# A General Purpose Expert System for Image Processing

*Robert A. Schowengerdt*
Office of Arid Lands Studies, Electrical and Computer Engineering Department and Optical Sciences Center, University of Arizona, Tucson, AZ 85721
*Han-lung Wang*
Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ 85721

ABSTRACT: A prototype expert system, developed from two existing software packages, one an expert system development tool and the other an image processing library, is described. In contrast to application-specific expert systems, this system addresses generic problems in image processing. With knowledge about image processing coded in the expert system, the inexperienced user is able to perform useful image processing tasks. Two applications of the system, contrast enhancement and noise suppression, are discussed in detail.

## INTRODUCTION

REMOTE SENSING of the Earth from satellites and aircraft is intimately dependent on computer image processing technologies for analysis of the data. Related fields, such as geographic information systems (GIS), depend directly on remote sensing images for map revision, environmental analyses, temporal monitoring of changes on the Earth's surface, and numerous other Earth science studies.

Remote sensing and GIS have developed in concert with computer technology since the late 1960s. In many cases, new research fields in mapping and remote sensing have been opened and driven by advances in computer capabilities. This is particularly evident today with personal workstations that have large memory and mass storage capacity combined with high resolution color graphics. These computers represent a new direction for image processing at many laboratories, i.e., one in which large data sets can be shared via networks and processing is distributed over multiple workstations at different sites. As this capability is distributed directly to Earth scientists who may not have an in-depth knowledge of image processing techniques, there is a need for intelligent software advisers (knowledge-based or "expert" systems) to assist in data processing and analysis.

The application of artificial intelligence techniques to the general problem of image analysis (image interpretation or "understanding") has been of interest for some time (Duda and Hart, 1973; Nagao and Matsuyama, 1980; Ballard and Brown, 1982; Marr, 1982; Levine, 1985). However, the promise of this research has remained largely unfulfilled, because computer image analysis is an extremely difficult problem and the analogies with human vision are poorly understood at best (Rosenfeld *et al.*, 1986; Haralick, 1986). A more manageable goal in the near term is the application of knowledge-based systems to image processing, i.e., the pixel-level processing required to calibrate, rectify, and enhance images for interpretation. The field of knowledge-based systems is a relatively mature technology that can produce useful results in specific, albeit limited, applications (Duda and Shortliffe, 1983; Bobrow *et al.*, 1986; McKeown *et al.*, 1985; Perkins *et al.*, 1986; Goodenough *et al.*, 1987; Nichol, 1987; Schowengerdt and Mehldau, 1987) and offers considerable potential benefit for image processing.

## DESCRIPTION OF THE EXPERT SYSTEM

Computer programming tools available to scientists have progressed from assembly languages in the 1950s, through high-level languages like FORTRAN and LISP in the 1960s and C in the 1970s, to complete programming environments and standardized tools, such as the Macintosh user interface, in the 1980s. As these tools become more sophisticated, we can increasingly avoid the recoding of commonly used algorithms for use in new software systems, and we can access existing codes from higher level tools that can merge widely different systems.

We have taken this approach to construct a prototype expert system for image processing. The expert system structure, consisting of a database of facts, a knowledge base of rules, and an inference engine for control of the process, is contained in the C Language Integrated Production System (CLIPS) recently developed at the NASA Johnson Space Center (Giarratano, 1987). CLIPS was originally intended to run on personal computer systems wtih less than 256K bytes of memory. It is highly portable, and we have installed CLIPS Version 4.10 with little modification on a VAX system under VMS; the system development described here was performed on this system.

The image processing system used was the System at Arizona for Digital Image Experimentation (SADIE), a FORTRAN library of subroutines that has been developed over more than ten years at the University of Arizona and other locations. SADIE contains routines to do most of the commonly needed image processing functions, such as a variety of contrast enhancements, spatial filtering for edge enhancement or noise removal, geometric warping for rectification, and so forth. The integration of CLIPS and SADIE was accomplished by writing a C language interface through which CLIPS could call SADIE subroutines as appropriate. SADIE has recently been entirely rewritten in C as Version 4.0 and should, in that form, be more readily interfaced with CLIPS, although the present interface was not difficult to build. The overall structure of the CLIPS-SADIE system is shown in Figure 1.

As will be seen in the following discussion, our goal was to provide an intelligent assistant to the inexperienced user. However, the system allows full override of the "expert" if the user wishes to proceed on his own. We have also taken the philosophy that the computer system should not be an automaton that takes the raw image and creates an enhanced image with no user interaction or input. Such a goal is unrealistic in the near future because of our lack of understanding about visual processes. We have therefore restricted the problem domain in the prototype system and made use of the user's cognitive and visual skills when computer processing would be unfeasible. A similar cooperative approach to user-computer interaction has been applied to interactive cartographic feature extraction from images (Schowengerdt and Pries, 1988).

PHOTOGRAMMETRIC ENGINEERING AND REMOTE SENSING,
Vol. 55, No. 9, September 1989, pp. 1277–1284

0099-1112/89/5509–1277$02.25/0

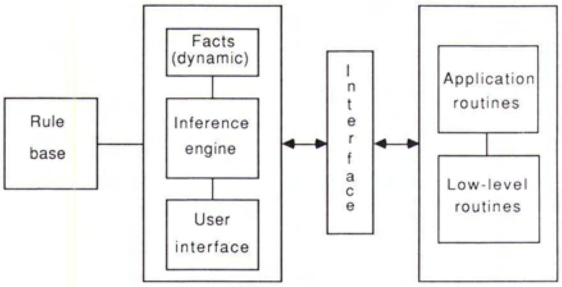


FIG. 1. CLIPS-SADIE structure.

An expert system for statistical data analysis, similar in philosophy to the one presented here in that it provides assistance in using a general purpose "toolbox" of routines, has also recently been described (Martin *et al.*, 1988). Another expert system, written in PROLOG, has been developed as a user interface to the Land Analysis System Landsat image processing system (Doescher *et al.*, 1988). This system is oriented to producing specific output products, such as shaded relief images of digital terrain models, and employs a similar question and answer user-interface to that described here.

## CLIPS

CLIPS was designed to be a portable, efficient, and capable expert system development tool (Giarratano, 1987; Culbert, 1987). It is essentially a very high level programming language, with characteristics of both C and LISP, in which the user does not need to be concerned about details of a program's logic flow and control. A CLIPS program consists of facts and rules defined by the user (Figure 1). The inference engine that relates the two is primarily a forward-chaining production system that reaches a goal by a monotonically progressing series of decisions. There is a less developed capability for backward-chaining inference in CLIPS where a goal is given and the facts required to reach that goal are determined. The system described here utilizes only the forward-chaining capability.

The rules in CLIPS are expressed as a combination of facts on the left-hand-side (LHS) and actions on the right-hand-side (RHS) of an IF-THEN rule:

IF (facts exist) THEN (execute actions).

The facts on the LHS are compared to the currently existing facts in the system, as stored in a *fact-list*; if there is a match, the particular rule is placed in an *agenda* to wait for execution. When a rule is executed, it is said to have "fired." The RHS may cause further rules to be placed in or removed from the agenda, thus forming a multilayer decision tree. Also, unlike conventional procedural language programs, the physical order of the rules in a CLIPS program is irrelevant. Thus, CLIPS programs are easily expanded by adding new rules and/or facts, or by modifying selected existing rules or facts. The inference engine mechanism, based on the Rete algorithm (Culbert, 1987), simply matches the LHS of all rules with all facts in the fact-list. The relative firing priority of multiple rules in the agenda may be controlled by the user, however, with a *salience* statement.

## SADIE

SADIE is a FORTRAN library of application subroutines and image file management utilities that contains most of the commonly required image processing capabilities. SADIE Version 3.1, used in this research, was derived in 1985 from earlier versions developed over about ten years. The capabilities of SADIE are listed in Table 1. Traditionally, SADIE users wrote a FORTRAN program that called the various subroutines in the required order to achieve the desired processing (Figure 2). Thus, all of the intelligence about image processing, as well as that about using SADIE and writing FORTRAN programs, had to be supplied by the user.

## THE CLIPS-SADIE INTERFACE

Passing of parameters from CLIPS to SADIE was implemented by writing a C routine that collects the parameters, either defaults or those specified by the user, required by each routine

TABLE 1. SADIE APPLICATION ROUTINES AND THEIR FUNCTIONS (VERSION 3.1)

| | |
|---|---|
| ADD | weighted linear combination of images |
| BIL2BSQ | converts band-interleaved by line format to band-sequential format |
| BSQ2BIL | converts band-sequential format to band-interleaved by line format |
| COVAR | calculates the covariance matrix of a multi-band image |
| DESTRIPE | removes line-to-line striping by bias adjustment |
| DUPL | enlarges an image by pixel and line replication |
| EDGE | edge gradient of an image with one of six algorithms |
| EIGEN | calculates the eigenvectors and eigenvalues of a covariance matrix |
| EXPAND | enlarges an image by pixel and line resampling |
| FCONVL | filters an image by Fourier transform methods |
| FFT1D | 1-D Fast Fourier Transform |
| FFT2D | 2-D Fast Fourier Transform |
| FILTER | filters an image in the Fourier frequency domain |
| INIT | initializes a SADIE program |
| FINI | terminates a SADIE program |
| FUNCTION | creates several types of images for testing |
| GEOMWARP | geometric warp of an image using resampling |
| GEOMCOEF | calculates a polynomial warp function fitted to pairs of image control points |
| GRADIENT | edge gradient of an image with user-specified filter |
| SCATTER | 2-D scattergram between two images |
| HISTEQ | histogram equalization contrast stretch |
| HISTOGRM | calculates grey level histogram of an image |
| HMOSAIC | horizontal mosaic of two images |
| HSV2RGB | transforms hue-saturation-value image to red-green-blue image |
| IMG2VECT | converts image from raster to vector format |
| LVLSLICE | level-slice classification of two images |
| MATCH | transforms histogram of one image to match that of another |
| MEDIAN | median filters an image |
| MOSAIC | 2-D mosaic of several images |
| NOISE | adds random noise of two types to an image |
| PCT | calculates principal components of a set of images |
| PROQUO | calculates product or quotient of two images |
| PSPECT | calculates Fourier power spectrum of an image |
| RESAMPL | smooths and resamples an image |
| RGB2HSV | transforms red-green-blue color image to hue-saturation-value representation |
| RMSDIFF | calculates root-mean-squared difference between to images |
| SCONVL | filters an image with spatial convolution |
| STATS | calculates grey level statistics of an image |
| STRETCH | contrast stretch with a variety of transformations |
| SUBSAMPL | windows and subsamples an image |
| VECT2IMG | converts image from vector to raster format |
| VMOSAIC | vertical mosaic of two images |
| VSTRETCH | spatially-adaptive contrast stretch |
| WINDOW | transforms an image with user-supplied moving window function |

```
program MAIN

call INIT                              ! initialize SADIE program
in = 1                                 ! define input image file
iout = 2                               ! define output image file
inc = 1                                ! pixel and line increment
nlev = 256                             ! number of histogram bins
call HISTEQ (in, iout, inc, nlev)      ! perform histogram equalization
call FINI                              ! terminate SADIE program

end
```

FIG. 2. Example SADIE program written in VAX FORTRAN.

and passes them to SADIE. Because SADIE Version 3.1 is in FORTRAN, it was necessary to convert string variables from C arrays to FORTRAN string descriptors.

Each subroutine of SADIE that is to be called by CLIPS requires specification as one of the CLIPS user functions ("usrfuncs" in CLIPS terminology). For example, use of the SADIE subroutine MEDIAN, which implements a median filter, would require a CLIPS definition as follows:

```
usrfuncs()
{ define_function("median",'i',c_median, "c_median");
}
```

The first argument of a define_function is the CLIPS name for the routine and is used in the CLIPS rules, the second argument is the type of variable returned by MEDIAN (always defaulted to 'integer' for FORTRAN subroutines, which do not return variables), the third argument is a pointer to the FORTRAN subroutine MEDIAN, and the fourth argument is provided for potential use by later versions of CLIPS (this argument is unused in CLIPS Version 4.10).

## EXAMPLE APPLICATIONS

We have implemented two common image processing capabilities in the CLIPS-SADIE expert system: contrast enhancement and noise suppression. Both are limited domains, for which the knowledge base can be relatively easily defined. Implementation of these two applications has served as a learning and testing mechanism for us in developing the expert system. Other applications, such as edge enhancement, or control point selection for geometric rectification, can be added to the expert in a similar manner. The major effort for any application is in the specification of explicit knowledge about the problem and appropriate techniques to address it. Implementation in CLIPS is then straightforward.

### CONTRAST ENHANCEMENT

Contrast enhancement is perhaps the most ubiquitous operation in image processing. It is invariably one of the first capabilities to be included in any software package for image processing. Contrast enhancement is also somewhat unique in that spatially invariant stretching is entirely amenable to interactive execution because it is simply a table-lookup transformation of each pixel's grey level. Thus, contrast enhancement is often done with hardware lookup tables, with the transformation parameters supplied from the instantaneous position of a user-controlled cursor (Schowengerdt, 1983).

Although contrast enhancement can be implemented interactively, the choice of linear stretch breakpoints or of transformations other than the simple linear stretch is still entirely made by the user. This is where an expert system can provide advice on appropriate alternatives and assistance in using the suggested technique. The portion of our expert system that does contrast enhancement is shown in a decision tree structure in Figure 3. There are two modes: the manual mode, where the

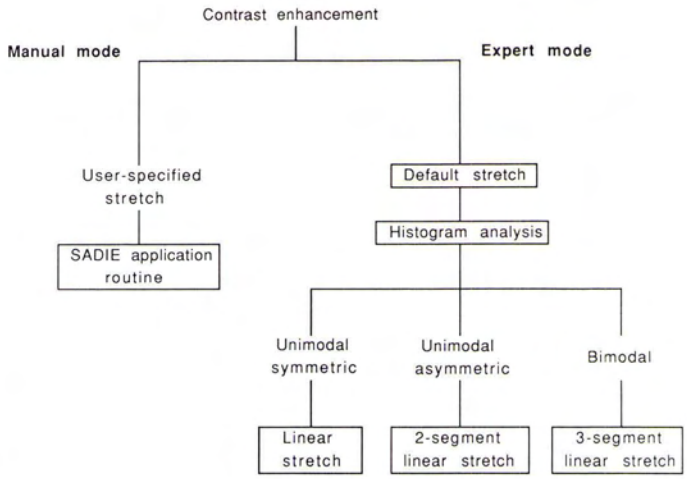


FIG. 3. Logic tree for contrast enhancement.

user is permitted to do whatever transformation is available, and the expert mode, where the system first provides a default stretch (histogram equalization) and then analyzes the original image histogram shape and suggests alternative transformations. One of these may then be used if the default stretch is unsatisfactory to the user.

The default stretch, histogram equalization, works reasonably well for a wide variety of images and requires no user input. Basically, the cumulative distribution function (CDF) is calculated from the histogram by integrating it from one direction, and after appropriate scaling, is used as the grey level transformation function (Figure 4a). Unfortunately, histogram equalization generally produces contrast which is too harsh, with a large number of pixels redistributed to the low and high ends of the grey level scale. Histogram equalization is also not particularly well-suited to images that have skewed (asymmetric) histograms with large populations of pixels in a limited grey level range.

Other common contrast enhancement techniques are simple linear transformations with saturation and piecewise linear transformations that stretch different parts of the grey level scale differently. Thus, histogram asymmetry or multimodality can be accomodated (Figure 4). To implement this type of contrast stretch, the expert system first evaluates the symmetry and shape of the histogram using the following statistical moments (Burford, 1968):

$$\begin{aligned} \text{mean (average) } \mu &= \sum_i GL_i/N \\ \text{variance (width) } \mu_2 = \sigma^2 &= \sum_i (GL_i - \mu)^2/N \\ \text{skewness (asymmetry) } \mu_3 &= \sum_i (GL_i - \mu)^3/N \\ \text{kurtosis (peakness) } \mu_4 &= \sum_i (GL_i - \mu)^4/N \end{aligned} \tag{1}$$

Skewness and kurtosis may be normalized to be independent of data units as follows:

$$\begin{aligned} \text{normalized skewness } \alpha_3 &= \mu_3/\sigma^3 \\ \text{normalized kurtosis } \alpha_4 &= \mu_4/\sigma^4 \end{aligned} \tag{2}$$

We may then characterize an image histogram with the last two parameters as follows (Burford, 1968):

normalized skewness = 0: symmetric
normalized skewness > 0: skewed to the right
normalized skewness < 0: skewed to the left
normalized kurtosis = 3: gaussian distribution

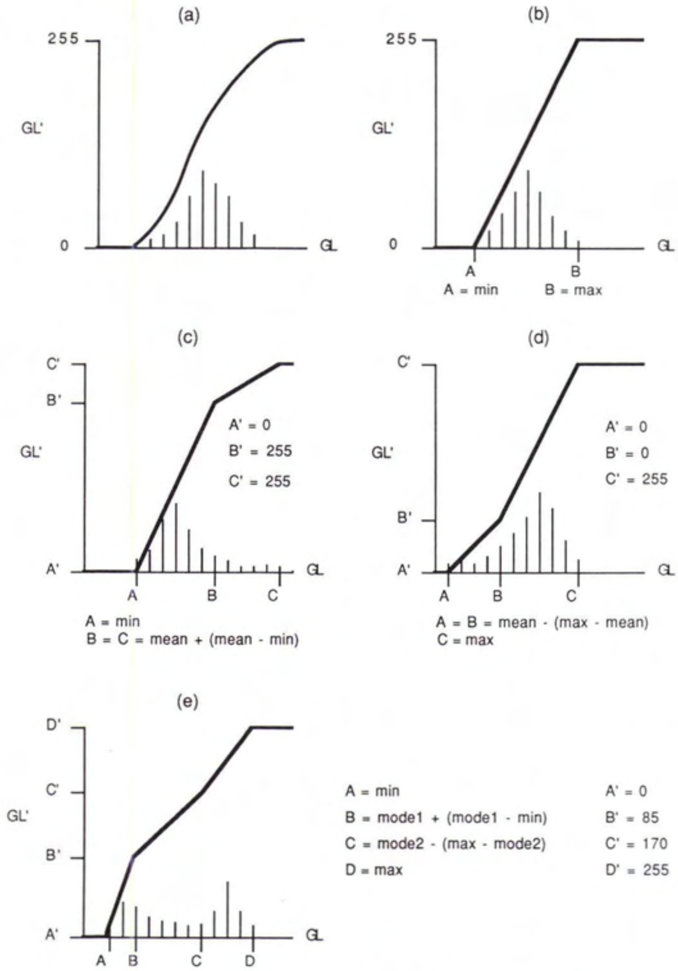


FIG. 4. Default and suggested contrast stretches for different image histograms. The values specified for the parameters A, B, C, D, A', B', C', and D' are those suggested by the expert system and may be changed by the user. (a) Default histogram equalization. (b) Symmetric histogram - linear stretch. (c) Right-skewed histogram - two segment linear stretch. (d) Left-skewed histogram - two segment linear stretch. (e) Bimodal histogram - three segment linear stretch.

With real data histograms there must be an allowance for statistical variation. Some of the above conditions are therefore implemented in the expert system with predefined thresholds; e.g., a normalized kurtosis between 2.9 and 3.1 indicates a Guassian distribution. In addition, we need a way to detect multimodes (multipeaks) in the histogram that arise in certain types of images. For example, images of bodies of water and land often exhibit a bimodal histogram. We check for multimodes with a simple algorithm that moves along the grey level scale and looks for changes in histogram slope from positive to negative that indicate a significant peak. Using the above histogram measures, the expert system then suggests an appropriate stretch with breakpoints as shown in Figure 4.

An example CLIPS rule is shown in Figure 5. All statements prior to the "=⟩" symbol represent the LHS and all statements after the "=⟩" symbol represent the RHS. CLIPS first compares the LHS with the fact-list. If all facts match, the RHS is executed. At the end of the RHS, all facts, and the rules dependent on them, are removed from the fact-list and agenda.

An example of contrast enhancement using the expert system

```
(defrule skewed_right "R9"         ; define rule R9 called skewed_right
 (declare (salience 10))           ; set agenda priority
 ?f1 <- (no_of_peak 1)             ; bind number of peaks = 1 to fact address f1
 ?f2 <- (alph3 ?a3)                ; bind normalized skewness to fact address f2
 ?f3 <- (glmin ?min glmax ?max glmean ?mean)    ; bind image minimum,
;                                    maximum, and mean GLs to fact address f3
 ?f4 <- (mode ?mode $?)            ; bind histogram mode GL to fact address f4
 (test (> ?a3 0))                  ; is normalized skewness greater than zero?
=>
 (bind ?brk (+ ?mean (- ?mean ?min)))      ; create new variable ?brk
 (printout " Since the histogram of the input image is skewed right,    "crlf)
 (printout " it is suggested that you use the piecewise linear option.  "crlf)
 (printout " The suggested transformation is:                           "crlf)
 (printout "                                                       "crlf crlf)
 (printout "      C'  +                        *                         "crlf)
 (printout "          |                                                  "crlf)
 (printout "          |                                                  "crlf)
 (printout "      B'  +              *                                   "crlf)
 (printout "          |                                                  "crlf)
 (printout " GL'      |                                                  "crlf)
 (printout "          |                                                  "crlf)
 (printout "          |                                                  "crlf)
 (printout "          |                                                  "crlf)
 (printout "          |                                                  "crlf)
 (printout "      A'  +------*------+-------------------+--------> GL    "crlf)
 (printout "                 A      B                   C                "crlf)
 (printout "                                                       "crlf crlf)
 (printout " with the values:                                            "crlf)
 (printout "                                                       "crlf crlf)
 (printout "                 A = " ?min                                    crlf)
 (printout "                 B = " ?brk                                    crlf)
 (printout "                 C = " ?brk                                    crlf)
 (printout "                 A' =    0.0                                 "crlf)
 (printout "                 B' = 255.0                                  "crlf)
 (printout "                 C' = 255.0                                  "crlf)
 (printout "                                                       "crlf crlf)
 (printout " You may use the above suggested values for the contrast     "crlf)
 (printout " stretch, or modify them as you wish.                        "crlf)
 (printout "                                                       "crlf crlf)
 (printout " To decrease high GL saturation:                             "crlf)
 (printout "                     increase B and C or                     "crlf)
 (printout "                     decrease B' and increase C              "crlf)
 (printout " To increase high GL saturation:                             "crlf)
 (printout "                     decrease B and C                        "crlf)
 (retract ?f1 ?f2 ?f3 ?f4))          ; retract facts from fact-list
```

FIG. 5. Example CLIPS rule from the contrast enhancement expert system.

is shown in Figure 6. The original image (Thematic Mapper band 2 image of Dulles Airport in Virginia) is not only dark, but has a skewed histogram caused by the relatively bright runway areas (Figures 6a and 7a). The skewness of the image histogram is 3.85 and the kurtosis is 35.6, indicating strong asymmetry and a non-Gausssian distribution. Histogram equalization (Figures 6b and 7b) certainly enhances the contrast, but so strongly for low grey levels that noise is enhanced and areas surrounding the runways become almost indistinguishable from the runways. With the suggested linear stretch (Figure 7c), the image maintains a more "normal" appearance, even with significant contrast enhancement (Figure 6c). If the user deems the saturation at higher grey levels is excessive, the initial suggested parameters can be adjusted (Figure 7d) until the desired contrast is acheved (Figure 6d).

## NOISE SUPPRESSION

Although noise in digital images is defined as an unexpected, spurious signal, most commonly encountered noise artifacts can be categorized into one of the following types (Schowengerdt, 1983):

- random — grey level noise at every pixel, usually originating in the image detectors
- isolated — extreme noise at relatively few pixels, usually results in zero or maximum grey level at the affected pixels, sometimes called "salt and pepper" noise, usually caused by bit loss in data transmission
- periodic-stationary — consistent periodic pattern across the entire image, sinusoidal or combinations of sine patterns, usually

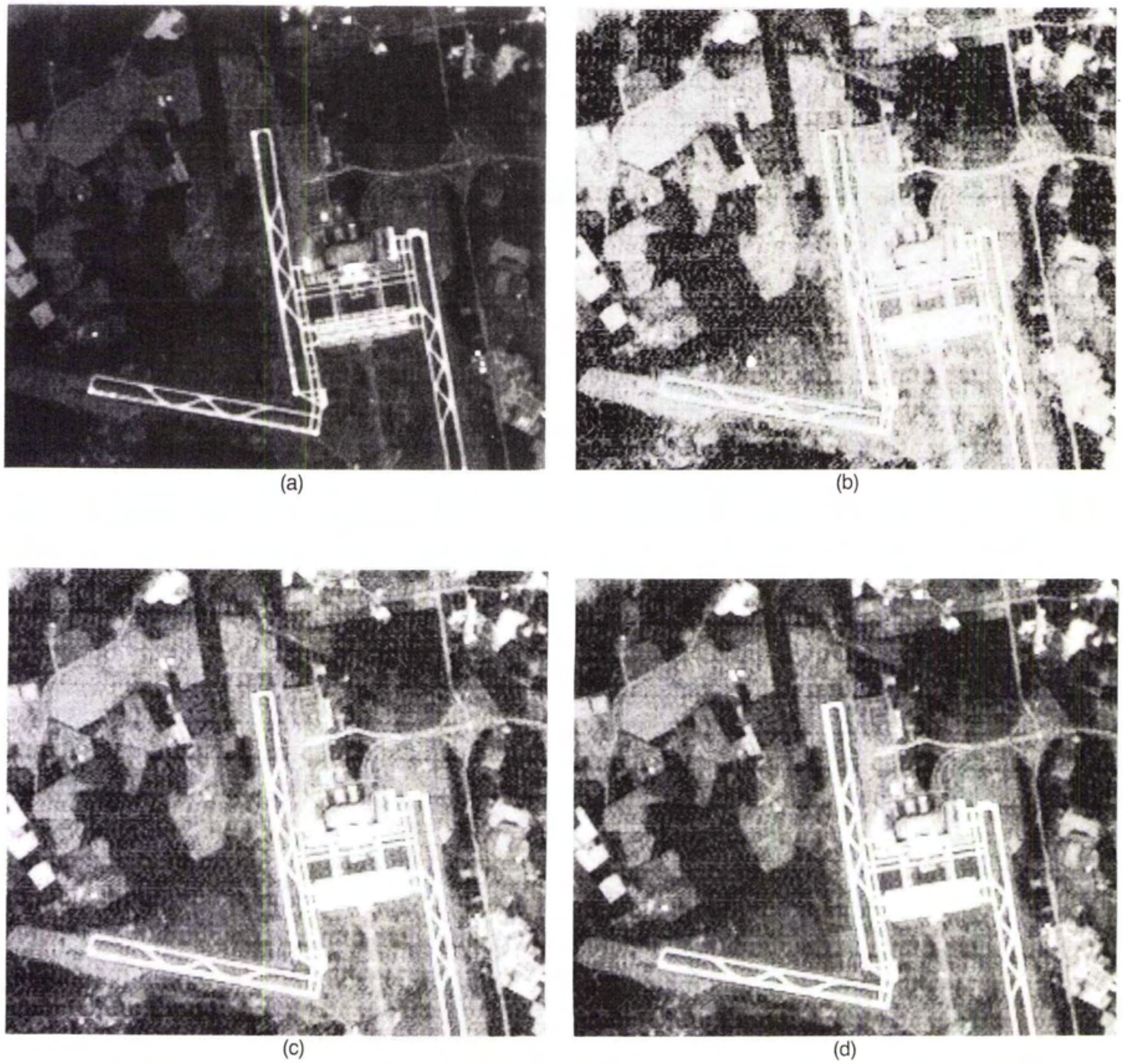


FIG. 6. Examples of contrast enhancement. (a) Original TM band 2 image of Dulles Airport. (b) Result of default histogram equalization. (c) Result of linear stretch suggested by the expert system. (d) Result of modification (gain change) of the suggested stretch by the user to reduce high end saturation.

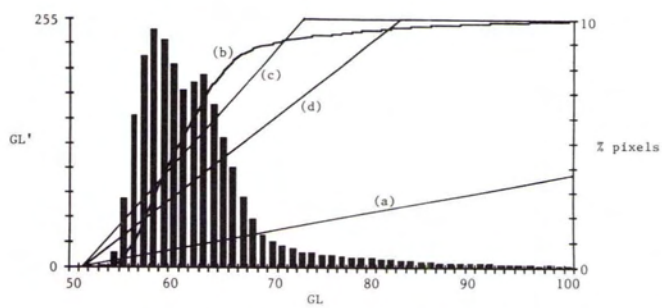


FIG. 7. Image histogram and contrast stretches for Figure 6. (a) Original image histogram and linear minimum (51) and maximum (237) GL stretch. (b) Default histogram equalize stretch. (c) Suggested linear stretch. (d) User-modified linear stretch.

resulting from electronic interference

periodic-nonstationary – periodic pattern that changes characteristics (modulation, frequency, phase) across the image, usually indicates time varying electronic interference

This reasonably well-defined classification of noise types makes a particularly good domain for processing with an expert system. We have implemented the suppression of isolated, periodic-stationary, and a particular type of periodic-nonstationary noise (line-to-line striping) in the CLIPS-SADIE expert system. This portion of the expert operates somewhat differently from that used for contrast enhancement and relies more on user input to guide the process.

The user is first asked to view the image and specify the noise category. The decision tree in Figure 8 is then followed according to the user's categorization. These procedures for noise suppression represent proven approaches to suppressing these

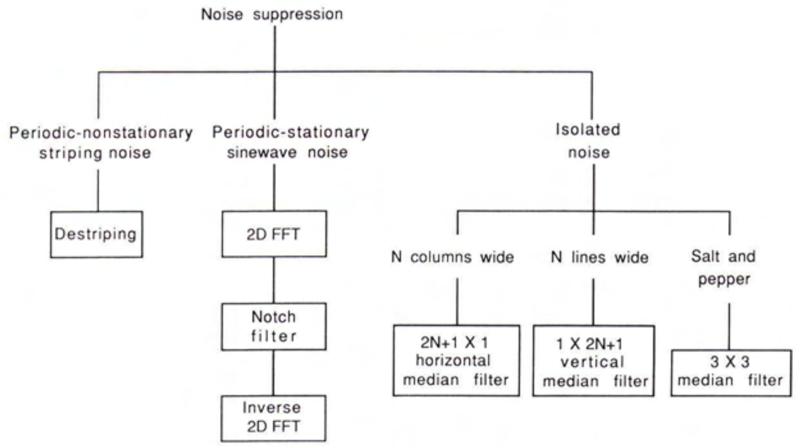


FIG. 8. Logic tree for noise suppression.

common types of noise (Pratt, 1978; Schowengerdt, 1983). Furthermore, they explicitly represent the image processing knowledge embedded in the expert system.

Examples of the different types of noise that can be processed by the system are shown in Figure 9. The destriping algorithm used to remove the periodic-nonstationary noise in the video image of Figure 9a is a simple bias adjustment applied to each line if its mean grey level differs from that of the previous line by an amount greater than a specified threshold. The mean grey level of the current line is then adjusted to be equal to that of the previous line. Periodic-stationary noise is found automatically by the expert system by looking for significant peaks in the image power spectrum (Figure 9d) and removing them with a notch filter. The Fourier transforms and noise filtering are all handled without interaction from the user; the expert does however report the procedures that are being used. The problem of periodic noise removal (Figures 9a through 9d) was addressed for early planetary images (Chavez and Soderblum, 1975) and continues to be of interest (Crippen, 1989; Srinivasan *et al.*, 1988).

For isolated noise, the expert automatically sets the size of the median filter to accomodate the type and width of the noise, as specified by the user (Figure 8). For example, if the user says the noise consists of horizontal bad lines, two lines wide, the expert will set the median filter window to be one pixel by five lines. If the noise consists of single isolated pixels, the median filter will be three pixels by three lines.

Other types of noise, random and general periodic-nonstationary, could be similarly addressed in the expert, although the image processing algorithms required are more complex, particularly for the latter type of noise. The level of noise categorization required of the user also could become prohibitively high. It is likely that some form of semi-automated noise classification, difficult as it may be, would be necessary.

## USER INPUT TO THE EXPERT SYSTEM

In both examples of the expert system described above, we have attempted to use the user's cognitive and visual facilities whenever it was more appropriate or efficient than computer processing. For example, in the case of noise suppression, the user is asked to categorize the noise and then provide easily determined visual characteristics of the noise to the system. If, instead, we had assigned these tasks to the expert system, considerable computer time and resources would have been expended to obtain the same information. Moreover, it is questionable whether the computer could autonomously determine some of this information (for example, deciding whether the noise was statonary or non-stationary). In fact, this is exactly one of the central issues in computer vision that makes it such a difficult problem.

In the case of the contrast enhancement expert, the system relies less on the user's input. It is relatively easy to measure the asymmetry of the grey level histogram and probably more reliable than human judgement in this case. The user, however, is the ultimate judge of the contrast enhancement and therefore is allowed to continue experimenting with the stretch parameters until satisfied. Basically, the expert system provides a starting point that is reasonable for the given image.

## SUMMARY DISCUSSION

We have created an expert system for image processing by interfacing two existing software systems, one an expert system development tool written in C and the other an image processing library written in FORTRAN. The expert system serves as a "friendly" interface to the image processing software and as an image processing advisor that has been coded with specific knowledge for two common image processing applications: contrast enhancement and noise suppression. It makes use of the user's visual diagnostic capabilities where possible to avoid time-consuming and difficult computer vision analysis.

The expert system is easily expanded to include other applications, the major work required being specification of expert knowledge in a form suitable for a rule-based system. For example, an expert subsystem could be added to support control point selection for geometric registration of an image to a map, a typically laborious process with little automation in most current image processing systems. Considerable sophistication could be built into such an expert system to provide quantitative support to the user. To improve the distribution of control points, the system might suggest specific additional control points in both the image and map coordinate systems, based on initial operator selection of three control points and a low order warping polynomial. The user would then precisely locate points, if possible, in the vicinity of the suggested points. The expert system could also automatically eliminate selected control points based on a goodness-of-fit criteria and select the polynomial order for the final warping funciton based partly on knowledge of the type of sensor that produced the image.

Our goal in the long term is to relieve the user as much as possible from having to acquire detailed knowledge about image processing. After all, the user should not (necessarily) have to know these details to accomplish what he wants to do. In many research environments, this is accomplished with a sup-

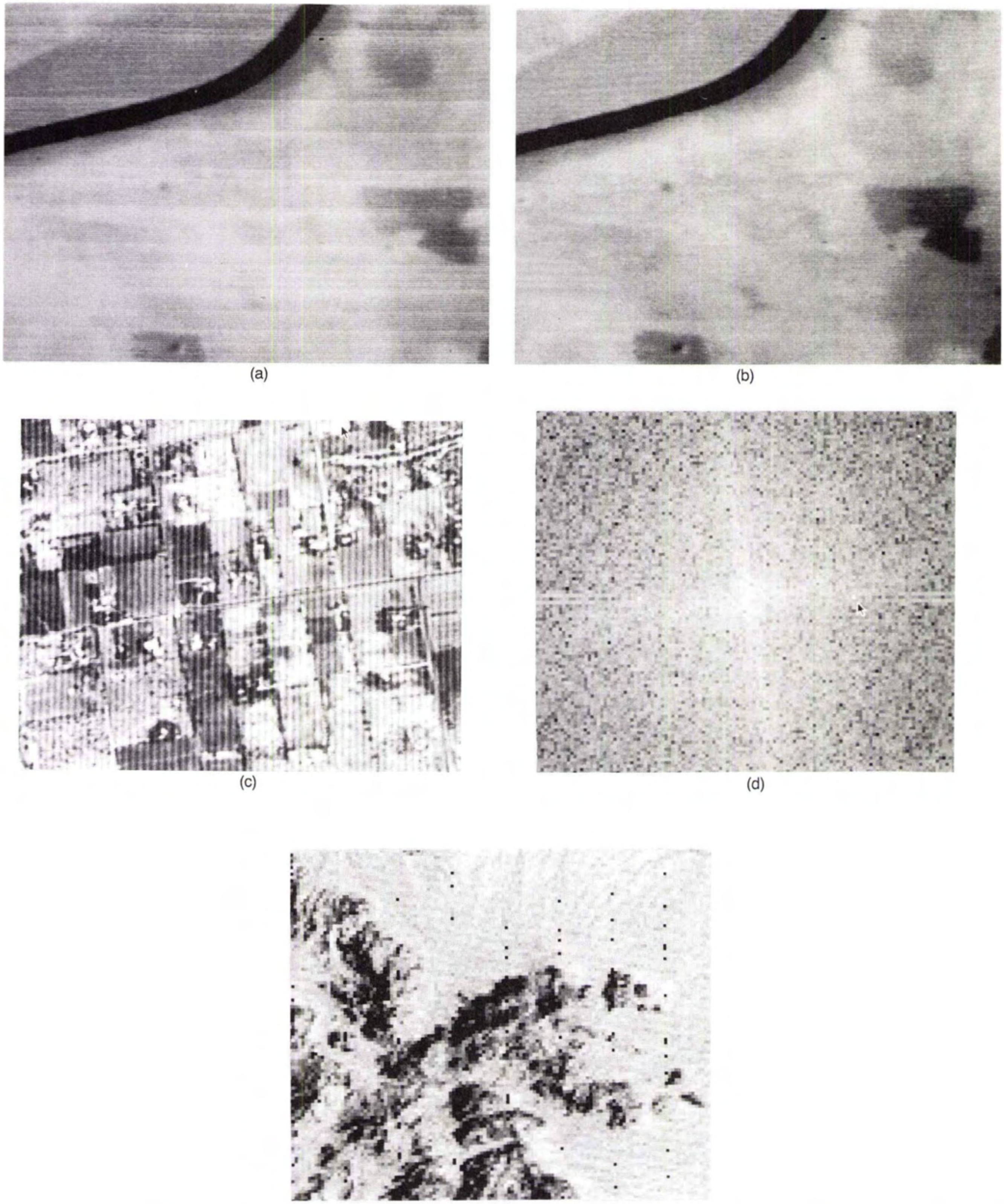


FIG. 9. Examples of noise types that can be processed by the system. (a) Aerial video image with periodic-nonstationary noise in the form of time varying line-to-line striping. (b) Destriped video image. (c) Digitized aerial photograph with synthetic periodic-stationary noise added (period equal to four pixels). (d) Power spectrum of a portion of (c) enlarged four times to show the spike (small arrow) at the frequency (1/4 cycles/pixel) of the noise. The center spike is at zero frequency and the left-hand spike is the negative frequency counterpart of the right-hand noise spike. (e) Landsat MSS image with isolated noise.

port staff of image processing experts who do the actual computing under direction of the remote sensing application scientists. With the widening distribution of powerful image processing capabilities in desktop computing systems, there is a need to have surrogate experts in the form of software systems that embody significant knowledge about image processing.

It is the expert system developer's obligation to provide reliable and robust systems that are truly "expert." We do not claim that the image processing techniques described in this paper are the best for each problem. They do, however, demonstrate the nature of such an expert system. A significant problem for all expert systems is the acquisition of appropriate knowledge, e.g., what image processing techniques to use or what parameters to suggest to the user. The robustness of image feature extraction algorithms, such as the histogram peak counter in our contrast enhancement expert system, is also important. A great deal of research remains to be done in these areas before expert systems can be widely accepted and trusted in applications such as image processing.

## ACKNOWLEDGMENTS

The authors would like to thank the reviewers and Gerhard Mehldau of the University of Arizona Computer Science Department for valuable comments on this paper.

## REFERENCES

Ballard, Dana H., and Cristopher M. Brown, 1982. *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jersey.

Bobrow, D. G., S. Mittal, and M. J. Stefik, 1986. Expert systems: perils and promise, *Communications of the ACM*, Vol. 29, No. 9, pp. 880–894.

Burford, R. L., 1968. *Statistics: A Computer Approach*, Charles E. Merrill Publishing, Columbus, Ohio.

Chavez, Pat S., Jr., and Larry A. Soderblum, 1975. Simple high-speed digital image processing to remove quasi-coherent noise patterns, *Proc. Am. Soc. Photgramm, Annual Convention*, pp. 595–600.

Culbert, C., 1987. *CLIPS Reference Manual, Version 4.10*, Artificial Intelligence Section, NASA Johnson Space Center.

Crippen, Robert E., 1989. A Simple Spatial Filtering Routine for the Cosmetic Removal of Scan-Line Noise from Landsat TM P-Tape Imagery, *Photogrammetric Engineering and Remote Sensing*, Vol. 55, No. 3, pp. 327–331.

Doescher, S. W., D. K. Scholz, and B. K. Quirk, 1988. An Expert-System Front End for the Land Analysis System, Chapter A, *Selected Papers in the Applied Computer Sciences* (D. A. Wiltshire, ed), U. S. Geological Survey Bulletin 1841.

Duda, R. O., and P. E. Hart, 1973. *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York.

Duda, R. O., and E. H. Shortliffe, 1983. Expert systems research, *Science*, Vol. 220, No. 4594, pp. 261–268.

Giarratano, J., 1987. *CLIPS User's Guide, Version 4.10*, Artificial Intelligence Section, NASA Johnson Space Center.

Goodenough, D. G., M. Goldberg, G. Plunkett, and J. Zelek, 1987. An expert system for remote sensing, *IEEE Trans. Geosci. and Rem. Sens.*, V. GE-25, No. 3, pp. 349–359.

Haralick, Robert M., 1986. Computer vision theory: the lack thereof, *Computer Vision, Graphics and Image Processing*, Vol. 36, pp. 372–386.

Levine, Martin D., 1985. *Vision in Man and Machine*, McGraw-Hill, New York.

Marr, David, 1982. *Vision*, W. H. Freeman, New York.

Martin, G. G., J. Lienhart, T. Hyman, and G. C. Levy, 1988. Statistical Expert System for Data Analysis, *American Laboratory*, February.

Matsuyama, Takashi, 1987. Knowledge-based aerial image understanding systems and expert system for image processing, *IEEE Trans. Geosci. and Rem. Sens.*, Vol. GE-25, No. 3, pp. 305–316.

McKeown, D. M., Jr., W. A. Harvey, Jr., and J. McDermott, 1985. Rule-based interpretation of aerial imagery, *IEEE Trans. Pattern Anal. and Mach. Intell.*, Vol. PAMI-7, No. 5, pp. 570–585.

Nagao, Makoto, and Takashi Matsuyama, 1980. *A Structural Analysis of Complex Aerial Photographs*, Plenum Press, New York.

Nazif, Ahmed M., and Martin D. Levine, 1984. Low level segmentation: an expert system, *IEEE Trans. on Patt. Anal. and Mach. Intell.*, Vol. PAMI-6, No. 5, pp. 555–577.

Nichol, David Gawith, 1987. Autonomous extraction of an eddy-like structure from infrared images of the ocean, *IEEE Trans. Geosc, and Rem. Sens.*, Vol. GE-25, No. 1, pp. 28–34.

Perkins, W. A., T. J. Laffey, and T. A. Nguyen, 1986. Rule-based interpreting of aerial photographs using the Lockheed Expert System, *Optical Engineering*, Vol. 25, No. 3, pp. 356–362.

Pratt, W. K., 1978. *Digital Image Processing*, John Wiley and Sons.

Rosenfeld, Azriel, and various authors, 1986. DIALOG- "expert" vision systems: some issues, *Comp. Vision, Graphics and Image Processing*, Vol. 34, pp. 99–117.

SADIE Image Processing Software, V3.1 Users Manual, Digital Image Analysis Lab, University of Arizona, 1986.

Schowengerdt, Robert A., 1983. *Techniques for Image Processing and Classification in Remote Sensing*, Academic Press.

Schowengerdt, R. A., and G. Mehldau, 1987. Classification of Multispectral Imagery with a Rule-Based System, *Proc. 40th Annual Conf. Soc. Photographic Scientists and Engineers, Symposium on Hybrid Imaging Systems*, pp. 186–189.

Schowengerdt, R. A., and R. A. Pries, 1988. Interactive image feature compilation for geographic information systems, *Recent Advances in Sensors, Radiometry, and Data Processing for Remote Sensing*, Proc. SPIE, V. 924, Orlando, Florida.

Srinivasan, R., M. Cannon, and J. White, 1988. Landsat data destriping using power spectral filtering, *Optical Engineering*, Vol. 27, No. 11, pp. 939–943.