.-0497

Linear Feature Extraction from Radar Imagery: SBIR Phase II, Option I

Gary D. Conner
David L. Milgram
Daryl T. Lawton
Christopher C. McConnell

Advanced Decision Systems
201 San Antonio Circle, Suite 286
Mountain View, CA 94040-1289

April 1988

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

Prepared for:
U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060-5546

UNCLASSIFIED 2117 1430

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188 Exp. Date Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY None	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE None	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR-3139-02	5. MONITORING ORGANIZATION REPORT NUMBER(S) ETL-0497
6a. NAME OF PERFORMING ORGANIZATION Advanced Decision Systems 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U.S. Army Engineer Topographic Laboratories
6c. ADDRESS (City, State, and ZIP Code) 201 San Antonio Circle Suite 286 Mountain View, CA 94040	7b. ADDRESS (City, State, and ZIP Code) Code: CEETL-RI-A Fort Belvoir, VA 22060-5546
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Engineer Topographic Laboratories 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DACA72-86-C-0004
8c. ADDRESS (City, State, and ZIP Code) Code: CEETL-RI-A Fort Belvoir, VA 22060-5546	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)

Linear Feature Extraction from Radar Imagery: SBIR Phase II, Option I (UNCLASSIFIED)

12. PERSONAL AUTHOR(S)

Gary D. Connor, David L. Milgram, Daryl T. Lawton, & Christopher C. McConnell

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final Technical Report	FROM 10/16/86 TO 6/15/87	1988, April	41

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	SAR, Feature Extraction, Edge Detection, Terrain Analysis, Image Understanding

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

The goal of this effort is to develop and demonstrate prototype processing capabilities for a knowledge-based system to automatically extract and analyze linear features from synthetic aperture radar (SAR) imagery. This effort constitutes Phase II funding through the Defense Small Business Innovative Research (SBIR) Program. Previous work examined the feasibility of and technology issues involved in the development of an automated linear feature extraction system. This Option I Final Report documents this examination and the technologies involved in automating this image understanding task. In particular, it reports on a major software delivery containing an image processing algorithmic base, a "perceptual structures" manipulation package, a preliminary hypothesis management framework and an enhanced user interface.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL E. James Books	22b. TELEPHONE (Include Area Code) (202) 355-3039 — 22c. OFFICE SYMBOL CEETL-IM-T

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

PREFACE

This report describes work performed under contract DACA72-86-C-0004 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060-5546 by Advanced Decision Systems, Mountain View, California 94040-1289. The Contracting Officer's Technical Representative has been Dr. P. F. Chen.

The authors would like to thank Dr. Pi-Fuay Chen and Mr. Richard Hevenor for their many helpful suggestions throughout the course of the effort.

TABLE OF CONTENTS

LIST OF FIGURES

1. EXECUTIVE SUMMARY

Advanced Decision Systems (ADS) is pleased to submit this final technical report on research undertaken during the Option I portion of this three part, two year effort (contract #DACA72-86-C-0004). The goal of this second portion is to develop and demonstrate prototype processing capabilities for a knowledge-based system to automatically extract and analyze linear features from synthetic aperture radar (SAR) imagery. This effort constitutes Phase II funding through the Defense Small Business Innovative Research (SBIR) Program. The previous Phase I (contract DACA72-84-C-0014) work examined the feasibility of and technology issues involved in the development of an automated linear feature extraction system. The current Option I extension of the base contract effort which was reported in [Conner - 87] continues this examination and is developing the technologies involved in automating this image understanding task.

1.1 BACKGROUND OF PROBLEM

A vitally important problem facing the Department of Defense is the ability to quickly and efficiently analyze remotely sensed image data. This analysis is used for a variety of applications ranging from automated map making/updating to a variety of surveillance tasks, to other military and commercial remote sensing applications. An increasingly important and useful sensing capability is provided by synthetic aperture radar (SAR) imagery.

Imaging radar sensors provide all-weather, cloud penetration capability for a variety of applications. Technical capabilities now allow enormous volumes of such imagery to be automatically produced in relatively short periods of time. However, the current methods for analysis and interpretation of radar imagery largely consist of manual examination by human experts. As the quantity of imagery expands, the requirements for timely and efficient feature classification and the scarcity of radar image interpreters point to the need for an automated system for feature extraction and classification.

Linear features such as roads, rivers, bridges, and railroads are major landmarks in such imagery. Extracting and analyzing such features are prerequisites for most analysis applications. Traditional linear feature extraction techniques (edge detection and region segmentation) tend to perform adequately for low noise, high resolution visible imagery. However, the relatively poor quality and the complexity of the observed scenes in radar imagery make these feature extraction techniques less effective.

The ability to automatically detect and analyze linear features will have a major payoff for numerous applications. Technology to provide such an automated capability is emerging from the fields of image understanding (IU) and artificial intelligence (AI). Such a system could incorporate knowledge about the scene and use context (from the image or external sources such as digital terrain maps or terrain object models) to intelligently guide and interpret the extraction process. The results of the Phase I SBIR effort were encouraging in showing the feasibility of this approach. An automated system would greatly enhance the Army's capability for aerial cartography, change detection, aerial surveillance, and autonomous navigation. The goal of this effort is to pave the way for such a system by developing a largely automated terrain/image analysis workstation prototype.

There has been much work in artificial intelligence, computer vision, and graphics that satisfies the individual requirements for object modeling capabilities. Little has been done to integrate these diverse fields, especially for the domain of SAR imaging. To date, the only vision systems that can interpret natural scenes limit themselves to very restricted environments [Hanson - 78] while other systems are restricted to artificial objects and environments. A system which uses well defined shape attribute inheritance among a set of progressively more complex object models, and which generalizes affixment relations to handle uncertainty begins to fulfill the basic requirements. This system must also generate constraints on image features from object models. Care must be taken so that constraints on image structures generated from the abstract instances of object models are specific enough to generate initial correspondences between models and image structures. A rich set of image feature descriptions and robust object models that can adjust the segmentation process directly during their instantiation are also crucial to an automated system. Object models will be produced by ADS during the Option II phase of this effort for a limited set of features. A minimal object model must be able to direct constrained searches against image data. Models must eventually be capable of supporting learning and handling uncertainty in the matching of image feature descriptions to multiple terrain features.

The basic motivations for such a system stem from the poor results associated with the undirected application of low level image processing techniques. Environmental objects such as roads and rivers are semantic entities whose extraction requires contextual and object-specific knowledge which cannot be easily incorporated into, for example, low level filtering operations. In fact, it has become clear that a general and expandable system will have to incorporate processing which reflects the actual reasoning involved in expert SAR image interpretation.

The purpose of the Phase II Effort is to complete the design of an automated linear feature extraction system for SAR imagery and to demonstrate this design in a prototype software embodiment.

1.2 APPROACH

The major steps of the Phase II effort are as follows:

1. Develop the appropriate working environment to manipulate and process imagery.

2. Develop and experiment with various segmentation and feature extraction algorithms.

3. Determine significant terrain object feature properties and construct representative object models.

4. Experiment and evaluate model to image feature matching schemes.
5. Develop an approach for managing competing and conflicting hypothesis matches.
6. Develop feature finders/predictors to support or contradict an expected terrain feature's existence.
7. Implement a display interface to support the above processing steps.

Once the proper environment is established, the system for determining and extracting terrain features can be extensively tested. These experiments will further establish the role of autonomous feature extraction from SAR imagery and, indeed, the importance of SAR imagery to map generation.

1.3 PROGRESS TO DATE

1.3.1 Phase I

The major accomplishments of the Phase I effort were:

- Reviewed and implemented several edge and region extraction routines from optical image processing on SAR aerial imagery. Routines were evaluated for their performance in order to determine which would be valuable for integration into the general system.
- Obtained a better understanding of the nature of SAR aerial imagery and its requirements for interpretation.
- Considered a variety of techniques for representing the properties of environmental objects such as roads and rivers in SAR imagery.
- Designed and began component implementation of a model-based vision system for the extraction of linear features from SAR aerial imagery. In particular, ADS implemented an initial image structure data base and experimented with associated perceptual grouping rules and simple SAR object models.

A comprehensive report of Phase I results is available [Lawton - 85].

1.3.2 Phase II - Base Contract

The work performed by ADS under the Base Contract addressed three different problem areas.

The primary work area focused on the continuation of the design produced in the Phase I SBIR effort. The results of that design are described in [Conner - 87].

The second major area in which ADS pursued the project goals was the development and the design of a software environment in which to perform experiments and begin to build the eventual prototype system. The basic framework of this software was delivered to ETL in May 1987. The delivery emphasized neighborhood and display operations. The software also contained the necessary software "hooks" for future expansion into the other system components.

Finally, the last area of work undertaken as part of the Base Contract was the continued experimentation with the government provided radar imagery. Experimentation included algorithm surveys, hand processing sample imagery, and actual algorithm implementation. This work and ADS's general understanding of machine vision, has been continually supporting the design and development of the components of a model based vision system for linear feature extraction.

1.3.3 Phase II - Option I

The bulk of the work accomplished under this effort pertained to the continuing effort to embody the system design in software. A major software delivery to ETL of the processing framework was made in September 1987. The software included the following:

- Many of the relevant image processing routines used at ADS (see note below on operating system version compatibility).
- The software for creating, manipulating, accessing, and editing image structures (also called "perceptual structures").
- The preliminary framework of the hypotheses database. (This database contains hypotheses about extended image structures. Functions that provide for the creation of these structures are embodied in the "filter" functions.)
- Enhanced user interface to display the image structures.

The software was also accompanied by a "User's Guide." The guide was written with the expert Symbolics Lisp Machine user in mind. At the suggestion of ETL, a supplemental guide was issued to address the needs of those users not intimately familiar with the Symbolics environment. In addition to the documentation, two sessions were held at ETL. The first session was a general "demonstration" of the software delivered. The second session was oriented towards familiarizing the user with the software. Given the size and complexity of the development environment, a subsequent visit was scheduled in December 1987 to further assist ETL personnel in the use of the system. During this visit some software "bug" fixes were also accomplished.

As expected, the system design continues to evolve as more of the system becomes realized in software. An updated system design will be submitted in the Option II final report.

Work was also initiated on the recognition procedures. The details of the various terrain features were studied. In addition to the standard properties of the individual features, of particular interest is both the internal and external structures of the features. For example, the apparent image-based structure of a patch of forest may be comprised of the textured area representing the bulk of the forest, the bright leading edge of the patch, and the trailing shadowed region. All three portions have entirely different "visual" characteristics, but each is an important component of the recognition of the forest patch. An example of external structures is best illustrated by a bridge. Typically, a bridge is detected as a long, thin bright region. Unfortunately however, this is not a unique signature by itself. If this bright region has roads extending from both ends and is surrounded on each side by water, then a unique signature for a bridge begins to form. Because this work in image object structure is only preliminary, details will not be provided until the final report for the Option II phase which will specifically address the area of recognition procedures.

A continuing source of difficulty facing the Linear Feature II project is the compatibility of software environments at ADS and ETL. Much of the Linear Feature I work was performed on a Symbolics system running the Version 6 OS operating system. At the beginning of the Linear Feature II contract both ETL and ADS were running Version 6 OS. Since then ETL has installed Version 7 while ADS has not. ADS made a commitment early on to deliver software in Version 7 OS. This extra effort and overhead requires additional time and money to port software between versions, thus delaying delivery of important additions and bug fixes to ETL.

1.4 ORGANIZATION OF THIS DOCUMENT

Section 2 provides the technical foundation for the framework in which the Linear FEature (LFE) system is being developed and prototyped. It begins with a general discussion of object oriented programming, and then continues with how these principles are applied to the Image Understanding problem. It concludes with details of the LFE framework implementation.

Section 3 provides a discussion of Synthetic Aperture Radar (SAR) and its characteristics and capabilities.

Section 4 is a depiction of the processing scenario presented to ETL with the software delivery. In addition to this some discussion/postulation as to the direction of the remaining effort is presented.

Section 5 closes with the status of the Phase II effort and a recap of the accomplishments of the effort to date.

Section 6 provides a list of references used in the compilation of this document.

2. OBJECT-ORIENTED IMAGE UNDERSTANDING

This section discusses the concepts and merits of performing Image Understanding (IU) tasks within an object-oriented programming environment. We begin with a discussion of object-oriented programming. This is followed by a description of our realization of an object-oriented programming environment. We close by discussing uses of this environment to perform the bottom-up process of recognizing significant image structures within a given scene.

2.1 OBJECT-ORIENTED PROGRAMMING

In object-oriented programming, a program is thought of as being built around a collection of objects. These objects represent conceptual or physical objects in the real world. For example, a text editing program might have objects such as "windows" and "words". Objects may be organized into homogeneous groups that all exhibit the same behavior and can perform the same operations, though each object may also have unique information associated with it. Object-oriented programming provides a lucid and modular style of programming allowing the user to perform generic operations on objects.

Object-oriented programming is a programming methodology that is guided by well-defined software engineering principles. It is especially well suited for use in large programming projects. The software engineering principles of abstraction, information hiding, modularity, localization, uniformity, completeness, and confirmability are supported by object-oriented programming. Levels of specialization of objects contain the essential features at each level of abstraction of the software. Individual objects define local functions and storage that can be hidden from other objects. Objects localize all the pertinent code by defining all the possible operations on a given data type. In object-oriented programming, each object is a module that encapsulates the behavior of each data type as well as providing (and hiding) the representation of that type. A software system defined using such objects will embody the uniformity of the object notation. All objects are of equal status. Completeness together with abstraction insures that each module is a necessary and sufficient solution to a component of the programming problem. Confirmability is supported by the modularity of programs written using an object-oriented methodology. Each module can be independently verified and tested.

Object-oriented programming languages are characterized by the following features:

- Data and procedures are encapsulated in modules called objects. Object implementations are (or can be) hidden so that the only permissible operations are those operations that the object itself defines. This facilitates easily changing an object's representation. Encapsulation of representation and operations in an object minimizes interdependence by defining a strict interface.

- Computation occurs by means of messages sent between objects. Which operation actually gets performed is specified by the combination of the message name and the object.

- A hierarchy of objects permits specialization. Specialized objects inherit behavior from more general objects. Default behavior can be specified at the top of this inheritance to be overridden by specializations. Objects may inherit behavior from many other objects.

2.1.1 Contrast with Functional Languages

The current Linear FEature (LFE) System is implemented using Symbolics' ZetaLisp and an internally developed IU environment (POWERVISION). Both traditional functional programming (as in Lisp) and object-oriented programming are supported in this environment.

In functional or procedural languages, where the emphasis is on activity rather than on the data abstraction, functions may be overloaded. Overloaded or generic functions may be applied to many different types of objects. A typical case of an overloaded function is the "plus" function. The same plus function can be called with two integers, two reals, or a real and an integer. In contrast, an object-oriented language might define integers and reals to be two different objects that each responded differently to the plus message.

Such a reorganization benefits the construction of large systems. All the behavior pertinent to a given data abstraction is available at the same place in an object-oriented language. In programming languages this is called an object. Objects define data abstractions and localize all the code which affects that object. Since only the specification (the format of acceptable messages) needs to be known by other objects or other programmers, the implementors responsible for an object are free to modify the implementation as they choose.

2.1.2 ~~Inheritance and Prototypes~~

Traditional procedural languages define objects in terms of types. Many object-oriented languages define objects in terms of classes. First the characteristics of the type or class are specified, then objects (called instances of the type or class) are created that have those characteristics. In procedural languages, functions and procedures operate on instances of particular types. Many object-oriented languages distinguish between two different levels of objects. They define class objects that specify the behavior of a set of instance objects of that class. To provide flexibility, those languages also usually define a third level of object (often called meta-class objects) that define behavior of class objects.

One way to understand the characteristic of specialization is by analogy with set theory concepts. Figure 2-1 illustrates this comparison. Defining the behavior of instances of a class is equivalent to defining the behavior of members of a set. In disjoint inheritance, the "apple" class contributes a complete set of attributes (to its instances) which are disjoint form those contributed by the "orange" class. Common behavior of two objects may be inherited from a third

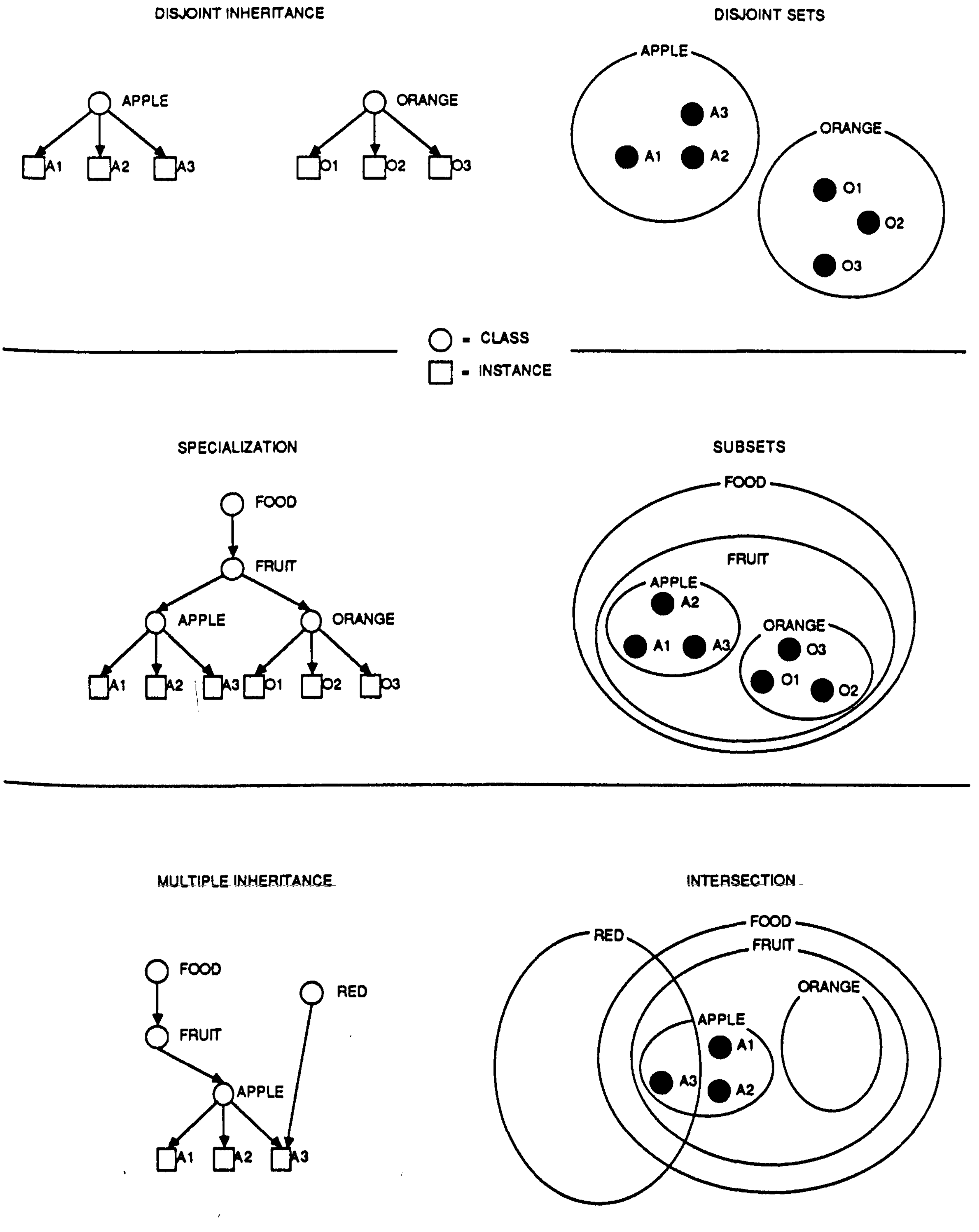

Figure 2-1: Illustrating Object Classes using Set Theory

object, in this case a class object. "Specialization" allows common attributes for different classes to be shared since apples and oranges while different are both instances of "fruits". Common behavior of instances of two classes may be inherited from an enclosing class.

Objects may inherit behavior from the objects that they designate. The designated objects may in turn inherit their behavior from other objects. This may result in an object inheriting behavior from far away in the inheritance hierarchy. In complex programs that use far inheritance, care must be taken when defining or modifying the behavior of objects at the top of the hierarchy.

2.1.3 A Simple Example of Object-Oriented Programming

The specification of an object is the declaration of its message handlers and their arguments. Message handlers for objects perform two functions. They define the behavior with which an object will respond to a message, and they store the current state of the object. A handler may either store a value, or a Lisp function. Allowing handlers to store values is shorthand for using Lisp functions that return the same data. Function handlers may refer to other handlers as part of their definition. For instance, a rectangular object might define four handlers: height, width, perimeter, and area. The first two would likely be value handlers, the last two would be functions of the height and width handlers.

Object Example

```
CREATE-OBJECT    rectangle
    :HEIGHT      4
    :WIDTH       17
    :PERIMETER   (2 * (SEND(self, HEIGHT) + SEND(self, WIDTH)))
    :AREA        (SEND(self, HEIGHT) * SEND(self, WIDTH))
```

A useful feature of object oriented languages is that objects may be specialized to form other objects. A specialization of a rectangle is a square. A square can be defined as a rectangle with height equal to width. Here is an example of that definition:

Specialized Object Example

```
CREATE-OBJECT square INHERITS-FROM(rectangle)
    :SIDE      22
    :HEIGHT    SEND(self, SIDE)
    :WIDTH     SEND(self, SIDE)
```

Since square inherits from rectangle, it delegates any messages that it does not know how to handle to rectangle. If we were to send a square object the :AREA message, square would delegate the handling of that message to rectangle. These inherited handlers get executed in the context of the square system. This means that when the :AREA handler attempts to determine the width by sending a message to itself, the message gets sent to square instead of rectangle. Notice that both rectangle and square define handlers named :WIDTH and :HEIGHT. Square's definitions hide those made by rectangle. Another way this is described is by saying that the :WIDTH and :HEIGHT handlers of rectangle are "shadowed" by square.

2.2 LFE SYSTEM APPROACH

Early image processing research focused on pixels as the primitive information units. The bulk of image processing was concerned with enhancing the "visual" appearance of the image. Given the computing resources of the time, even this was a formidable task. The paradigm for image processing used individual pixels (or small windows around pixels) to compute enhanced values, edges, and classifications based solely upon local neighbor properties. Edge operators processed an image and produced an image. Today this type of processing is referred to as "low-level" computer vision [Marr 1982].

As the field of computer vision matured, its progress was paralleled by a maturing computer hardware field. Additional computing power enabled researchers to explore new possibilities. The pixel images produced from low-level processes were transformed further via various segmentation and connected component processes. This step, characterized by a progression away from pixel-based reasoning is termed "medium-level" computer vision.

A large part of the research community is still involved with these two areas of research.

The next level of processing is termed "high-level." This level is characterized by relating extracted image structures ("perceptual" objects or "image" objects) to meaningful chunks of real-world objects. Sometimes individual perceptual objects can be mapped onto world objects; other times, groups or collections of perceptual objects must be mapped into world objects. The mapping between perceptual objects and world objects is accomplished through the use of a model. The model can describe the world objects, the sensor, the environment, or the imaging process. Models can be used to "measure" the match of perceptual objects to a particular object model. Using both a sensor and object model, predictive methods can be used to generate synthetic scenes.

This method of object recognition will be referred to as "graphic modeling". It attempts to generate images using basic sensor physics and the image formation process. For example, in order to determine the reflectance of a particular truck panel, its appearance would be computed from the spectral properties of the material, the panel's orientation, the sensor's particular operating mode (polarity, wavelength, etc.), shadowing objects, distance/elevation, and multibounce effects. There has been a volume of good work in this area. Most, if not all of it dealing with manufactured objects in controlled settings (such as tanks positioned on a laboratory turntable). This work is encouraging for the small set of relatively well-structured objects that have been examined. However, it does have significant shortcomings if extended to radar scenes of natural geographic features. These features do not have "cookie cutter" engineering type (CAD) models; therefore, graphics modeling is not directly applicable. People have tried to extend this approach to terrain by using abstract mathematical models such as fractals and Markov models to model natural features like mountains, forests, and fields. An argument offered for this approach says that if a human cannot distinguish the graphical version from the imaged scene, then a vision algorithm that matches the graphics function to the image data will extract image segments that a human would choose to label as the object for which the graphics were developed.

There are several fundamental logical flaws in this argument. One is that even if the above statement is true about human performance, the fact that there may be multiple graphics models that match the same image segments from the point of view of human perception, does not imply that any of the models actually match quantitatively; so segments that a human might label are not necessarily extracted by the default graphics model. The second and more basic flaw is that, just because two images are indistinguishable to a human does not imply that a machine algorithm can be developed that performs the match between the graphics model and the imaged instance. Naively, the ideal such algorithm duplicates human perception, and this is clearly beyond the current state of the art. Finally, the purpose of a terrain feature extraction system is to create a database which corresponds to ground truth, not to a human performance baseline.

The approach taken in the LFE system is closely related to the schema-based approach described by Lawton [Lawton - 87]. This approach is based on a general object model called a "schema." A schema can represent perceived, but unrecognized visual events, as well as recognized objects and their relationships in natural scenes. Schemas are related to similar concepts found in [Hanson - 78] and [Ohta - 80]. Schemas can depict a continuum of hypotheses. At one extreme hypotheses may be as general as "a perceptual object has been detected" at a particular location. At the other extreme hypotheses may be as specific as "this perceptual object is a portion of the left bank of the Potomac River."

Object models are used to organize perceptual processing by integrating descriptive representations with recognition and segmentation control. One aspect of this is the use of different types of attributes and inheritance relations between generic schemas for representation in IS-A and PART-OF hierarchies. A particular object attribute relates world properties of an object in general qualitative terms. These attributes are inherited and modified according to different object types as described in the earlier section on object-oriented programming. Objects are treated as having lists of attributes that are matched against extracted image features. In addition to this feature description, objects may contain information specifying an active control process that directs image segmentation by specifying grouping procedures to extract and organize image structures.

2.3 LFE SYSTEM ARCHITECTURE

The use of computational processes for perceptual organization is basic to computer vision. Researchers have discovered over the past decades that active, intelligent processing must occur at all levels of image understanding. Undirected segmentation and feature extraction processes have proven to be too brittle and narrowly focused, resulting in a meagre structure for interpretation of the world. Early work reflected gestalt principles; e.g., in the line trackers and region growers which optimized curvature or compactness to form more complete contours and regions. More recent research in perceptual grouping has involved two major trends in computer vision. The first of these is the modern framework which stresses the fundamental role of symbolic and relational representations at all levels of vision ([Marr - 82], [Binford - 81]). Perceptual organization in this framework, is expressed as rule-based operations applied to a rich set of extracted symbolic relations and objects. This is in contrast to earlier approaches where image processing was treated more or less as a sequence of pixel filtering operations which resulted in image-to-image transformations but not in an explicit structural database. This made the manipulations necessary for shape recognition, for example, quite difficult. Interestingly, psychologists working in perceptual organization are developing rule-based models independently of work in computer vision [Rock - 84].

The second trend stresses the extraction of robust, qualitative information from images as opposed to exact quantitative information about the environment. Researchers ([Witkin - 83], [Lowe - 86], [Binford - 81], and [Lawton - 87]) are attempting to establish more reliable, qualitative structures which can be extracted from images. The processes proposed for doing this are non-semantic grouping operations sensitive to such things as coincidence, symmetry, pattern repetition. This approach involves an object modeling methodology in which objects and events are represented in a form compatible with predictions of qualitative image structures.

The LFE approach to perceptual processing is concerned with organizing images into meaningful chunks. From a data-driven perspective, the definition of "meaningful" and the development of explicit criteria to evaluate segmentation techniques requires the chunks to have characterizing properties, such as regularity, connectedness, and fragmentation resistance. From a model-driven point of view, "meaningful" is defined as the extent that chunks can be matched to structures and predictions derived from object models. From either perspective, a basic requirement is that image segmentation procedures find significant image structures, independent of world semantics, in order to initialize and cue model matching. This allows for the extraction of world events such as regions, boundaries, and interesting patterns independent of understanding perceptions in the context of a particular object. These, in turn, are useful abstractions of image information to match against object models or describe the characteristics of novel objects.

The Perceptual Structure Data Base (PSDB), depicted in Figure 2-2, contains several different types of information. These are classified as images, perceptual objects, and grouping (or groups). Images are the arrays of numbers obtained from the different sensors (SAR sensors for the LFE system) and the results of low level image processing (such as smoothing operators or median filters) that produce such arrays. It is difficult for the symbolic/relational representations used for object models, such as schemas, and the processing rules in computer vision systems, to work directly with an array of numbers. Therefore, there are many spatially-tagged, symbolic representations used in image understanding systems that describe extracted image structures. These include the primal sketch [Marr - 82], the RSV structure of the VISIONS system [Hanson

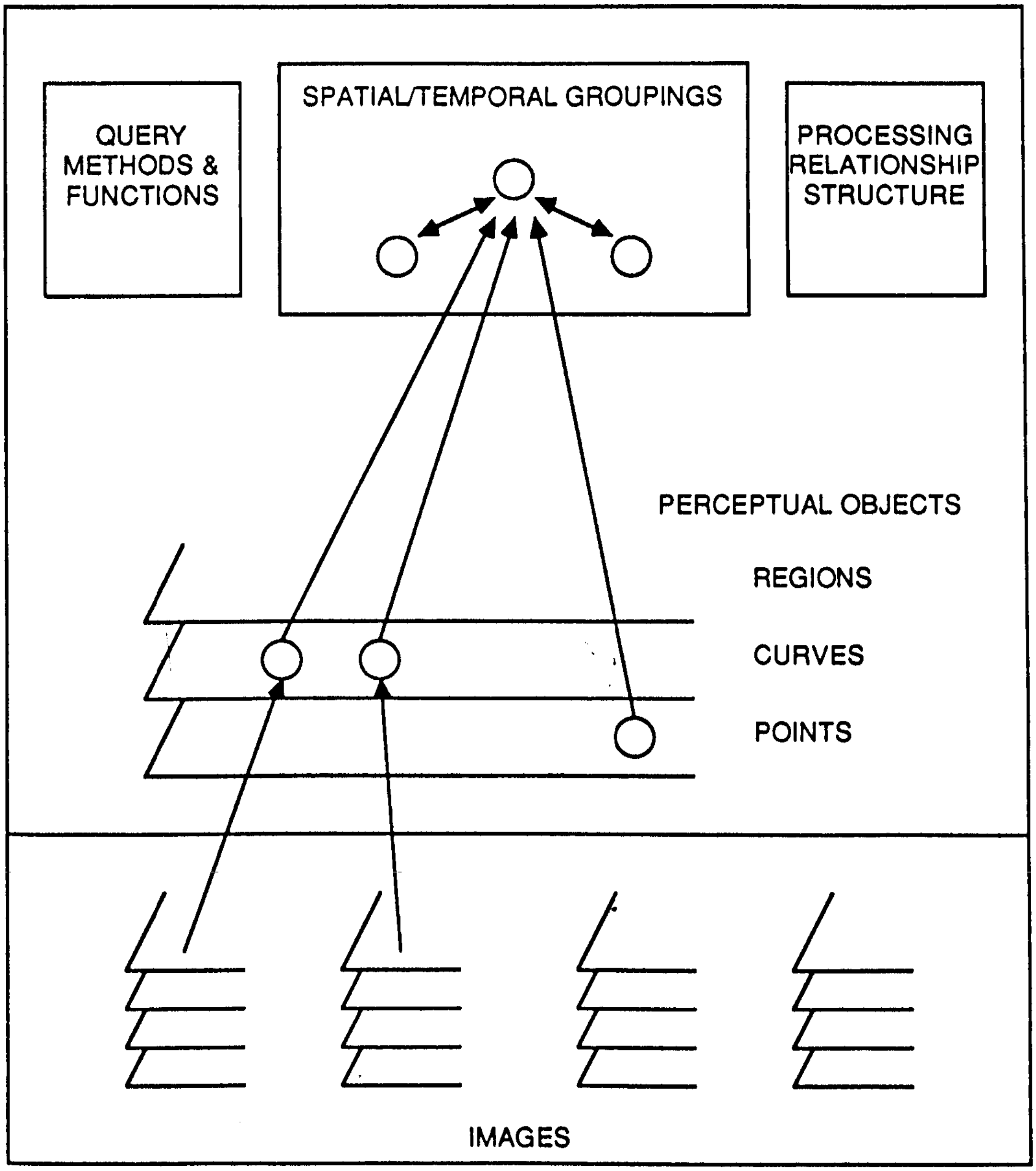

Figure 2-2: Perceptual Structure Data Base (PSDB)

- 78], and the patchery data structure of Ohta [Ohta - 80]. The LFE representation has been built around a set of basic perceptual objects corresponding to points, curves, regions, and other basic shape descriptions.

Groupings are recursively defined to be a related set of such objects. The relation may be exactly determined, as in representing which edges are directly adjacent to a region, or it may require a grouping procedure to determine the set of objects that satisfy the relationship. Groupings are typically defined spatially, e.g., linking texture elements under some shape criteria such as compactness and density.

Whenever new sensor data are obtained, a default set of operations is performed to initialize the PSDB. For example, edges could be extracted at multiple spatial frequencies and decomposed into linear subsegments. The edges could then be grouped into distinct connected curves, and general attributes such as average intensity, contrast, and variance of contrast are computed. Similar processing could be performed to extract regions. For example, thresholds could be selected with respect to a wide range of object-based and image-based characteristics (e.g., gray level, homogeneous intensity, homogeneous texture). Prespecified operations are used to initialize bottom-up grouping processes and schema instantiations to piece together lower level structures. These, in turn, determine significant structures using heuristic interestingness rules to prioritize the structures for the application of grouping processes or object instantiations.

2.3.1 Defined Perceptual Objects

Several types of objects in the initial environment have been defined, including images, points, curves, and regions. There are also composite objects, stacks, and groups that abstractly combine collections of other objects. These objects support common properties and additional properties particular to each class.

Because objects are defined as abstract types, interesting combinations of these initial objects are possible. For example, a raster grid, structured as an image, can be created where each pixel is an object instance and is not restricted to being just a number. This is called a "label plane" and is used extensively for geometric reasoning. A grid consisting of histogram "pixels" can provide a representation for hierarchical segmentation.

A general attribute of all non-image objects such as points, junctions, curves, and regions, is the representation of their spatial characteristics. Any representation should be compact, provide fast access, and should facilitate most common operations. However, there is no optimal format; each must trade off between time and space considerations.

Thus, the primitive objects may have several possible predefined representations: arrays, segments and lists. These variations are possible in object-oriented programming. When a representation for an object is added, the object inherits the procedures which performed the operations associated with the relevant messages. Thus, any newly defined object can immediately be manipulated in the environment.

In addition to spatial access, objects can be stored in a more conventional feature-based database. Database queries can then be performed on the objects such as sorting, ranking, attribute matches, and range queries. The objects are stored in the Perceptual Structure Database (PSDB).

The Perceptual Structure Data Base stores extracted image structures such as curves and regions, as well as associated groupings of them. These structures result from processes as simple as low-level edge extraction or as complex as a multi-schema search and linking procedure. The structures can be formed from recursive algorithms. All of the objects are stored together with their inferred or measured attributes.

Database queries are expressed in terms of "filter functions" and, in special cases, lists of objects. A filter function takes a list of objects as its first parameter and, optionally, some additional parameters. The function produces a list of objects as its result. The functions enabled by queries can range from simple attribute checking to complex search or pattern matching operations. The objects returned are usually a subset of the original objects. The filter functions are combined by using the "filter" macro. The macro takes an input specification in terms of logical operations and other filter functions and generates the bindings and additional functions required to execute the query. A large library of general utility and special purpose filter functions have been written.

A filter is a macro that generates LISP from filter functions and the logical combiners AND, OR, NOT. A filter function takes a list as its first parameter, and together with optional additional parameters, returns a list as its result. The logical combiners specify how the results of filter functions are piped into other filter functions. This includes the generation of temporary bindings and set union (OR), or difference (NOT) code. Filters are efficient because they expand into some optimal, but, perhaps, harder-to-understand LISP code. They are very flexible because the only convention is that the first parameter and the result must be lists. There are no constraints on the elements of these lists, although they are usually objects. The objects returned are usually a subset of the original objects, but can be a superset or even a completely different set of objects. There are three different general classes of filter functions: selectors, transformers, and modifiers.

"Selectors" produce a subset of the original list as their result. This is the most common type of filter function. "Transformers" take in a list and produce a list of completely different objects. "Modifiers" change some aspect of the objects and then return the objects as their result. Side effects in a query can be very useful. For example, long edges can be chosen, their orientation computed and stored, and then only the horizontal edges selected.

3. THE SAR ENVIRONMENT FOR FEATURE EXTRACTION

This section briefly discusses some background on the history of Synthetic Aperture Radar, discusses the Model-based Reasoning paradigm in the SAR domain, and concludes with an overview of SAR features.

3.1 SAR BACKGROUND

Several good reference texts and introductory papers describing the process of synthetic aperture radar (SAR) imaging have been written over the years including [Skolnik - 62], [Brown - 67], and [Brown - 69]. Synthetic aperture radar is sometimes referred to in older texts as synthetic array radar or simulated array/aperture radar. The first SAR systems were developed in the late fifties to early sixties. Prior to that, real-aperture imaging radar systems existed and are still used today for some applications [Stimson - 83]. SAR provides significant improvement in along track resolution over real-aperture systems. The SAR concept employs a coherent radar system and a single moving antenna to simulate the function each antenna which would comprise a real linear array. This single antenna is used to occupy sequentially the spatial positions of the non-existent linear array. The received signals are stored and then processed at a later time to re-create the image of the illuminated area as seen by the radar [Eaves - 87]. This technique can be used to synthesize an antenna array which may be thousands of feet long, thereby increasing the effective resolution.

Designers of early applications of radar technology whose objectives were to locate or determine speed and direction of man-made targets considered the backscattering from terrain as a nuisance. This attitude toward terrain backscatter coined the term "radar clutter." It was this clutter signal that was later used to perform radar remote sensing. "More specifically, the variation of the scattering coefficient with the physical properties of terrain and water surfaces is the key to extracting useful information from radar images" [Ulaby - 82].

Of particular interest to remote sensing applications was the introduction of spaceborne SAR. Seasat-A was the first spacecraft to carry an imaging radar into orbit. It was originally intended to detect and map ocean waves. However, it was subsequently used to generate large volumes of data covering land surfaces [Ulaby - 82]. Renewed interest in space-based remote sensing has been generated by the highly successful SIR-A and SIR-B missions on the NASA space shuttle flights of 1981 and 1984. The future promises additional SIR-X missions, along with potential efforts by the European Space Agency(EAS) and the Japanese ERS-1 for free flying earth orbiting imaging radars [Leberl - 85].

Also of interest to those interested in the interpretation of radar imagery is an effort by AFWAL which is attempting to bring together the Image Understanding (symbolic reasoning) and the radar Signal Analysis (signal processing) communities [Milgram - 87].

3.2 MODEL-BASED REASONING AND SAR

Over the past twenty years, research in image processing has built up a large compendium of approaches and algorithms for extracting and interpreting structure from optical images. Any standard textbook (e.g., Rosenfeld and Kak [Rosenfeld - 82]) in the field will list dozens of methods for peak extraction, edge detection, region segmentation, line finding and the like. With modifications to account for the different characteristics of SAR, many of these techniques can apply to SAR image understanding.

In attempting to apply existing algorithms to SAR, it is important to recognize a significant characteristic of SAR; that a known system impulse response is convolved with every scattering center in the scene to form the complex image. Two situations should be considered: cultural features and terrain features. Cultural features tend to be "hard" and scatter the radar energy in particular directions which are predictable from an analysis of the geometry. If scattering centers are separated by more than the system resolution, the image of the cultural feature takes on a blob-like appearance with blobs of known shape (the impulse response) but unknown location, phase, and height (radar cross section). If the scattering centers are unresolved (i.e., no one scattering element dominates over the others), the image is again blob-like except that now the blobs are due to the mutual interference of complex returns. This causes the feature image to assume somewhat the nature of a Rayleigh distributed nonhomogeneous 2-d random process. If resolved scatterers do not behave like stable point scatterers over the imaging interval, the image blobs are perturbed. Causes of instability include, specular scattering from slightly curved surfaces, radar focusing imperfections, and complex multibounce scattering paths. The perturbations may form a useful signature for the cultural feature which may be extracted by processing of the complex signal or image.

Natural terrain features tend to scatter energy in a diffuse way. The degree of diffusion is related to the natural surface "texture" of the feature. Gravel will provide a more diffuse response than asphalt; bare soil is more diffusing than gravel, etc. Thus, the SAR image of natural terrain will tend to resemble an optical image of the same area. Texture-based processing is therefore appropriate in both sensor domains. However, the behavior of the image at borders of regions may differ due to the imaging geometrics and specular conditions. Also, the image formation process of SAR is very different from optical imagery, artifacts such as slant range presentation, near range compression, layover, etc. must be taken into account.

SAR image processing converts the SAR image (either real- or complex-valued) into various spatial data structures. These describe image features by location and various shape and structural properties. These data structures can be stratified into a hierarchy typical for most systems which interpret mid-level image structures. The hierarchy and the discussion which follows is subdivided into five levels:

- Pixel Grids
- Point Features

- Linear Features
- Region Features
- Structures (Group Features)

3.2.1 Pixel Grids

Pixel grids are the spatial structures which most commonly represent input imagery. As part of the preparation of the imagery for feature extraction, it has been common to apply a number of operators to "clean up," restore or enhance the imagery. The operators range from simple gray level histogram transformations to local statistical smoothing to adaptive relaxation techniques. The result of this preprocessing step is another image which serves as the "real" input to the system.

In SAR, the input image is derived (or synthesized) from the radar signal history. In general, any reconstruction or restoration is more properly applied to the signal domain prior to or as part of the image formation. Nonetheless, situations arise which necessitate pixel grid operations.

3.2.2 Point Features

Many man-made objects or their components in SAR imagery are characterized by point features. These appear as image peaks with associated shape, location, and intensity. These features can be reliably detected with a peak detector (e.g., local maximum) followed by extent and attribute measurements.

3.2.3 Linear Features

Generally, feature extraction work in the optical domain has focused on edge extraction and region extraction. Edge extraction techniques [Canny - 83] are based upon the basic concept that grey levels will change radically near region boundaries, and furthermore that these boundaries can be detected by derivatives operating on the image as though it were a 3-D surface. Starting with this basic concept, a great number of edge extraction techniques have been developed over the last three decades. Edges may occur in optical imagery because of occlusions between three-dimensional objects, because of folds and junctions that occur within an object, because of texture elements within a region, because of shadows, specular reflections, or because of spurious noise introduced during the propagation of energy through the atmosphere, during the image formation process, or during image preprocessing steps. Processing at the levels of boundary, junction, and surface interpretation, and limited relative height inference and recognition, must take into account the possible different interpretations of these edges.

Since the amount of illumination of any given point is highly dependent on the angle of incidence, long linear features often appear to be broken into smaller line segments or tend to fade completely as the linear feature becomes occluded, shadowed or changes direction with respect to the source of illumination. Image smoothing can sometimes connect the smaller segments together; however, this tends to become highly unreliable with unrelated segments being joined as well. The amount of smoothing is also extremely dependent on the scatterer spacing, material composition (dielectric properties) and radar cross section separations. A better strategy may be to group image structures (e.g., point, line and region) into linear groups rather than to attempt extracting linear features directly with conventional edge detection operators. Hough transforms and line tracking are two traditional techniques for detecting and grouping together linear structures. Recent trends are towards extending/replacing these techniques with perceptual grouping schemes [Lowe - 85] [Lawton - 87]. These approaches are discussed in the "Structures" section below.

3.2.4 Region Features

Region extraction operators look to segment the image into regions that are homogeneous according to some measure. The two basic sorts of homogeneity that regions may possess are intensity homogeneity and texture homogeneity. In intensity homogeneity the region operators look for areas whose pixels are nominally within the same grey level range as compared to surrounding regions. Texture operators rely on measures that characterize the textures such as statistics, micro edge densities, etc. These feature-based measurements in local neighborhoods of pixels are then compared to see if their values are nominally within the same neighborhood compared to surrounding regions.

The intensity of a SAR image typically varies rapidly and widely from pixel to pixel so that intensity homogeneity is practically limited to bright (above a threshold) and dark (below a threshold) regions. Bright regions can be used to segment entire objects from the scene or individual peaks (another technique for extracting point features). Dark regions, or regions of no-return, can be caused by occlusion (image shadows), reflection away from the radar (common for water bodies and road surfaces, and parking lots), or absorption.

Intensity homogeneous regions can be found by combinations of filtering, thresholding, and connected components. Regions defined by multiple thresholds can be integrated into so-called containment trees of connected components.

Texture homogeneous regions should be especially useful in segmenting natural terrain regions such a forest canopies, fields, and orchards in low resolution SAR imagery. Within such regions, the SAR image tends to approximate homogeneous random processes. The process parameters define the "texture" of the region. See Rosenfeld [Rosenfeld - 81] for relevant research papers applied to the optical domain.

Region extraction (and grouping below) are not bound by any fixed neighborhood radius and so can respond to information at any distance. This is in contrast with window-based pixel processing which cannot respond to the true shape and extent of the data features. Measurements of the regions such as area, location and 2-D orientation, are made during the processing and attached as attributes to the region descriptors. Regions may also be related to other features and regions by explicit links. For example, thresholding a gray scale image at a sequence of values and linking the resulting region yields a containment tree [Morgan - 87].

3.2.5 Structures

Structures are collections of other features such as point, linear, region, and even other structures. Structures represent features or component structures linked by geometric relations. For example, a sequence of bright linear blobs along the leading edge of a forest form a linear structure. A set of lines which intersect form a junction structure. Structures may be adjacent (e.g., a mosaic of regions), connected (e.g., edges in a continuous edge), or disconnected (e.g., points in a dotted line; a series of power line support towers). Typically, only simple relations occur with sufficient frequency to make them worth searching for. These may result from image structures that are related by proximity, linearity, symmetry, and the like.

Lowe [Lowe - 85] demonstrated the utility of perceptual organization of line structures within the SCERPO optical vision system.

Lawton [Lawton - 86] has defined a set of grouping operators of this sort for ground level forward-looking color optical imagery. He has enlarged the concept to "notice" configurations based on a measure of "interestingness" and to tune the bottom up processing to discover repetitions of the interesting configurations. This grouping process aids linear feature extraction since terrain features often have unpredictable image level descriptions but are, nonetheless, regular (i.e., interesting) in structure.

Other researchers [Nevatia - 82] [Binford - 82] have also studied the extraction of extended image structures in optical data.

3.3 LIMITATIONS OF CURRENT APPROACHES TO RECOGNITION

It is appropriate to analyze the distinction between the model-based recognition approach and other formulations. Describing the statistical pattern recognition approach first will motivate the need for model-based vision. This section briefly describes the statistical approach and discusses its capabilities and failings.

Statistical pattern recognition has been one of the traditional methods of identifying features in remotely sensed imagery for over twenty years. It rests on the assumption that structure can be recognized implicitly and characterized by limitations on statistical variability. Typically this method begins with a "training" set of imagery reflecting the expected variability. The targeted set of features to be recognized then have various descriptive properties measured. These feature properties are then used to characterize the target object classes in the set of images to be analyzed. This technique has had some success, particularly in the domain of multi-spectral imagery. This technique has had limited success for SAR imagery, but proves not to be robust due to lack of structural descriptive capability.

Statistical pattern recognition uses measured feature values to directly relate the appearance of image features to object classes. A typical feature might be characterized by the statistical covariance of object classes and extracted image features. The parameters(ranges) of the feature values (such as the covariance) are established by training. Training may be done either on real data collected with ground truth or (less successfully) with simulated data from an object and sensor model. The approach is illustrated in Figure 3-1.

The simplicity and apparent generality of statistical pattern recognition can be quite attractive:

- Decision rules are usually easy to implement.
- Training procedures are explicit and easy to follow.
- Any apparent system failure to recognize an object class can be "patched up" with more and better training data.

However, a closer look at the methodology for representing object features and the procedures for recognizing them statistically shows that there are fundamental drawbacks which cannot be remedied with simple patches.

"Recognition adequacy" is a system's ability to use the stored object feature information to interpret the data. This information needs to be chosen and structured so that data can be processed within time and accuracy constraints. The choice of features to model also affects recognition adequacy. For instance, recognizing an object from the set of its pixel values alone may be impossible; recognizing it from its spatial structure may be relatively straightforward.

Maintaining recognition adequacy depends on using the most useful data features at each stage of recognition. Choices of features include:

- Individual pixels.
- Low-level image features such as peaks and regions.
- Structures of low-level image features.

Each feature type provides its information to a portion of the analysis. For instance, statistical features of individual pixels tend to be useful at the outset of image exploitation; image structures provide strong information about scene layout, and about specific object classes. Statistical pattern recognition generally exploits pixel level features. Since structural features often have many parameters, their statistical models either become needlessly complex or are restricted to operating over fairly simple sets of features.

The following list describes additional shortcomings of statistical pattern recognition:

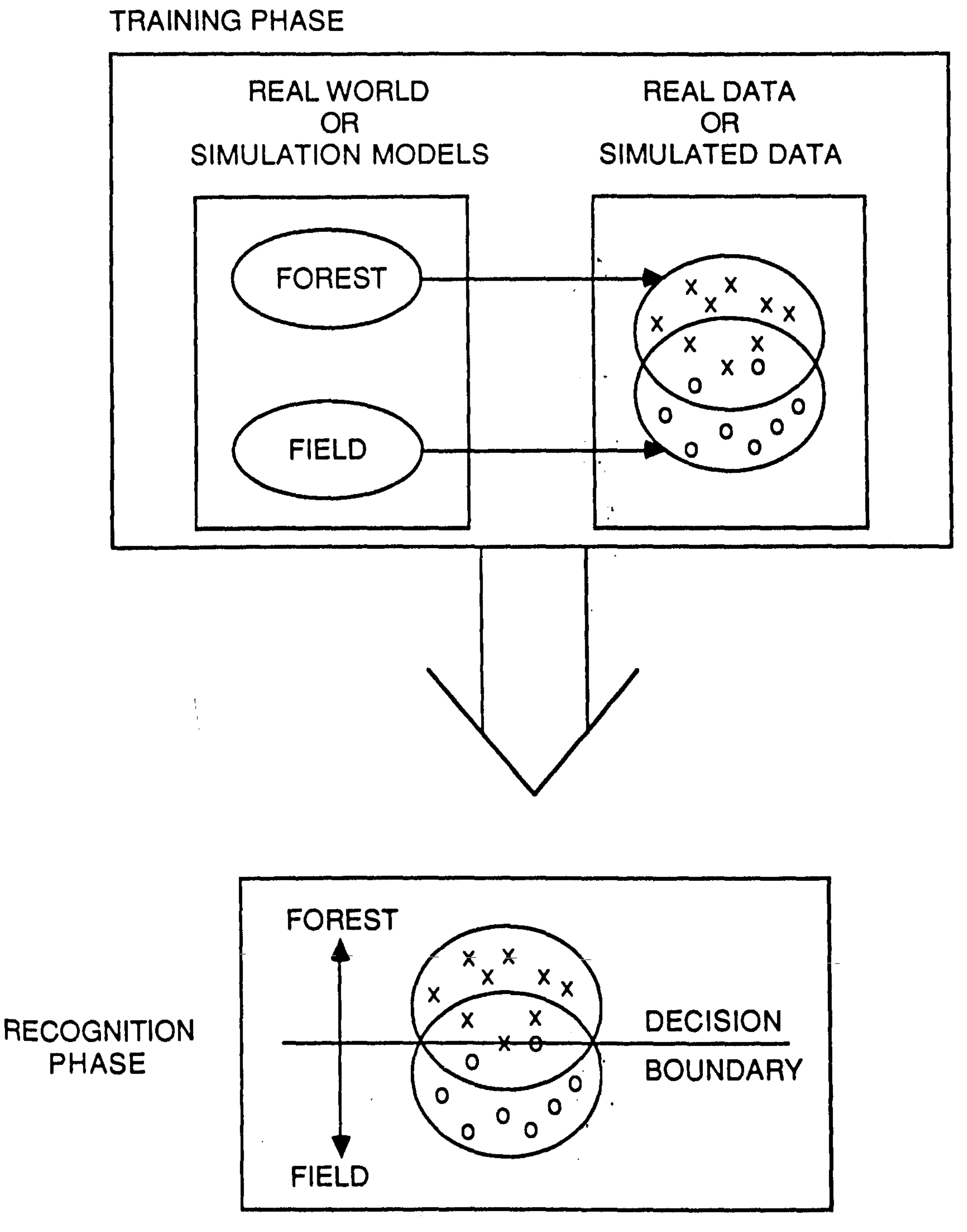

Figure 3-1: Statistical Pattern Recognition Paradigm

- Obtaining training data for a sufficient set of cases to span the real world is often quite difficult. The training set must have enough examples of each desired discrimination state to be statistically sound. Furthermore, the addition of new cases causes an inordinate requirement for new training data and verification.

- The complex joint variability of factors in the real world is hard to capture statistically in a train/test paradigm.

- The ability to completely train the system is uncertain at best since multidimensional statistical decision spaces are hard to visualize and explore. Incomplete training results in a system with limited and often unpredictable real world performance.

- Statistical methods do not provide a means for incorporation of collateral (or map) information into the decision. This is a very serious shortcoming in systems whose performance is expected to improve as additional information accumulates.

- Feature discrimination does not improve as higher resolution imagery becomes available. In fact, performance often deteriorates.

- Similarly, knowledge of the presence of other objects within the scene is not easily integrated with the statistical classification approach.

3.4 SAR FEATURES

This section describes the "meaning" of SAR imagery, the requirements for a processing system, and the features that are of interest.

3.4.1 Radar Signatures

The value at a given pixel in a SAR image is directly proportional to amount of energy returning to the receiver/transmitter that results from the backscattering produced by the ground area corresponding to that pixel. According to Ulaby [Ulaby - 82], The received power is determined by: 1) system factors including the transmitter power level and antenna gain; 2) propagation losses that account for propagation from the radar antenna to the ground and back; and 3) the reflectivity factor of the ground area. Gray level differences for features on the image (such as two agricultural fields) are due to differences in their individual reflectivities, since system and propagation factors are essentially the same for both features. The reflectivity factor of a terrain feature is called the backscattering radar cross section per unit area, and is abbreviated as "scattering coefficient"." Ulaby goes on to point out that the scattering coefficient for a region does not always contain enough information to discriminate between terrain features, but when combined with textural information becomes increasingly powerful.

The return signal from terrain (i.e., backscatter) is composed of two primary components, surface and volume scattering. Surface scattering is due to the dielectric difference between air and the terrain surface. The incident wave is scattered by the terrain in many directions (Figure 3-2) and the radar measures the part of the scattering pattern in the backscatter direction. The backscattering coefficient is strongly dependent upon surface roughness. "Volume scattering, e.g., as caused by foliage in a forest canopy, is caused by spatial inhomogeneity in a volume at a scale comparable to that of the wavelength of the incident wave" [Ulaby -82] (see Figure 3-3).

The dielectric constant of the surface being imaged figures prominently in the volumetric component. For soils and vegetation, the dielectric constant is strongly dependent upon moisture content. This helps explain some of the effects seen on river and creek banks and irrigated vs. non-irrigated fields.

Terrain feature models and segmentation strategies will have to incorporate knowledge of the physics or radar. The methodology chosen for the LFE system is a "heuristic modeling" scheme. In heuristic modeling the complicated underlying physical and mathematical relationships are reduced in complexity and embodied into "general rules of thumb." These rules of thumb encapsulate complex interactions like volumetric and surface backscattering by relating the physics to recognizable image features or attributes. For example, a patch of forest may be characterized by a bright leading edge (a specular reflection from the dihedral effect of tree trunks), an area of rough texture (corresponding to the volumetric scattering of the canopy), and a trailing dark region (caused by the shadowing of the terrain surface by the tree canopy). While heuristic modeling avoids much of the complexity of mathematical SAR modeling, it does limit the descriptive power of the representation where explicit volumetric and surface material composition information exists. Nonetheless, heuristic modeling has the advantage of being more intuitive by depicting a model that can be "visualized" by a human and of providing a solution in the absence of material composition information.

Similar conclusions were reached by Autometric, Inc., approaching a similar problem from a different direction. In 1984, Autometric performed a study [Pascussi - 84] in which SAR imagery analysts were asked to describe various man-made features in qualitative terms. The descriptions of what they saw were formalized in a number of tables which served as valuable inputs toward developing the requisite heuristic models for the LFE system.

"The principle of least commitment" is an important perspective on the rules which perform feature extraction and segmentation that has been espoused most notably by Rosenfeld (UMd). It states the conservative position that transformations which compress information should avoid selecting a single choice from among the range of possible alternatives. In other words, each stage of processing should make as little commitment to a single selection as is possible while remaining consistent with a goal directed strategy. The justification for this principle is that confidence in a decisive choice rests on the accumulation of evidence which enters into the decision. In the early stages of processing, decisions about features (and their interpretations) are based mainly on local evidence and therefore are subject to greater risk of error than will be present later once processes with wider scope are employed. If close alternatives are eliminated too soon, it becomes impossible to recover from bad choices. Therefore, the principle of least commitment suggests that multiple alternatives be retained until interpretations

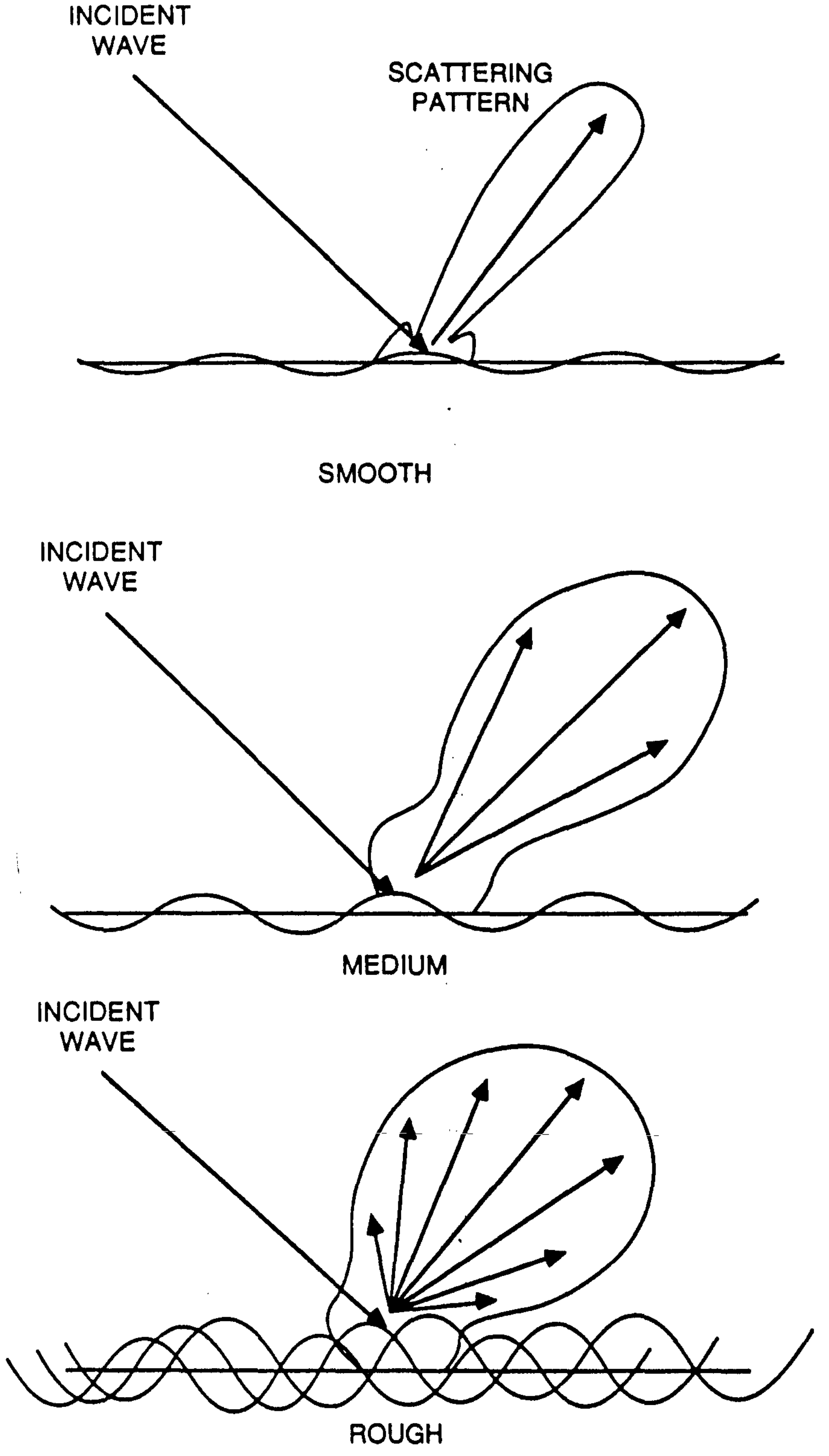

Figure 3-2: Examples of Surface Scattering Patterns

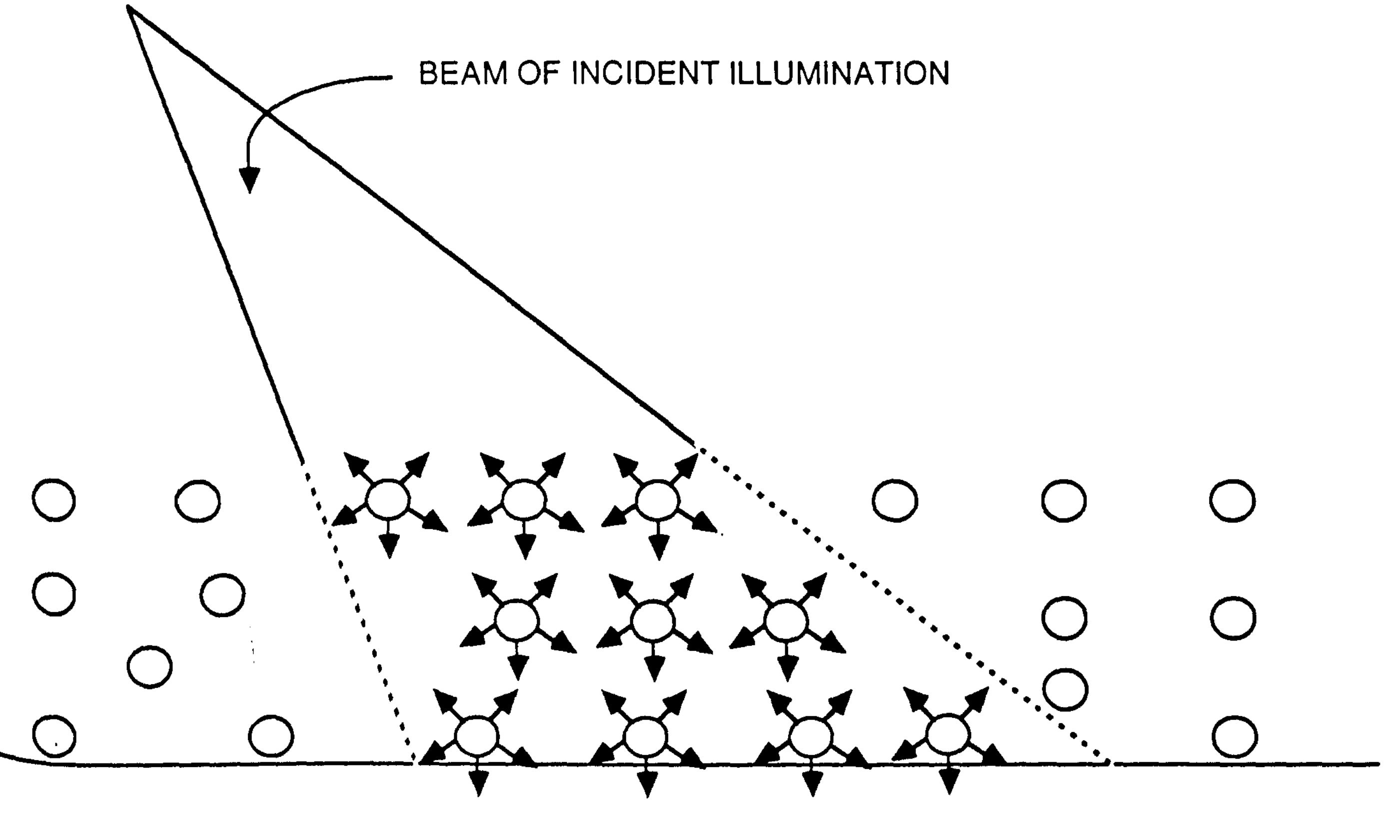

Figure 3-3: Volumetric Scattering, as in a Vegetation Canopy or Snowpack

made in later stages can be shown to be well founded (e.g., are well supported by evidence or by conformance to applicable theories or heuristic models). The result is that information and alternatives are produced (in great volume) and carried forward with little need for algorithmic "backing up". Computationally, this increases memory requirements but otherwise simplifies the architecture.

4. PROCESSING SCENARIO

This section describes a demonstration that was used to illustrate the capabilities of the software delivered to ETL in September 1987. The demonstration illustrates the power of an object-oriented image understanding environment. The following example highlights only some of the display and database capabilities found in the LFE system.

The example begins by displaying the image chip that will be used for processing throughout this example, Figure 4-1. The image is a 256 X 256 pixel SAR image of an area near La Crosse, Wisconsin. The approximate resolution is 7.5 meters per pixel. The upper and lower third of the picture depict areas that are largely undeveloped and primarily covered with forest. The middle portion of the image contains a river with a very large island in the middle. The two bright elongated regions to the left of center are bridges which connect the upper and lower land masses via the tip of the island. The direction of radar illumination is from the top of the image.

The original data can be transformed or processed by a number of routines that exist in the system. Images can be preprocessed using algorithms implementing symmetric convolutions, median filters, edge preserving filters, and simple thresholds. In this example, the image is convolved repeatedly with a gaussian mask in order to remove some of the interference caused by the high frequency noise inherent in the image. This series of convolutions also removes some of information originally in the image. This information still remains in the original image from which it can later be extracted when needed. After this preprocessing is performed a region segmentation is performed. Figure 4-2 shows the boundaries of the segmented regions.

After the segmentation procedure, the image regions undergo a process of extraction and description. The extraction process transforms the pixel data structure into an image object data structure. This process begins by performing connected-component analysis on the segmented image. The result is then used to create database objects for each region. Each region object then has a number of properties and features computed for it. This process is sometimes called the signal-to-symbol transformation. In this case, the signal is the two dimensional representation of the returned energy, i.e., a pixel intensity image; the symbols are image structures representing regions extracted from the imagery. The image structures are stored as objects in a data base.

The image structures have a number of properties that are computed and associated with them. Measurements such as the average and variance of the intensity of the pixels making up a region are readily computed using pointers back to the pixel locations that comprise the object. Also provided are measurements describing the shape of the region. Shape descriptions can range from simple bounding boxes, (i.e., raster-oriented rectangles of minimum area which contain the region) to minimum bounding rectangles (i.e., non-raster oriented bounding boxes) to polygonal approximations. Spatial and topological relations are also stored with the object. These include contained regions, such as "holes", adjacent regions, and the line segments comprising the perimeter. An important advantage of the image structure data representation is the ease with which new properties can be computed and added to the region description. In addition, the researcher has control over which features are computed and when they are computed. Standard functions describing the intensity properties and simple shape descriptions are computed by default during region extraction. Properties that are computationally expensive are usually reserved for only a few select regions.

The extracted regions shown in Figure 4-2 have had the default set of properties calculated after their creation. These image structures are then stored in an image structure database that allows standard database queries to be answered. An example query is "Return all regions with area greater than X or with an average intensity of between Y and Z". Figure 4-3 shows a display of the top fifteen "brightest" regions. First, the average intensity was computed for each region. The regions were then sorted according to the value of the region's average intensity. The first fifteen elements in the list were then selected resulting in the display of the fifteen brightest regions.

The previous discussion has centered primarily on region extraction and the region image structure objects produced. Similar capabilities exist for line segments. Figure 4-4 shows the edges produced using an algorithm based on edge detection techniques developed by Canny [Canny - 83]. One of the advantages of this technique is that it is extremely sensitive to weak edges. It may appear in Figure 4-4 that the technique actually produces too many edges. This concern would be justified if the results of edge extraction were viewed as an undifferentiated set of edges. This is not, however, the case. Edges have a number of properties associated with them, such as edge strength, length, orientation, average and variance of underlying original image pixels, etc., that can be used to select and prioritize the extracted edges into more useful information.

Each of the blue lines displayed in Figure 4-4 represents an entry in the image structure data base. The data structures representing edges share many of the properties of the region objects such as pixel count, average intensity, etc. Edge objects also have a number of additional unique properties such as endpoints and orientation.

Figure 4-5 shows all of the edges resulting from the segmentation procedure. The lines in red represent the database objects which have a count of between ten and twenty pixels. This example shows the results of a simple query. A more complex query is depicted in Figure 4-6. The image structure database was asked to return all edges that are between 10 and 100 pixels long with an average intensity of between 45 and 200. The results are displayed in green. Although not depicted in this example, an edge can also have an average edge strength (e.g., contrast) associated with it.

The LFE system is designed to be extremely interactive. Figure 4-7 shows the manual selection of a single curve. Upon selecting the curve its database properties can be reviewed. For the remainder of this example the selected curve will act as the "model" curve.

Figure 4-8 illustrates the results of querying the database for edges that have an orientation similar to the model curve. The model curve is represented in red, while the query results are displayed in yellow. Figure 4-9 show the results of querying the database for edges that are approximately the same size as the model curve. The actual query selected any edge that was within five pixels in length of the model curve.

It is clear that image structure objects defined by their attribute values can be found quickly and easily within this database. However, this facility also extends to queries embodying spatial relations, e.g., location and nearness. The next portion of this discussion will illustrate these aspects of the database. The conclusion of this section will show the power of combining different image structure databases generated using two different segmentation techniques.

Figure 4-10 shows in yellow the edges that have an endpoint within ten pixels of one end of the model edge (red). Figure 4-11 shows the results of a more powerful search constraint. The results shown are derived from using the "cone" filter. The cone filter gets its name from the shape of the search space. A search space is generated by taking the line segment shown in red and extending it in both directions to infinity. The line is then rotated around the segment midpoint by plus and minus a fixed angle (five degrees in this case). The area swept out by the infinite line is then used as an area restriction in a database query. The results are shown in green.

The following discussion emphasizes the power which comes from combining the results of different segmentation algorithms. Figure 4-12 is the result of querying the image structure database for the "brightest" region. The results are displayed in red. Figure 4-13 shows an image where the intensity value of a pixel is proportional to its distance from an object. The technique used is called the distance transform, or "chamfering" [Barrow - 78]. Using this image as a measure of nearness, the database is requested to produce edges which are "near" the region (Figure 4-14). The discovery of relationships among objects derived using different segmentation techniques is a valuable tool. It permits guidance from different detection and extraction algorithms to be combined to strengthen the confidence in the correctness of the image interpretation. This combination is called "convergence of evidence".

Database queries produce output that can easily be used as input to other queries. As a more refined interpretation is attached to an image structure, more powerful queries can be made. In this example, the "bridge segment" (elongated bright region) could serve as a starting point for finding the roads and rivers usually associated with a bridge. In this way, powerful algorithms can be constructed from simple primitives to reason about terrain features. It is beyond the resources available to this effort to seriously address this level of reasoning, although most of the requisite primitive capabilities are resident in the system.

This example was generated in the absence of terrain object models. Work pertaining to object models and recognition procedures will be performed and reported on in the Option II portion of the contract. Because no object models are currently in the system, no labeling of perceptual objects as terrain features has as yet been implemented.

Note: Colors in original of report do not reproduce here.

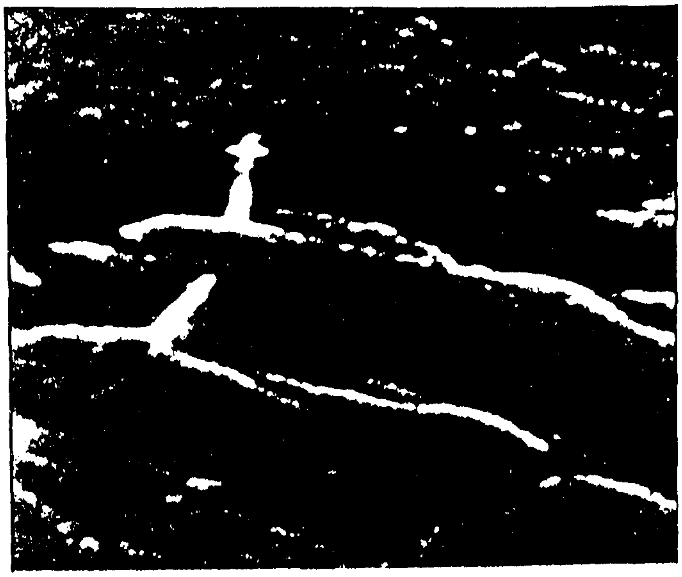

Figure 4-1: Original Image Chip

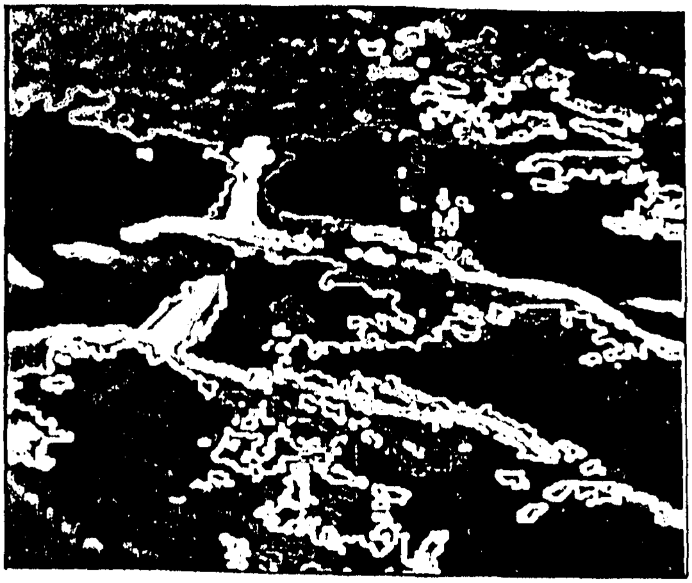

Figure 4-2: Extracted Regions

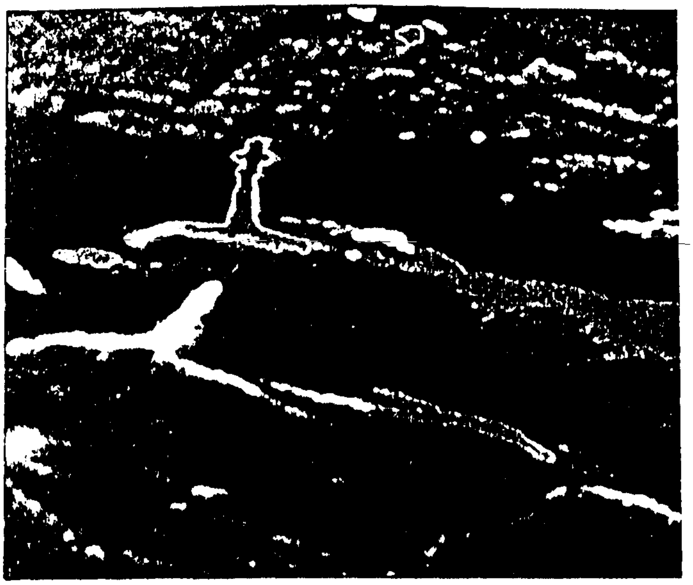

Figure 4-3: The Fifteen Brightest Regions

Note: Colors in original of report do not reproduce here.

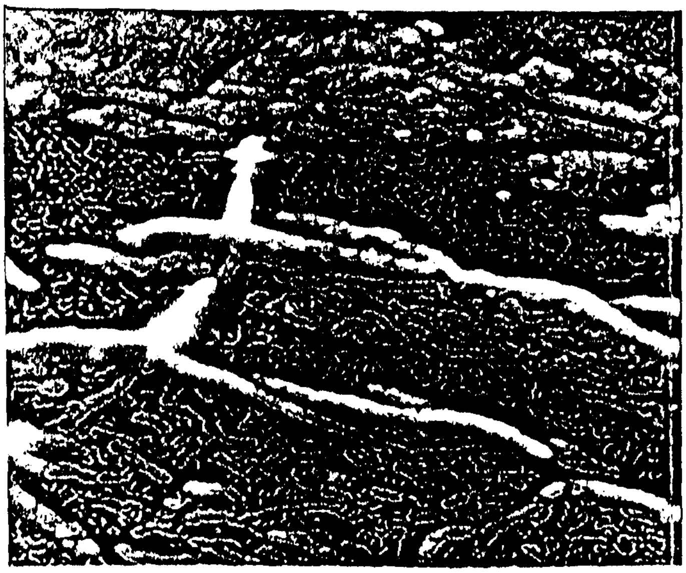

Figure 4-4: Canny Edge Extraction Results

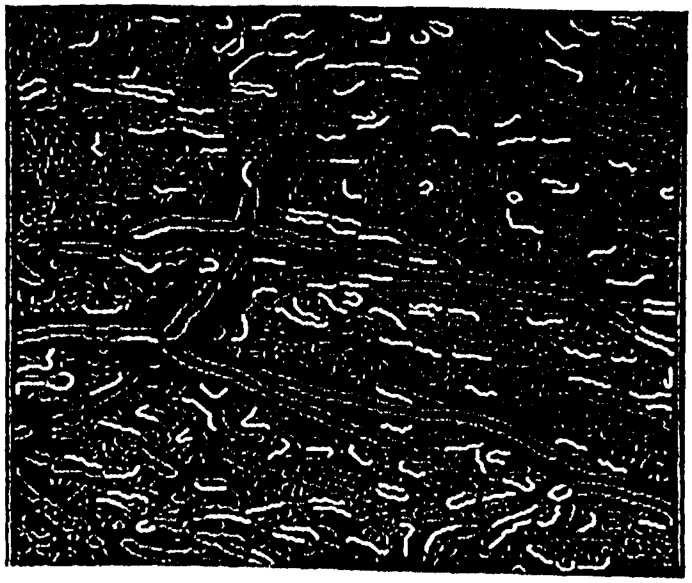

Figure 4-5: Curves Between 10 and 20 Pixels Long

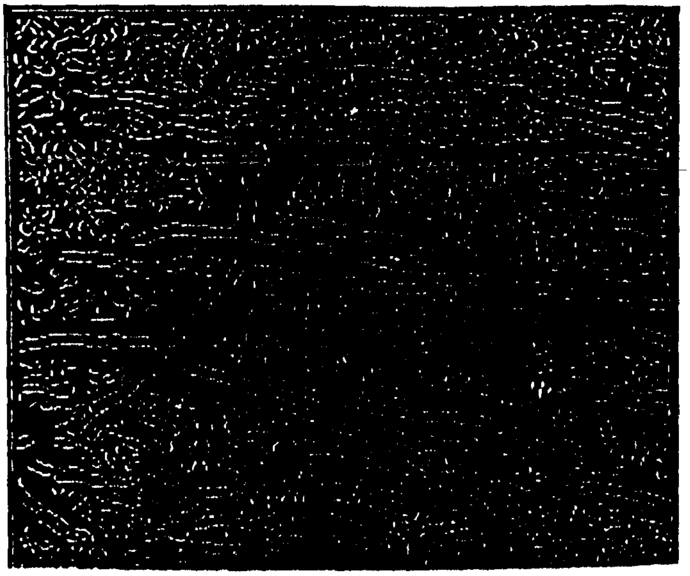

Figure 4-6: Multiple Attribute Query

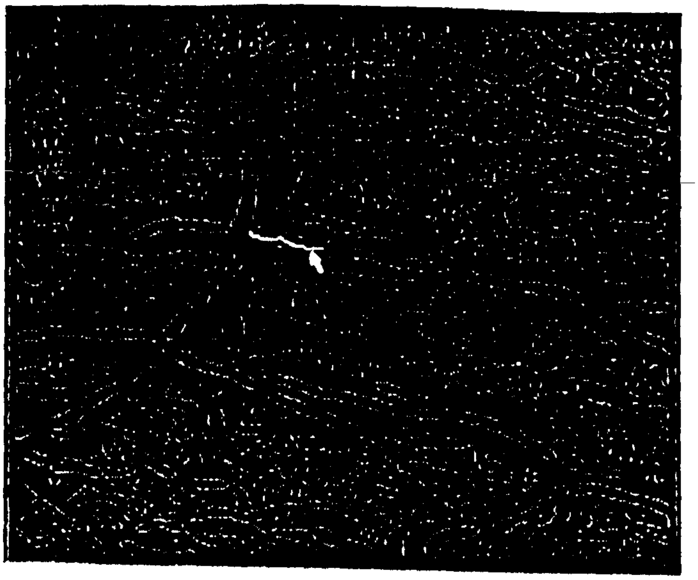

Figure 4-7: Manual Selection of an Interesting Curve

Note: Colors in original of report do not reproduce here.

Figure 4-8: Curves With a Similar Orientation

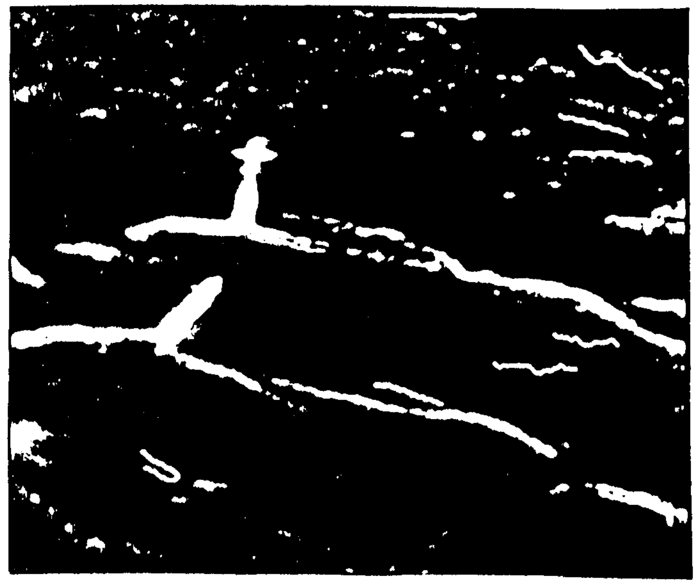

Figure 4-9: Curves With a Similar Size

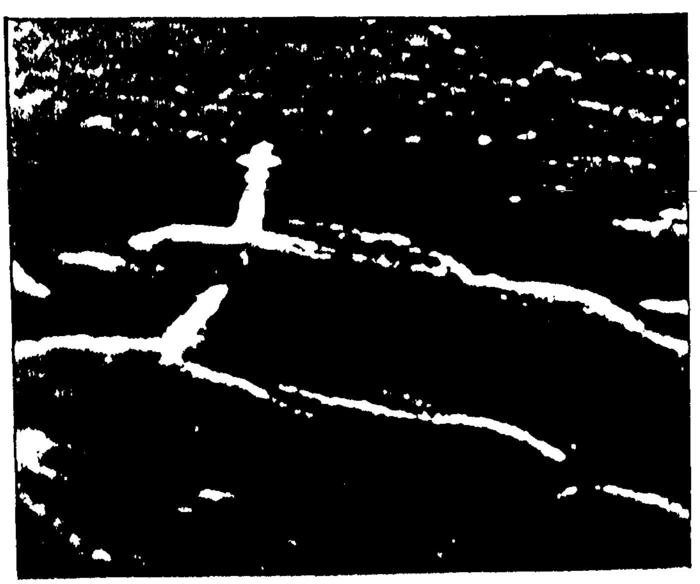

Figure 4-10: Curves Near the First Endpoint

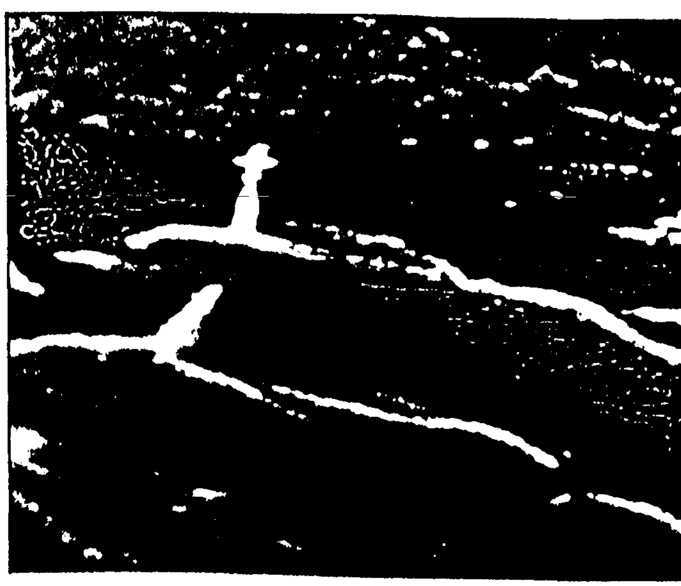

Figure 4-11: Curves Within a Projected Cone

Note: Colors in original of report do not reproduce here.

Figure 4-12: "Brightest" Region

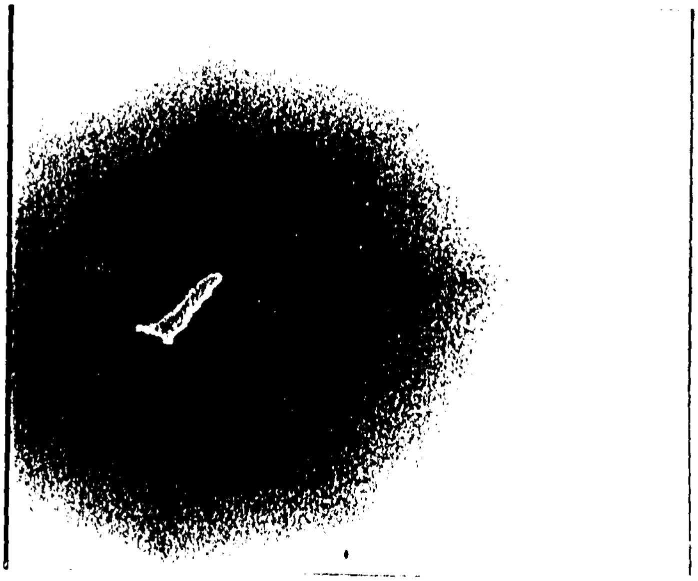

Figure 4-13: Distance From Blob Displayed as a Function of Intensity

Figure 4-14: Curves that are Close to the Selected Region

5. PROJECT STATUS

5.1 PROJECT PLAN

The goal of the Linear Feature Extraction Phase II SBIR is to develop an automated linear feature extraction system for radar imagery.

The major steps in achieving a capable linear feature extraction system are as follows:

1. Develop the appropriate working environment to register, manipulate, and process imagery.
2. Develop and experiment with various segmentation and feature extraction algorithms.
3. Determine significant terrain object feature properties and construct representative object models.
4. Experiment and evaluate model to image feature matching schemes.
5. Develop an approach for managing the competing and conflicting hypothesis matches.
6. Develop feature finders/predictors to support or contradict an expected terrain feature's existence.
7. Implement a display interface to support the above processing steps.

This project is divided into three parts.

Base Contract - (6 months) Undertake and complete the design of an automated linear feature extraction system for SAR imagery.

Option I - (9 months) Undertake and complete the development of all necessary software for the core system components of such a system. Work will also begin for recognition technique development and the system development. (This option overlaps the previous phase by three months.)

Option II - (12 months) Complete the work on the recognition technique development and the system development work began in the previous effort.

5.2 REVIEW OF PROGRESS

5.2.1 Base Contract

The work performed by ADS under the Base Contract and continued under Option I has addressed three problem areas.

The primary work performed under this contract has been the continuation of the design produced in the Phase I SBIR effort. The results of that design are described in the Linear Feature Extraction from Radar Imagery Base Contract Final Technical Report [Conner - 87].

The second major area in which ADS pursued the project goals was the development and design of a software environment in which to perform experiments and begin to build the eventual prototype system. The basic framework of this software was delivered to ETL in May 1987. The delivery emphasized neighborhood and display operations. The software also contained the necessary software "hooks" for future expansion into the other system components.

Finally, the last area of work undertaken as part of the Basic Contract was the continued experimentation with the government-provided radar imagery. Experimentation included algorithm surveys, hand processing of sample imagery, and actual algorithm implementation. This work and ADS's general understanding of machine vision has continually supported the design and development of the components of a model based vision system for linear feature extraction. The work described above corresponds to significant progress in Steps 1, 2 and 7, and has established the infrastructure for continuing work on the other steps of the Project Plan.

As the proper environment has been established, this system for determining and extracting terrain features is being developed and extensively tested. These experiments further establish the role of autonomous feature extraction from SAR imagery and, indeed, the importance of SAR imagery to map generation.

5.2.2 Option I Contract

The bulk of the work accomplished under this effort pertained to the continuing effort to embody the system design in software. A major software delivery to ETL of the processing framework was made in September 1987. The software included the following:

- Many of the relevant image processing routines used at ADS (see note below on operating system version compatibility).
- The software for creating, manipulating, accessing, and editing image structures (also called "perceptual structures").
- The preliminary framework of the hypotheses database. (This database contains hypotheses about extended image structures. Functions that

provide for the creation of these structures are embodied in the "filter" functions.)

- Enhanced user interface to display the image structures.

The software was also accompanied by a "User's Guide." The guide was written with the expert Symbolics Lisp Machine user in mind. At the suggestion of ETL, a supplemental guide was issued to address the needs of those users not intimately familiar with the Symbolics environment. In addition to the documentation, two sessions were held at ETL. The first session was a general "demonstration" of the software delivered. The second session was oriented towards familiarizing the user with the software. Given the size and complexity of the development environment, a subsequent visit was scheduled in December 1987 to further assist ETL personnel in the use of the system. During this visit some software "bug" fixes were also accomplished.

As expected, the system design continues to evolve as more of the system becomes realized in software. An updated system design will be submitted in the Option II final report.

Work was also initiated on the recognition procedures. The details of the various terrain features were studied. In addition to the standard properties of the individual features, of particular interest is both the internal and external structures of the features. For example, the apparent image-based structure of a patch of forest may be comprised of the textured area representing the bulk of the forest, the bright leading edge of the patch, and the trailing shadowed region. All three portions have entirely different "visual" characteristics, but each is an important component of the recognition of the forest patch. An example of external structures is best illustrated by a bridge. Typically, a bridge is detected as a long, thin bright region. Unfortunately however, this is not a unique signature by itself. If this bright region has roads extending from both ends and is surrounded on each side by water, then a unique signature for a bridge begins to form. Because this work in image object structure is only preliminary, details will not be provided until the final report for the Option II phase which will specifically address the area of recognition procedures.

A continuing source of difficulty facing the Linear Feature II project is the compatibility of software environments at ADS and ETL. Much of the Linear Feature I work was performed on a Symbolics system running the version 6 OS operating system. At the beginning of the Linear Feature II contract both ETL and ADS were running Version 6 OS. Since then ETL has installed Version 7 while ADS has not. ADS made a commitment early on to deliver software in Version 7 OS. This extra effort and overhead requires additional time and money to port software between versions, thus delaying delivery of important additions and bug fixes to ETL.

6. REFERENCES

[Barrow - 78] Barrow, H.G., Tenebaum, J.M., Bolles, R.C., and Wolf, H.C., "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching." *Proceedings of the DARPA Image Understanding Workshop,* May 1978.

[Binford - 81] Binford, T.O., "Inferring Surfaces from Images", *Artificial Intelligence,* Vol. 17, August 1981.

[Binford - 82] Binford, T.O., "Survey of Model-Based Image Analysis Systems", *The International Journal of Robotics Research,* Vol. 1, No. 1, Spring 1982.

[Brown - 67] Brown, W.M., "Synthetic Aperature Radar." *IEEE Transactions on Aerospace and Electronics Systems,* Vol. AES-3, No. 2, pp. 217-229, March 1967.

[Brown - 69] Brown, W.M. and Porcello, L.J., "An Introduction to Synthetic Aperature Radar." *IEEE Spectrum,* pp. 52-62, September 1969.

[Canny - 83] Canny, J., "A Variational Approach to Edge Detection." *AAAI-83,* pp. 54-58, August 1983.

[Conner - 87] Conner, G.D., Lawton, D.T., McConnell, C.C., and Milgram, D.L., *Linear Feature Extraction from Radar Imagery,* Report No. ETL-0469, U.S. Army Topographic Laboratories, Fort Belvoir, Virginia, July 1987.

[Eaves - 87] Eaves, J.L. and Reedy, E.K., *Principles of Modern Radar,* Van Nostrand Reinhold Company, New York, New York, 1987.

[Hanson - 78] Hanson, A.R. and Riseman, E.M., "VISIONS: A Computer System for Interpreting Scenes." *Computer Vision Systems,* Academic Press, New York, New York, 1978.

[Lawton - 85] Lawton, D.T., Glicksman, J., Conner, G.D., and Drazovich, R.J., *Linear Feature Extraction from Radar Imagery,* Report No. TR-3075-01, Advanced Decision Systems, Mountain View, California, August 1985.

[Lawton - 86] Lawton, D.T., Glicksman, J., Levitt, T.S., and McConnell, C.C., "Terrain Models for an Autonomous Land Vehicle", IEEE International Conference on Robotics and Automation, California, April 1986.

[Lawton - 87] Lawton, D.T., Levitt, T.S., McConnell, C.C., Nelson, P.C., Black, M.J., Edelson, D.J., Koitzch, K.V., Dye, J.W., Binford, T.O., Chelberg, D.M., Kriegman, D., and Ponce, J., *Knowledge-Based Vision Techniques,* Report No. TR-1093-02, Advanced Decision Systems, Mountain View, California, November 1987.

[Leberl - 85] Leberl, F.W., Domik, G., and Kobrick, M., "Mapping with Aircraft and Satellite Radar Images." *Photogrammetric Record*, Vol. 11, No. 66, pp. 647-665, October 1985.

[Lowe - 85] Lowe, D.G., *Perceptual Organization and Visual Recognition*, Kluwer, Massachusetts, 1985.

[Marr - 82] Marr, D., *Vision*, W.H. Freeman, San Francisco, California, 1982.

[Martelli - 76] Martelli, A., "An Application of Heuristic Search Methods to Edge and Contour Detection." *Communication of the ACM*, Vol. 19, No. 2, pp. 73-83, February 1976.

[Milgram - 87] Milgram, D.L., Morgan, D.R., Miltonberger, T.W., and Binford, T.O., *Symbolic RF Signature Prediction*, Report No. TR-3145-01, Advanced Decision Systems, Mountain View, California, September 1987.

[Morgan - 87] Morgan, D.R., Chestek, R.A., Miltonberger, T.W., Neveu, C.F., Drazovich, R.J., Smith, F., Mostafavi, H., and Froeberg, P., *Broad Area Search (BAS) System*, Report No. 1055-01, Advanced Decision Systems, Mountain View, California, September 1987.

[Nevatia - 82] Nevatia, R., *Machine Perception*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[Ohta - 80] Ohta, Y., "A Region-Oriented Image Analysis System By Computer", Ph.D. Thesis, Kyoto University, Department of Information Science, Kyoto, Japan, 1980.

[Pascussi - 84] Pascussi, R.F. and Huffman, E.T., *Development of Description sets for the Unambiguous Characterization of Geographic Features on SAR Imagery*, Report No. ETL-0369, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, August 1984.

[Rock - 84] Rock, I., *Perception*, Scientific American Library, New York, 1984.

[Rosenfeld - 81] Rosenfeld, A. (Editor), *Image Modeling*, Academic Press, New York, New York, 1981.

[Rosenfeld - 82] Rosenfeld, A. and Kak, A.C., *Digital Picture Processing* (second ed.), Academic Press, New York, New York, 1982.

[Skolnik - 62] Skolnik, M.I., *Introduction to Radar Systems*, McGraw Hill, New York, New York, 1962.

[Stimson - 83] Stimson, G.W., *Introduction to Airborne Radar*, Hughes Aircraft Company, El Segundo, California, 1983.

[Ulaby - 82] Ulaby, F.T., "Radar Signatures of Terrain: Useful Monitors of Renewable Resources." *Proceedings of the IEEE*, Vol. 70, No. 12, December 1982.

[Witkin - 83] Witkin, A.P. and Tenebaum, J.M., "On the Role of Structure in Vision", *Human and Machine Vision*, Academic Press, New York, New York, 1983.

[Wong - 79] Wong, R.Y. and Hall, E.L., *Edge Extraction of Radar and Optical Images*, IEEE Document Number CH1428 - 2/79/0000015000.

[Witkin - 83] Witkin, A.P. and Tenebaum, J.M., "On the Role of Structure in Vision", *Human and Machine Vision*, Academic Press, New York, New York, 1983.

[Wong - 79] Wong, R.Y. and Hall, E.L., *Edge Extraction of Radar and Optical Images*, IEEE Document Number CH1428 - 2/79/0000015000.