

MASTER

UCRL- 85225
PREPRINT

CONF-810424--2

"A Digital Image Processing Software
System Using An Array Processor"

R. J. Sherwood
M. R. Portnoff
C. H. Journey
R. E. Twogood

This Paper Was Prepared For
Presentation At the
1981 ARRAY Meeting
St. Louis, MO, April 26-29, 1981

March 10, 1981

The logo for Lawrence Livermore Laboratory is a large, stylized 'V' shape. The top horizontal bar of the 'V' is white, while the two slanted sides are filled with a dark, stippled pattern. On the right-hand slanted side, the words 'Lawrence Livermore Laboratory' are printed in a white, sans-serif font, oriented vertically from top to bottom. A small black arrowhead points downwards at the top of the right slanted side.

Lawrence
Livermore
Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED -

A DIGITAL IMAGE PROCESSING SOFTWARE SYSTEM USING AN ARRAY PROCESSOR

R. J. Sherwood, M. R. Portnoff, C. H. Journeay, and R. E. Twogood
Lawrence Livermore National Laboratory

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, currency, or usefulness of any information, apparatus, product, or process disclosed, or represents its usefulness or endorses any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ABSTRACT

We have developed a versatile array processor-based system for general purpose image processing. At the heart of this system is an extensive, flexible software package which incorporates the array processor for effective interactive image processing. The software system is described in detail, and we briefly discuss its application to a diverse set of applications at LLNL.

INTRODUCTION

The digital processing of image data has found increasing importance in many aerospace, scientific, industrial, and biomedical applications. Typical applications include compensation for sensor and transmission errors from satellites and deep space probes; and resolution improvement, contrast enhancement, and motion blur removal from scientific, industrial and biomedical imagery such as radiographic, thermographic, ultrasonic and radionuclide images. Further processing is often performed for the enhancement and detection of specific features such as fractures or tumors within a given image, or changes in such features among an ensemble of images.

In this paper we describe a software system for interactive image processing using a minicomputer and the FPS AP-120B array processor (AP). Following a brief description of the hardware environment, the structure and capabilities of this software system for image processing will be discussed. Particular attention will be devoted to describing the software design philosophy, the classes of image processing algorithms implemented, and the performance of the array processor.

LLNL IMAGE PROCESSING RESEARCH FACILITY

A block diagram of the Image Processing Research Facility (IPRF) at LLNL is

shown in Figure 1. The host processor, a DEC PDP-11/70, is connected to a large assortment of peripherals and image display systems. In addition to 9-track tapes and large disks (1 300-Mbyte and 3 80-Mbyte) used for mass storage, the system has two image display systems - a Comtal 8000 and a Stanford Technology Corp. (STC) Model 70/E. A video digitizing system is interfaced to the STC display system for real-time image digitization, and a high-speed (4 megabit/sec.) link to the extensive LLNL central computing facility exists. Note also that the FPS AP-120B, which handles virtually all of the computational burden imposed by the image processing applications programs, is connected to both the PDP-11 and to the disks. This device-to-device transfer capability between the AP and disk avoids the traditional I/O overhead associated with system configurations in which data must pass through the host compute. to reach the AP.

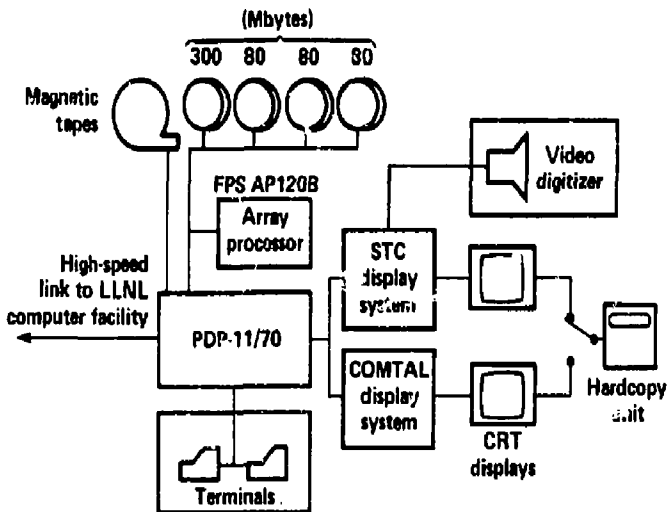


Figure 1. LLNL Image Processing Research Facility

IMAGE PROCESSING SOFTWARE

System Design Philosophy

The image processing software package, called IMG, is designed to perform a wide variety of image processing functions, employing a common user interface which is easy to learn and use. One of the primary design goals was to develop a system which accommodates both the occasional, inexperienced user and the regular user who is familiar with the system. An additional goal was to develop a multi-user system in which several users, each operating from his own computer terminal, could access either of the image display systems and the array processor with a minimal amount of interference from other users.

An additional major design objective was the design of a modular system which is flexible and easily extendable. The software was designed in a modular fashion so that a useful set of operations could be implemented early in the development cycle, thus allowing users to be processing images while system developers are adding new processing algorithms and improved control functions to the system. This approach has been well received by the users, and permits straightforward system maintenance and enhancement.

Image Processing Capabilities

The IMG software is organized to perform disk-based processing, and is largely dependent on the array processor. An image is generally represented on disk as an array of 32-bit floating-point numbers, and as a processing function takes place, the results are written to disk. Normally a new disk file is generated containing the output image but many operations optionally allow the user to perform the function in-place; i.e., the output data replace the data in the input file.

The image processing commands can be divided into four major categories: image system display and control, pointwise operators, neighborhood operators, and global operators. Image system display and control operations include such operations as loading an image system refresh memory from a disk file, displaying a refresh memory on the video monitors and digitizing an image on the video digitizer. In addition, the display systems have pseudocolor capability, which assigns a distinct color to each intensity value. Several pseudocolor patterns are available, and their assignment to the grey levels

can be modified under control of a trackball to produce optimum results. Furthermore, nonlinear intensity mapping can be used to interactively enhance areas of subtle intensity difference. Graphics overlays are available for displaying grids, curves, and alphanumeric information, and a trackball-controlled cursor is used to determine coordinates and pixel intensities. A zoom capability is available for enlarging the size of a specified portion of an image. Several other display system control functions are available, and more are being added as the system continues to develop.

The pointwise operators process image files on disk, and nearly all are performed using the array processor. Examples include unary operators such as absolute value, logarithm, exponential and square, and binary operators such as pointwise addition, subtraction, multiplication and division.

Neighborhood operators specify the value of a pixel in the output image as a function of the values of the pixels in the immediate neighborhood of the corresponding pixel in the input image. The neighborhood used is typically a rectangular window centered at the specified pixel in the input image. Examples include such operators as spatial derivatives, median window filters, convolution, and two types of contrast stretching algorithms.

The fourth category of operators, the global operators, are those for which the value at a given point in the output image is dependent on all points in the input image. Examples of these operators include such items as the 2-D Fast Fourier Transform (FFT) and its inverse, convolution using the FFT, alpha-rooting, and histogram equalization.

Software Structure

The software was designed and implemented in a modular manner. The amount of software is large and continuing to grow, yet runs on a 16-bit minicomputer in which a given program ("task") can directly access only 32K words of memory. The system must support two or more people doing image processing, as well as others involved in software development; consequently a prime concern is minimizing the amount of minicomputer memory required. Another major concern is providing a system in which orderly growth can take place. Image

processing users must be able to use the software, even while software development is taking place concurrently. Both of these requirements dictate the need for modular software.

The software is structured as a tree, with the root of the tree being a supervisor program which controls and monitors a number of processing tasks (Figure 2).

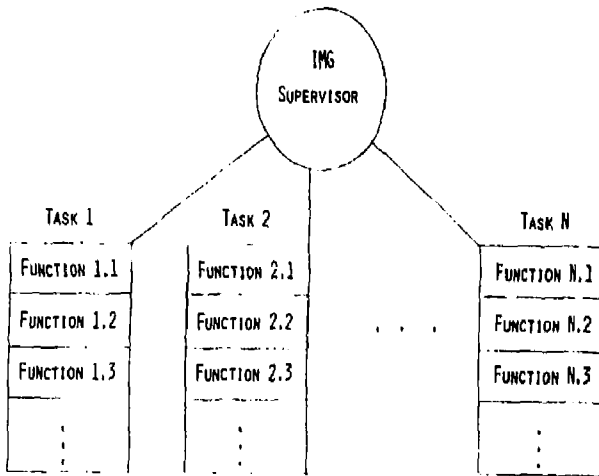


Figure 2. IMG Software Structure.

A typical processing task can handle a number of different commands. The supervisor maintains a symbol table which it reads from disk at startup time, and this table associates one of the processing tasks with each of the processing commands. When a command is entered, the supervisor searches its symbol table to determine which processing task should service that function, then starts the appropriate task and informs it which command was entered. The processing task runs that function to completion, then notifies the supervisor and exits. Thus, the processing tasks, which typically are large, occupy memory only during the time that they are needed. Between commands, while the user is examining displayed results or thinking, only the supervisor

(which is relatively small) is in computer memory. Thus, the system remains more available for other image processing and software development activities than it would if all the software were resident. New commands can be inserted into the system by simply writing the processing software, inserting it in a processing task, and inserting the command in the symbol table. No modification of the supervisor is required.

Supervisor Services

The IMG supervisor program provides a number of useful services to the user. One of these is the handling of command parameters. A command typically requires one or more input parameters. These parameters may either be requested interactively through prompts, or entered on the command line. The occasional user, who may not remember the exact parameter sequence, can simply type the function name followed by a carriage return. The system then prompts him for the required parameters. Alternately, the experienced user who is familiar with the parameter sequence can type the command and all the parameters on one line, and the system immediately processes the command without any prompting. Another service of the supervisor is the handling of default device and directory for the storage of images, and the selection of the image display system.

The supervisor can also handle several commands simultaneously. If the user places the supervisor in parallel mode, he may type several commands and they will be processed simultaneously where possible. This mode is useful for performing interactive activities on the display system, while an involved calculation is taking place. If the user does not desire this parallelism he can use the supervisor in serial mode, in which case commands that are "typed ahead" will merely be queued until the previous command runs to completion.

The system also supports the processing of commands issued from a disk file. This capability is useful for a long series of compute-intensive commands, and can be handled simultaneously with interactive commands. Additionally, a command chaining facility is provided allowing any task to compose and dispatch a command as though it had been typed at the user terminal.

Performance

The addition of the array processor to the system has greatly increased the processing speed of the system. Table 1 shows the computation (CPU + I/O) time for two processing algorithms applied to a 512x512 pixel image. The first algorithm is a contrast stretching algorithm in which a 2x2 pixel window is moved over the entire image and a contrast stretch performed. The second algorithm is a 2-D FFT. Notice that the 9.3 sec time required for the contrast stretch algorithm using the array processor and minicomputer is less than 10% of the time required to do the same operation using only the minicomputer. Notice also that the processing time for the contrast stretch algorithm using the minicomputer and array processor is less than twice the time for a stand alone CDC 7600, at a small fraction of the cost.

SYSTEM	ALGORITHMS	
	CONTRAST STRETCH: 512 x 512 2 x 2 WINDOW	FAST FOURIER TRANSFORM 512 x 512
PDP-11/70	3 MIN	122 SEC
PDP-11/70 WITH ARRAY PROCESSOR	9.3 SEC	18 SEC* * ~9 MIN FOR 2048 x 2048
CDC-7600	5.1 SEC	3.3 SEC

Table 1. Sample Image Processing Algorithm Performance Timings.

APPLICATIONS

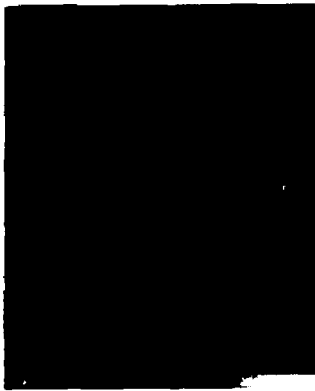
Image processing applications at LLNL span nearly the entire range of image processing problems. These applications may be grouped into the categories of image display, image enhancement, image restoration, and feature extraction.

In the area of image display, pseudocolor display of grey-scale images has proven to be a useful and effective technique. Since the human observer is often the best feature detector, we have developed a family of pseudocolor mappings designed to enhance the visibility of various classes of features. Our pseudocolor display is routinely used both for displaying two-dimensional data as color-coded contour maps and for displaying real images for the purpose of detecting or measuring certain features. Applications of the former include display of global ozone concentrations, characterization of underground rock structures and aquifers, and display of geomorphographs. Applications of the latter include interactive analysis of ultrasound images of the human chest, analysis of aerial imagery attempting to detect underground nuclear tests, and analysis of radiographs for nondestructive evaluation of precision parts.

In the area of image enhancement, we are concerned with developing and applying numerical algorithms to enhance or suppress certain features of an image. One class of algorithms is concerned with improving the contrast of low contrast images. Such images arise frequently as the result of the limited dynamic range and nonlinearities present in many photographic and electronic image recording systems. Our contrast manipulation algorithms include linear and nonlinear remapping of grey levels, histogram modification, and alpha rooting. Other frequently used enhancement algorithms employ linear and nonlinear filters in order to enhance features such as edges or to suppress features such as noise. Specific applications of image enhancement techniques have included contrast and edge enhancement of radiographs, suppression of high spatial frequencies in aerial imagery to reduce sensor noise, and contrast enhancement and noise suppression in photographs of hypervelocity rail gun projectiles.

An example of contrast enhancement of a radiographic exposure of a rail gun diagnostic image is shown in Figure 3. Figure 3a shows the original radiograph, and Figure 3b is the enhanced image resulting from a nonlinear gray scale histogram mapping technique.

Image restoration problems are essentially model-based estimation problems in which we operate on an observed image that was obtained by means of a



(a)



(b)

Figure 3. Enhancement of Rail Gun Diagnostic Radiograph. a) Original radiograph. b) Enhanced result using nonlinear histogram mapping.

non-ideal image formation or recording system. Based on a model of the degradation, we seek to estimate the image that would have been observed were there no degradation present. Typical techniques used for image restoration include simple space and frequency domain filtering, as well as Wiener filtering and its variants. Image restoration problems that we have been concerned with at LLNL have been reconstruction of Fresnel zone plate coded images, removal of degradation in infrared aerial images caused by malfunctioning sensor elements, and compensation for motion blur in radiographs obtained during actual weapons testing.

An example of an image restoration application is shown in Figure 4. Figure 4a shows an infrared aerial image that has been severely degraded by horizontal striations caused by sensor instrumentation problems. Figure 4b shows the corresponding transposed spectrum calculated using the 2-D FFT. The sensor noise appears in the spatial-frequency domain as energy along the horizontal axis of Figure 4b. Scaling the portion of the spectrum dominated by this noise such that local statistics in that area are made comparable to local statistics away from the noisy region results in the spectrum of Figure 4c. Performing the inverse 2-D FFT on this processed spectrum gives the



(a)



(b)



(c)



(d)



(e)

Figure 4. Aerial Infrared Imagery. a) Original image. b) FFT.
c) Filtered FFT. d) Filtered image. e) Removed noise.

output shown in Figure 4d in which the sensor noise has been successfully suppressed. Figure 4e shows the removed noise, which is simply the difference between the input and output images.

Feature extraction problems at LLNL have involved both interactive and automatic algorithms. An example of interactive feature extraction is the definition of the chest wall boundaries on an ultrasound image of the chest. Using a pseudocolor enhanced image formed from an ultrasound scan, an operator specifies a number of points on the chest wall and the computer fits a smooth (spline) curve through these points. An example of automatic feature extraction is the estimation of waveform values from their trace on an oscilloscope photograph. The oscilloscope photograph is first digitized as an image; an estimation algorithm is then applied to each column of the image to determine the center of the beam trace.

CONCLUSION

We have developed a comprehensive image processing facility whose key computational component is an array processor. An extensive, flexible, maintainable, general purpose image processing software package has been implemented to perform a large number of image processing operations. The software system is actively being used in a number of image processing applications, examples of which have been described above.

"Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-ENG-48."

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and should not be used for advertising or product endorsement purposes.