

PHOTOGRAPHIC TO GRAPHIC TRANSFORMS

by

GRAHAM COPELAND

A.A.Dipl. Architectural Association, London
1975

Submitted in Partial Fulfillment
of the Requirements for the
Degree of
Master of Architecture in Advanced Studies
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1978

Signature of Author

Department of Architecture
January 19, 1978

Certified by

Nicholas P. Negroponete, Thesis Advisor.
Associate Professor, Department of Architecture, MIT

Accepted by

Professor Julian Beinart, Chairman.
Departmental Committee for Graduate Students

-1-

Rotch



PHOTOGRAPHIC TO GRAPHIC TRANSFORMS

by

Graham Copeland

Submitted to the Department of Architecture on January 19, 1978
in partial fulfillment of the requirements of the Degree of
Master of Architecture in Advanced Studies.

ABSTRACT

This report documents a package of computer programs that form part of an ongoing research effort on the M.A.G.I.C. system at the Architecture Machine Group in MIT's Department of Architecture. This research is involved with the development of a Spatial Data Management System.

The programs are designed to allow a user to study and perform transformations from photographic to graphic images utilizing interactive, digital computer-based image processing techniques.

The user is able to generate a photographic image on to a Ramtek RM-9300 raster-scan display and then filter, infill, reduce, enlarge, change gray-scale values and operate a determinant function on the image in any desired order and/or combination to achieve the required transform.

With reference to the Architecture Machine Group's Spatial Data Management System's requirements, particular emphasis is placed on the effect of miniaturization of images and the simulation of changing the image to 1-bit per pixel resolution.

The report initially gives an introduction to the field of image processing, explains the hardware configuration used, the software usage and concludes with a discussion of the programs and possible future developments. Two appendices are included; a user's manual and an index of illustrations used in the report.

THESIS SUPERVISOR: Nicholas P. Negroponte.

TITLE: Associate Professor of Computer Graphics.

ACKNOWLEDGEMENTS

I would like to take the opportunity to acknowledge the help of my thesis advisor, Professor Nicholas P. Negrofonte, and for providing me with the opportunity of studying with the Architecture Machine Group and experiencing, first hand, life and education in America.

I thank Bill Donelson for his help, suggestions and advice in the preparation of this report.

I thank the staff and graduate student body of the Architecture Machine Group for their unfailing help whenever I ran into problems.

I would like to dedicate this report to the staff of MIT's Foreign Student Office for their friendly and understanding help to an Englishman in America.

The work documented in this report has been sponsored by a full-time research assistantship from the Defense Advanced Research Projects Agency, contract number MDA903-77-C-0037.

Graham Copeland

January 1978

TABLE OF CONTENTS

ABSTRACT	2
ACKNOWLEDGEMENTS	3
1.0. INTRODUCTION	6
1.1. Spatial Data Management System	6
1.2. Thesis Objectives	10
1.3. Image Processing: The State of the Art	12
2.0. HARDWARE AND SOFTWARE ORGANIZATION	17
2.1. Hardware Description	17
2.1.1. Ramtek RM-9300 Display	17
2.1.2. Imlac and Tektronix Auxiliary Displays	20
2.1.3. Vidicon Camera	21
2.1.4. Interdata Mini-computer and M.A.G.I.C. System	23
2.2. Software Description	24
2.2.1. Transform	26
2.2.2. The Digitize Program	28
2.2.3. The Filter Program	30
2.2.4. The Infill Program	52
2.2.5. The Reduce Program	57
2.2.6. The Enlarge Program	63
2.2.7. The Grays Program	67
2.2.8. The Silhouette Program	71
2.2.9. The Information Files	75
2.2.10. Linkit and Runit	76
3.0. CONCLUSION AND FUTURE DEVELOPMENTS	76

REFERENCES 80

APPENDIX A - Programmer's Manual 81

APPENDIX B - Index of Illustrations 95

BIBLIOGRAPHY 100

1.0 INTRODUCTION

1.1 Spatial Data Management System¹

At the time of this report's writing the Architecture Machine Group has begun its second year of research into the concept of a computer-based Spatial Data Management System. This report forms part of the Architecture Machine Group's ongoing research in this field, henceforth referred to in this report as SDMS.

At the outset I feel it important to acquaint the reader with what has already been achieved in the first year of the SDMS research by the AMG.

By computer-based SDMS the concept of user-directed storage, search, and retrieval of data according to a user model of an interactive, high-bandwidth informational "space" at the user/computer interface is implied. This paradigm assumes the user's experienced sense of spatiality in order to organize the desired data inside a "virtual space" beyond the physical interface. The paradigm relies heavily upon such notions as multi-media cueing, response compatible user controls and an appropriate method of presenting the data images to the user for perusal.

In the SDMS the user effectively "travels" through a virtual

space user-defined by a gamut of visual, auditory, and haptic cues and encounters individual items of data carefully and methodically defined within that space.

As in the real world, vision is utilized as the primary means of travel; vision is the means of sensing where to go in the space by detecting at a distance the data of interest or indications of the location of that data.

The computer-based SDMS is thus defined as the intersection of:

- a) the user's sense of spatiality, developed and articulated over a lifetime of negotiating real space;
- b) a set of virtual spaces, generated at the user/computer interface by coordinated, interactive high-bandwidth media.

These virtual spaces are negotiated through by means of response compatible user controls and a secondary (auxiliary) navigational display.

The user controls consists of joy-sticks, joy-pads, and a touch sensitive tablet to allow direction and velocity to be communicated to the computer system. This report, however, is primarily concerned with the generation of a more usable

auxiliary navigational display. At present the matrix of the SDMS is a very large "data plane", a virtual x,y plane upon which the user's world of data is organized. The large 10-foot diagonal screen used in the Architecture Machine Group's "media room" displays the user's location as the user travels over this virtual plane. It is a $2\frac{1}{2}$ -D, full color, raster-scan display. This plane is 64 feet wide and 48 feet high (3072 square feet in area), but only displays a small section of the entire data plane at any one moment. In order to arrive at that particular section of the data plane, a navigational aid is employed that serves two primary functions:

- a) as an indication to the extent of the entire data plane and the user's relative location on it;
- b) as a mnemonic of the sites already visited so that the user may quickly return to a location on the data plane already perused, and at which the user feels a relationship exists to the current location.

At present this auxiliary display consists of a Tektronix color monitor (see Section 2.1.2.) which represents the macro view of the SDMS data plane as a (highly) symbolic configuration of rectilinear forms. A moving, transparent square negotiates this auxiliary macro view representing the location of the current image on the main screen in the data plane.

Figure 1 shows typically the media room and its multi-media

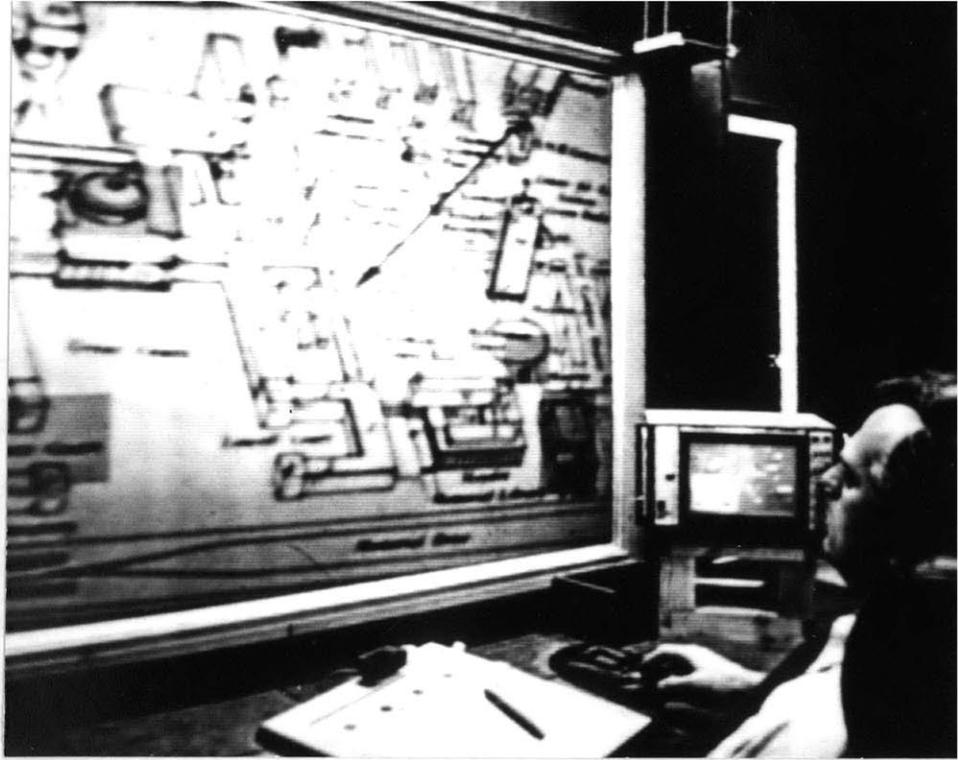


Figure 1.



Figure 2.

functions in operation.

1.2. Thesis Objectives

The existing auxiliary display previously described performs its two primary functions (relative location on the data plane, a memory "trigger" to already perused locations) with marginal success, a situation that the entire group working on the SDMS is particularly aware of. However, a higher resolution and immediately more useful auxiliary view has been considered as a principal development in this second year of research. The existing auxiliary view in no way reflects the high quality of the current SDMS as a whole.

The objectives of this report are to create an easily used, interactive method of generating miniaturized graphic representations of the original images in the data plane. The auxiliary view would, therefore, not only depict a miniaturized macro view of the entire data plane such that the user could continue to navigate throughout the data plane with greater appreciation of the data (both locally and universally available), but also would remember and recognize with far greater accuracy the location of the entire collection of data images. The advantage of a miniaturized graphic representation over that of a symbolic representation cannot be over-stressed. The innate problems of symbolic validity are well understood.² For example,

not only is it more difficult to create a valid symbol of the image with relative ease and speed, but also what is thought to be an obvious symbol to one user will often be quite opaque to another. The other users are also equally convinced that their symbols are similarly obvious. Indeed, an individual is more likely to forget over time exactly what these symbols represent.

This report, therefore, looks at various methods of image processing to create miniature graphic transformations of the data image to greatly improve the quality of the SDMS auxiliary display. The desired miniature image size would be approximately postage stamp dimensions (not including commemorative issues!), or something approximately $\frac{1}{100}$ th of the 12-inch diagonal Tektronix screen.

It should be noted also that another important advantage would accrue from utilizing such an approach to the auxiliary display. A small, black and white 1-bit image may be automatically created from each data image that the user stores into the data plane. Such an image can be easily stored and quickly retrieved from the hardware of the existing Architecture Machine Group's M.A.G.I.C. system (see Section 2.1.4.) without consuming precious disk space. This would allow more rapid perusal of the large screen display by freeing more usable computer-core for the running of the SDMS.

1.3. Image Processing: The State of the Art

The evolution of computer-based digital image processing can be viewed as a small, but important, subject in the larger realm of data processing. Just as the subject of this report may be viewed as a sibling of the SDMS, so image processing may be seen as a microcosm of data processing, relying considerably and consistently upon the techniques spawned by the field of data processing as a whole.

Image processing was originally established, and developed, as a batch-type computer operation. At its inception in the early 1960's only a handful of practitioners and applications were identifiable, but like the field of data processing as a whole it has grown with a steadily increasing rate. The limitations of batch-type operations have been overcome, and in common with most fields of computer-based endeavor, improvements in specific areas such as price/performance ratios of memory and increased speed of data input and processing times have contributed to a great stimulus in the widespread growth of applications.

The growth of image processing can be seen as having three (so far) distinct periods:

- a) the 1960's--Nasa's unmanned lunar landers sent back pictures from the Moon that astonished the world, not

least for the amazing clarity and sharpness of the images themselves. However, to technicians and their associates at NASA's Jet Propulsion Laboratory the quality came as no surprise. The incoming television pictures had been digitally enhanced by JPL, often imparting information, clarity, and detail into the images missing from the received photographs.

This application was essentially the first serious enlistment of digital image processing technology. Although created to lend clarity and precision to unmanned space probes, image processing (in those days referred to as "enhancement") was later nourished by the requirements of the American intelligence community for remotely sensed data, by photographic or radiographic images and by side-looking radar.

The very high volume data processing requirements (even by today's standards) dictated batch-mode processing during these early pioneering efforts.

Typical processors were the IBM 704 with magnetic tape input and multiple (or "overstrike") standard impact-type line printer hardcopy output. The density of overstrikes corresponded to the data at any point.

Initial use of graphic displays was severely limited to the CRT technology then available. The cost of sufficient storage capacity to allow data to be available in a storage medium with sufficient accessibility to provide an acceptable refresh rate^{was} the over-riding factor. Typical compromises for all but the most expensive systems involved long-exposure photography of an image built up line by line, and limiting the spatial resolution of the image typically to a display of 80 x 80 spatial resolutions

The breakthroughs in the 1960's were technological. The fixed-head disk, with pre-amplifiers to each head, provided the first cost effective solution to refresh memory. The development of algorithms continued, but with great frustration, since the large arrays of data to be processed with the equipment available produced results typically within hours, and often within days!

- b) the early 1970's--The first years of the 1970's saw the arrival of memory components and circuitry that allowed the reality of rapid processing of large quantities of data. The expansion of the technology of digital data processing and the proliferation of image processing applications nurtured the rapid evolution to a situation in which a user would interactively analyze and process

digital image data.

An initial expansion resulted from the adaption of pseudo-color such that small changes in intensity could be transformed into substantial subjective changes in color. However, the development of function processing, in which each pixel of the image may be changed in brightness, allowed the user the capability to digitally stretch and/or compress local portions of the value curves to subjectively affect contrast.

The small groups of image processing devotees rapidly enlarged as faster and cheaper circuits allowed a greater degree of interactive, real-time processing to be available to an ever larger, eager research community.

- c) the late 1970's--Two dominant trends have appeared in the development of image processing in the most recent years, trends that seem to be indicative of the present development and future direction of the field. These trends are the development of cheaper and larger computer systems allowing more difficult and memory-consuming problems to be researched, and the ability to more rigorously provide solutions to existing problems. Secondly, the emergence of small, yet more stylized systems adapted to the particular solution of a specific

set of problems. The subject of this report is seen as an example of this second trend.

Users in the field of image processing feel that the future will also be greatly influenced by the concept of distributed processing in which a large number of users are able to share a remote data base lending the increased abilities of group participation to processing problem-solving.

In terms of application, the field of diagnostic medicine has witnessed the largest single growth in the use of image processing to date. In the specific use of head or body scans the digitally processed output images are being increasingly used on the populace in general in preference to the much less attractive, yet familiar, X-ray.

Today many systems daily perform some kind of enhancement technique on images that may be validly classified as digital image processing. An ever increasing number of users has created an electronics industry (as yet in its infancy) specifically producing digital image processing technology that has allowed the horizons of application to rapidly expand.

However, there is still much to be understood and explored.

It is hoped that this report can add to the body of knowledge so far accumulated in this infantile, but quickly maturing, topic of digital image processing.

2.0. HARDWARE AND SOFTWARE ORGANIZATION

2.1. Hardware Description

The suite of programs described in this report runs on the AMG M.A.G.I.C. system. A diagram of the system is shown in Figure 3, the heavy-lined items being the hardware used specifically by the system in this report.

A cursory description and introduction to the operation of this hardware is useful to the (particularly naive) user. However, it is necessary to explain that the M.A.G.I.C. system is in a constant state of (necessary) flux, and particular users may find the hardware in various stages of evolution.

Figure 2 shows a typical layout of the hardware described below currently used by the author.

2.1.1. Ramtek RM-9300 Display

All of the photographic illustrations in this report were

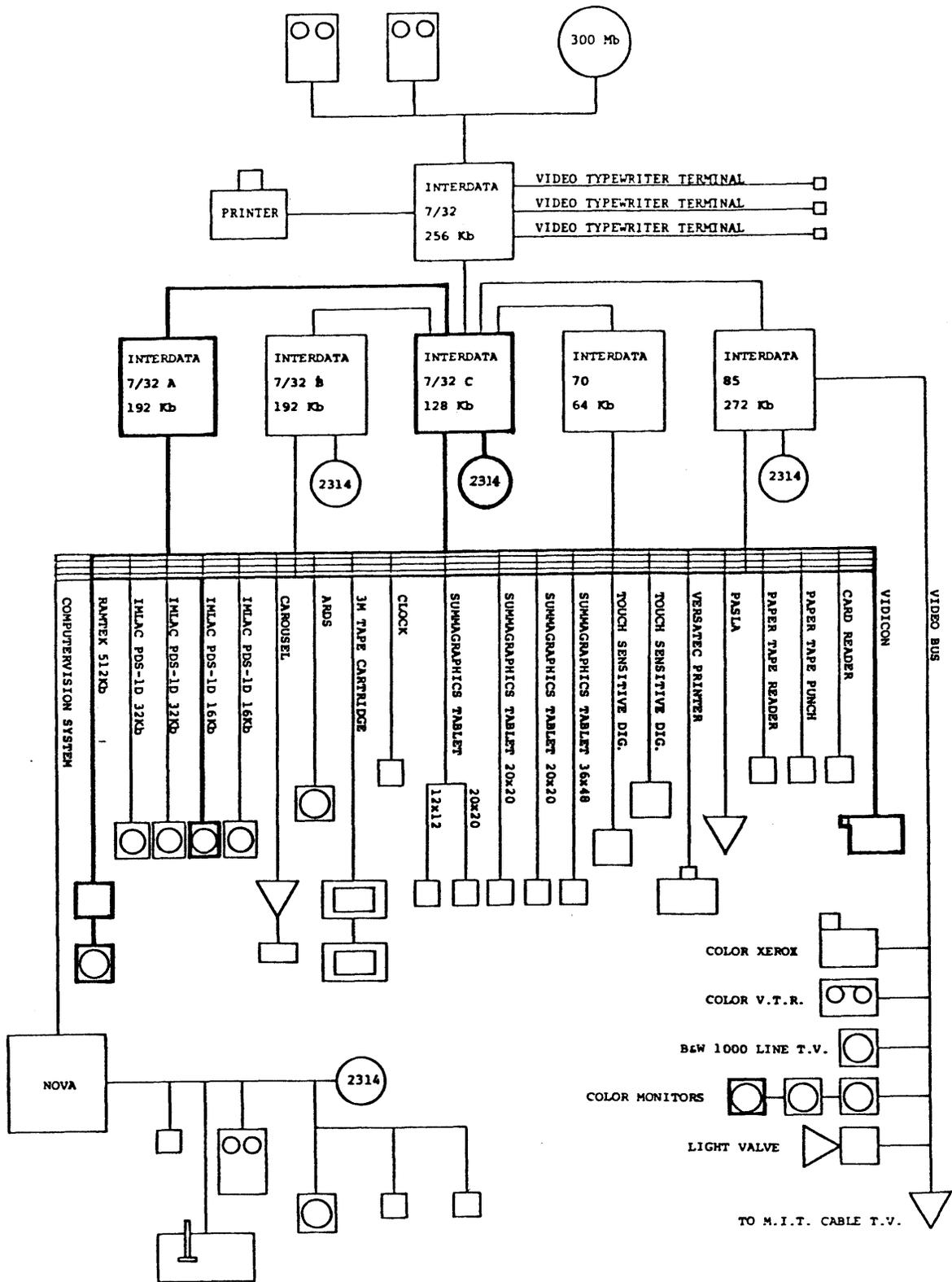


Figure 3.

taken from the Ramtek RM-9300 display (to allow a brighter image the tinted safety screen was temporarily removed; a Nikkormat with a Nikkor 55 mm, f3.5 macro lens was used for the photographs).

This display is effectively the "heart" of the sub-system used for the operation of the programs described in this report. The Ramtek RM-9000 series displays are microprocessor-controlled, full color, raster-scan graphic and image display systems. In order to "remember" its image, the display stores the image by 4k RAM refresh in binary, raster-scan pixel format allowing one to ten bits per pixel. Pseudo-color is achieved by means of a 1024 slot matrix (also variously described as "look up table", "random access table", etc.) which provides for the arbitrary equations of user-defined colors and any refresh memory bit pattern.

The RM-9000 series memory also provides, among others, for simultaneous read/write, eight scan sequences and single pixel addressing for write/read/erase. The particular model used by the AMG, the RM-9300, has a resolution of 480 lines by 640 pixels.

The microprocessor used in the Ramtek display is not used by the software in this report for any software operations. It is used solely for its primary function of memory refresh.

The pixel origin is in the top left hand corner of the image, but may be relocated in standard locations by changing the Ramtek microprocessor environment.

2.1.2. Imlac and Tektronix Auxiliary Displays

Although the Ramtek is used for the display and operation of images generated by the program suite, two further displays are utilized in order to:

- a) request information, commands and data from the user at the appropriate time;
- b) monitor the "Vidicon" image before digitization (see Section 2.1.3.).

The programs in this report are written to demand interactive use by the user. Certain commands, requests and instructions are given to the user by using an Imlac vector-display CRT (cathode-ray tube). No graphics are used in the Imlac display in this operation, only character.

The Imlac CRT display has a refresh memory of 16k-bytes and is linked to the mini-computer processor (see Section 2.1.4.) via a keyboard. All user commands are entered into the M.A.G.I.C. system through this Imlac/keyboard configuration. All information and requests for information from the processor are made to the user through the Imlac/keyboard configuration. The Imlac display,

therefore, is the "command level" of the system.

The second important auxiliary display used by the system is a Tektronix 12-inch diagonal, full-color, raster-scan television display. The Tektronix is used as a monitor for aligning and focusing the Vidicon camera (see Section 2.1.3.) prior to digitization on to the Ramtek. The Tektronix display uses a standard "Trinitron" color tube, but is capable of displaying the image in American standard NTSC or RGB. User-controls allow switching from monitoring of the Vidicon image to monitoring of the current Ramtek image. Probably the only other significant user control is switching of the frame to "smaller than screen" size. The significance of this capability is the ability to view the extreme edges of the image that might otherwise be obscure at "screen normal" image size.

These two auxiliary displays are located adjacent to the Ramtek display and very near to the Vidicon camera and mini-computer processor.

2.1.3. Vidicon Camera³

Sierra Scientific Corporation's video standard high resolution CCTV camera used in the AMG consists of two separate units:

- a) a camera head model number LSV-1 which contains a 1-inch

- antimony trisulfide vidicon and a standard "C" lens mount;
- b) a camera control unit (CCU) which allows panel control of power, focus, beam, gain, etc.

The two parts of the camera are linked together under the raised floor by means of a specially fabricated interconnecting cable. The signal/noise ratio with this configuration is typically 43dB.

The "C" lens mount is fitted with an adaptor to accept the Nikkor 55mm, f3.5 macro lens. Although the particular mask in the Ramtek creates a high-definition image (by American standards), the configuration of the Vidicon camera and Nikkor macro lens is, in fact, beyond the resolution of the Ramtek and is therefore perfectly adequate.

Two photographer's flood lamps with deflectors are used to illuminate the photograph to be digitized into the Ramtek display, and the Vidicon is supported on a tracked stand immediately between the flood lamps.

One important point should be noted by the user. The Vidicon camera should never be switched on or left switched on while the Ramtek is switched off, as the Vidicon/Ramtek interface may easily be burnt out in this situation. The Vidicon power switch on the front of the CCU is located above the mini-computer processor.

2.1.4. Interdata Mini-computer and M.A.G.I.C. System

The aforementioned displays and camera are linked to the M.A.G.I.C. system via an Interdata Model 7/32 192k-byte mini-computer processor. This processor itself runs off a common shared bus which is controlled by another Interdata Model 7/32 128k-byte mini-computer processor, and at present access to the 2314m-byte fixed head disk is through this second processor.

It is because of the somewhat tortuous route between Ramtek and disk that no image storage is performed by the programs in this report. The time needed to store/retrieve full screen images prohibits this facility.

The hardware configuration used to operate the program suite in this report consists, therefore, of:

- a) Ramtek RM-9300 raster-scan high resolution color display;
- b) Imlac vector display;
- c) Tektronix raster-scan color display;
- d) Vidicon two-piece camera;
- e) Interdata Model 7/32 192k-byte mini-computer processor;
- f) Interdata Model 7/32 128k-byte mini-computer processor;
- g) 2314m-byte fixed head storage disk.

2.2. Software Description

The suite of programs described below consists of twenty-one programs, seven information files and an execution file linker and run programs. The programs are linked together to create an execution file that is run by the user (see Figure 4).

All the programs are written in the PL1 programming language and are compiled with no-debug (this allows the programs to operate as quickly as possible).

The source files (PL1 and Text) are in the directory "SIMPLE."

The execution file and the information files are in the directory "SIMPLER." The execution file is called "TRANSFORM."

To start the programs on the current M.A.G.I.C. system it is necessary to run on the Interdata Model 7/32 processor. Read/write actions to the disk storage is performed via the Interdata Model 7/32C processor, but all other functions are performed via the 7/32A machine.

To acquaint the naive user with the startup procedure, it is necessary to:

- a) login to the system at the Imlac by typing "L SIMPLER";

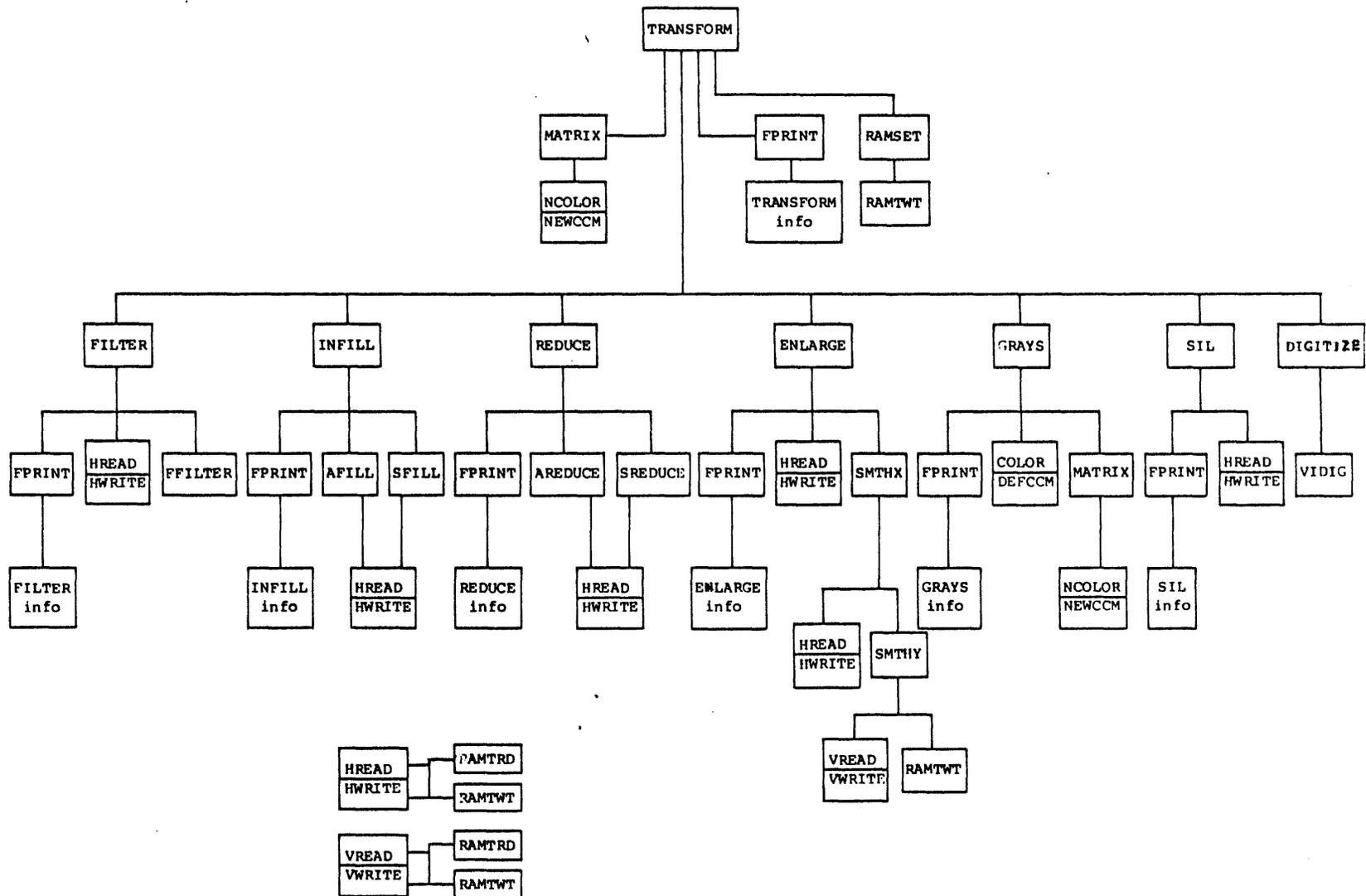


Figure 4.

- b) switch on the Ramtek and Tektronix displays;
- c) switch on the Vidicon camera and attach the lens;
- d) at the Imlac type "TRANSFORM"

From the last command onwards, the system requests information and commands from the user at the appropriate time in the operation of the routines. In this sense the routines are interactive; indeed the routines could be described as pseudo-realtime since the current transform occurs in front of the user.

All requests for information and commands to the user are in upper case characters. All responses from the user are in upper case letters also. However, information that the current transform is in progress is given to the user in lower case characters.

The user is informed at all times the present state of the programs.

Figures 5 and 6 illustrate the original images used for the transforms in this report.

2.2.1. Transform

This program allows the user to call the six programs that are responsible for the "Photographic to Graphic Transforms" and



Figure 5.

...and even a dry 1970 CBS
of the Beethoven Op 18 quartets
life. I went mad, what could I try
ped Edison off the radio, the 'Mary-
e-lamb' routine, and again, Edison,
limited frequency range voice (doc-
still looking into it) had a good try
resent in the room. The only kind
el that didn't seem to take to this
was some kinds of rock—it usually
the effect of a great band on stage
'Skin Music' by Ry Cooder, for
was very amenable—but occasion-
of the kind of sound, on drums for
that recording engineers sweat
get rid of.
r level is important. If it is too loud,
n vanishes and the sound becomes
although it was fun to put on a
d Lee disc—not mine I hasten to
get the effect of them singing to
a football stadium—well almost).
he unit in circuit the music col-
ck behind the speakers and it be-
arduous not having to use the
s, when critically listening to records
... because they all became pleas-
blew the air hoses
wondered also if there was any
noise in the output as both my cat
very disturbed when I was using the
Pulse.)
My heart sank when I sent the Au-
back to Pyser, but it lifted again w
ceived a Sansui QSD-2 four-channel
synthesiser. This is based on the s
A vario-matrix circuit used in the
receiver (see FM Radio, p. 99) w
provide up to 20 dB separation betw
cent channels and 30 dB on diagon
QS encoded material. It has alter
facilities apart from front stereo
SQ, matrix H, and two QS synthe
tions—'Hall' and 'Surround', for us
channel sources. It's the operatio
two functions that I wanted to lo
merits of the four channel matri
have been examined *ad nauseam* a
want to get embroiled in further d
especially now with Ambisonics
large. The 'QS Hall' and 'QS
functions provide enhancement
sources by using the information
Sansui originally developed the Q

Figure 6.

for digitization of the photographic image itself.

The program always returns to the high level program when each transformation program is complete. The user is asked to choose yet another transform, or conversely to exit from the programs.

As in the other programs described below it is only necessary for the user to answer the programs' requests with a single, upper case character response. Longer character responses, however, will not cause any grief to the program.

Every time the user is asked to choose a new transform the list of options is given. The chosen transform is then immediately called and the user is made aware that the new transform is in operation.

2.2.2. The Digitize Program

This program is the means by which a photographic image may be digitized into the Ramtek display.

The program is the most "hardware involved" of the programs. Assuming that the steps outlined in Section 2.2.2. have been completed, it remains necessary to set up the photograph ready

for digitization:

- a) the photograph is placed on the table below the Vidicon and the lights switched on. It is a safeguard to set the lens initially to f16 or f22 to avoid flaring of the image in the Tektronix. The Tektronix is used to manipulate the photograph into the required position; the Vidicon camera is moved up or down its track and the lens adjusted until the photograph is framed and focused properly;
- b) the lens on the Vidicon camera must now be adjusted by rotating the f-stop collar until the image in the Tektronix is sufficiently bright. The brightness required will become "second nature" to the user after a little experience using the Vidicon/Ramtek set-up;
- c) the user starts the digitization of the image by following the instructions in the Imlac display.

The "digitize" program calls a sub-routine entitled "vidig". This allows the digitization to be accomplished. If the lens's f-stop was set incorrectly the user can adjust and digitize the image over and over again until a correct "exposure" is achieved in the Ramtek display.

The program requires approximately 20 seconds to digitize the image into the Ramtek display.

The image digitized into the Ramtek is an 8-bit matrix preset, black and white (gray scale) image. The transforms described below, therefore, are also black and white in nature, and no attempt is made in this report to become involved in colored images.

The program returns to "transform" and the user is asked to choose another transform.

2.2.3. The Filter Program

This program consists of two programs "filter" and "ffilter" and calls to various sub-routines as described above. This program is seen as providing the principal proportion of the research and description of the author's work in the field of image processing.

The programs directly convolve a 3 x 3, 2-dimensional pixel kernel with the Ramtek display, and assumes an 8-bit, preset matrix.

The number of operations required to convolve a 480 x 640 pixel image with a n x n kernel is approximately:

$$\frac{n^2}{3} \times 10^6$$

As one would expect, only small kernels (i.e. $n=2, 3$) are reasonably usable for implementation, even when using realtime hardware convolutions. While the degree of freedom offered by a 3×3 convolution kernel are obviously limited, its use in the application described in this report should not be underestimated. For the linear, non-adaptive convolutions described below, there are 11 degrees of freedom available for user adjustment (the nine slots of the kernel plus the normalization and offset factors). There are essentially three classes of 3×3 convolution kernels available to the user by these programs:

- a) the isotropic kernels (mean, Guassian and Laplacian filters);
- b) the non-isotropic kernels (compass gradient filters);
- c) the "yet to be defined" kernels, allowing the user to invent and test new filters beyond the standard, well-documented filter described and illustrated in this report.

The isotropic filters are symmetrical about north/south and east/west axes. The action of low pass filters upon the kernel effectively reduce the number of gray-levels by cleaning up the local variances in the image. The mean (low pass) filter is given by the kernel

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

and is illustrated in Figure 7. This filter performs a simple averaging of the pixels in the 3×3 kernel. It is also used

as the default kernel in the routines.

The Guassian (low pass) filter is given by the kernel

$$\begin{array}{ccc} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{array}$$

and is illustrated in Figure 8. This filter performs a de-focussing operation and is particularly useful for obtaining local threshold values in subsequent edge detection. The Guassian filter will be used in the remainder of the description of these filter operations due to its superior sidelobe structure to its mean counterpart.

The family of three Laplacian (high pass) filters described below should be used on images already filtered by at least one pass of the Guassian filter (to clean up the image).

The Laplacian filters are omnidirectional derivatives and are useful where there exists no preferred orientation in the image. The Laplacian filters are used as edge detectors without regard to edge direction, but have also been used in coding and pattern recognition applications. In the examples and illustrations below, the Laplacian filter is used to improve the analog gradient image.

The Laplacian filter (a) is not illustrated here as it performs the least well of its family; the author hopes that the



Figure 7.



Figure 8.

description and illustration of the better filters will suffice the reader. The high pass Laplacian (a) filter is given by the kernel

$$\begin{array}{ccc} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{array}$$

If this filter kernel is used, the author recommends a normalization factor of -2 and an offset of 64. The Laplacian (b) filter is given by the kernel

$$\begin{array}{ccc} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{array}$$

Figure 9 shows the effect of the combination of Guassian and Laplacian (b) filters, and Figure 10 shows the change to a simulated 1-bit (black and white) image.

The Laplacian (c) filter is given by the kernel

$$\begin{array}{ccc} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{array}$$

Figures 11 and 12 are similar to Figures 9 and 10, but illustrate the combination of Guassian and Laplacian (c) filters. The author feels that the illustrations confirm that the analog gradient image using the combination of Guassian and Laplacian (c) filters is slightly superior to the alternative. The illustrations in Figures 13 to 20 depict various combinations of Guassian and Laplacian (c) filters, and the effect of changing to simulated 1-bit images (see Appendix B for the description of these illustrations).



Figure 9.



Figure 10.



Figure 11.



Figure 12.



Figure 13.



Figure 14.



Figure 15.



Figure 16.



Figure 17.



Figure 18.



Figure 19.



Figure 20.

Direction of Gradient	Prewitt	Kirsch	3-level Simple	5-level Simple
North	1 1 1	5 5 5	1 1 1	1 2 1
	1 -2 1	-3 0 -3	0 0 0	0 0 0
	-1 -1 -1	-3 -3 -3	-1 -1 -1	-1 -2 -1
North-West	1 1 1	5 5 -3	1 1 0	2 1 0
	1 -2 -1	5 0 -3	1 0 -1	1 0 -1
	1 -1 -1	-3 -3 -3	0 -1 -1	0 -1 -2
West	1 1 -1	5 -3 -3	1 0 -1	1 0 -1
	1 -2 -1	5 0 -3	1 0 -1	2 0 -2
	1 1 -1	5 -3 -3	1 0 -1	1 0 -1
South-West	1 -1 -1	-3 -3 -3	0 -1 -1	0 -1 -2
	1 -2 -1	5 0 -3	1 0 -1	1 0 -1
	1 1 1	5 5 -3	1 1 0	2 1 0
South	-1 -1 -1	-3 -3 -3	-1 -1 -1	-1 -2 -1
	1 -2 1	-3 0 -3	0 0 0	0 0 0
	1 1 1	5 5 5	1 1 1	1 2 1
South-East	-1 -1 1	-3 -3 -3	-1 -1 0	-2 -1 0
	-1 -2 1	-3 0 5	-1 0 1	-1 0 1
	1 1 1	-3 5 5	0 1 1	0 1 2
East	-1 1 1	-3 -3 5	-1 0 1	-1 0 1
	-1 -2 1	-3 0 5	-1 0 1	-2 0 2
	-1 1 1	-3 -3 5	-1 0 1	-1 0 1
North-East	1 1 1	-3 5 5	0 1 1	0 1 2
	-1 -2 1	-3 0 5	-1 0 1	-1 0 1
	-1 -1 1	-3 -3 -3	-1 -1 0	-2 -1 0

Figure 21.

The non-isotropic filters (or "compass-gradient" filters) are shown in Figure 21. The filters are non-symmetrical and yield different analog gradients in each direction. For each class of filter (Prewitt, Kirsch, 3-level simple and 5-level simple) there are 8 compass gradient levels resembling the north, north-east, east, etc. sides of compass. These edge detection filters are particularly useful where there exists a preferred direction in the digitized image. The compass names indicate the slope direction of maximum response, for example the north-gradient kernel produces a maximum response for vertical luminance changes, i.e. a horizontal edge. The compass direction is determined such that the brighter side of the edge is always to the left as the user moves in the direction of the edge.

Unlike the isotropic filters, the compass gradient filters have more than one "optimum" value for normalization and offset factors.

Figures 22 to 37 illustrate the use of the four compass-gradient filters, together with the Sobel operator. In all of the cases, the top half of the illustration shows the effect of a north gradient filter with a subsequent, superimposed west gradient filter--the Sobel operator. The bottom half of the illustration depicts the use of the north gradient filter only, the top half the effect of the Sobel operator. Each pair of

illustrations shows the filters and the effect of the simulated 1-bit image.

The Prewitt filter is a 3-level filter with de-emphasis on the central pixel of the 3 x 3 kernel. Similarly, the Kirsch filter is a 5-level filter with de-emphasis on the central pixel, but the 0 value of this pixel suppresses the jitter on the line where an edge occurs. The 3-level simple filter approximates the partial derivatives in order to detect edges by placing de-emphasis on the entire central line of pixels, dependent upon the current compass gradient. The 5-level simple filter, however, has certain advantages:

- a) the filter approximates the partial derivatives in order to detect edges as does the 3-level simple filter;
- b) unlike the Prewitt and Kirsch filters, only four of the eight compass gradients are needed to obtain the optimum filters;
- c) a higher gradient yield is obtained in the diagonal direction than in horizontal or vertical directions. It is known that perception in horizontal and vertical direction is superior to perception in diagonal directions for a large range of spatial frequencies.⁴

The structure and integer weights of the 3-level and 5-level simple kernels make them particularly suitable for first digital computation of gradient magnitudes and directions.



Figure 22.



Figure 23.



Figure 24.



Figure 25.



Figure 26.

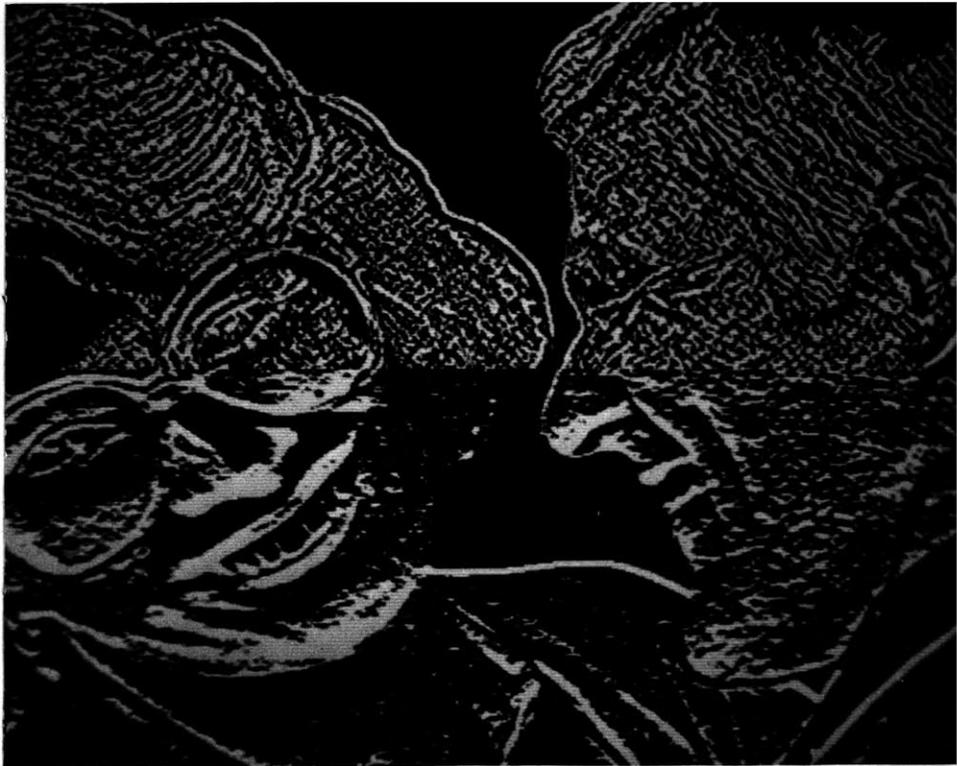


Figure 27.



Figure 28.



Figure 29.



Figure 30.



Figure 31.



Figure 32.



Figure 33.



Figure 34.



Figure 35.



Figure 36.



Figure 37.

The advantage of the Sobel operator is that a gradient modulus is obtained by taking the magnitude of the two orthogonal filters, and any dramatic inconsistencies due to missing edges are more easily corrected.

The advantage of user-defining the filter kernel, normalization and offset values lies in the ability of accurately "calibrating" the filter transform for each individual image. Whilst necessarily the photographic image is consistent in all of the figures to allow ready comparison, the particular values enumerated in Appendix B should not be taken as optimum values for all applications.

The filter program requires approximately 2 minutes and 40 seconds to complete each image transform.

The program returns to the program "transform" when the user decides to dis-continue filter transforms, and the user is asked to choose another transform.

2.2.4. The Infill Program

The infill program consists of three programs "infill", "afill" and "sfill". The user is able to choose between the

two methods of infilling the Ramtek display. The infill program is included to allow the user to increase the "strength" of image sections that, when reduced, would otherwise be lost or unrecognizable. The principal use of the infill program is seen in the transform of characters before reduction.

The "infill" program allows the user to choose between filling by averaging ("afill") or by skipping ("sfill"). The two programs are included to show the different results yielded by the techniques.

There is a marked difference in the operation times of the two programs, and Figure 38 illustrates in graphical form this difference. The range of infill values is consistent in both graphs, from 2 to 100.

It can be seen that the program "sfill" requires approximately $\frac{2}{3}$ the time to complete the transform than does "afill".

The illustrations shown in Figures 39 to 42 depict the action of the two programs on a typical example of print. Figures 39 and 40 depict a pixel fill number of 5 suitable for small print operated by "afill" and "sfill" respectively. It can be seen that at these resolution levels, the effect of "afill"

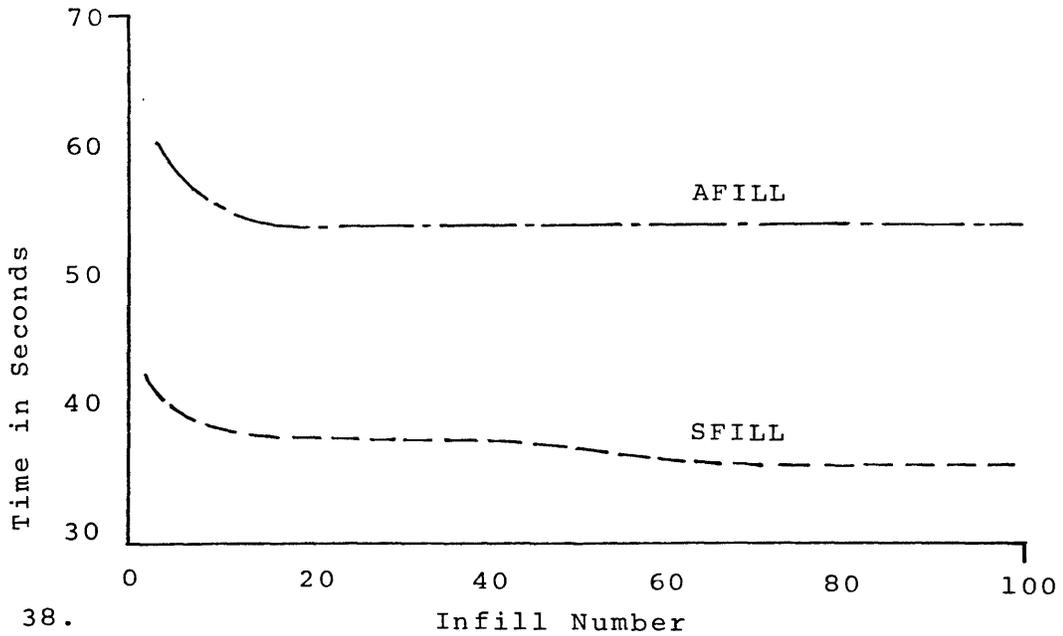


Figure 38.

is to more closely approach a "bar" effect on the print, whereas the "sfill" program results in a highly fractured image. As may be imagined, upon reduction the two images are quite successful in producing a transform that creates the impression of large or small print whilst still retaining the pattern of the original.

Operation of the infill program upon a natural image is relatively unsuccessful. At full screen size the image is disturbingly "faceted", and reduction gives inferior results to simple reduction at the outset.

The infill program, therefore, is suitable in lending "strength" to characters in print prior to reduction, and the

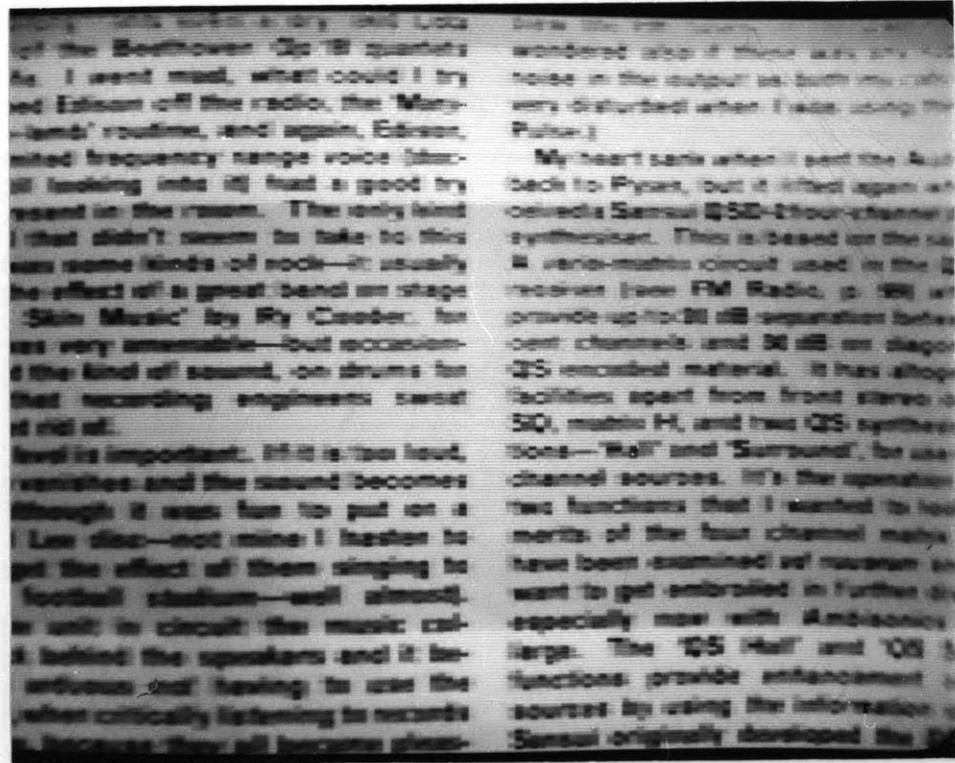


Figure 39.

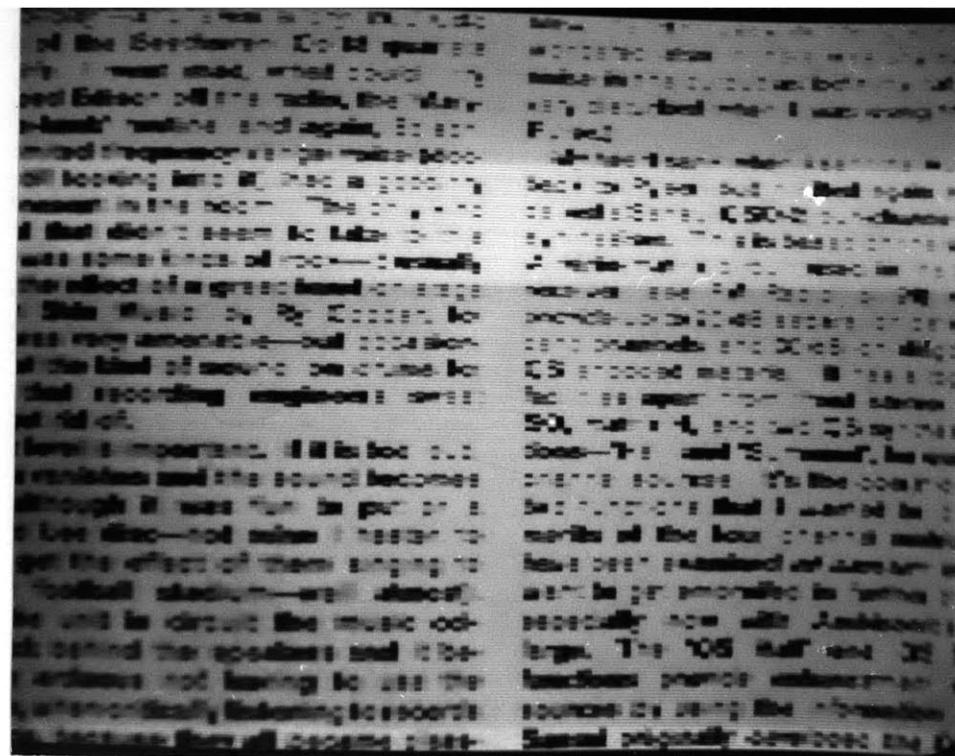


Figure 40.

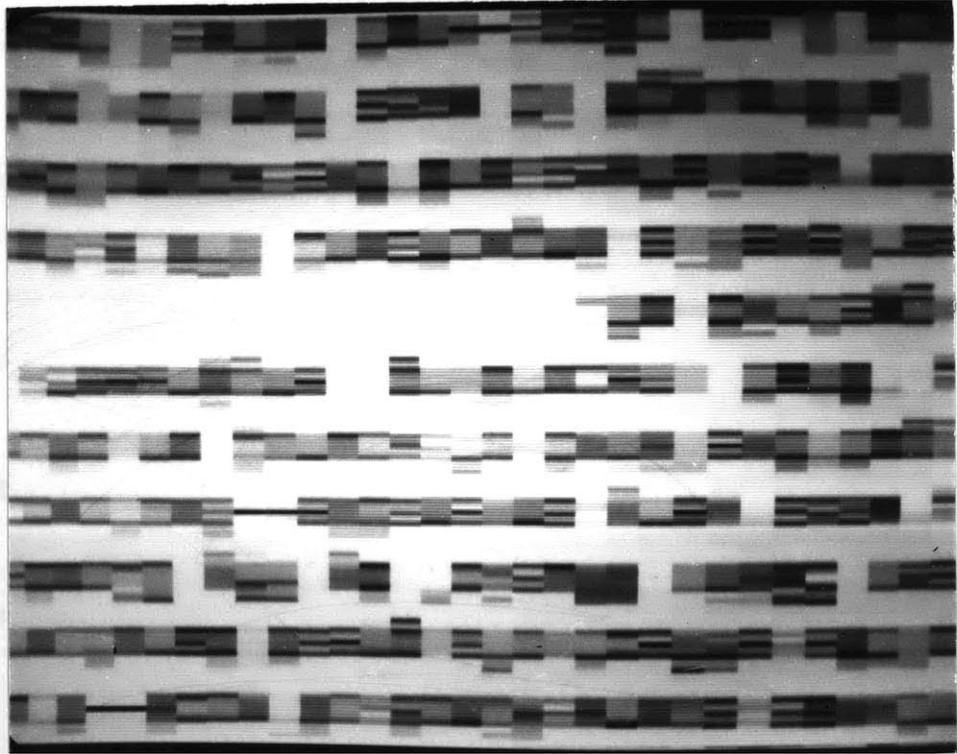


Figure 41.

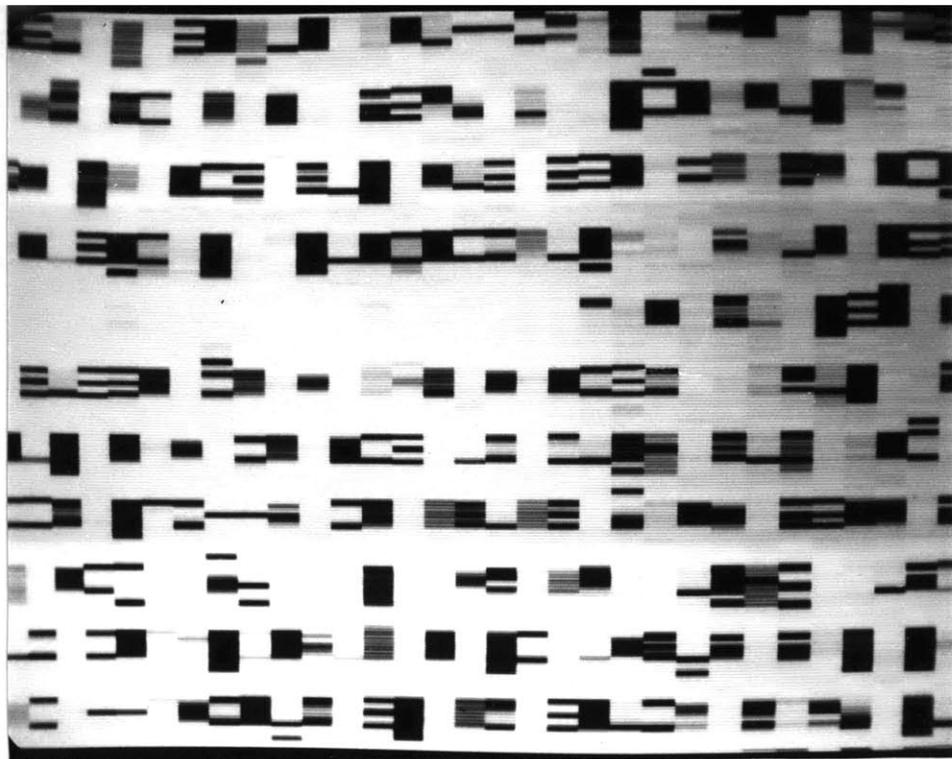


Figure 42.

averaging program "afill" is particularly recommended to the user.

Upon completion, the program returns to the program "transform" and the user is asked to choose another program.

2.2.5. The Reduce Program

This program consists of three programs "reduce", "areduce", and "sreduce" and allows the user to reduce the current image in the the Ramtek display.

The program "reduce" allows the user to access the two programs that reduce the image by averaging ("areduce") and by skipping ("sreduce").

Although the ultimate use of the programs described in this report is to allow the user to create small images for the auxiliary display of the SDMS, only the illustrations to this program are given in reduced form. This is due to the limits of photographing such small images in the limited environment of a working, computer laboratory.

There is considerable difference in the operation times of the two reduction programs, and Figure 43 illustrates graphically the difference between the run-times of these two programs for reduction ratios from 2 to 20.

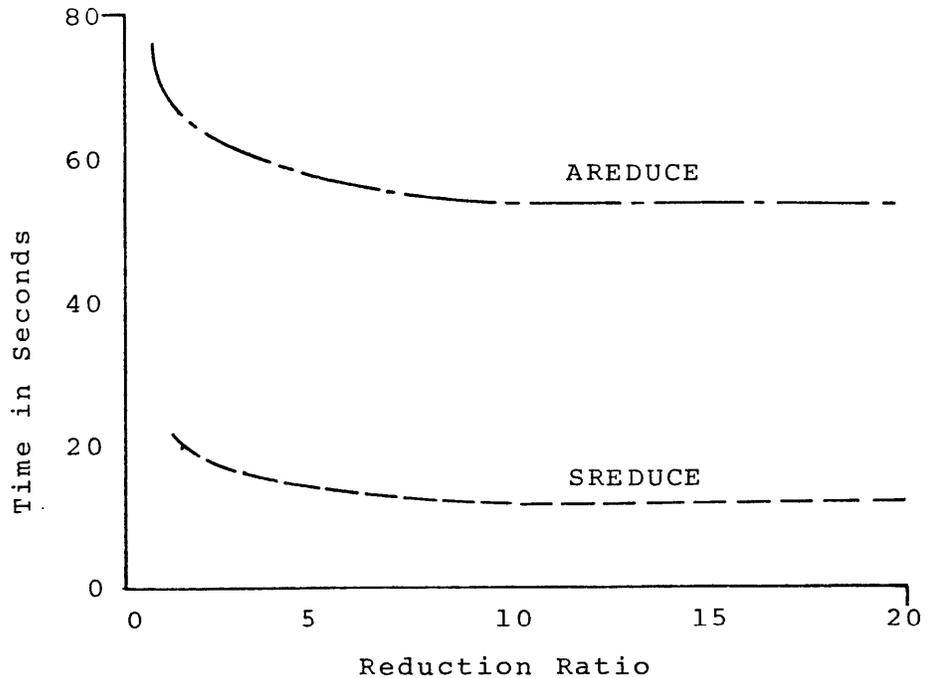


Figure 43.

Figures 44 and 45 illustrate the effect of reducing the image by a ratio of 4 ($\frac{1}{16}$ original image size) by "areduce" and "sreduce" respectively. Similarly Figures 46 and 47 illustrate the transform with a ratio of 8 ($\frac{1}{64}$ original image size) by "areduce" and "sreduce" respectively.

The reduced image using "areduce" is as sharp as the alternative but maintains the contrast more closely resembling the original than does "sreduce". Indeed the averaging program uses all of the pixel values to arrive at the reduced image and is more able to maintain the original information.

The illustrations in Figures 48 and 49 depict a photographic



Figure 44.



Figure 45.



Figure 46.



Figure 47.

enlarged image reduced from the original by a ratio of 10 ($\frac{1}{100}$ original image size). The illustration has been re-enlarged for the purpose of this report to give the user a more precise indication to the quality of the reduced image at such reduction levels. The figures illustrate the use of "areduce" and "sreduce" respectively.

While the problems of photographing and printing such small images somewhat detracts from the spirit of this report, the author would point out that these illustrations are used only to depict the possibilities of the system in order that the user may understand, becoming involved with the programs "on the fly", their capabilities.

It should be noted that the image is centered on the screen. This is done so that:

- a) the user is always able to easily calculate, knowing the reduction ratio, the corner coordinates of the reduced image (also true if the image's top left hand corner had been located at the coordinate origin);
- b) the reduced image does not violate any unprocessed parts of the original image;
- c) to allow the enlarge program to operate without resort to further hardware (see Section 2.2.6.)

Upon completion the routine returns to the program "transform"



Figure 48.



Figure 49.

and the user is asked to choose another transform.

2.2.6. The Enlarge Program

As a corollary to the reduce program, for the author's sense of symmetry and to provide the user with the ability to achieve reductions with greater clarity, an enlarge routine is provided consisting of "enlarge", "smthx" and "smthy" programs that allow the user to enlarge the central area of the image in the Ramtek display.

The user chooses an enlargement ratio and for every pixel in the original display a ratio-number of pixels on a ratio-number of lines is written into the Ramtek display to create the enlarged image. So as not to violate any unprocessed parts of the original image, the enlarged image is written into the display from top and bottom simultaneously.

After the enlargement transform is complete, two further programs "smthx" and "smthy" are consecutively run to "smooth" out the enlarged image. As can be imagined, for enlargements involving a large ratio in particular (greater than 4), the new image is extremely "faceted". The "smthx" program smooths out the facets in the horizontal direction and "smthy", after changing the Ramtek microprocessor environment, smooths out in

in the vertical direction.

The run-time of the enlarge program is illustrated below in Figure 50, showing the difference between running with and without the smoothing operation. The enlarge ratio varies between 2 to 20.

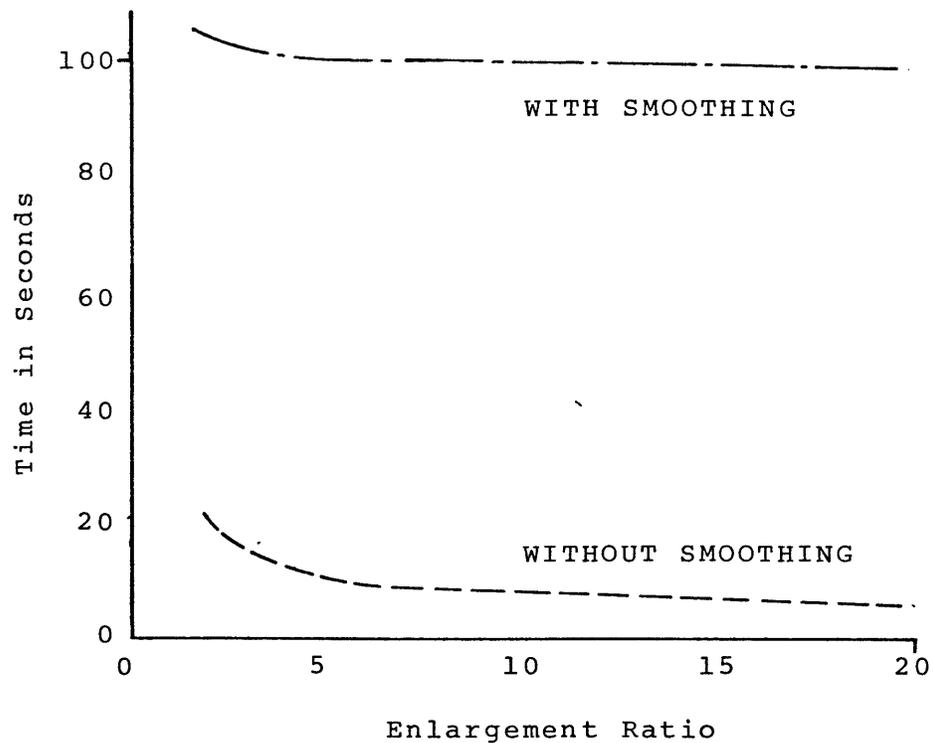


Figure 50.

Figure 51 illustrates the effect of enlarging with a ratio of 4 (the central $\frac{1}{16}$ of the original image) and Figure 52 with a ratio of 8 (the central $\frac{1}{64}$ of the original image). As can be seen, the images are not particularly useful in themselves. The value of this program, however, lies in the procedure when reducing an image still further from an existing reduced image. For example, an image reduced by a ratio of 4 will impart more information,



Figure 51.



Figure 52.

clarity, and contrast when reduced by a ratio of (for example) 8 if immediately preceded by an enlargement, using the enlarge program, back to the original full screen size. This is particularly true when using the "areduce" program since this program's success increases with the quantity of information available.

The enlarge routine concerns itself only with a transform from the central portion of the screen due to inadequacies in the present hardware system. As explained above (see Section 2.1.4.), all read/write operations to the 2314 disk are made via a second, time-share processor (the Interdata Model 7/32C). In order to enlarge from parts of the screen other than the center it is necessary to store all the pixel values onto disk before processing can begin. This is due to the fact that it would be impossible not to violate unprocessed parts of the original image whilst writing the new, enlarged image into the Ramtek display. Since read/write operations to disk are extremely slow, the time involved becomes prohibitive.

As a secondary motive for not dealing with non-centered images, the problem of defining the area to be enlarged would be best served by using the Summagraphic tablet and pen and the Ramtek cursor to indicate the center of enlargement. The author feels that enough hardware is already tied up by the programs

at present and wishes to avoid adding a further piece of hardware for such a limited use.

Upon completion the program returns to the program "transform" and the user is asked to choose another transform.

2.2.7. The Grays Program

This program consists of the program "grays" and is used to simulate the change to less than an 8-bit, but not less than 1-bit, image.

As explained in Section 2.2.5 above, the ultimate objective of this report is to enable the user to produce clear, minute images resembling the original photograph. It is also important, however, that these images be defined as only 1-bit per pixel so as to occupy the minimum necessary space on the disk.

To this end illustrations in the filter program have shown the effect of changing from an 8-bit to a 1-bit image. This was achieved by operating the program "grays" on the filtered image and changing the Ramtek color matrix to only two gray-scale values-- black and white.

It is most important to note that the grays program does not

change the 8-bit image to a lesser number of bits per pixel but (simply) simulates this change. The author feels that it is not part of the requirements of the programs in this report to actually effect that change, only to depict what would accrue if the change were to happen.

The user is asked to define the number of gray-scale values that are to be loaded into the color matrix. The following number of values correspond to the appropriate simulated bit per pixel images:

2	gray-scale	values	simulate	a	1-bit	image
4	"	"	"	"	2-bit	"
8	"	"	"	"	3-bit	"
16	"	"	"	"	4-bit	"
32	"	"	"	"	5-bit	"
64	"	"	"	"	6-bit	"
128	"	"	"	"	7-bit	"

In order to change the number of gray-scale values it is possible to return to the original image (256 gray-scale values) each time, although not necessary if the number is being successively reduced.

Figures 53 to 56 illustrate the simulation of changing the number of gray-scale values to 128, 16, 4, and 2 respectively (simulating 7, 4, 2, and 1-bit images).

It will be seen that the higher number examples (particularly the 128, 7-bit simulated image) are more confusing than any other illustration in this report. It is only the 4 and 2 level gray-

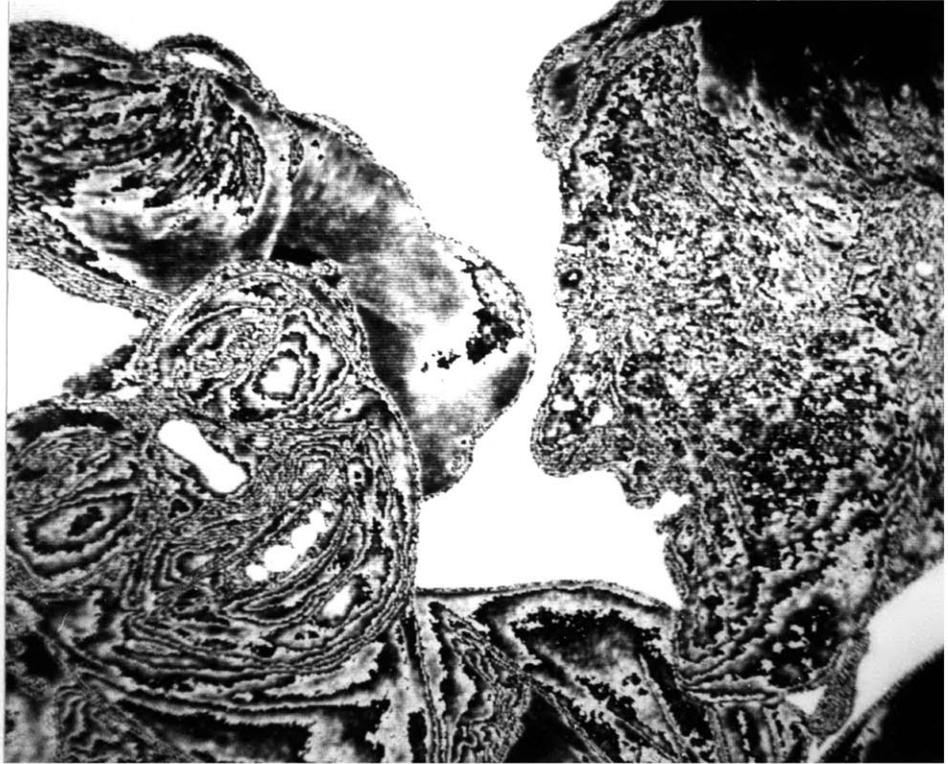


Figure 53.

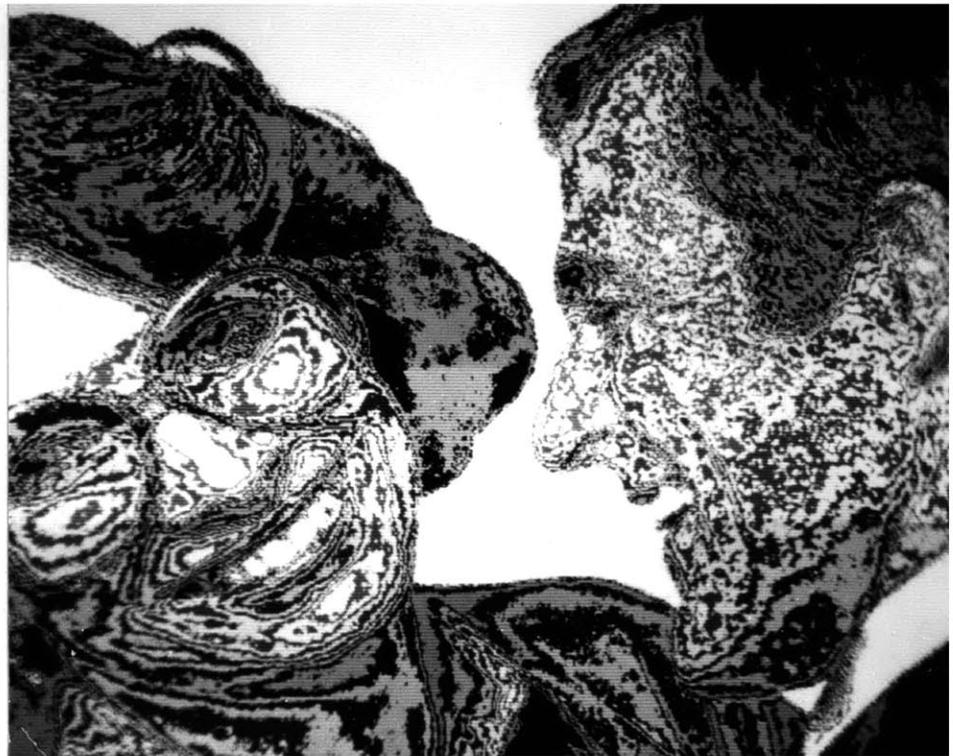


Figure 54.



Figure 55.



Figure 56.

scale value illustrations that are really useful to the user. Indeed it may be argued that the 2 level, 1-bit simulated image is the clearest of all the illustrations in the report. The user will find most effective results in changing to 2 gray-scales and using "areduce" to reduce the image by averaging in producing clear, recognizable miniature images.

The run-time of the grays program in changing the matrix (and hence the image) is virtually immediate (less than 1 second). However, the return to the original image requires approximately 6 seconds to complete and the user is advised that this operation is in progress.

Upon completion this routine returns to the program "transform" and the user is asked to choose another transform.

2.2.8. The Silhouette Program

This program operates a determinant algorithm upon a 2 x 2, 2 dimensional pixel kernel, and is an old and well tried formula for edge detection. The algorithm lacks the finesse and capability of the Laplacian and compass gradient 3 x 3 filter kernels described in Section 2.2.3. above. The author includes the program here more as a comparison to the 3 x 3 filter kernel than as a serious transform contender.

As with the other transforms, however, the user must gain experience in using the silhouette program in order to realize the best image transforms.

Because only a 2 x 2 kernel is being processed, the runtime of the program is faster than the 3 x 3 filter kernel at approximately 1 minute and 35 seconds for completion.

The illustrations given in Figures 57 to 60 depict user-defined determinant threshold values of 500, 1000, 2000 and 3000 respectively. The illustrations clearly show the "speckled" nature of the image after the program is complete and the black lines in the larger white areas. Although no illustrations are shown the user is recommended to convolve the image initially with a 3 x 3 Gaussian filter to clean up the digitized image. In this way the extremes of local contrast are reduced.

The silhouette program, however, unlike the 3 x 3 filter edge-detection kernels is less susceptible to extremes of contrast at edges. This is because the algorithm exaggerates the contrast at the edge in order to detect that edge (relative to the user-defined threshold value).

The silhouette program is useful for determining the relative image brightness. By applying a large value to



Figure 57.



Figure 58.



Figure 59.



Figure 60.

the image, the user is able to verify, by experience, the overall contrast of the image by the resultant operation of the silhouette program.

However, the user will probably determine that the use of the silhouette program will not yield results as valuable as with the edge detection by the 3 x 3 filter kernel.

It is interesting to note that the silhouette program is a relative of the first filters used by the JPL to enhance photographs from space probes; the evolution of image processing techniques is indicated by the value of this program in relation to contemporary programs also described in this report.

Upon completion the program returns to the program "transform" and the user is asked to choose another transform.

2.2.9. The Information Files

It is particularly useful to the user to have information available upon entering the seven programs. To this end an information file is accessible to the user for each program.

These information files explain by the quickest possible means the rudiments of the current program. These files are

in the same directory as the execution file. They are:

- a) TRANSFORM.info
- b) FILTER.info
- c) INFILL.info
- d) REDUCE.info
- e) ENLARGE.info
- f) GRAYS.info
- g) SIL.info

2.2.10 Linkit and Runit

These two programs link and load the programs (described in Sections 2.2.1. to 2.2.8. above) to create an execution file "TRANSFORM".

The programs are linked together with the text "vidig" (L. (linked directly from another directory) and editions of Ramlib, Iolib, Implib, Utilib, and PL1RTL.

The execution file resides in core from 2C00 to 9A03.

3.0. CONCLUSION AND FUTURE DEVELOPMENTS

The author feels that the package of programs described and

illustrated in this report go a long way to fulfilling the AMG's need for an interactive, flexible, and extendable image processing system. However, there is always room for development, and this will be discussed below.

The author's programs often appear to be very short. This lends support to the truth that quality may not always be measured in terms of quantity. Most of the programs are kept as short as possible to alleviate problems in debugging; other programs appear to be short but in reality involve nested loops responsible for many hundreds of thousands of calculations.

Perhaps it is in this concept of large profits from small investments that image processing has gained such a reputation in digital, computer-based application in the "real world." Relatively short, modest programs are capable of performing impressive processing operations (with equally impressive amounts of data). Recent hardware evolution has allowed the user (at a price) to perform the processes described in this report at real-time rates (faster than the $\frac{1}{30}$ second television refresh period).

The programs have been designed in order to:

- a) allow complete interaction between user and system;
- b) allow the user to define personal procedures and combinations of transforms;

c) allow the user to include new programs simply.

Throughout the description of the programs, the author has attempted as little as possible to give subjective comment. The needs of the user have been uppermost in the requirements of the system's design. It is the subjective decision of the user that is seen to be important in the ultimate use of the system.

The author sees future developments to the system principally in the field of hardware. If the AMG (or any serious research group) is intent in pursuing image processing (or a sub-set of the field), or intends to implement an image processing facility permanently, the author feels that recourse to special hardware is the only reasonable action. With processing times greater than two minutes for each transformation, software procedures offer inefficient methods of completing transforms. In the same period hardware processors can achieve hundreds of transforms.

The future of image processing generally, however, is much less in doubt. Many of the "birth pangs" of the technology have now subsided. Many thousands of systems are performing various tasks considered to be digital image processing, and users are constantly invoking new and more sophisticated applications. The overwhelming fact of the field is that there is constantly large amounts of data that must be sensed, generated, processed, stored,

and displayed. The fertile area of data compression techniques has, to some extent, been approached in this report, in order to substantially reduce the size and extent of stored data. The specific benefit to the SDMS is not unique; data compression is also particularly rewarding in the areas of document retrieval and storage. A digitized document may be stored in a "compressed" form, retrieved and enlarged using present digital techniques.

Finally, the author would lay the strongest possible emphasis on the value of user-experience in the operation of this package of programs. It is impossible for the author to describe and illustrate all the combinations of the transforms obtainable. However, the author feels that a majority of the more useful combinations have been explored and that the user may have an insight into the capabilities of this package.

It is for the user to become acquainted with the delicate intricacies of the various programs to obtain the maximum possible value from this sub-set of the field of digital image processing.

REFERENCES

1. Parts of this section are paraphrased from Dr. R. A. Bolt's interim report to DARPA concerning the present state of the SDMS research under the same sponsor and contract as this report.
2. For discussions of the techniques and psychology of graphic symbolism see:

Bowman, W.J. Graphic Communication, New York: Wiley Press, 1968;
Garrett, L. Visual Design, New York: Krieger Publ., 1975;
Karo, J. Graphic Design, New York: van Nostrand Reinhold Publ., 1975.
3. For a more detailed description, see Sierra Scientific's User Manual.
4. See Bibliography under Cornsweet, T.N. and Duda, R.O. et al.

APPENDIX A
Programer's Manual

File name	Called From	Other Subroutine Called
AFILL	INFILL	HREAD HWRITE
AREDUCE	REDUCE	HREAD HWRITE
COLOR	GRAYS	
DEFCCM	GRAYS	
DIGITIZE	TRANSFORM	VIDIG
ENLARGE	TRANSFORM	FPRINT HREAD HWRITE SMTHX
ENLARGE.info	FPRINT	
FFILTER	FILTER	
FILTER	TRANSFORM	FFILTER FPRINT HREAD HWRITE
FILTER.info	FPRINT	

File Name	Called From	Other Subroutine Called
FPRINT	ENLARGE FILTER GRAYS INFILL REDUCE SIL TRANSFORM	ENLARGE.info FILTER.info GRAYS.info INFILL.info REDUCE.info SIL.info TRANSFORM.info
GRAYS	TRANSFORM	COLOR DEFCCM FPRINT MATRIX
GRAYS.info	FPRINT	
HREAD	AFILL AREDUCE ENLARGE FILTER SFILL SIL SMTHX SREDUCE	RAMTRD RAMTWT
HWRITE	AFILL AREDUCE ENLARGE FILTER SFILL SIL SMTHX SREDUCE	RAMTWT
INFILL	TRANSFORM	AFILL FPRINT SFILL
INFILL.info	FPRINT	

File Name	Called By	Other Subroutine Called
MATRIX	GRAYS TRANSFORM	NCOLOR NEWCCM
NCOLOR	MATRIX	
NEWCCM	MATRIX	
RAMSET	TRANSFORM	RAMTWT
RAMTRD	HREAD VREAD	
RAMTWT	HREAD HWRITE RAMSET VREAD VWRITE	
REDUCE	TRANSFORM	AREDUCE FPRINT SREDUCE
REDUCE.info	FPRINT	
SFILL	INFILL	HREAD HWRITE
SIL	TRANSFORM	FPRINT HREAD HWRITE
SIL.info	FPRINT	

File Name	Called By	Other Subroutine Called
SMTHX	ENLARGE	HREAD HWRITE SMTHY
SMTHY	SMTHX	RAMTWT VREAD VWRITE
SREDUCE	REDUCE	HREAD HWRITE
TRANSFORM		DIGITIZE ENLARGE FILTER FPRINT GRAYS INFILL MATRIX RAMSET REDUCE SIL
TRANSFORM.info	FPRINT	
VREAD	SMTHY	RAMTRD RAMTWT
VWRITE	SMTHY	RAMTWT

AFILL

This program infills the current image in the Ramtek display by averaging pixel values.

The program procedes through the image, line by line.

A user-defined "pixel infill number" is used to fill that number of pixels along each line with the average value of those pixels.

The Imlac display informs the user that infilling by averaging with the user-defined infill number is in progress.

The program returns to the second level INFILL program.

```
dcl AFILL entry;  
call AFILL;
```

AREDUCE

This program reduces the current image in the Ramtek display by averaging pixel values.

A user-defined reduction ratio operates on both axes of the image.

In order not to violate unprocessed portions of the original image, the reduced image is centered on the screen and is created outwards in both vertical directions at once.

The Imlac display informs the user that reduction by averaging with the user-defined ratio is in progress.

The program returns to the second level REDUCE program.

```
dcl AREDUCE entry;  
call AREDUCE;
```

DIGITIZE

This program digitizes a photographic image through the Vidicon camera into the Ramtek display.

The Imlac display informs the user that digitizing is in progress.

The user is given the option of re-digitizing or returning to the first level TRANSFORM program.

```
dcl DIGITIZE entry;  
call DIGITIZE;
```

ENLARGE

This program enlarges the current image in the Ramtek display.

A user-defined enlargement ratio operates on both axes of the current image.

An information file is available to the user.

In order not to violate unprocessed portions of the original image, the enlarged image is based on the center of the screen and is created from the top and bottom of the screen and towards the center simultaneously.

The Imlac display informs the user that enlarging with the user-defined ratio is in progress.

The program returns to the first level TRANSFORM program.

```
dcl ENLARGE entry;  
call ENLARGE;
```

FFILTER

This program calculates the pixel values of the output-line array for the FILTER program.

The program is passed the three input-line arrays, the output-line array and the coefficients structure (the 3 x 3 kernel, normalization and offset variables).

Values calculated beyond the matrix limits are re-valued at those limits.

The program returns the output-line array to the second level FILTER program.

```
dcl FFILTER entry ( , , , , );  
call FFILTER (line1,line2,line3,output,coefficients);
```

FILTER

This program convolves a 3 x 3, 2-dimensional pixel kernel.

It is a command program, allowing the user to:

- a) see an information file;
- b) set the kernel, normalization and offset variables;
- c) set the normalization variable only;
- d) set the offset variable only;
- e) type the current variables;
- f) exit the program;
- g) initiate the filtering procedure (call FFILTER,etc)

In the case of (g) above, the Imlac display informs the user that filtering is in progress.

A break handler is included.

The program returns to the first level TRANSFORM program.

```
dcl FILTER entry;  
call FILTER;
```

GRAYS

This program re-defines the Ramtek display matrix.

The user defines the number of gray levels to be in the matrix in order to simulate n-bits per pixel.

An information file is available to the user.

The Imlac display informs the user that gray-scale matrix change is in progress.

The user is able to return to the original image, and the Imlac display informs that this is in progress.

The program returns to the first level TRANSFORM program.

```
dcl GRAYS entry;  
call GRAYS;
```

HREAD

This program reads a horizontal line (raster) of pixel values from the Ramtek display.

The user calls the program with the line number required and the program returns the 640 pixel values in that line in an input-line array.

```
dcl HREAD entry (fix,bit(16));  
call HREAD (y,input_line);
```

HWRITE

This program writes a horizontal line (raster) of pixel values into the Ramtek display.

The user calls the program with the line number required and the output-line array. The program writes the 640 pixel values into that particular line of the Ramtek display.

```
dcl HWRITE entry (fix,bit(16));  
call HWRITE (y,output_line);
```

INFILL

This program allows the user access to the two INFILL programs AFILL and SFILL.

An information file is available to the user.

The Imlac display informs the user of the choice.

The program returns to the first level TRANSFORM program.

```
dcl INFILL entry;  
call INFILL;
```

MATRIX

This program sets the matrix of the Ramtek display.

256 slots are loaded with a continuous gray-scale from black to white in the locations 0 to 255, 256 to 511, 512 to 767 and 768 to 1023.

```
dcl MATRIX entry;  
call MATRIX;
```

RAMSET

This program sets the Ramtek microprocessor environment.

The environment is set with the origin in the Ramtek's top, left-hand corner at 0,0 and the end in the Ramtek's bottom, right-hand corner at 640,480. The scan variable is set to left-to-right, top-to-bottom (0).

```
dcl RAMSET entry;  
call RAMSET;
```

REDUCE

This program allows the user access to the two REDUCE programs AREDUCE and SREDUCE.

An information file is available to the user.

The Imlac display informs the user of the choice.

The program returns to the first level TRANSFORM program.

```
dcl REDUCE entry;  
call REDUCE;
```

SFILL

This program infills the current image in the Ramtek display by skipping pixels.

The program proceeds through the image, line by line.

A user-defined "pixel infill number" is used to fill that number of pixels along each line by skipping pixels and loading the intermediate pixels with the skip-count pixel value.

The Imlac display informs the user that filling by skipping is in progress with the user-defined infill number.

The program returns to the second level INFILL program.

```
dcl SFILL entry;  
call SFILL;
```

SIL

This program operates a determinant function on a 2 x 2, 2-dimensional pixel kernel.

An information file is available to the user.

The user defines a threshold value and this is compared

to the determinant value.

The Imlac display informs the user that silhouetting with the user-defined threshold value is in progress.

The program returns to the first level TRANSFORM program.

```
dcl SIL entry;  
call SIL;
```

SMTHX

This program compliments the ENLARGE program; it smoothes the current image in the Ramtek display horizontally.

The program is called by ENLARGE and is passed the enlargement ratio.

The Imlac display informs the user that horizontal smoothing is in progress.

The program returns to the second level ENLARGE program.

```
dcl SMTHX entry (fix);  
call SMTHX (ratio);
```

SMTHY

This program compliments the ENLARGE and SMTHX programs; it smoothes the current image in the Ramtek display vertically.

The program is called by SMTHX and is passed the enlargement ratio.

The Imlac display informs the user that vertical smoothing is in progress.

The program returns to the third level SMTHX program.

```
dcl SMTHY entry (fix);  
call SMTHY (ratio);
```

SREDUCE

This program reduces the current image in the Ramtek display by skipping pixel values.

A user-defined reduction ratio operates on both axes of the current image.

In order not to violate unprocessed portions of the original image, the reduced image is centered in the screen and is created from the center outwards in both vertical directions simultaneously.

The Imlac display informs the user that reduction by skipping with the user-defined ratio is in progress.

The program returns to the second level REDUCE program.

```
dcl SREDUCE entry;  
call SREDUCE;
```

TRANSFORM

This program allows the user access to the second level programs DIGITIZE, FILTER, INFILL, REDUCE, ENLARGE, GRAYS and SIL.

An information file is available to the user.

The Imlac display informs the user of the choice.

The program allows the user to re-access the second level programs or may exit from the transforms.

TRANSFORM is not declared or called.

VIDIG

This program performs the actual image digitization. The program is called by DIGITIZE and is linked from another directory.

```
dcl VIDIG entry (fix);  
call VIDIG (error);
```

VREAD

This program reads a vertical line of pixel values from the Ramtek display.

The user calls the program with the line number required and the program returns the 480 pixel values in that particular line in an input-line array.

The program assumes a pre-set Ramtek environment.

```
dcl VREAD entry (fix,bit(16));  
call VREAD (x,input_line);
```

VWRITE

This program writes a vertical line of pixel values into the Ramtek display.

The user calls the program with the line number required and the output-line array. The program writes the 480 pixel values into that particular line of the Ramtek display.

```
dcl VWRITE entry (fix,bit(16));  
call VWRITE (x,output_line);
```

APPENDIX B
Index of Illustrations

DIAGRAMS:

Figure	Page	Description
3	18	Existing M.A.G.I.C. System.
4	25	Software Organisation.
21	41	Compass Gradient Filters.
38	54	Comparison of AFILL and SFILL run-times.
43	58	Comparison of AREDUCE and SREDUCE run-times.
50	64	Comparison of ENLARGE run-times with and without smoothing (operation of SMTHX and SMTHY).

PHOTOGRAPHS:

Figure	Page	Description
1	9	Media Room in operation running SDMS.
2	9	Author using hardware described in report.
5	27	Original unprocessed image (people).
6	27	Original unprocessed image (characters).

Figure	Page	Transform	Type	Values
7	33	Filter	Mean	norm=9, offset=32
8	33	Filter	Guassian	norm=16, offset=16
9	35	Filter	Guassian + Laplacian(b)	norm=-1, offset=128
10	35	Grays	Guassian + Laplacian(b) + Grays	grays=2
11	36	Filter	Guassian + Laplacian(c)	norm=1, offset=64
12	36	Grays	Guassian + Laplacian(c) + Grays	grays=2

Filter	Page	Transform	Type	Values
13	37	Filter	Guassian + Guassian + Laplacian(c)	norm=1, offset=64
14	37	Grays	Guassian + Guassian + Laplacian(c) + Grays	norm=1, offset=64 grays=2
15	38	Filter	Guassian + Guassian + Laplacian(c) + Laplacian(c)	norm=1, offset=64 norm=1, offset=64
16	38	Grays	Guassian + Guassian + Laplacian(c) + Laplacian(c) + Grays	norm=1, offset=64 norm=1, offset=64 grays=2
17	39	Filter	Guassian + Laplacian(c) + Guassian	norm=1, offset=64
18	39	Grays	Guassian + Laplacian(c) + Guassian + Grays	norm=1, offset=64 grays=2
19	40	Filter	Guassian + Laplacian(c) + Guassian + Laplacian(c)	norm=1, offset=64 norm=1, offset=64
20	40	Grays	Guassian + Laplacian(c) + Guassian + Laplacian(c) + Grays	norm=1, offset=64 norm=1, offset=64 grays=2

Figure	Page	Transform	Type	Values
22	44	Filter	Prewitt (top=North+West) (bot=North)	norm=1, offset=64
23	44	Grays	Prewitt + Grays	norm=1, offset=64 grays=2
24	45	Filter	Prewitt (top=North+West) (bot=North)	norm=1, offset=128
25	45	Grays	Prewitt + Grays	norm=1, offset=128 grays=2
26	46	Filter	Kirsch (top=North+West) (bot=North)	norm=1, offset=64
27	46	Grays	Kirsch + Grays	norm=1, offset=64 grays=2
28	47	Filter	Kirsch (top=North+West) (bot=North)	norm=1, offset=128
29	47	Grays	Kirsch + Grays	norm=1, offset=128 grays=2
30	48	Filter	3-level Simple (top=North+West) (bot=North)	norm=1, offset=64
31	48	Grays	3-level Simple + Grays	norm=1, offset=64 grays=2
32	49	Filter	3-level Simple (top=North+West) (bot=North)	norm=1, offset=128
33	49	Grays	3-level Simple + Grays	norm=1, offset=128 grays=2
34	50	Filter	5-level Simple (top=North+West) (bot=North)	norm=1, offset=64

Figure	Page	Transform	Type	Values
35	50	Grays	5-level Simple + Grays	norm=1, offset=64 grays=2
36	51	Filter	5-level Simple (top=North+West) (bot=North)	norm=1, offset=128
37	51	Grays	5-level Simple + Grays	norm=1, offset=128 grays=2
39	55	Infill	Afill	ratio=5
40	55	Infill	Sfill	ratio=5
41	56	Infill	Afill	ratio=20
42	56	Infill	Sfill	ratio=20
44	59	Reduce	Areduce	ratio=4
45	59	Reduce	Sreduce	ratio=4
46	60	Reduce	Areduce	ratio=8
47	60	Reduce	Sreduce	ratio=8
48	62	Reduce	Areduce (enlarged photographically)	ratio=10
49	62	Reduce	Sreduce (enlarged photographically)	ratio=10
51	65	Enlarge	Enlarge + Smthx + Smthy	ratio=4
52	65	Enlarge	Enlarge + Smthx + Smthy	ratio=8
53	69	Grays		grays=128
54	69	Grays		grays=16
55	70	Grays		grays=4
56	70	Grays		grays=2
57	73	Silhouette		threshold=500
58	73	Silhouette		threshold=1000
59	74	Silhouette		threshold=2000
60	74	Silhouette		threshold=3000

BIBLIOGRAPHY

Andrews, H.C. Computer Techniques in Image Processing, New York: Academic Press, 1970.

Introduction to Mathematical Techniques in Pattern Recognition, New York: Wiley Press, 1972.

Image Processing Application Note, Los Angeles: USC, 1977.

Andrews, H.C. and Hunt, B.R. Digital Image Restoration, New York: Wiley Press, 1973.

Bibermann, L.M. and Nudelman, S. Editors Photoelectronic Imaging Devices, Vol.2 Devices and their Evaluation, New York: Plenum Press, 1971.

Bolt, R.A. Spatial Data Management - Interim Report, report to DARPA, contract no. MDA903-77-C-0037; MIT, Nov. 1977.

Brock, G.C. Image Evaluation - An Appraisal of Current Techniques, London: Focal Press, 1970.

Cornsweet, T.N. Visual Perception, New York: Academic Press, 1971.

Duda, R.O. and Hart, P.E. Pattern Classification and Scene Analysis, New York: Wiley Press, 1973.

Freeman, H. Picture Processing and Psychopictorics, New York: Academic Press, 1970.

Huang, T.S. Editor Picture Processing and Digital Filtering, Berlin and New York: Springer-Verlag, 1975.

Kirsch, R. Computer Biomed. Res. Vol.4 Computer Determination of the Constituent Structure of Biological Images, 1971.

Kowaliski, P. Applied Photoelectronic Theory, London: Wiley Press, 1973.

Morrell, A.M., Law, H.B., Ramberg, E.G., and Herold, E.W. Color Television Picture Tubes, Supplement 1, Advances in Image Pickup and Display, New York: Academic Press, 1974.

Patterson, C.L. Editor, Computer, P8-69, Vol.10, No.8, Aug. 1977.

Pratt, W.K. Digital Image Processing, New York: Wiley Press, 1970.

RCA Astro-Electronics Div., Electronic Photointerpritation Study - Final Report, Vol.1, Sensor and Display Analysis, Princeton: 1975.

Robinson, G.S. Detection and Coding of Edges Using Directional Masks, proceedings of the SPIE Conference on Advances in Image Transmission Techniques, P117-125, San Diego: Aug. 24-25, 1976.

Edge Detection by Compass Gradient Masks, Computer Graphics and Image Processing, Vol.6, P492-501, New York: Academic Press, 1977.

Rosenfeld, A. Picture Processing by Computer, New York; Wiley Press, 1969.

Wilson, R.M., Teuber, P.L., Watkins, J.R., Thomas, D.T. and
Cooper, C.M. An Image Data Processing System for Solar
Astronomy, Appl. Opt., Vol.16, P944-949, 1977.

Wyszecki, G. and Stiles, W.S. Color Science - Concepts and
Methods, Quantitative Data and Formulas, New York: Wiley
Press, 1967.