ISG 1983-9

A Knowledge Guided
Image Interpretation Program

by

Mark R. Seyler

January 1983

File No. UIUCDCS-F-83-903

ISG 83-9

A Knowledge Guided

Image Interpretation Program

BY

Mark R. Seyler

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

January 1983

ISG 83-3

File No. UIUCDCS-83-F-903

This research was supported in part by a National Science Foundation Grant No. NSF MCS 82-05166.

Abstract

The task of photo-interpretation relies primarily on knowledge derived fron sources outside the images themselves. This paper describes a programmable image analysis system designed to integrate these knowledge sources with techniques used in machine photo-interpretation. An experiment is presented that illustrates how the system might be used.

Acknowledgments

I would like to express my gratitude to my advisor, Prof. R. S. Michalski, for providing the inspiration, guidance and resources that made this research possible.

Table of Contents

Illustrations

CHAPTER 1

Introduction

The task of automatic image interpretation is often cast as a problem in statistical pattern recognition. The techniques that have been developed proceed in general through the following steps. (Moik, 1980) :

(1) *Feature Extraction* : A set of relevant attributes are derived from the input image(s).

(2) *Develop Interpretation Rules* : Two approaches are used here (Steiner, 1970). In the first approach, test data from targets of interest are collected and a discriminant technique (usually statistical) is used to generate decision rules. Alternatively, a grouping technique such as cluster analysis is performed on the image (or some subset of it) and the resulting groups matched to targets. The statistical characteristics of each group can then be used to guide the interpretation.

(3) *Testing* : Decision rules generated from step 2 are applied to a set of test data and steps 2 and 3 are iterated until good results are obtained.

(4) *Classification* : Subsequent images of interest can then be classified.

These methods have been well studied and give good results for many interpretation problems. They differ considerably, however, from methods used by human photo interpreters.

Estes and Simmonett (1975) point out that "most problems in [human] image interpretation require the interpreter to have at his command knowledge derived not from the images themselves but from the relevant field or fields of study." It is the focus of this research to explore means by which the experts own problem specific knowledge (domain knowledge) can be used more explicitly in machine interpretation methods. In the pursuit of this objective the following specific issues will be explored :

CHAPTER 2

A Programmable Image Analysis System (PIAS)

2.1. Introduction

In a discussion of some pragmatic issues involved in machine vision Reddy (1978) states "the slow progress over the last decade [in vision research] is directly attributable to non-visual computational issues such as limitations of architectures, file structures, image databases and so on. Before we can prove or disprove a conjecture regarding a representation or control structure we must have a system providing rapid response, within which one can design experiments to understand the contributions of competing hypotheses." It was with regard to these sorts of issues that PIAS was designed. The design criteria called for a program with the following characteristics :

- *Simplicity* : The program was designed to be fast and highly interactive. It consists of only 8000 lines of Pascal.
- *Extensibility* : The PIAS system is programmable via a very high-level language.
- *Modularity* : The components of the system were designed independently. The picture data-base manager, for example, can be easily de-coupled from the rest of the system in order to use it alone or make design changes.
- *Versatility* : The program was designed as a tool for conducting a variety of image analysis experiments.

Perhaps the greatest simplification in the design of PIAS is with regard to the representation of scenes. Pictorial data is stored as a set of 2-dimensional arrays or frames. Each frame represents a specific attribute or feature and is referenced by a unique identifier. For example, an analysis task may begin with three frames representing the yellow, magenta rules are useful in that they represent compact 'chunks' of knowledge and explicitly reflect the decision making (inference) process. The rule representation used in this research (VL_{IA}) is a modification of the variable-valued logic calculus VL_1 developed by Michalski (1974). The calculus lies somewhere between propositional logic and first-order predicate logic (FOPL). It extends propositional logic by allowing variables with domains of different sizes. It is easier for non-logicians to use since its syntax is closer to natural language than FOPL. For more details concerning VL_1 see Michalski (1974). A similar formalism useful in pattern recognition problems can be found in Michalski (1980). In order to illustrate the use of VL_1 as a picture processing language, a number of examples will be presented.

Consider the following decision rule:

[color(n1) = green] -> [identity(n1) = trees]

The above formula can be read as "If the color of the current pixel is *green*, then assign the value *trees* to the same pixel in the frame representing identity." Suppose this rule were applied at every pixel in a frame representing color, having the values blue, green and red. The result would be another frame representing identity that has the value or label *trees* everywhere its color is *green*. The text enclosed in square brackets ([ ]) is called a selector and represents some relation between a variable and its values. The arrow symbol (->) represents a decision symbol. When a selector occurs on the left-hand side (LHS) of a decision symbol it represents a state that must be true in order for the action on the right-hand side (RHS) to occur. In the example, a true LHS condition results in the assignment of the value *trees* to the frame *identity*. Observe that the formula follows quite naturally from its natural language counterpart.

Consider the similar formula :

logical OR and between selectors OR is symbolized as a plus sign (+).

Functions over neighborhoods are allowed as in the formula :

true -> [tone(n1) = average(tone(n1..n9))]

The current pixel is assigned the average of the current pixel and its eight nearest neighbors. The constant TRUE on the LHS of the rule means its RHS should always be executed. When this rule is applied to every pixel in the frame *tone* it implements a smoothing operation. When the same frame (tone in this case) is searched for values and also assigned values within a rule, the rulegroup can be iterated.

2.3. The VL_{IA} Language Compiler/Interpreter

2.3.1. Design Considerations

As mentioned earlier, the problem of air photo interpretation requires considerable specialist's knowledge. A body of research has emerged recently concerned with just such problems. The programs that have resulted are generally referred to as knowledge-based expert systems (See Michie, 1978 for a review). The computational formalism favored in much of this work is the production system (Duda and Garvey, 1980). A production system consists of three major components: (i) a global database within which are represented facts concerning the problem at hand (ii) a set of rules that read and write from the global database and (iii) a control module that determines which rule to consider next. Such systems gain their power from their explicit separation of domain knowledge (rules) from the mechanism which controls their application (Davis and King, 1977). The knowledge embedded in the program (represented as rules) is declarative rather than procedural. As a consequence, each rule is independant of the others and can be modified with no effect on the other rules.

(1) *Identification Block* (ID) : Consists of the keyword ID followed by a alpha-numeric string of 10 characters or less. It is terminated by a semicolon. The identification block is used to reference a rule group and initiate its executation.

(2) *Informational Block* (INFO) : Consists of the keyword INFO followed by any 64 characters beginning at the first non-blank character. It is terminated by a semicolon. The informational block provides space for additional text describing the function of the rule group.

(3) *Frame Declaration Block* (VARS) : Consists of the keyword VARS followed by a frame identifier, an equals symbol and the type of the frame enclosed in parentheses. The frame can be of type INTeger, REAL, CHARacter or defined by a string of labels. Each declaration is terminated by a semicolon *except* the last. When the rulegroup is executed, each frame declared is bound to a frame in local memory.

(4) *Rules Block* (RULES) : Consists of the keyword RULES followed by one or more rules of the type described in section 2.2. Each rule is terminated by a semicolon *except* the last one.

(5) *End Keyword* (END) : Consists of the keyword END followed by a semicolon. This symbol marks the end of a rule group.

Comments can be embedded anywhere in the text of a rulegroup by enclosing them in (* and *). An example of a rule group is presented below :

2.3.4. Intrinsic Functions

As in the neighborhood averaging rule presented earlier, the VL_{IA} compiler/interpreter recognises a number of intrinsic functions. These intrinsic functions serve two purposes : (i) as a means of implementing arithmetic operations between pixels and (ii) to mask specific pixels in the current neighborhood. The functions currently implemented in the program are :

- **MASK** : This function eliminates all pixels whose values are different than the current pixel.
- **UNMASK** : This function eliminates all pixels whose values are the same as the current pixel.
- **MODE** : Returns the value of that pixel that occurs most frequently.
- **MAXFREQ** : Returns the number of occurrences of the most frequently occurring value.
- **AVERAGE** : Returns the arithmetic average of the values.
- **MIN** : Returns the minimum of the values.
- **MAX** : Returns the maximum of the values.
- **COUNTS** : Returns the number of valid values. A value may have been eliminated (set to don't care) by a masking function.
- **MEDIAN** : Returns the median of the values.
- **EDGE1** : Calculates Roberts gradient.
- **EDGE2** : Calculates the Prewitt edge operator.
- **CONTRAST** : Calculates the contrast texture measure as presented in Tamura et al. (1977).

CHAPTER 3

Image Analysis Techniques

4. Introduction

In many knowledge-based programs the facts needed to manipulate its knowledge are asked of the user. The problem is compounded in automatic image interpretation by the need to extract the facts (features) from the images themselves, as well as perform the interpretation. This section presents some of the variety of techniques that can be implemented in VL_{IA}. Many of these operations are described in Rosenfeld (1982).

4.1. Image Enhancement Operations

This class of operations includes smoothing, sharpening and grey-scale modification. They are often used to compensate for the effects of quality degradation during the transfer of images from one form to another. Although they represent the backbone of low-level operations necessary to perform digital image processing, they can be viewed in another light. For example, the edge detection techniques can be used to identify roads or rivers in aerial images (a relatively high-level task). Histogram specification can be used to threshold and segment an image.

Image averaging is implemented using the AVERAGE intrinsic function. For example, the rule for nine neighbor averaging is :

true -> [frame(n1) = average(frame(n1..n9))]

The neighborhood over which averaging takes place can be varied and the operator iterated.

Another class of features useful in image interpretation are geometric measures. Such measures rely on the concept of a connected group of pixels (segment). In order to implement these measures, the intrinsic function CONNECT4 has been defined in VL_{IA}. It marks all those pixels in the local neighborhood that are four-connected to the current pixel. An example of a rule that calculates the size of a segment is :

[size(n1) = unknown]
->
[size(connect4(frame(n1..n49))) = counts(connect4(size(n1..n49)))]

The condition that size be unknown in the rule is for efficiency, since all pixels that are four-connected to the current pixel are assigned the measurement. Other shape measures can be calculated using template matching techniques and by varying the neighborhood size and shape.

Another approach to calculating geometric features is by utilizing a simple set of shrink, fill and boolean operations on binary images (Juhn, 1981). For example, if it is known that a particular target is roughly circular and of a particular size, all of the segments in the image can be shrunk and expanded (filled) according to the radius of those circular targets. The result effectively filters out all targets of smaller size. The PIAS system in its simpliest configuration can function as a eight-connected cellular array processor that handles binary images.

4.3. Contextual Processing

One class of operations that deal with local, contextual relations are cooperating or relaxation processes (See Davis and Rosenfeld, 1980 for a review). The relaxation process attempts to reduce the ambiguity in an image where the labelling assignment is not unique for all objects in the image. Knowledge in the system is represented by relations that represent valid neighborhood conditions (constraints). Certainty is propogated through the image either by discarding labels (discrete relaxation) or having some mechanism for updating certainty values assigned to labels (probablistic relaxation). Spatial information is propogated through

$$
\begin{array}{c}
[\mathrm{veg}(N1) = A..B]\ \ [\mathrm{soil}(N1) = A..C]\ \ [\mathrm{size}(N1) = D..E] \\
+ \\
[\mathrm{veg}(N1) = A..F]\ \ [\mathrm{soil}(N1) = A..B]\ \ [\mathrm{size}(N1) = G] \\
+ \\
[\mathrm{veg}(N1) = D]\ \ [\mathrm{soil}(N1) = C..D]\ \ [\mathrm{size}(N1) = C] \\
+ \\
[\mathrm{veg}(N1) = A..E]\ \ [\mathrm{soil}(N1) = A..B]\ \ [\mathrm{size}(N1) = A] \\
+ \\
[\mathrm{veg}(N1) = A]\ \ [\mathrm{soil}(N1) = A]\ \ [\mathrm{size}(N1) = G..I] \\
\text{->} \\
[\mathrm{class}(n1) = \mathrm{water}];
\end{array}
$$

in this example, the attributes *veg*, *soil* and *size* are interval scaled from A to I. The letter A represents a low value for that attribute and the letter I the highest value. The range operator (..) between domain values means that the attribute can take on any of the values in the range to be satisfied.

Techniques of machine learning can be used to identify patterns in large, complex data sets. The results serve to augment the domain experts understanding of the problem. As Michalski (1981) points out, the development of knowledge-based expert systems requires efficient methods for acquiring and refining knowledge. Such tools will become increasingly important as knowledge bases increase in size and scope.

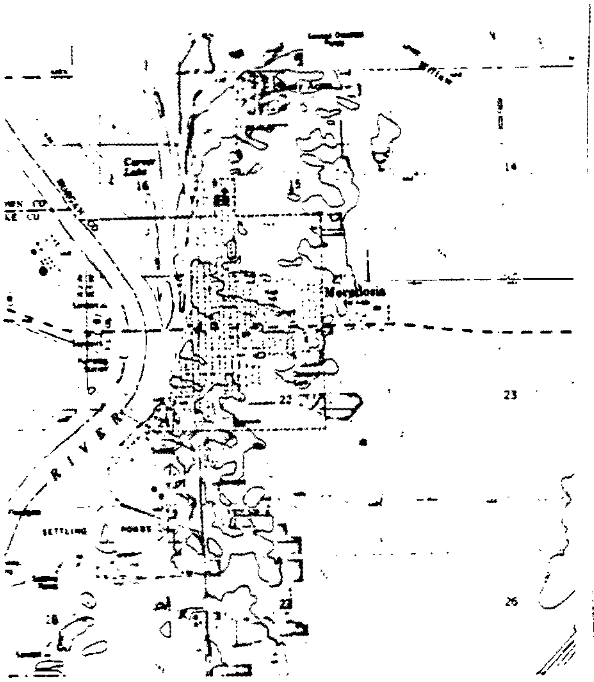

Figure 1 : Map of the Meredosia area from a portion of the Meredosia, Illinois USGS topographic quadrangle.

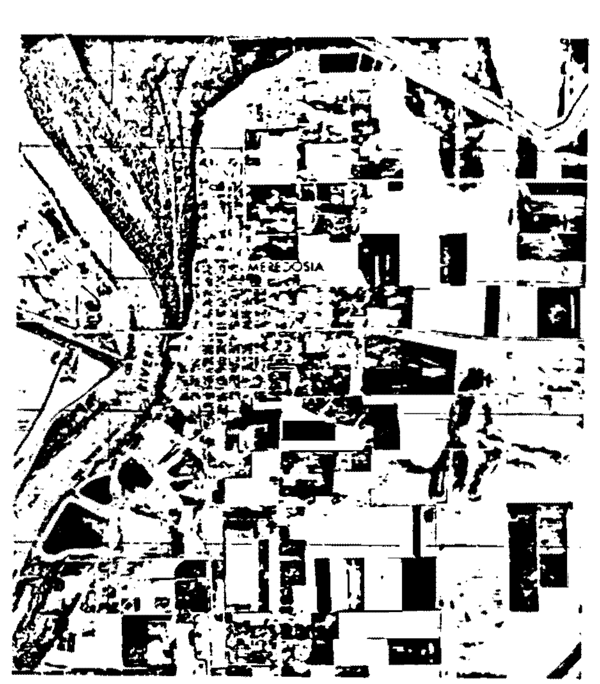

Figure 2 : Map of the Meredosia area from a portion of the Meredosia SE, Illinois USGS orthophoto quadrangle.

attributes will be referred to as *spectral* features in the remainder of the paper. The histograms of these two attributes were then equalized to compress grey levels in the extreme portions of the distribution. The histograms were scaled to nine grey levels and the resulting *spectral* attributes pictured in Figures 3 and 4. Dark regions in the *brightness* image correspond to absorptive targets in the visible spectrum, and dark regions in the *greenness* image represent very low concentrations of vegetation. For example, the Illinois River is apparent in both images as the dark, curved region on the left. It appears dark because water absorbs radiation in both the visible and near-infrared portions of the spectrum.

combination of trees and grass lawns (high *greenness* response, low to high *brightness*) and streets, roof tops and other high *brightness*, low *greenness* targets. Distinguished in a large part by textural qualities at medium to high resolution.

- **Urban Commercial (COMMERCIAL)** : Commercial areas used primarily for the sale of products and services. Characterized by very high *brightness* and very low *greenness*. In the midwest United States, the commercial district is often centralized in the urban core and in strips along major transportation routes.

- **Urban Services, Other Urban (SERVICE)** : This category includes institutions such as schools and hospitals, parks, golf courses and the like. Usually located on the urban periphery, particulary in medium to small towns. Characterized by open grass lands (high *brightness*, very high *greenness*) with associated buildings and parking lots.

- **Industrial (INDUSTRIAL)** : Light to heavy manufacturing and other light industries. Usually located outside urban areas and in particular along the Illinois river in this study area.

- **Woods (WOODS)** : Forested areas distinguished by a high *greenness* response and a low *brightness* response (due to shadows). The assocaited texture can often be used to identify species and stages of growth.

- **Forested Wetlands (WETVEG)** : Wetlands dominated by woody vegetation, possibly seasonally flooded areas. A medium response to *greenness* and very low response to *brightness*. Associated with rivers and large bodies of water.

- **Fallow (FALLOW)** : Indicates any undeveloped land dominated by grasses, shrubs and the like. Pastures, ditches along transportation arteries, and farmsteads are included in this class. Medium *greenness* and *brightness* responses.

- **Row Crop (ROWCROP)** : In Illinois the principle row crops are corn and soybeans.

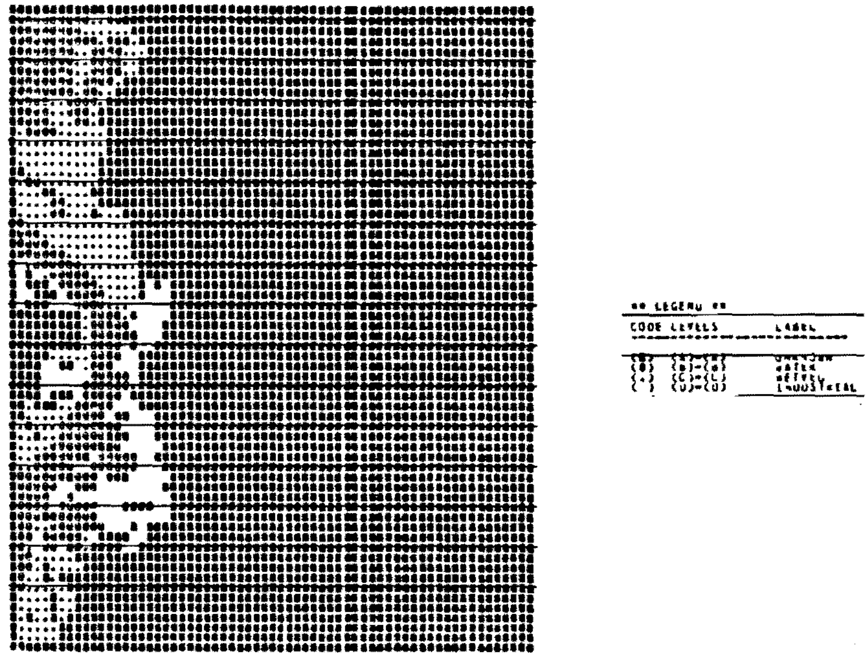

Figure 5 : The identification of wetland vegetation and industrial sites using water as a context.

examination of the *greenness* and *brightness* images. On neither image is any meaningful texture evident. As Tamura et al. (1977) concluded "we should apply textural features only to a problem that we perceive in advance to be solvable from the textural point of view." In this particular study area, texture would have played a more important role had the resolution of the Landsat data been higher.

The rules generated in the previous step using *spectral* features alone were successful in identifying the following classes :

- WATER
- NONROW

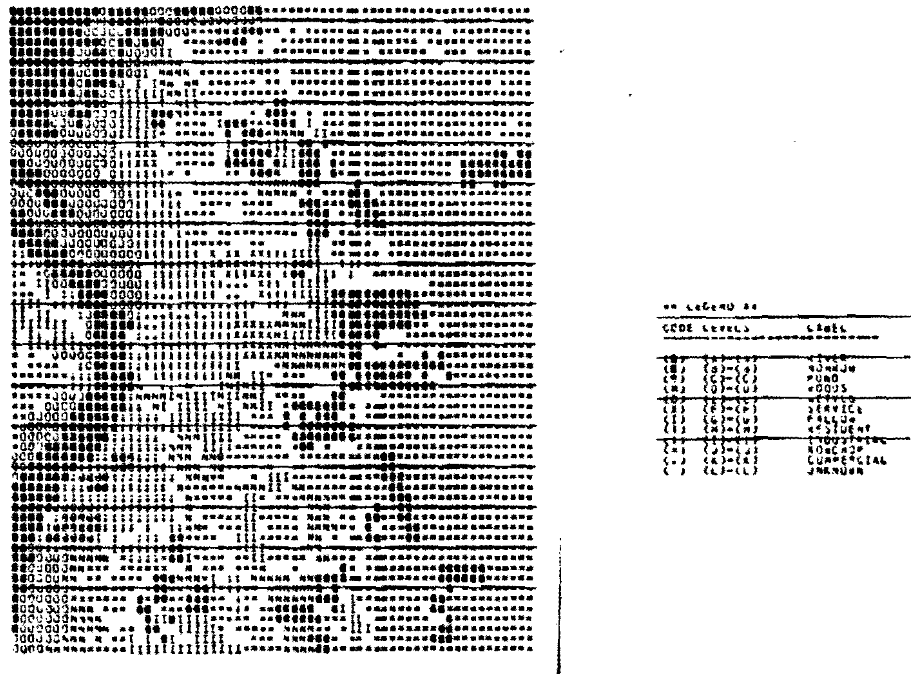

Figure 6 : The machine classification after ordering all classes by reliability and combining them.

At this point there were 10 frames each corresponding to one of the classes used in the study. These frames were ordered according to reliability of class identification and combined. This ordering represented the confidence in that identification. The combination process assigned each pixel to that class that had the highest confidence. The results (Figure 6) demonstrate that what remained to be identified were primarily border pixels. These were assigned to that class that occured most frequently among the unknown pixels nearest thirteen neighbors. As a final step, the size of the four-connected regions identified as water were calculated and regions of less than 25 pixels labelled as pond.

Contingency Table													
	Riv	Nrw	Pnd	Wds	Wvg	Ser	Flw	Res	Ind	Row	Com	Total Error	Percent Error
Riv	219	0	0	0	13	0	0	0	7	8	0	28	11.3
Nrw	0	281	0	44	0	0	17	0	0	42	0	103	26.8
Pnd	0	0	31	0	4	0	0	0	0	12	0	16	34.0
Wds	0	74	0	103	8	5	16	0	3	10	0	116	53.0
Wvg	7	3	0	3	255	1	5	2	1	3	0	25	8.9
Ser	0	0	0	0	0	48	2	2	0	1	2	7	12.7
Flw	0	25	0	25	1	18	139	3	1	33	0	106	43.3
Res	0	0	0	2	2	20	0	160	2	3	8	37	18.8
Ind	5	1	14	1	6	0	11	0	114	35	1	65	34.5
Row	0	55	0	7	5	2	72	3	12	1548	0	156	9.2
Com	1	0	0	0	0	0	0	5	5	0	23	8	23.0

Table 2 : Contingency table comparing the results of the expert and machine interpretations.

Some care must be taken when comparing the results of a machine classification with that of a photo interpreter. Considerable error can occur along borders, since in both cases these pixels are assigned arbitrarily. Suppose 20% of an image consisted of edges. Also assume that edge pixels are randomly assigned to one of the two classes that occur on either side. If this process were repeated twice on the same image and the results compared, one would expect about 10% disagreement in the identification of edge pixels. It is clear from examining the final frames that much of the error in machine classification occured along border areas.

The development of the final set of procedures used in the machine classification invloved considerable experimentation. Some general principals emerged during the course of this experimentation that might prove useful when addressing similar problems. Little was lost by reducing the quantization of variables to a small set of values. The raw spectral data was quantized to 128 grey levels, but this was reduced to 9 grey levels throughout the experiment with no apparent loss of information. It proved valuable to make general observations first and resolve details later. For example, many of the edge pixels in the experiment were assigned

CHAPTER 5

Summary and Further Directions

In this research the question of knowledge utilization in the development of automatic image interpretation strategies has been explored. An interactive environment (PIAS) was created to assist in developing such strategies and performed well, given a difficult problem in aerial image interpretation. The experiences gained while designing and working with the program suggest numerous avenues for improvement and further research.

Although the VL_{IA} language proved adequate for representing a variety of image operations, two extensions to the language would improve its usefulness. The first is to allow for quantification over neighborhoods. This would provide a general mechanism for referencing only certain values in a local neighborhood. A second improvement would allow functions to be defined that have more than one argument. With this extension a more sophisticated set of intrinsic functions could be defined.

One consequence of the pixel-level scene representation was that much of the processing involved low-level operators. This sort of processing is foreign (at least consciously) to human photo interpreters. The objects manipulated by VL_{IA} were single pixels or groups of pixels within a local neighborhood. Local spatial relations such as adjacency were easily captured using this scheme but it became inefficient when attempting to represent global and structural spatial relations. There is a clear need for a higher level of representation both of objects within the scene itself and the language that manipulates these objects. However, as argued in section 2.1, access to the low-level, spatially isotrophic representation (the two dimensional array) should be allowed at all levels of representation. An example of such a system is described in Lemmer (1982).

Literature Cited

Abdou, Ikram E. and Pratt, William P. (1979) "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," *Proc. IEEE* , vol. 67, no. 5.

Anderson, James R., Hardy, Ernest E., Roach, John T. and Witmer, Richard E. (1976) "A Land Use and Land Cover Classification System for Use with Remote Sensor Data," Geological Survey Professional Paper 964.

Barrow, Harry G. and Tenenbaum, Jay M. (1981) "Computational Vision," *Proc. IEEE* , vol. 69, no. 5.

Davis, R. and King, J. (1977) "An Overview of Production Systems." In Elcock, E.W. and Michie, D. (eds) *Machine Intelligence 8: Machine Representations of Knowlege* Ellis Horwood.

Davis, Larry S. (1980) "A Logic Model for Constraint Propogation," Univ. of Texas Computer Sciences Dept., TR-137.

Davis, Larry S. and Rosenfeld, Azriel (1981) "Cooperating Processes for Low-Level Vision: A Survey," *Artificial Intelligence* , 17.

Duda, Richard O. and Garvey, Thomas D. (1980) "A Study of Knowlege-Based Systems for Photo Interpretation," U.S. Army Topographic Laboratories Report ETL-0235.

Estes, John E. and Simonett, David S. (1975) "Fundementals of Image Interpretation." In Reeves, Robert G. (ed) *Manual of Remote Sensing,* American Society of Photogrammetry.

Fischler, Martin A. (1978) "On the Representation of Natural Scenes." In Hanson, Allen R. and Riseman, Edward M. (eds) *Computer Vision Systems* , Academic Press.

Haralick, Robert M. (1979) "Statistical and Structural Approaches to Texture," *Proc. IEEE,* vol. 67, no. 5.

Hanson, Allen R. and Riseman, Edward M. (eds)(1978) "Segmentation of Natural Scenes," *Computer Vision Systems,* Academic Press.

Michie, Donald (ed)(1979) *Expert Systems in the Micro Electronic Age* , Edinburgh University Press.

Moik, Johannes G. (1980) *Digital Processing of Remotely Sensed Images* , NASA.

Nevatia, Ramakant and Price, Keith E. (1982) "Locating Structures in Aerial Images," *IEEE Trans. PAMI* , PAMI-4, no. 5.

Philipson, Warren P. and Liang, Ta (1982) "An Airphoto Key for Major Tropical Crops" *Photogrammetric Engineering and Remote Sensing*, vol. XLVIII, no. 2.

Reddy, Raj (1978) "Pragmatic Aspects of Machine Vision." In Hanson, Allen R. and Riseman, Edward M., *Computer Vision Systems* , Academic Press.

Rosenfeld, Azriel and Kak, Avinash L. (1982) *Digital Picture Processing* , Academic Press.

Steiner, D. (1970) "Automation in Photo Interpretation," *Geoforum* , vol. 2.

Tamura, H., Mori, S. and Yamawaki, T. (1976) "Psychological and Computational Measurements of Basic Textural Features and their Comparison," *Third Intl. Joint Conf. on Pattern Recognition* , Corondo, California.

Tamura, H., Mori, S. and Yamawaki, T. (1977) "Effectiveness of Textural Features for Classification of Aerial Multispectral Images," *Proc. Conf. Pattern Recognition and Image Processing* , Troy, New York.

Wirth, Nicklaus (1976) *Algorithms + Data Structures = Programs*. Prentice-Hall.

all purged disk frames (garbage collection).

- **RETURN**,<F1>,<F2>,...,<F3>: Deletes a frame from working memory.
- **CLEAR**: Deletes all frames from working memory.
- **DIRECT**,<F1>,<F2>,...,<F3>: Displays a directory of all frames on disk.
- **LFNS**,<F1>,<F2>,...,<F3>: Prints a directory of all frames in working memory.
- **READ**,<F1>,<F2>,...,<F3>: Copies a frame from the file 'ALT1' to working memory. This command is used to initially load a disk data base.
- **CHANGE**,<F1>,<F2>,...,<F3>: Changes header information of a frame in working memory.
- **COPYM**,<F1>,<F2>: Copies frame F1 into F2. F1 must be in working memory. If F2 is in working memory it is overwritten, otherwise a new frame F2 is created.
- **NEW**,<F1>,<F2>,...,<F3>: Creates a new frame(s) in working memory. It is initialized to zero and all ancillary fields are blank.

3. Image Processing Utilities

A variety of utilities are available for processing, displaying and summarizing frames in working memory. Many of these operations are programmable in VL/IA but are implemented as separate commands for reasons of efficiency.

- **PICTURE**,<F1>,<F2>,...,<F3>: Prints a picture of a frame(s). Each value in the frame is assigned a character, or set of overprint characters, corresponding to its intensity. For example, a low value is assigned a dark character.
- **HISTOGRAM**,<F1>,<F2>,...,<F3>: Prints a histogram of a frame(s). A histogram gives the frequency of occurence of each value in the frame.
- **TYPE**,<F1>,<F2>,...,<F3>: Prints a frame as it is stored internally (as an array of characters). The 'ord' of each character corresponds to its intensity value. The character set used is Cyber Display Code.

generate test data and the results can be input into AQ-11.

4. Miscellaneous Commands

Included here are commands that ease interaction with the program and perform miscellaneous operations on frames.

- **HELP or HELP,<command>**: Prints the list of commands known to the system. If HELP is followed by a command name a short description of that command is given.
- **END**: Terminates the program. The user is asked whether he wishes to eliminate all purged files from disk.
- **STOPWATCH,ON or STOPWATCH,OFF**: If the stopwatch is on the cpu processing time after each command is given.
- **RULEGROUPS**: Prints a list of the rulegroups known to the system.
- **HEADER,<F1>,<F2>,...,<Fn>**: Prints header information about a specified frame(s).
- **LABEL,<F1>,<F2>,...,<Fn>**: Allows the user to assign 10 character labels to each value in a frame(s).
- **COMMENT**: Allows the user to insert a comment into the message file (MSG).

```
* includes a number of averaging operators using several
* neighborhood sizes (AVE5,AVE9,etc.), edge measures, texture
* measures and some size measures.
*

COMMAND?? help,get

 GET,<F1>,<F2>,...,<FN>:

 LOAD A FRAME(S) FROM DISK TO THE PROGRAM MEMORY.

COMMAND TIME... ELAPSED =  0.001  TOTAL =  0.142 (SECONDS)

COMMAND?? direct

PERMANENT FILES :
   CLASS      ID        FEATURE
   -------------------------
 1) LANDSAT    LS11      BRIGHTNESS
   LANDSAT 2, JUNE 2 1978, TUSCOLA-SE, CHEMICAL PLANT
 2) LANDSAT    LS12      GREENNESS
   LANDSAT 2, JUNE 2 1978, TUSCOLA-SE, CHEMICAL PLANT
 3) MISC       BOX       TONE
   4 BOXES AND A DOUGHNUT, REF. GONZALEZ AND WINTZ
 6) LANDSAT    LS31      BRIGHTNESS
   LANDSAT 2, JUNE 2 1978, TUSCOLA-SW, ATWOOD(NORTH) LAKE FORK(EAST
 7) LANDSAT    LS32      GREENNESS
   LANDSAT 2, JUNE 2 1978, TUSCOLA-SW, ATWOOD(NORTH) LAKE FORK(EAST
 8) LANDSAT    LS41      BRIGHTNESS
   LANDSAT 2, MAY 18 1976, MEREDOSIA-SE, MEREDOSIA
 9) LANDSAT    LS42      GREENNESS
   LANDSAT 2, MAY 18 1976, MEREDOSIA-SE, MEREDOSIA
10) LANDSAT    LS51      BRIGHTNESS
   LANDSAT 2, MAY 18 1976, BEARDSTOWN-SW, HILL,PLAIN,RIVER
11) LANDSAT    LS52      GREENNESS
   LANDSAT 2, MAY 18 1976, BEARDSTOWN-SW, HILL,PLAIN,RIVER
12) LANDSAT    LS61      BRIGHTNESS
   LANDSAT 2, AUGUSTA-NW, STRIP MINE
13) LANDSAT    LS62      GREENNESS
   LANDSAT 2, AUGUSTA-NW, STRIP MINE
14) LANDSAT    LS71      BRIGHTNESS
   LANDSAT 2, DANVILLE-NW, RANTOUL
15) LANDSAT    LS72      GREENNESS
   LANDSAT 2, DANVILLE-NW, RANTOUL
COMMAND TIME... ELAPSED =  0.012  TOTAL =  0.149 (SECONNNDS)

*
* The 'direct' command lists the frames available in the
* current IMAGES file. In this case there are a number
* of LANDSAT images and a picture of some boxes.
* Each frame is assigned an image class, id and feature.
* Frames are referenced by their id.
* An additional line of text is used to describe in
```

```
 1)(A)(   5)
 2)(B)(  49)**
 3)(C)( 380)***************
 4)(D)(1441)***********************************************************
 5)(E)( 590)************************
 6)(F)(  83)***
 7)(G)(  30)*
 8)(H)(  91)***
 9)(I)(  64)**
10)(J)(  53)**
11)(K)(  40)*
12)(L)( 153)******
13)(M)( 183)*******
14)(N)( 201)********
15)(O)(  46)*
16)(P)(   7)
17)(Q)(   1)
18)(R)(   2)
19)(S)(   4)
20)(T)(   8)
21)(U)(   8)
22)(V)(  17)
23)(W)(  45)*
24)(X)(  88)***
25)(Y)(  99)****
26)(Z)( 263)**********
27)(0)( 131)*****
28)(1)(  12)
29)(2)(   2)
```

EACH SYMBOL(*) REPRESENTS...24.42 COUNTS.

STATISTICS FOR FRAME...BOX

```
MIN.........    1
MAX.........    29
MEAN........    9.28
VARIANCE....  62.80
STND. DEV...   7.92
```

COMMAND TIME... ELAPSED = 0.212 TOTAL = 0.424 (SECONDS)

*

* *Each pixel in each frame is stored internally in six*

* *bits. Each line in the histogram represents the number*

* *of pixels in the frame of that particular value. For example,*

* *Line 1 of the histogram indicates that their are 5*

* *pixels of value 1. Frames are manipulated internally as*

* *arrays of characters. The letter 'A' associated with the*

* *value 1 is the character associated with postion 1 in*

* *the CYBER's collating sequence.*

*

COMMAND?? **prewedge**

FILE.....TONE

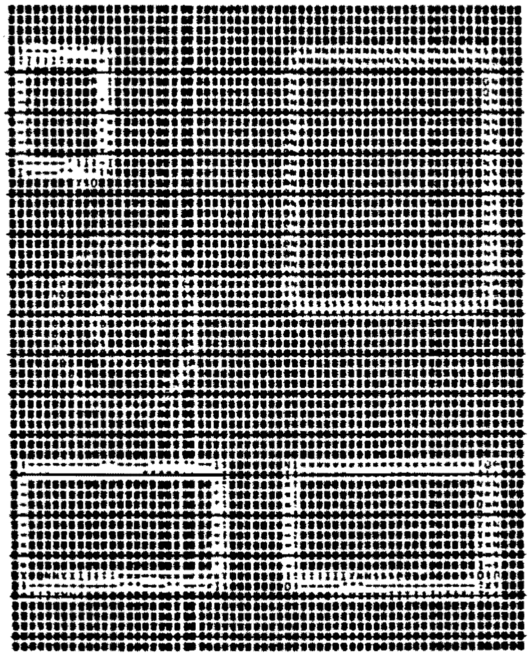

Figure 9 : Results of the Prewitt edge detector on the boxes image.

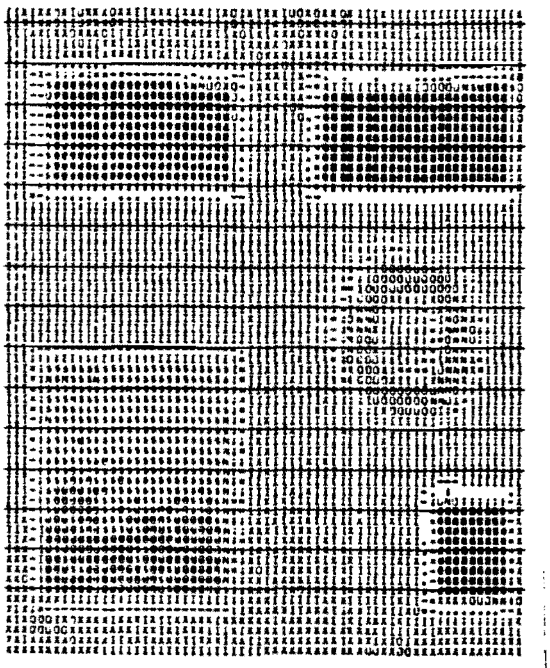

Figure 10 : The result of subtracting the original boxes image from its edge image.

RESULTS IN FILE...EDGE
COMMAND TIME... ELAPSED = 0.311 TOTAL = 4.540 (SECONDS)

COMMAND?? **picture,edge**
*
* *See Figure 10*
*
*
* *The results of subtracting a frame from the edge*
* *image is was created from is an enhancement of*
* *border regions.*
*

COMMAND?? **end**
COMMAND TIME... ELAPSED = 0.000 TOTAL = 4.542 (SECONDS)
DISK STATUS :
VALID FILES = 13
PURGED FILES = 2
BLANK FILES = 15
GARBAGE COLLECTION ON IMAGE FILE?(Y OR N) **n**
*
* *Any frames purged from disk can be physically deleted*

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-F-83-903	2.	3. Recipient's Accession No.

4. Title and Subtitle A Knowledge Guided Image Interpretation Program	5. Report Date January 1983
	6.
7. Author(s) Mark R. Seyler	8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois Urbana, IL	10. Project/Task/Work Unit No.
	11. Contract/Grant No. NSF MCS 82-05166
12. Sponsoring Organization Name and Address National Science Foundation Washington, DC	13. Type of Report & Period Covered
	14.

15. Supplementary Notes

16. Abstracts

The task of photo-interpretation relies primarily on knowledge derived from sources outside the images themselves. This paper describes a programmable image analysis system designed to integrate these knowledge sources with techniques used in machine photo-interpretation. An experiment is presented that illustrates how the system might be used.

17. Key Words and Document Analysis. 17a. Descriptors

Digital Image Processing
Pattern Recognition
Expert Systems
Machine Learning

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 54
	20. Security Class (This Page UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71