensors and Algorithms for 3-D Machine Perception. Whashington, D.C., August 1983

2. F. Tomita and T. Kanade. "A 3D Vision System: Generating and Matching Shape Descriptions in Range Images," presented at the 2nd International Symposium of Robotics Research. Kyoto. Japan. August 1984.

3. D. Smith and T. Kanade. "Autonomous Scene Description with Range Imagery", presented at the 15th DARPA Image Understanding Workshop. October 1984.

4. M. Herman. "Generating Detailed Scene Descriptions from Range Images," presented at the 1985 IEEE International Conference on Robotics and Automation. St. Louis. MO. March, 1985.

7. SUMMARY

In Chapters 2-5, we have described our progress towards achieving a combination of pattern recognition, image understanding, and artificial intelligence techniques for space-based image processing, using both optical and digital processing methods. We have achieved results in the areas of optical feature extraction and sub-pixel target detection, hybrid digital/optical representation and matching, and model-based three-dimensional scene interpretation. The remainder of this chapter summarizes the results achieved over the past year.

7.1 OPTICAL FEATURE EXTRACTION AND SUB-PIXEL TARGET DETECTION HIGHLIGHTS

The highlights of optical feature extraction work include :

- A new optical processor for detection of in-plane distortion parameters from optically generated chord distributions.
- A new optical/digital moment processor concept.
- A new hierarchical non-ad-hoc tree structure formulation.
- Successful initial tests of the moment processor on ship and pipe part data bases
- Promising initial quantifications of the accuracy to which the distortion parameters of the object can be produced in the hybrid moment processor.
- Development of new correlation SDFs.
- Promising initial ATR test results on correlation SDFs.

The highlights of our sub-pixel fast time change detection/recognition effort include :

- A more unified and accurate image generation software for producing detector images containing sub-pixel moving targets, correlated noise and uncorrelated noise.
- Detailed quantitative results of the performance of sub-pixel shift estimators.
- Detailed quantitative results of the performance of various interpolation schemes.
- Introduction of a new and better performance measure for the characterization of background suppression.
- Analytical and experimental investigation of the use of double differencing for background suppression.
- Initial formulation of the more general "space/time filtering" to enhance the sub-pixel target and suppress the background

- Investigation of the effects of detector limitations such as limited dynamic range and detector noise.

- Initial efforts of multi-region image generation.

7.2 ALGORITHMS FOR HYBRID DIGITAL/OPTICAL REPRESENTATION AND MATCHING

This phase of the project has focussed on the development and evaluation of methods which yield representations of structural and textural information in an image, and may be used for matching images to scene models. The principal results achieved in this research include:

- *Probabilistic Graph Matching* - Attributed graph structures are used as models of structural and statistical information in the image. Matching of these graph structures using probabilistic similarity methods poses a number of interesting problems in the mathematical formalism, in the computational matching algorithms, and in the application of these methods to real images. We have investigated methods of subgraph decomposition which permit branch-and-bound search of the matching tree and provide efficient pruning of the possible matches.

- *Multiple Resolution Rotation-Invariant Operators* - The MRI (Multiresolution Rotation Invariant) operator and the MRD (Multiresolution Difference) transform have been introduced to extract structural and textural features of images for use in matching and interpretation phases of analysis. The MRI is a complex operator derived from derivative expansions of Gaussian kernels and will have magnitude of response independent of feature orientation and phase angle of response which provides information about orientation. The spatial and frequency domain properties of these operators have been studied and an approximate MRI operator which uses difference of shifted Gaussian kernels has been derived and shown to be computationally efficient due to the scaling and shift properties of the Gaussian kernel. The MRI operators have been applied to aerial images of objects and textures.

- *Texture Analysis* - The MRI operators described above have been used to characterize and classify textures from aerial images. This set of multiresolution operators permits classification of texture independent of the size and orientation of the texture pattern itself. The statistical distribution of the magnitude responses is analyzed across the set of operators for regions of the image. Correlation with the corresponding magnitude range and the corresponding phase distribution provides information on the relative scale and the relative orientation. Experiments on textures from aerial images and textures from simple patterns have been carried out and compared to previous texture energy operators.

The algorithms studied in this section reflect the interdisciplinary nature of the project. The MRI operators and associated texture measures are particularly well-suited to parallel or optical processor implementation. They will be implemented and evaluated on the array processor with *RAPIDbus* host. Our formulation of the *recursive model matching algorithms* is also intended for implementation on this type of architecture with extensions which may integrate symbolic and numerical processing. The interactive use of parallel and optical preprocessing with hypothesis formation and adaptive search strategies will be natural continuation of the work completed.

7.3 IMAGE UNDERSTANDING TECHNIQUES FOR 3D SCENE INTERPRETATION

Our effort this year has resulted in techniques dealing with two levels of processing required for the task of describing 3D scenes: the 2D image level, detecting features, such as edges, lines, and corners, in images, and the 3D scene level, representing, constructing, and updating the 3D scene model. Our principal results include:

- *Stereo Correspondence using Dynamic Programming (2D Imaqe Level)* - We have described a method to match the epipolar line pairs in a stereo pair and determine a rather dense depth map of the scene, using intra- and inter-scanline search. *Intra-scanline* search determines the correspondence between edges in the same scanline of the left and right images. This search can be treated as the problem of finding a matching path on a 2D search plane whose axes are the right and left scanlines. Vertically connected edges in the images provide consistency constraints across the 2D search planes. *Inter-scanline* search in a 3D search space, which is a stack of the 2D search planes, finds the vertically connected edges and applies the constraints. By considering both intra- and inter-scanline searches, the correspondence problem can be cast as that of finding in a three-dimensional search space the matching surface that has the best match scores from intra-scanline search and also satisfies the consistency constraints from inter-scanline search. This problem is solved using dynamic programming for both searches.

- *Three-Dimensional Model Building and Maintenance (3D Scene Level)* - We have investigated model building using rangefinder data, which is already three dimensional, bypassing the problem of generating a 3D description from 2D data. We have developed techniques for representing, constructing, and updating the scene model. The model is in the form of 3D faces, edges, vertices, and their topology and geometry. A range image is segmented into edge points to which linear segments are fit. The original line segments are refined to eliminate gaps. Faces are then fit to the line drawing. The final model is represented as a graph in terms of the symbolic primitives *line*, *face*, *edge*, and *vertex*. Although the final description is three-dimensional, most of the processing is done in the two-dimensional image space. Future work will combine model information to obtain a full symbolic description of a scene from range data obtained from multiple viewpoints.

In the future, we will continue our work on both high-level and low-level image processing that is required for the 3D scene analysis task. Our effort will focus on analyzing and extracting 2D repetitive textural features from images, improving our stereo algorithm, and representing and matching 3D scene models.

- *Texture* - Aerial images of urban scenes contain a large amount of textures made of repetitive patterns, such as windows on the building fces. The ability to find and characterize such textures is essential to analyze complex images of man made structures. We will study the problem of detecting and segmenting the regions made of regular arrays of repeated patterns in images by using the analysis of variation.

- *Stereo* Stereo is one of the most important ways of extracting 3D features from images. A fast, robust stereo capability would greatly enhance any 3D scene interpretation system, and would result in a significant step towards an effective change detection system. We expect to continue work on the stereo algorithm based on the dynamic programming technique described in this report to increase its speed and improve the quality of its matching results. Our next step will be to incorporate multi-resolution techniques into this algorithm. This should improve matching quality because it is easier and more reliable to match at lower-resolution (smaller) images and the results can be propagated to higher resolution (larger) images. Speed should also be improved, since results from smaller images can be used to limit the range of search in larger images.

- *3D Model Acquisition and Matching* - Once 3D features have been extracted from the images, they must be accumulated into a coherent model and matched with previous models to determine whether changes in the structure of the scene have occurred. Matching is also necessary when merging two scene descriptions of the same scene, perhaps obtained from different viewpoints, into a single consistent description, or when identifying the same 3D objects, such as moving objects, in different scenes. We will continue our investigation into this problem of reconstructing and matching 3D descriptions from a dense depth map which will be obtained either from stereo or from direct range finding sensors.

Appendix A. Hierarchical Feature-Based Object Identification

David Casasent and R. Lee Cheatham*
Carnegie-Mellon University
Department of Electrical and Computer Engineering
Pittsburgh, Pennsylvania 15213

*Present Address: Battelle Northwest, Computers & Information Systems
Section, Richland, Washington 99352

ABSTRACT. A multi-level classifier for multi-class 3-D distortion-invariant recognition is described. New real imagery and distortion parameter estimation accuracy data are presented.

1. INTRODUCTION

A feature space processor for multi-class distortion-invariant pattern recognition is detailed in Section 2. A moment feature vector space is considered. Test data [1 2] on a robotic database are summarized in Section 3. Results on a ship database, using real input imagery with references from models is presented with attention to preprocessing, distortion parameter estimation, and class identification are advanced in Section 4.

2. PROCESSOR

A moment feature space is easily generated optically [3,4,5] or digitally [6]. Its outputs can easily be corrected for processing errors in post-processing [3]. Moments are jointly Gaussian random variables [2] due to sampling with respect to in-plane distortions. Thus, they allow use of a Bayesian classifier and thus can minimize P_e. To determine the class i (object class c and aspect view φ) and the object's distortions (described by a distortion parameter $\underline{b}$) for each computed input moment vector $\hat{\underline{m}}$, we calculate

$$\varepsilon_i = [\hat{\underline{m}} - \underline{m}_i(\underline{b})]^T \Sigma^{-1} [\hat{\underline{m}} - \underline{m}_i(\underline{b})].$$

with $\underline{b}$ calculated iteratively (k is the iteration index) using

$$\underline{b}^{k+1} = \underline{b}^k + [(\underline{J}^k)^T \Sigma^{-1} \underline{J}^k]^{-1} (\underline{J}^k)^T \Sigma^{-1} [\hat{\underline{m}} - \underline{m}_i(\underline{b})].$$

The class i that minimizes (1) defines c and the out-of-plane rotation angle (aspect) φ of the input, whereas $\underline{b}$ provides estimates of translations, scales, and in-plane rotations. The number of iterations k can be reduced to 4-6, $\Sigma = \underline{I}$ can be used in (1) and (2), and $\underline{J}$ in (2) calculated as an update [1,2]. This significantly reduces the computational load per class/aspect i.

The major problem is the large number of aspect-classes i that need potentially be searched. To relieve this, we use two first-level estimators [1,2] to estimate the aspect (this is achieved by $I = \hat{\mu}_{31}/\hat{\mu}_{02}$) and class (a hierarchical tree is used for this, with the node structure chosen from a multi-class Fisher projection and with a two-class Fisher discriminant vector used per node). As we show in Section 3, this reduces the number of aspect-classes i to be searched and thus makes the processor very computationally efficient. A block diagram of the system is shown in Figure A-1.

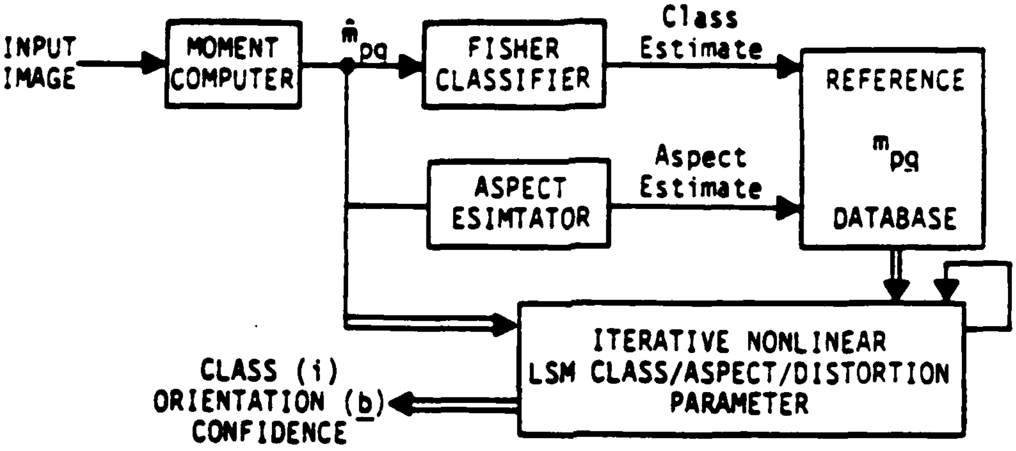


Figure A-1: Block diagram of a multi-level moment feature-space classifier

3. PIPE PART TEST RESULTS

Nine different pipe parts (4 classes) viewed from a 50° depression angle were digitized (128 x 128 pixels) with 36 images per part (one image every 10° in aspect) and used as our test database. Test results are summarized in Table 1. They show: 9 out of 36 references are adequate (Test 1). Use of the first-level estimator reduces the number of i to be searched in (1) to 10 (Test 2). The number of iterations k in (2) is only 6 over a large Δg_i range (Test 3) and $\Sigma = I$ in (1) and (2) is adequate (Test 4). As seen in Table A-1, the system of Figure A-1 can correctly classify over 97% of the 324 images (using only 9 x 4 = 36 references).

4. DISTORTION PARAMETER ESTIMATION ACCURACY

Related tests on another database [2,7] showed comparable performance and similar operational parameters. In this database, the reference objects were obtained from models and in tests against real-world IR images, excellent recognition was obtained. The preprocessing required [7] used only simple 1D and 2D histogram operations and thresholding (to maintain low computational overhead).

We now consider the class c, aspect φ, scale α and translation x_0 estimation accuracy of the system for a second five-class database (36 images at 10° aspect intervals per class) using only four references per class. The true object was the 80° aspect view of the class 1 image. A real IR input

TEST NUMBER	CONDITIONS	PERCENT CORRECT (OUT OF 324)	REMARKS
1	No Aspect Estimator	97.5%	9 Aspect Refs each 40° Used 24 View-Class (Avg) Passed
2	Full First-Level Estimator	97.5%	10 View-Class (Avg) Passed
3	$\Delta g_i = 10^{-4}$ to 10^{-1}	98.2% more refs	6 Iterations k
4	Different $\underline{\Sigma}$	90-93.9%	$\underline{\Sigma} = \underline{I}$ (90%) Adequate

Table A-1: Representative Pipe Part Data (Different Test Conditions)

image (vs. references obtained from models) at a depression angle 10° different from that of the reference set was used with real IR noise present in the input. The tests (Table A-2) show perfect class and aspect classification for $\Delta g_i = 10^{-4} - 10^{-1}$ (for $\Delta g_i = 0.5$, errors resulted as expected) and excellent shift (x_0 in pixels) and scale factor (α) distortion parameter estimation. All distortion parameters were estimated within 5% accuracy, due to the input resolution, noise, etc. factors.

TEST NUMBER	TRUE SCALE α/PIXEL SHIFT x_0	α/x_0 ESTIMATE	CLASS/ASPECT ESTIMATE
1	1.0/0	1.0/0	1/80°
2	1.0/15	1.016/14.22	1/80°
3	1.0/25	1.023/23.22	1/80°
4	0.5/0	0.499/0.1	1/80°
5	0.75/0	0.750/0.07	1/80°
6	0.9/0	0.90/0.03	1/80°

Table A-2: Results of Class and Distortion Estimation Tests (True Class 1, Aspect 80°)

ACKNOWLEDGMENT. The support of this research by the Air Force Office of Scientific Research (Grants AFOSR-79-0091 and F49620-83-C-0100) are gratefully acknowledged.

APPENDIX A REFERENCES

1. D. Casasent and R.L. Cheatham, Proc. ASME, August 1984.

2. R.L. Cheatham and D. Casasent, Proc. SPIE, ***504***, August 1984.

3. D. Casasent, R.L. Cheatham and D. Fetterly, Applied Optics, ***21***, 3292, September 1982.

4. K. Wagner and D. Psaltis, Proc. SPIE, ***352***, 82, August 1982.

5. J.A. Blodgett, R.A. Athale, C.L. Giles and H.H. Szu, Optics Letters, 7, 7, 1982.

6. A P. Reeves and R.R. Seban, Proc. of 15th Annual Hawaii International Conference on Systems Sciences, pp. 388-396, 1982.

7. D. Casasent and R.L. Cheatham, "Image Segmentation and Real-Image Tests for an Optical Moment-Based Feature Extractor", Optics Communications, Submitted, April 1984.

END
11-87
DTIC