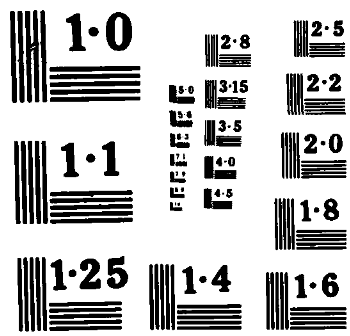
AD-A155 531 ANALYSIS TECHNIQUES FOR MICROWAVE DOSIMETRIC DATA(U) 1/1
TECHNOLOGY-USA INC OXON HILL MD M J CAMPBELL ET AL.
JUL 81 DAMD17-79-C-9151

UNCLASSIFIED F/G 6/18 NL

END
FILMED
DTIC



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AD-A155 531

AD____________

ANALYSIS TECHNIQUES FOR
MICROWAVE DOSIMETRIC DATA

ANNUAL REPORT

By
M.J. CAMPBELL, T.E.GOFF, and V.L. KALB

24 JULY 1981

Supported By
U.S.ARMY MEDICAL RESEARCH AND DEVELOPMENT COMMAND
Fort Detrick, Frederick, Maryland 21701

Contract No. DAMD17-79-C-9151

Technology-USA,Inc.
P.O. Box 55333 Oxon Hill, Maryland 20022

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RESEARCH IN ANALYSIS TECHNIQUES FOR MICROWAVE DOSIMETRIC DATA		5. TYPE OF REPORT & PERIOD COVERED Annual Report Oct 79 - June 81
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) M.J. Campbell, T.E. Goff and V.L. Kalb		8. CONTRACT OR GRANT NUMBER(s) DAMD17-79-C-9151
9. PERFORMING ORGANIZATION NAME AND ADDRESS Technology-USA Inc. Oxon Hill, MD 20022		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62771A.3E162771A805.00.011
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Medical Research and Development Command Fort Detrick Frederick, MD 21701		12. REPORT DATE July 1981
		13. NUMBER OF PAGES 43
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, Distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image Processing
Cubic Spline Interpolation
Two Segment Enhancement Curve

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A/1	

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

The overall objective of this research investigation is to develop a methodology for displaying microwave dosimetric data. Special emphasis was placed on developing a method for evaluating the spatial variation of absorbed microwave energy by specific organs and organ subdivisions. Techniques were developed by a computer and displayed on an image in pseudocolor. Manipulation of the data using the image system allows the operator to highlight organ subdivisions within

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Block 20 Continuation)

an image. A cubic spline interpolation program was developed to uniformly expand the data set for presentation on a large format display. A two segment enhancement curve was developed which gives the operator the capability of creating his own lookup table for pseudocolor translation of images stored in refresh memory. The results to date show that pseudocolor imaging is an effective method of analyzing microwave dosimetric data. Additional processing algorithms need to be developed and the system hardware should be optimized for image processing.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

The overall objective of this research investigation is to develop a methodology for displaying microwave dosimetric data. Special emphasis was placed on developing a method for evaluating the spatial variation of absorbed microwave energy by specific organs and organ subdivisions. Techniques were developed by a computer and displayed on an image in pseudocolor. Manipulation of the data using the image display system allows the operator to highlight organ subdivisions within an image. A cubic spline interpolation program was developed to uniformly expand the data set for presentation on a large format display. A two segment enhancement curve was developed which gives the operator the capability of creating his own lookup table for pseudocolor translation of images stored in refresh memory. The results to date show that pseudocolor imaging is an effective method of analyzing microwave dosimetric data. Additional processing algorithms need to be developed and the system hardware should be optimized for image processing.

SECTION I

Introduction

The research described in this Annual Report is the result of efforts performed during the first year of this contract. As a result of an inordinate delay in obtaining necessary hardware for this project, it was necessary to extend the time for performance of the contract beyond the original one year. The overall objectives of this research investigation is to develop a methodology for displaying microwave dosimetric data on a CRT type display for use in data interpretation and analysis. A color image processing system was procured and interfaced to the laboratory Hewlett-Packard computer system. Data obtained from the microwave scanner is processed by the computer and then displayed on the image system. Using the image system, the experimenter can manipulate an image to enhance areas of interest for better analysis. Shortly after this system was placed in operation, it became apparent that viewing the data set in a one-to-one aspect ratio produced an image which was too small for detailed analysis. A cubic spline interpolation program was developed to uniformly expand the image to a larger format. Also, a two segment enhancement curve was developed to allow the operator to define his own video lookup table for pseudocolor translation of images stored in refresh memory.

SECTION II

DEVELOPMENT OF DATA ANALYSIS SYSTEM

This section of the report provides an overview of the display system and a description of its application to data analysis in pictorial format. The operation of the image display system is straight forward and does not involve any computer programming on the part of the user. A brief description of the commands used by the operator to interact with the system is presented.

A. Selection of a Display Media

The objective of this phase was to select a method for presenting experimental results in a manner that maximizes the information content and interpretability of the data. To date, the data display has primarily involved an isometric presentation that represents energy absorption or phase shift as a height above a planar surface whose perspective can be varied. The major disadvantage of this approach is that the aerial perspective contains no information. It was felt that an aerial perspective was critical to the successful analysis of the data because an aerial perspective would allow a determination of the spatial relation between the organ subdivisions.

The data obtained from the microwave scanner system is in a digital format. Each data point represents the energy absorption of a specific physical location on the organ. Using the co-ordinate information associated with each intensity value, the data can be assembled to form a mosiac "picture" of the energy absorption of the organ. This process is identical to that employed in reconstructing an image with data from a satellite. Hence, a literature search of commercial image processing systems was undertaken. Systems from the following venders were considered:

- Ramtek Corporation, Sunnyvale CA
- Lexidata Corporation, Burlington MA
- Grinnel Systems, San Jose CA
- Comtal Corporation, Altadena CA

The Ramtek model 9351 Image Display System selected for this project for the following reasons:

- The modular architecture of the 9351 allowed the selection of a performance capability desired for this application. Also, field expansion or modification of the system at a later date would be easy to accomplish.

- The display refresh memory stores 16 bits per picture element.

- The function table provided for programmable definition of output intensity and color.

- The 9351 was on GSA schedule so it could be purchased at a substantial discount.

- Local service was available from the Ramtek office in Silver Spring, Maryland.

- An existing software package used for satellite image processing was available for the Ramtek which provided much of the processing necessary for display of the microwave data.

A price quotation was obtained from Ramtek to establish the price of the system. A letter proposal was then submitted to the sponsor recommending the Ramtek equipment and requesting additional funds for its procurement. While our proposal was being evaluated, two events occured which were to impact this project. First, Ramtek instituted a 10 percent price increase on all their equipment and second, the GSA contract expired. Thus, when we received authorization to proceed with acquisition of the equipment, we were facing a 25 percent over-run in material cost. Over the next few weeks, we proceeded to negociate an extension of the original price quotation from Ramtek in exchange for an extended delivery schedule.

Having defined the image processing system, the next task was to provide a hard copy capability. The most common method of making a hard copy at this time was to take a photogtaph of the color monitor. However, this required the services of a photographer and a darkroom capable of processing color prints. Not only was this a costly process but it was not well suited to a research environment where one is not able to predict when a hardcopy of the data will be required. In our search for an alternative, we contacted the Polaroid Corporation and were informed that they did not have a camera which would record the image directly. However, we were informed that two companies

. Dunn Instruments, San Francisco CA
. Matrix Instruments, Milan MI

made comeras for use with CAT scanners and used a newly developed 8½ by 11 Polaroid film. Subsequently, we were able to arrange for a demo of both camera systems. Each camera was fed an RGB video signal from the Ramtek and a number of photographs were taken.

The Matrix camera seemed more reliable and produced a slightly superior photograph. The Matrix camera also had a unique feature which allowed up to 16 images to be recorded on one 8½ by 11 sheet of film by reducing the size of each image. The capability of producing multiple images on one sheet of film was an extra cost option. Each time the number of images was doubled, the price increased by $2000.00. In discussions with the technical officer, it was decided that two images per sheet of film was the optimum in price versus capability consistent with their needs at that time. The camera could be upgraded later by field modification if the need arose.

On March 19, 1980, a purchase order was issued to Matrix Corporation for a model 2001 camera system for delivery in 90 days. The first delivery date was not met and we were informed that production difficulties have caused the delivery schedule to slip. Many phone calls later, we were informed that Matrix had stopped production of the 2001 camera and was in the process of redesigning the entire system. The new camera, called a model 4007, was to incorporate all the optional features of the 2000 series camera plus have increased reliability and ease of operation. If we would agree to a six month delivery, Matrix proposed to deliver the new model for the same price as the model 2001. In discussions with the technical officer, it was decided that the six month delay was accptable in return for a camera system with much improved performance. As a result of this change, the government received a camera system worth $4000 more that the price they were charged. Unfortunately, the schedule on the new camera also slipped and it was not delivered until April 3, 1981. It should be noted that the Matrix camera has lived up to all the expectations and is considered well worth the extended time necessary to obtain it.

B. System Description

Figure 1 is a detailed block diagram of the RM-9351 System. The various subsystems are briefly described in the following paragraphs.

Host Parallel Link - The host parallel link provides the high speed (up to 600 KHz) bidirectional, 16-bit parallel communications between the host computer and the display system. Four external interrupts are implemented. The computer uses direct memory access (DMA) to communicate with the Ramtek GPIF, which is typically contained within the control/video board.

Internal Processor Bus - The internal processor bus connects the host parallel link, Z80 microprocessor, registers, memories, video generators, and interfaces for optional peripheral equipment. This bus provides high-speed device-to-device communication and implements nonprocessor DMA.

Display Address and Data Registers - Display address and data

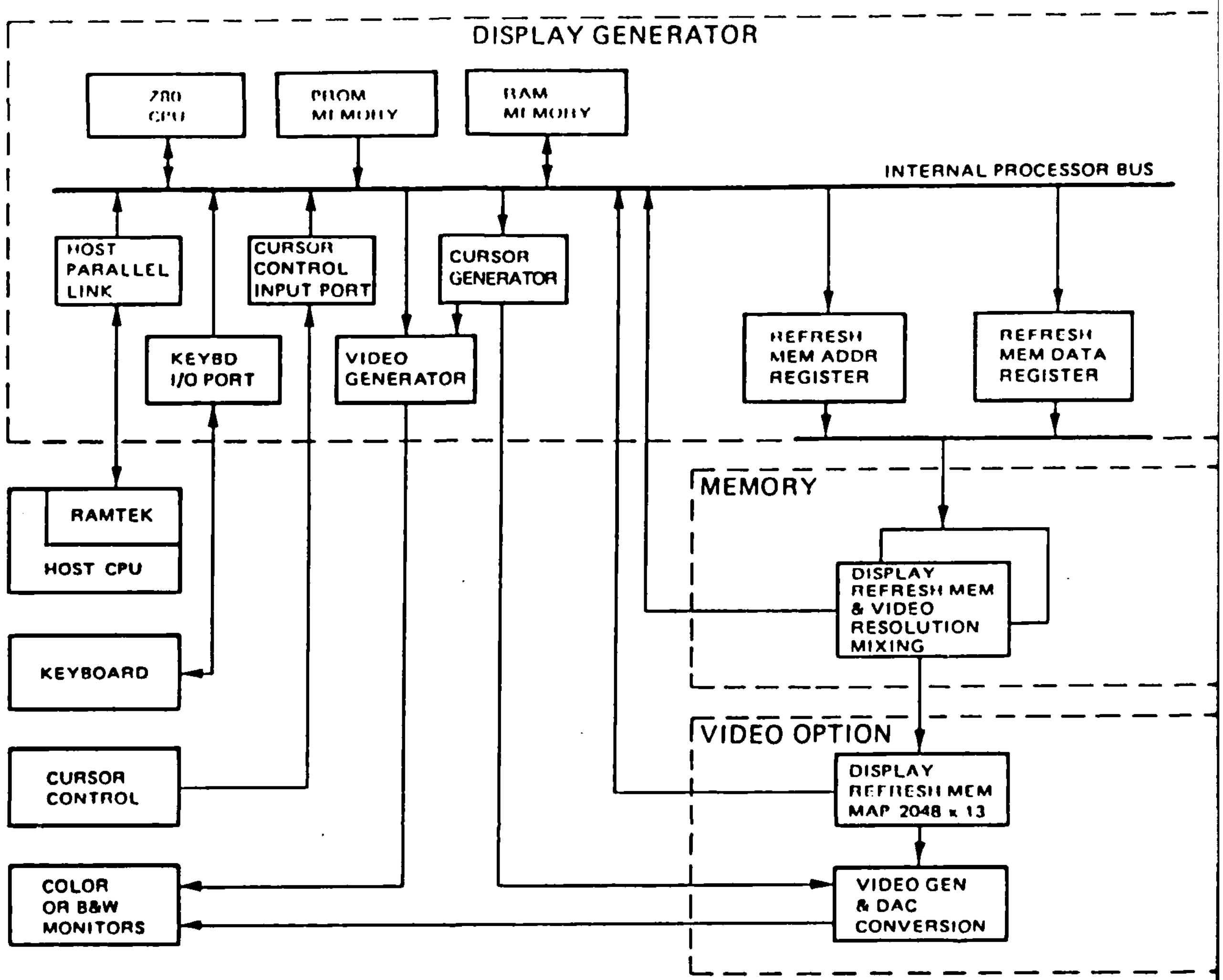


FIGURE 1

RM-9351 Image Display System

registers connect the bus and refresh memory for image generation and retrieval. This communication is in the form of a 16-bit word per pixel, with up to 16 bits being written in the z-axis. The X- and Y- addresses of the pixels are incremented as prescribed by the control registers and logic in the display generator. The display processor interface is optimized with respect to its internal algorithms for generating character and graphics data.

Graphic Display System - Graphics data is written under the direction of the current operating point (X-Y address), that is, up or down and/or left ot right. The color or intensity of font, raster, and graphics data (Z-axis) is assigned by the Z80 microprocessor. Foreground, background, reversal, and writing mode (replacement versus additive) may be specified. The Z80 microprocessor is a powerful tool for implementing not only these modes of writing but any other application-dependent mode.

Video Monitor - The video monitor decodes the generated video signal and displays the image by driving one or more cathode-ray beams in raster fashion. Tube refresh time is at 30-Hz for momestic systems.

Display Processor - This display processor interprets display-instruction information and presides over the bus. Secondary functions include character generation, vector generation, plot generation, raster mode, and raster margining. A Z80 microprocessor with 1,024 bytes random-access-memory (RAM) and 5,120 bytes of PROM is included. A basic instruction set is implemented, which provides imaging, graphics, and text-generating functions.

Refresh Memory - The refresh memory provides sixteen bits of storage for refreshing each pixel on the CRT. The refresh memory contains two memory boards, each board containing eight sections of memory. Each section contains sixteen 16K MOS RAMs that offer 512-line by 512-element refresh capability in the RM-9351 model. One plane of memory stores one bit of storage per pixel. For example, a refresh memory board in the RM-9351 contains sixteen sections of memory, i.e., one plane per section. Thus, the RM-9351 offers 16 planes or 16 bits of storage per pixel.

Video Generation - The video generation section of the control board can be modified through PROM coding to yield a multitude of possible configurations. Figure 2 shows the video generation network contained on the control/video board. By utilizing only the basic control/video board and one eight-plane memory board, the user can specify the PROM code so that the basic four video amps provide him with any of the following systems:

a. Four planes (plane 0 through plane 3) patched to four video amps to yield four black-and-white displays. This is the standard configuration.

b. Six planes patched to three video amps (plane 0

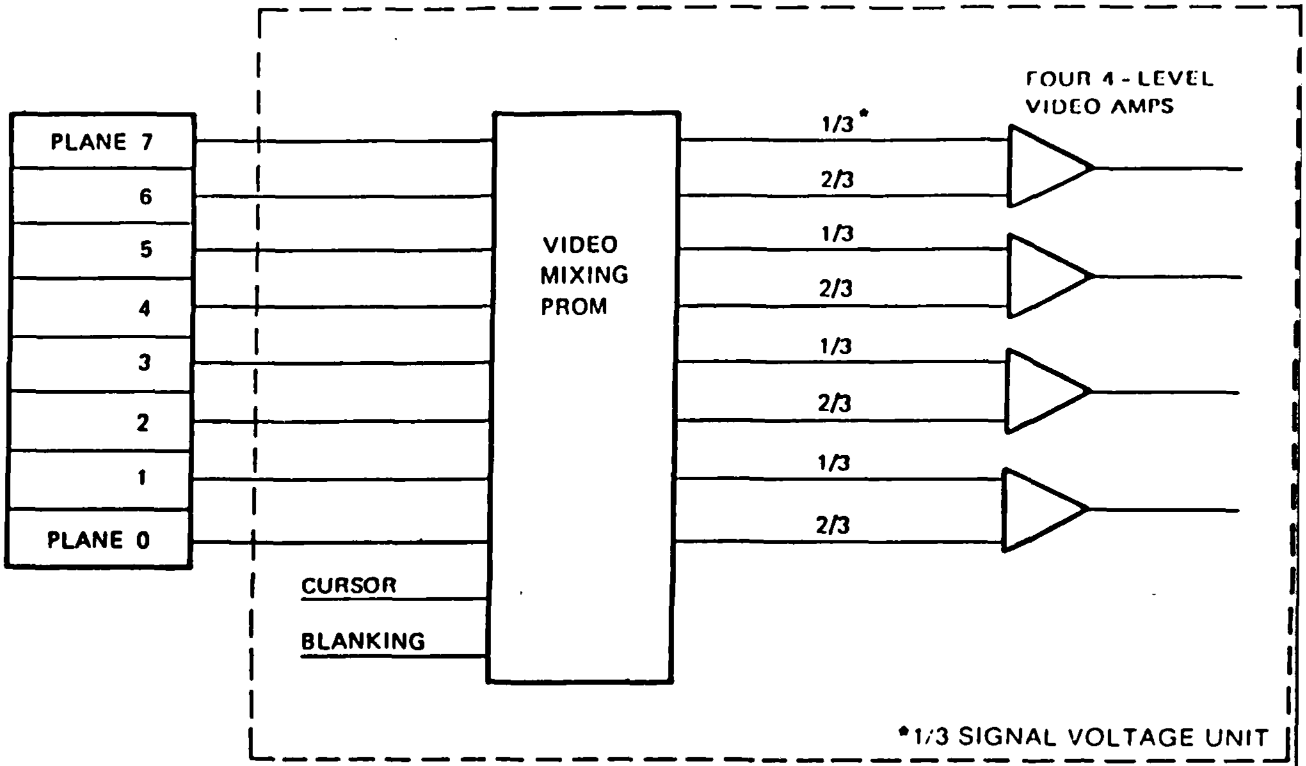


FIGURE 2

RM-9351 Control/Video Section

through plane 5) to yield 64 colors to drive an RGB monitor. The remaining video can be used to drive a composite signal of planes 0 through 5 to a B-W monitor which can provide four gray-scale levels.

C. Eight planes patched to provide 64 colors on an RGB monitor (planes 0 through 5) with two planes provided to yield two overlays (planes 6 and 7).

The remaining video amp may drive, using planes 0 through 7, a composite black-and-white picture.

Enhancement tables are video lookup tables that allow interactive pseudocolor or gray-scale translation of images stored in refresh memory. True color results when a color monitor is driven directly by memory data; pseudocolor results when memory data is manipulated with a RAM before being sent to a color monitor. When equiped with a lookup table, the stored refresh data is treated as an address to the lookup table, which is host programmable; that is, as each pixel is scanned from the refresh memory for video presentation, the contents of the corresponding cell is retrieved from the lookup table and this data is passed to the digital-to-analog converters and video amplifiers, instead of the refresh data itself. Thus, the refresh data addresses a host programmable lookup table that assigns the output intensity or color.

Lookup tables are most often used in imaging applications. Figure 3 illustrates a commonly used enhancement procedure termed windowing, or density (level) slicing. Here, a specified range or group of contiguous image intensity values are fitted to the available spectrum of output intensity levels. Pixels having values beneath or above the specified range (or window) are translated to black or white, respectively; while pixels within the window are translated to an appropriate gray level. Thus, the observer's attention is focused upon the window, and he can more easily distinguish between what were minor or negligible intensity differences in the original image. More important, the user is able to manipulate this window interactively by reloading the lookup table, and without affecting the image in refresh memory. The integrity of the original image is thus maintained while the visual presentation is varied to suite the needs of the observer.

Figure 3 illustrates the effect of a relatively simple contrast enhancement algorithm. It is important to note that the equipment supports sophisticated algorithms. For example, a gamma corrected output can be achieved by loading a non-linear function into the lookup table, or a pre-defined or computed set of pseudocolors might be assigned to a gray scale image, etc.

The enhancement table consists of 2,048 cells of nine or thirteen bits. Thus, up to 11 bits of image data (in refresh memory)

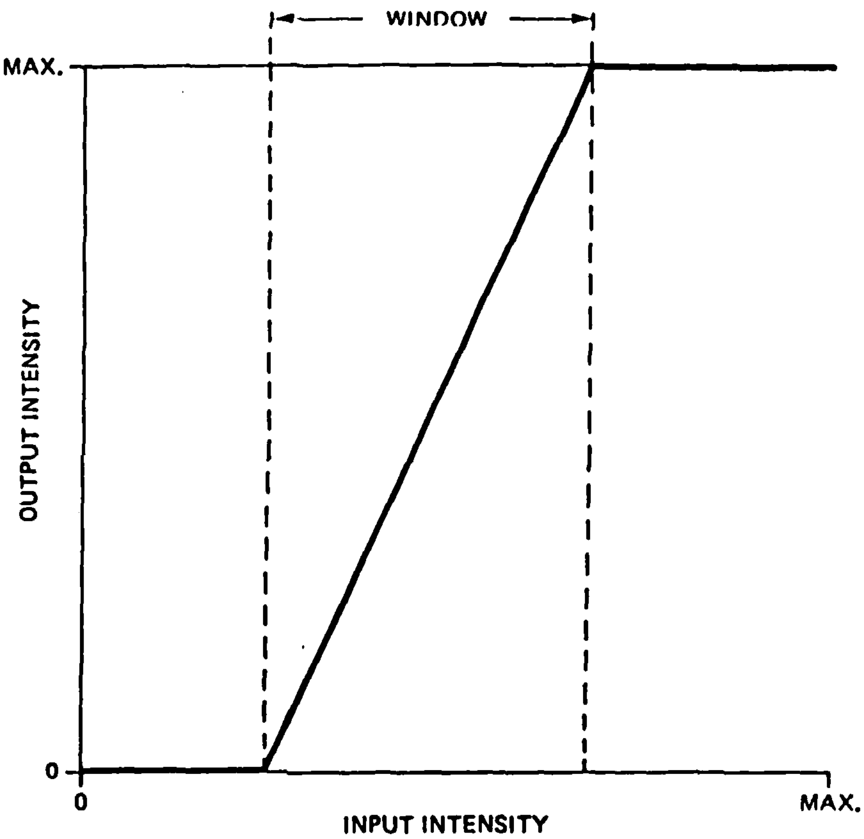


FIGURE 3

Example of Density Slicing

can address the table. For gray scale applications, bit 2_0 through 2_7 of the cells are output to a single 8-digital-to-analog converter (DAC). Thus any of 256 gray level intensities (I) may be produced, i.e., $I = N/255$ where $0 \leq N \leq 255$. For color applications, bits 2_0 through 2_3, 2_4, through 2_7, and 2_8 through 2_{11} are output to three seperate 4-bit DAC's corresponding to the red (R), green (G), and blue (B) primary inputs to an RGB color monitor. Thus, any of 4,096 colors (C) can be produced, i.e., $C = 2_8 \times R + 2_4 \times G + 2_0 \times B$ where $0 \leq R,G,B \leq 15$. Whether gray scale color, bit 2_{12}, when non-zero, causes the corresponding color or intensity to blink at a 2 Hz frequency.

C. Image System Operation

The operation of the image system was designed to provide the user with a very capable system that is simple to operate. All communication between the system and user takes place through the computer CRT console. A menu of the system commands is displayed on the CRT as an aid to the operator. To select a particular function, the operator need only type the first two characters of the particular command. The image system commands are listed below along with a brief description of their function.

STOP, END, OR EXIT

terminates program execution.

INITIALIZE

Program is initialized. All paramters are set to their default values. Next input command may now be chosen from list on CRT.

GRID

This gives the user the capability of varying either the picture or the grid intensity. The program will prompt the user with Input grid in (0 to 1). The user must input the desired grid brightness by punching in a number between 0.0 (black) and 1.0 (white). A number greater than 1.0 will cause the grid to become white and the brightness of the picture will decrease by an inversely proportionate amount. I.e., if a value of two is input the picture will become half as bright while the grid will become fully white. If a value of .5 is input the picture will be at maximum brightness while the grid will become half as bright. To return to ROLOR any of the input commands on the CRT may be selected.

TRANSFER

This command transfers a new picture from the disc to the RAMTEK display. The program will prompt the user with Picture disc Logical Unit. Punch in the LU number and press RETURN key. The program will then prompt "Picture number". Input the number of the picture you want transferred and press RETURN key. The appro- riate picture will be sent to display system. On the CRT the prompt picture nummber is returned in case another picture trans- fer is required. Any ROLOR's commands may also be input to return control to ROLOR and then to go to the specified command.

COLOR

This allows the user to select different color schemes for the pictures. The choice is varied and listed below. The program will prompt the user with choice (A through K, BW, A1 A9 or B1 B9). Choose a color scale (from the list below) and input your choice. The color schme of the picture will change at once. The input prompt will remain in case another color change is required. Any of ROLOR's commands may also be input to return control to ROLOR and then go to the specified command.

Key to color codes:

A- Spectral (ROYGBIV)
B- Half contouring of A
C- Full contouring of A
D- Green, Blue, Black, Green, Yellow, Red, Pink
E- From Bull of Amer Meteorlogical Soc. Vol. 52, #9 (Sept 71)
F- For NIMBUS 5 - ESMR pictures
G- Yellow, Dark Orange, Dark Blue, Light Blue
H- Blue, Green, Violet, Dark Orange White
I- Blue, Green, Yellow, Red, Pink, White
J- Variation of I
K- Pink, Orange, Yellow, Green, Blue
L- Microwave pseudocolor scale
BW- Black and white
A1 through A9- Contour intervals in green

A1 contours one interval (center of scale)
A2 contours two equally spaced intervals, etc.
Position of intervals contoured can be shifted by
changing the color scale limits using the SC command

B1 through B9- Same as A1 through A9, except black and white

BACKGROUND

This changes the background color for the title and color bar. The background color is given by three values, each ranging from 0 to 1, which specify the intensities of red, green and blue respectively in the background.

e.g. 0,0,0 for black background
0,1,0 for green background
.4,0,.6 for purple background

when the program prompts Input red, green, blue (0 tol) values then input your selection and press the RETURN key. The color change is immediate on the RAMTEK display system. The prompt will reappear in case another color change is required. If not, any of ROLOR's input commands (listed on CRT) may be used. Control is returned to ROLOR which then executes the specified command.

BLOW-UP

This allows the user to enlarge a portion of the picture. The program will prompt the operator with "Picture disc Logical Unit." Input LU number of the picture disc. Press the RETURN key. The program will then prompt "Input line #, column #, factor, (Pict #)." The line # and the column # specify the center of the area that is to be enlarged. Values from 1 to 511 may be input for the two parameters. "Factor" is the amount by which the picture is to be enlarged. I.e., a factor of two will make the picture twice as large. The picture # is an optional parameter for the number of picture that is to be enlarged. If not specified it defaults to the value of the current picture. Input your selection and press RETURN key. The prompt specified above will reappear in case another change is needed. If not, any of ROLOR's commands may also be used. That command will then be executed.

DATA or SCALE

This rpvides the user with the ability to change the limits of the data scale and has a kaleidoscope and upkill facility. The user must supply four parameters, spearated by commas, when the program gives the prompt "Input scaling for white 1/black = 0, upkill, klido"

1st para - Upper limit of data scale. If less than 1 the data with values beyond the limit will be displayed as white (or the current background color).

2nd para - Lower limit of data scale. If greater than 0 the data with values below the limit will be displayed as black.

3rd para - Upkill (0 or 1). If the value is 1 data with values above upper limit is displayed as black. If the value is 0 the data is white. It is an optional parameter and defaults to 0.

4th para - Kaleidoscope facility. This causes the picture to cycle through brightness scales automatically. The larger the number the slower the cycling. The length of time for which it runs is fixed and operator must wait for the whole sequence to end before issuing another command.

- Default values are 1,0,0, no kaleidoscope. The third parameter is optional. If not specified it equals 0. The fourth parameter is also optional. If not specified there is no kaleidoscope.

- If the lower limit is greater than the upper limit, the color scales are complimented.
- If an error occurs control is returned to ROLOR and you must specify the next input command from those listed on CRT.

ARCHIVE

This command references another program, program RATPT. It provides the user with the ability of transferring a complete 512 x 512 picture between the disc, the mag tape or the RAMTEK. To initiate a transfer, the operator must select one of the command options displayed on the CRT, type in the two letter mnemonic for the command, and press the RETURN key.

OVERLAY

This command references another program, program OVTPR. It provides the user with the capability of changing the grid and background colors of a picture to suit the picture requirements.

SECTION III

IMAGE PROCESSING PROGRAMS

A. CUBIC SPLINE INTERPOLATION

The microwave scanning system produces a 64 x 64 pixel image. The RAMTEK image system supports a 512 x 512 pixel image, so the raw image would occupy only one eighth of the screen, and little detail would be visible. There are numerous ways to enlarge an image, such as simple pixel replication or bilinear interpolation. While these methods are easier to implement than cubic spline interpolation, they have the disadvantage of degrading fine detail in the image. Thus, it is worth the effort to interpolate with cubic spline functions.

The particular method of cubic spline interpolation which was finally selected for implementation is outlined in Numerical Methods by Robert Hornback. This source describes how to fit a cubic spline to a set of points $(x_i, f(x_i))$. This is done as follows. Between each pair of adjacent points x_i and x_{i+1} it is necessary to find a cubic polynomial which passes through $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$. This polynomial is denoted by $F_i(x) = a_0+a_1x+a_2x^2+a_3x^3$ for $x_i \leq x \leq x_{i+1}$. There are 4 unknown constants $F_i(x_{i+1}) = f(x_{i+1})$. The remaining constraints are imposed by requiring that the first and second derivatives of F_i match those of the polynomial F_{i-1} used on the previous interval. This gives the cubic spline interpolator its characteristic smoothness. To actually calculate the cubics F_i, it is first necessary to calculate the second derivatives $g''(x_i)$ which can be found by solving the following set of simultaneous equations:

$$\left[\frac{\Delta x_{i-1}}{\Delta x_i}\right] g''(x_{i-1}) + \left[\frac{2(x_{i+1} - x_{i-1})}{\Delta x_i}\right] g''(x_i) + g''(x_{i-1})$$

$$+ 6\left[\frac{f(x_{i+1}) - f(x_i)}{(\Delta x_i)^2} - \frac{f(x_i) - f(x_{i-1})}{(\Delta x_i)(\Delta x_{i-1})}\right] \quad (i = 1,2,..,n-1)$$

where $x_i = x_{i+1} - x_i$.

Note that this procedure only n-1 equations in n + 1 unknowns $g''(x_0),\ldots, g''(x_n)$. The 2 additional equations are obtained by specifying conditions on $g''(x_0)$ and $g''x_n)$. For this purpose, take $g''(x_0) = 0$ and $g''(x_n) = 0$.

For image processing, consider the x_i to be at integer values and the functional values to be the pixel intensities. Thus $x_i = 1$, and the equations are simplified. The functions F_i can now be written in terms of g'' and f:

$$F_i(x) = \frac{g''(x_i)}{6}\left[(x_{i+1}-x)^3 - (x_{i+1}-x)\right] + \frac{g''(x_{i+1})}{6}\left[x - x_i)^3 - (x-x_i)\right.$$

$$+ f(x_i)\,(x_{i+1} - x) + f(x_{i+1})\,(x - x_i)$$

As an example, consider the effect of expanding a one dimensional edge which consists of five pixels with intensities 0,0,0, 512,512. In this case, $x_0 = 0,\ldots,x_4 = 4$ and $f(x_0) = 0,\ldots,\ f(x_4) = 512$.

We get the following conditions on g'':

$$\begin{vmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{vmatrix} \begin{vmatrix} g''(1) \\ g''(2) \\ g''(3) \end{vmatrix} = 6 \begin{vmatrix} 0 \\ 512 \\ -512 \end{vmatrix}, \quad g''(0),\ g''(4) = 0$$

hence
$$\begin{vmatrix} g''(1) \\ g''(2) \\ g''(3) \end{vmatrix} = \frac{6}{56} \begin{vmatrix} -5k \\ 12k \\ -19k \end{vmatrix} \qquad \text{where } k = 512$$

and the equations are:

$$F_0(x) = \frac{-5k}{56}\,(x^3 - x)$$

$$F_1(x) = \frac{-5k}{56}\left[(2 - x)^3 - (2 - x)\right] = \frac{12}{56}\left[(x - 1)^3 - (x - 1)\right]$$

$$F_2(x) = \frac{12k}{56}\left[(3 - x)^3 - (3 - x)\right] - \frac{19k}{56}\left[(x - 2)^3 - (x - 2)\right] + k(x - 2)$$

$$F_3(x) = -19k\left[(4 - x)^3 - (4 - x)\right] + k$$

To interpolate one pixel in between each original pixel we must compute $F_0(.5)$, $F_1(1.5)$, $F_2(2.5)$, and $F_3(3.5)$.

$F_0(.5) = .033k = 17$

$F_1(1.5) = -.047k = -24 = 0$ (for image processing)

$F_2(2.5) = .547 = 280$

$F_3(3.5) = 1.127k = 577$

The expanded edge is slightly enhanced and smoothed, as shown in Figure 4.

Since the cubic spline interpolator applys to one dimensional arrays of data and an image is two-dimensional, it is essentially necessary to make two passes with the interpolator. First, treat each column of the image as a set of data and interpolate the required number of pixels in each column. Then, treat each original plus interpolated row of the image as a one dimensional data set and interpolate horizontally.

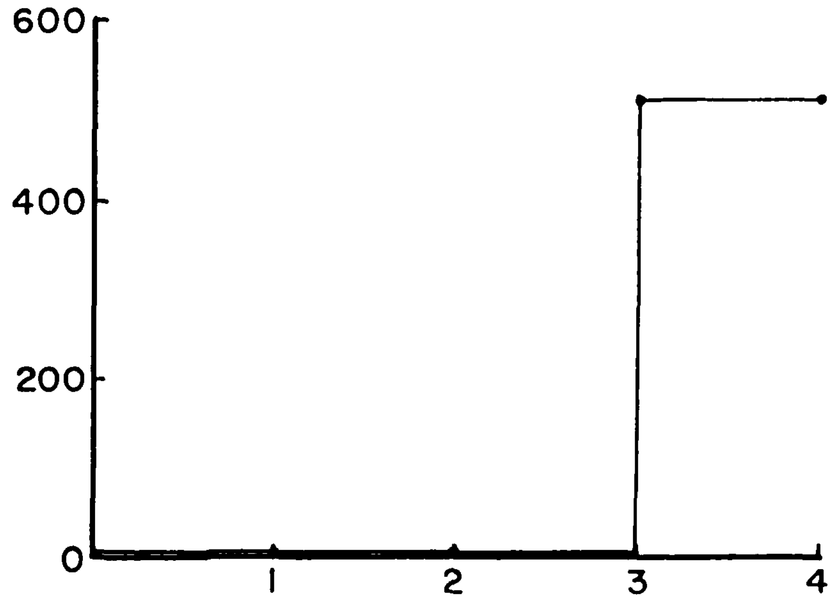
Conceptually, the mthod is shown in Figure 5.

In practice, to conserve memory requirements not all the vertical interpolation is done at once. Instead, all the information needed to construct the interpolating polynomials for each column is computed and stored. Then vertical interpolation is performed on two adjacent column pixels for every column, followed by horizontal interpolation across the original and interpolated rows. Then the data is flushed to the Ramtek and work begins on the next adjacent column pixels.

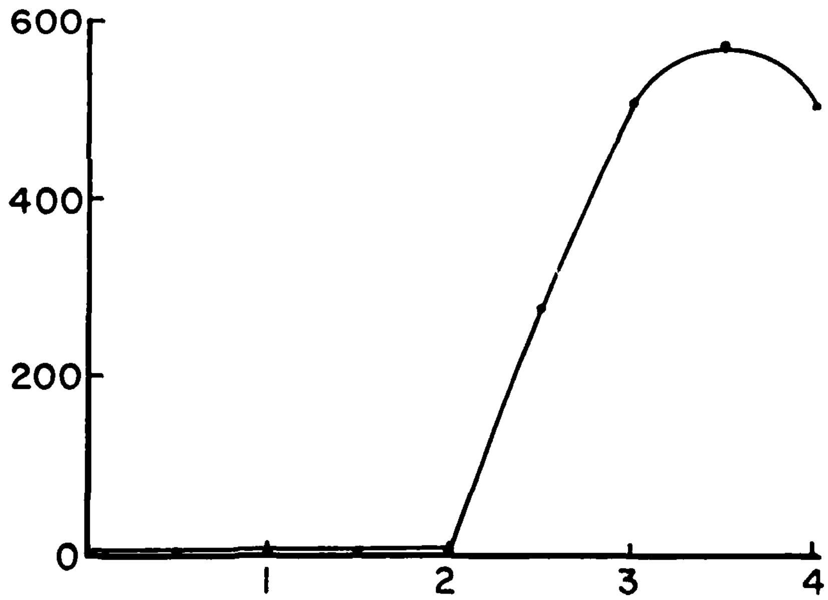
To allow the user maximum flexibility, this implementation of the cubic spline interpolation is not hard coded to any particular expansion factor. Due to screen size limitations it is not proactical to insert more than 7 pixels between the originals, but the program can handle any expansion factor between 1 and 8, inclusive.

B. TWO SEGMENT ENHANCEMENT CURVE

The addition of a two segment enhancement curve to the RAMTEK 9351 system was accomplished by implementing a new subroutine and operator calls to the previously existing RAMTEK software package. The implementation allows a choice of single line or dual line enhancement curves by appending additional information to the enhancement operator query. Because the enhancement curve must be single valued, various tests are included to prevent the generation of meaningless curves. Internally, the software multiplexes the requested color scale and the enhancement curve to produce the video lookup table bit format, which is then transferred to the Image System. The enhancement curve with descriptions is shown in Figure 6.

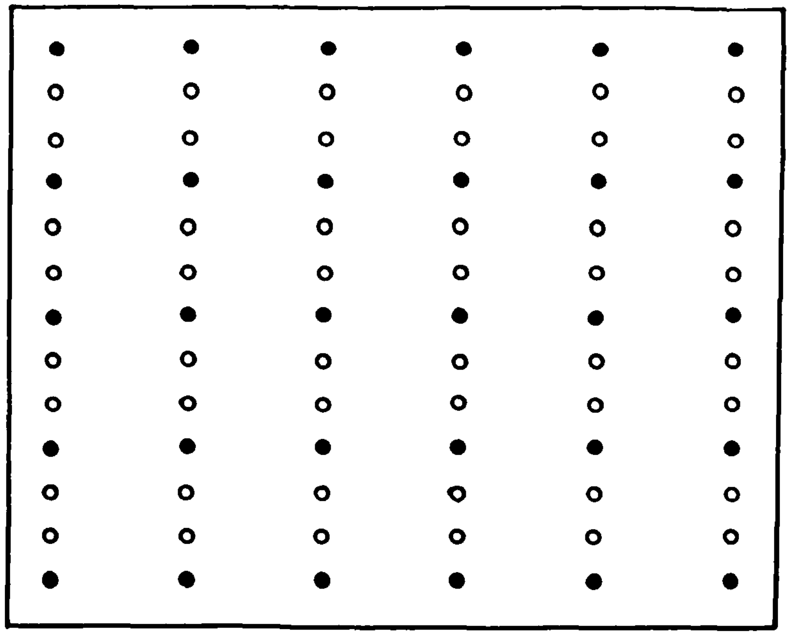


ORIGINAL

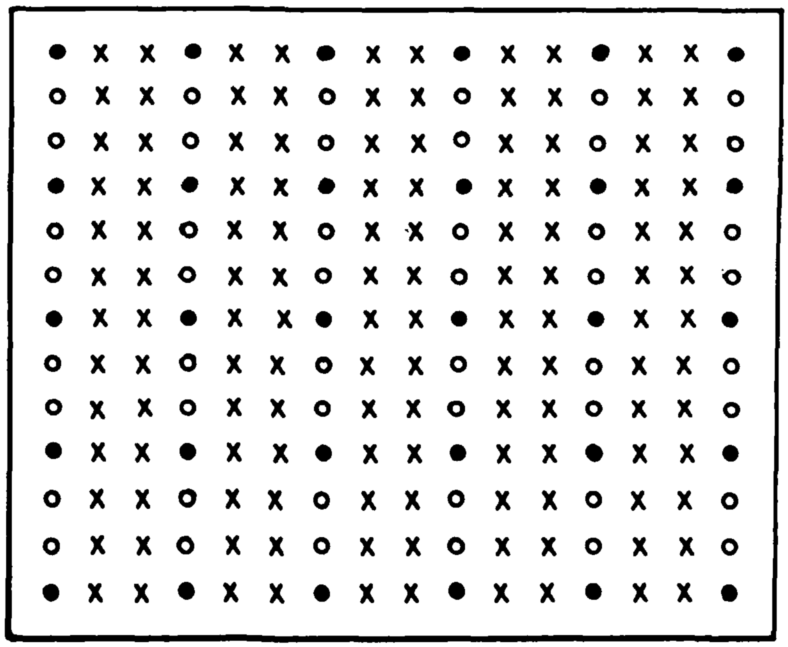


INTERPOLATED

FIGURE 4



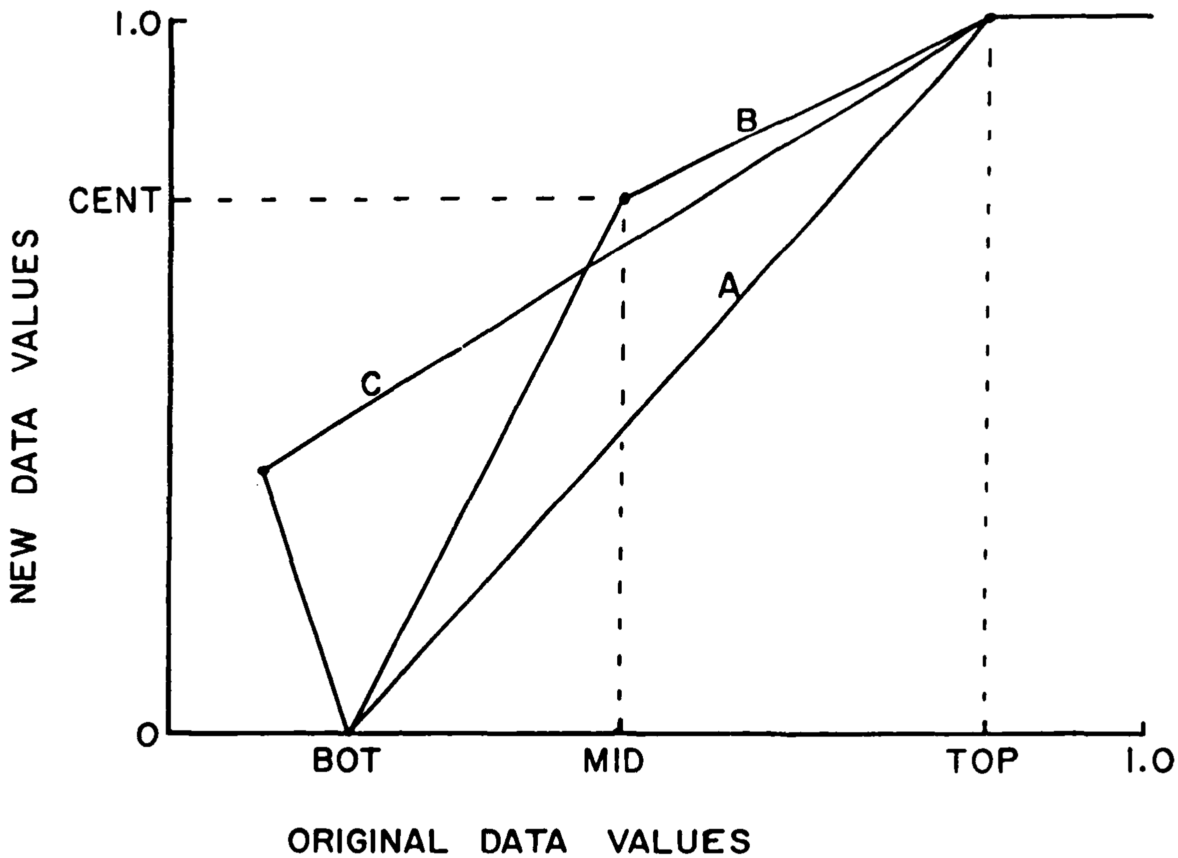
FIRST PASS



SECOND PASS

● Original Pixels
o Vertically Interpolated Pixels
x Horizontally Interpolated Pixels

FIGURE 5



A - Linear Enhancement Curve

B - Two Segment Enhancement Curve

C - Non-singular, Illegal Curve

FIGURE 6

****** T=00000 IS ON LU 04

SECTION IV

APPENDIX

```
FTN4,L
C*****************************************************
C*          Written by Ginny Kalb                     *
C*              Technology U. S. A.                   *
C*              P. O. Box 55333  Ph (301) 292-2592    *
C*              Oxon Hill Station                     *
C*              Washington D. C.   20022              *
C*       Rev.   29 Jan 1981                           *
C***************************************************************
C This program uniformly expands a square picture in both dim-
C ensions by using a cubic spline interpolation.  This is done
C by first interpolating along the columns of the input picture,
C then across the original plus interpolated rows. The resultant
C image is displayed on the RAMTEK or COMTAL screen and optionally
C stored on disc. Padding with zeros is used to obtain a full
C screen output picture.
C
C NOTE: Several array sizes will need to be altered for differ-
C ent input picture dimensions, as well as parameter IDIM =
C pixels/side of input picture. The array sizes are a function
C of IDIM and are located in COMMON blocks INPUT, UNIQ, and RROW.
C Subroutine TREAD is of course input-unique and must be altered
C for a new input picture size.
C Array sizes in subroutine INVRS, used by SOLVE, must be changed
C to A(IDIM-2,IDIM-2) and J(IDIM+19).
C
C COMMON /INPUT/IBUF(IDIM,IDIM)
C COMMON /UNIQ/NPTS,IDIM,CCOEF(IDIM,4),DELTA,ISHF,XSTEP,
C                 DERIV(IDIM,IDIM-2),T(IDIM-2,IDIM-2)
C COMMON /RROW/A(4),S(IDIM-2),Y(IDIM-2)
C
C The number of interpolated points is a user option at run time.
C The default is 3 which expands a 64x64 picture to 253x253. The
C maximum value for a 64x64 input picture is 7 because after that
C the output picture would exceed 512x512.
C***************************************************************
CξY
C
      PROGRAM EXPND
      COMMON /INPUT/IBUF(64,64)
      COMMON /OUTPT/IOPT,LUD,IPIC,L,JBUF(514),NPRT,LU3,ITK,ISECT
      COMMON /UNIQ/NPTS,IDIM,CCOEF(64,4),DELTA,ISHF,XSTEP,
     *               DERIV(64,62),T(62,62)
      COMMON /RROW/A(4),S(62),Y(62)
      DIMENSION KBUF(512)
      EQUIVALENCE (KBUF(1),JBUF(3))
      DIMENSION IP(5)
C
      CALL RMPAR(IP)
      WRITE(IP,1)
    1 FORMAT("ξmξhξJ","ξ&a17r0Cξ&dDEXPANSION VIA CUBIC SPLINE")
      WRITE(IP,2)
    2 FORMAT("ξ&a19r0CξKξ&dCNOTEξ&d@ program defaults to RAMTEK",
     * " screen on LU 12.","ξ&a20r0CξKDo you want to change either",
     * " default? Y/N _")
      READ(IP,30) ITEMP
      IF(ITEMP.EQ.1HN) GO TO 6
      WRITE(IP,3)
```

```
      3 FORMAT("ξ&a20r0CξK","ξ&a19r0CξKξ&dC.ξ&d@input ξ&dCCO",
       * "ξ&d@ for COMTAL or ξ&dCRAξ&d@ for RAMTEK _")
        READ(IP,4) NPRT
      4 FORMAT(A2)
        WRITE(IP,5)
      5 FORMAT("ξ&a20r0CξK","ξ&a19r0CξKξ&dC.ξ&d@input LU # _")
        READ(IP,*) LU3
      6 CONTINUE
        WRITE(IP,7)
        WRITE(IP,8)
      7 FORMAT("ξ&a19r0CξKξ&dC.ξ&d@specify expansion factor --",
       * " e.g. 2 will double input picture which is 64x64 ")
      8 FORMAT("ξ&a20r0CξK(or just RETURN and use the default of 4)")
        ITEMP = -1
        READ(IP,*) ITEMP
        IF(ITEMP.GE.0) NPTS = ITEMP-1
        IF(((IDIM-1)*NPTS+IDIM).GT.512) GO TO 995
        WRITE(IP,10)
     10 FORMAT("ξ&a20r0CξK","ξ&a19r0CξKξ&dC.ξ&d@specify inout ",
       * "picture tape unit _")
        READ(IP,*) MT
C
        CALL INVRS(T,62,DTNRM,DETM)
C
     15 WRITE(IP,20)
     20 FORMAT("ξ&a20r0CξK","ξ&a19r0CξKξ&dC.ξ&d@do you want ",
       * "to save output picture? Y/N _")
        READ(IP,30) IOPT
     30 FORMAT(A1)
        IF(IOPT.EQ.1HN) GO TO 50
        WRITE(IP,40)
     40 FORMAT("ξ&a19r0CξKξ&dC.ξ&d@specify output disc LU and",
       * " picture number _")
        READ(IP,*) LUD,IPIC
C
C Verify that the specified output disc is legitimate
C
        CALL CHECK(IP,LUD)
C
     50 CONTINUE
C
C Initialization of parameters
C
        L = -1
        ISHF = (512-((IDIM-1)*NPTS+IDIM))/2
        XSTEP = (NPTS+1.0)*DELTA
        ICNWD = MT+100B
C
C If using RAMTEK, issue a reset command
C
        IF(NPRT.EQ.2HCO) GO TO 65
        CALL EXEC(2,LU3,2400B,1)
C
C Output leading rows of zero
C
     65 LIM=(512-((IDIM-1)*NPTS+IDIM))/2
        LDSK = (480-((IDIM-1)*NPTS+IDIM))/2
        IF(NPRT.EQ.2HCO) LIM = LDSK
        IF(LIM.LE.0) GO TO 72
        DO 70 K=1,LIM
```

```
          IF(K.LE.LDSK) L = K-1
          CALL DSPLAY
          IF(IOPT.EQ.1HN) GO TO 70
          IF(K.GT.LDSK) GO TO 70
          CALL EXEC(2,LUD,KBUF,512,ITK,ISECT)
   70     CONTINUE
C
C Read in entire picture
C
   72 CONTINUE
        DO 75 I=1,IDIM
          CALL TREAD(ICNWD,I,IP)
   75     CONTINUE
C
C Solve for column-determined second derivatives
C
        DO 100 J=1,IDIM
          CALL SOLVE(0,J)
  100     CONTINUE
C
C Interpolate across first row
C
          DO 210 J=1,IDIM
            INDEX = ISHF+(NPTS+1)*J-NPTS
            KBUF(INDEX) = IBUF(1,J)
  210       CONTINUE
          CALL ROW(1)
C
C Loop on remaining rows, doing column followed by row
C interpolation
C
        DO 350 I=2,IDIM
          N = I-1
          DO 310 J=1,IDIM
            CALL GETCO(0,J,N)
  310       CONTINUE
C
          IF(NPTS.EQ.0) GO TO 331
          DO 330 K=1,NPTS
            X = (N-1)*XSTEP+K*DELTA
            DO 320 J=1,IDIM
              INDEX = ISHF+(NPTS+1)*J-NPTS
              A(1) = CCOEF(J,1)
              A(2) = CCOEF(J,2)
              A(3) = CCOEF(J,3)
              A(4) = CCOEF(J,4)
              KBUF(INDEX) = F(A,X,N)
              IBUF(I-1,J) = KBUF(INDEX)
  320         CONTINUE
            CALL ROW(N)
  330     CONTINUE
  331     DO 340 J=1,IDIM
            INDEX = ISHF+(NPTS+1)*J-NPTS
            KBUF(INDEX) = IBUF(I,J)
  340       CONTINUE
          CALL ROW(I)
  350     CONTINUE
C
C Write trailing rows of zero
C
```

```
      DO 400 I=1,512
        KBUF(I) = 0
  400   CONTINUE
      LSAVE = L+1
      DO 410 L=LSAVE,511
        CALL DSPLAY
        IF(IOPT.EQ.1HN) GO TO 410
        IF(L.GT.479) GO TO 410
        CALL EXEC(2,LUD,KBUF,512,ITK,ISECT)
  410   CONTINUE
C
      WRITE(IP,1)
      WRITE(IP,450)
  450 FORMAT("ξ&a19r0CξKit is finished")
      WRITE(IP,455)
  455 FORMAT("ξ&a20r0CξKξ&dC.ξ&d@do you want to expand the ",
     * "next picture? Y/N _")
      READ(IP,30) ITEMP
      IF(ITEMP.EQ.1HN) STOP
      ICNWD = MT+1300B
      CALL EXEC(3,ICNWD)
      GO TO 15
C
  995 WRITE(IP,996)
  996 FORMAT("ξ&a19r0CξKinvalid value for expansion factor",
     * "ξ&a20r0Cξ&KTry a value between 1 and 8, inclusive")
      STOP
      END
C
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ROW(I)
C
C This subroutine interpolates NPTS points between the gray levels
C in row I of the input buffer IBUF. This is either the original
C row or an interpolated-between-columns row.
C The resultant row is then displayed and optionally stored on disc.
C
      COMMON /OUTPT/IOPT,LUD,IPIC,L,JBUF(514),NPRT,LU3,ITK,ISECT
      COMMON /UNIQ/NPTS,IDIM,CCOEF(64,4),DELTA,ISHF,XSTEP,
     *                DERIV(64,62),T(62,62)
      COMMON /RROW/A(4),S(62),Y(62)
      DIMENSION KBUF(512)
      EQUIVALENCE (KBUF(1),JBUF(3))
C
      IF(NPTS.EQ.0) GO TO 201
      CALL SOLVE(I,0)
      LIM = IDIM-1
      DO 200 J=1,LIM
        CALL GETCO(I,0,J)
        DO 100 K=1,NPTS
          X=(J-1)*XSTEP+K*DELTA
          INDEX = ISHF+(NPTS+1)*J-NPTS+K
          KBUF(INDEX) = F(A,X,J)
  100     CONTINUE
  200   CONTINUE
  201 L = L+1
      CALL DSPLAY
      IF(IOPT.EQ.1HN) RETURN
      IF(L.GT.479) RETURN
```

```
      CALL EXEC(2,LUD,KBUF,512,ITK,ISECT)
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE SOLVE(I,J)
C
C This subroutine solves the IDIM-2 simultaneous equations
C for the second derivatives determined by the data values
C in row I or column J of the input buffer IBUF.
C
      COMMON /UNIQ/NPTS,IDIM,CCOEF(64,4),DELTA,ISHF,XSTEP,
     *                DERIV(64,62),T(62,62)
      COMMON /INPUT/IBUF(64,64)
      COMMON /RROW/A(4),S(62),Y(62)
C
      LL = IDIM-2
      IF(I.EQ.0) GO TO 200
C
C Compute second derivatives for row I of input buffer.
C
      DO 50 K=1,LL
        Y(K) = IBUF(I,K+2)-2*IBUF(I,K+1)+IBUF(I,K)
        Y(K) = Y(K)*6/(XSTEP*XSTEP)
   50   CONTINUE
C
      DO 100 K=1,LL
        S(K) = 0.0
        DO 75 M=1,LL
          S(K) = S(K)+T(K,M)*Y(M)
   75     CONTINUE
  100   CONTINUE
      RETURN
C
C Compute second derivatives for column J of input buffer.
C
  200 CONTINUE
      DO 250 K=1,LL
        Y(K) = IBUF(K+2,J)-2*IBUF(K+1,J)+IBUF(K,J)
        Y(K) = Y(K)*6/(XSTEP*XSTEP)
  250   CONTINUE
C
      DO 300 K=1,LL
        DERIV(J,K) = 0.0
        DO 275 M=1,LL
          DERIV(J,K) = DERIV(J,K)+T(K,M)*Y(M)
  275     CONTINUE
  300   CONTINUE
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE GETCO(I,J,N)
C
C This subroutine computes coefficients needed to interpolate
C span N along row I or column J.
C
      COMMON /UNIQ/NPTS,IDIM,CCOEF(64,4),DELTA,ISHF,XSTEP,
```

```
     *              DERIV(64,62),T(62,62)
      COMMON /INPUT/IBUF(64,64)
      COMMON /RROW/A(4),S(62),Y(62)
C
      IF(I.EQ.0) GO TO 100
C
C Row interpolation
C
      A(1) = 0.0
      A(2) = 0.0
      IF(N.GT.1) A(1) = S(N-1)/6.0
      IF(N.LT.(IDIM-1)) A(2) = S(N)/6.0
      A(3) = IBUF(I,N)
      A(4) = IBUF(I,N+1)
      RETURN
C
C Column interpolation
C
  100 CONTINUE
      CCOEF(J,1) = 0.0
      CCOEF(J,2) = 0.0
      IF(N.GT.1) CCOEF(J,1) = DERIV(J,N-1)/6.0
      IF(N.LT.63) CCOEF(J,2) = DERIV(J,N)/6.0
      CCOEF(J,3) = IBUF(N,J)
      CCOEF(J,4) = IBUF(N+1,J)
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION F(A,X,I)
C
C This routine computes the interpolated value for x coordinate X
C which lies in the Ith span.
C
      COMMON /UNIQ/NPTS,IDIM,CCOEF(64,4),DELTA,ISHF,XSTEP,
     *              DERIV(64,62),T(62,62)
      DIMENSION A(4)
      H = XSTEP
      DX1 = X-(I-1)*H
      DX2 = H-DX1
      F = A(1)*DX2*(DX2*DX2/H-H)
     *   +A(2)*DX1*(DX1*DX1/H-H)
     *   +A(3)*DX2/H+A(4)*DX1/H
      IF(F.LT.0.0) F=0.0
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE TREAD(ICNWD,I,IP)
C
C This subroutine reads one row of the input picture,
C assuming the following format:
C   1 row = 32 16-bit words = 64 8-bit pixels
C
      COMMON /INPUT/IBUF(64,64)
      COMMON /UNIQ/NPTS,IDIM,CCOEF(64,4),DELTA,ISHF,XSTEP,
     *              DERIV(64,62),T(62,62)
      DIMENSION LBUF(32),IR(2),IP(5)
```

```
      EQUIVALENCE (IR,REG)
      REG = EXEC(1,ICNWD,LBUF,32)
      IF(IAND(IR,200B).EQ.200B) GO TO 995
      LL = IDIM/2
      IF(IR(2).GE.LL) GO TO 75
      N = IR(2)+1
      DO 50 K=N,LL
   50   LBUF(K) = 0
C
   75 CONTINUE
      DO 100 J=1,LL
        IBUF(I,2*J-1) = IAND(LBUF(J)/256,377B)*4
        IBUF(I,2*J) = IAND(LBUF(J),377B)*4
  100   CONTINUE
      RETURN
C
  995 WRITE(IP,996)
  996 FORMAT("ξ&a19r0CξKInvalid # of records in picture file")
      STOP
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE DSPLAY
C
C This subroutine makes the appropriate EXEC call to write a line of
C the output picture to the selected screen, specified by NPRT and LU3.
C
      COMMON /OUTPT/IOPT,LUD,IPIC,L,JBUF(514),NPRT,LU3,ITK,ISECT
      DIMENSION KBUF(512)
      EQUIVALENCE (KBUF(1),JBUF(3))
C
      ITK = L/12+(IPIC-1)*40
      ISECT = (L-(ITK-(IPIC-1)*40)*12)*8
      IF(NPRT.EQ.2HRA) GO TO 200
C
      CALL EXEC(2,LU3,KBUF,512,L)
      RETURN
C
  200 CALL EXEC(2,LU3,JBUF,514)
      RETURN
CξZ
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      BLOCK DATA
      COMMON /INPUT/IBUF(64,64)
      COMMON /OUTPT/IOPT,LUD,IPIC,L,JBUF(514),NPRT,LU3,ITK,ISECT
      COMMON /UNIQ/NPTS,IDIM,CCOEF(64,4),DELTA,ISHF,XSTEP,
     *            DERIV(64,62),T(62,62)
      COMMON /RROW/A(4),S(62),Y(62)
      DATA NPTS/3/,IDIM/64/,T/0.0/,DELTA/0.5/
      DATA NPRT/2HRA/,LU3/12/
      DATA JBUF(1)/5001B/,JBUF(2)/1024/
      DATA T(1,1)/4.0/,T(1,2)/1.0/,
     *     T(2,1)/1.0/,T(2,2)/4.0/,T(2,3)/1.0/,
     *     T(3,2)/1.0/,T(3,3)/4.0/,T(3,4)/1.0/,
     *     T(4,3)/1.0/,T(4,4)/4.0/,T(4,5)/1.0/,
     *     T(5,4)/1.0/,T(5,5)/4.0/,T(5,6)/1.0/,
```

```
     *    T(6,5)/1.0/,T(6,6)/4.0/,T(6,7)/1.0/,
     *    T(7,6)/1.0/,T(7,7)/4.0/,T(7,8)/1.0/,
     *    T(8,7)/1.0/,T(8,8)/4.0/,T(8,9)/1.0/,
     *    T(9,1)/1.0/,T(9,9)/4.0/,T(9,10)/1.0/,
     *    T(10,9)/1.0/,T(10,10)/4.0/,T(10,11)/1.0/,
     *    T(11,10)/1.0/,T(11,11)/4.0/,T(11,12)/1.0/,
     *    T(12,11)/1.0/,T(12,12)/4.0/,T(12,13)/1.0/,
     *    T(13,12)/1.0/,T(13,13)/4.0/,T(13,14)/1.0/,
     *    T(14,13)/1.0/,T(14,14)/4.0/,T(14,15)/1.0/,
     *    T(15,14)/1.0/,T(15,15)/4.0/,T(15,16)/1.0/,
     *    T(16,15)/1.0/,T(16,16)/4.0/,T(16,17)/1.0/,
     *    T(17,16)/1.0/,T(17,17)/4.0/,T(17,18)/1.0/,
     *    T(18,17)/1.0/,T(18,18)/4.0/,T(18,19)/1.0/,
     *    T(19,18)/1.0/,T(19,19)/4.0/,T(19,20)/1.0/,
     *    T(20,19)/1.0/,T(20,20)/4.0/,T(20,21)/1.0/,
     *    T(21,20)/1.0/,T(21,21)/4.0/,T(21,22)/1.0/,
     *    T(22,21)/1.0/,T(22,22)/4.0/,T(22,23)/1.0/,
     *    T(23,22)/1.0/,T(23,23)/4.0/,T(23,24)/1.0/,
     *    T(24,23)/1.0/,T(24,24)/4.0/,T(24,25)/1.0/,
     *    T(25,24)/1.0/,T(25,25)/4.0/,T(25,26)/1.0/,
     *    T(26,25)/1.0/,T(26,26)/4.0/,T(26,27)/1.0/,
     *    T(27,26)/1.0/,T(27,27)/4.0/,T(27,28)/1.0/,
     *    T(28,27)/1.0/,T(28,28)/4.0/,T(28,29)/1.0/,
     *    T(29,28)/1.0/,T(29,29)/4.0/,T(29,28)/1.0/,
     *    T(30,29)/1.0/,T(30,30)/4.0/,T(30,31)/1.0/,
     *    T(31,31)/1.0/,T(31,31)/4.0/,T(31,32)/1.0/,
     *    T(32,31)/1.0/,T(32,32)/4.0/,T(32,33)/1.0/,
     *    T(33,32)/1.0/,T(33,33)/4.0/,T(33,34)/1.0/,
     *    T(34,33)/1.0/,T(34,34)/4.0/,T(34,35)/1.0/,
     *    T(35,34)/1.0/,T(35,35)/4.0/,T(35,36)/1.0/,
     *    T(36,35)/1.0/,T(36,36)/4.0/,T(36,37)/1.0/,
     *    T(37,36)/1.0/,T(37,37)/4.0/,T(37,38)/1.0/,
     *    T(38,37)/1.0/,T(38,38)/4.0/,T(38,39)/1.0/,
     *    T(39,38)/1.0/,T(39,39)/4.0/,T(39,40)/1.0/,
     *    T(40,39)/1.0/,T(40,40)/4.0/,T(40,41)/1.0/,
     *    T(41,40)/1.0/,T(41,41)/4.0/,T(41,42)/1.0/,
     *    T(42,41)/1.0/,T(42,42)/4.0/,T(42,43)/1.0/,
     *    T(43,42)/1.0/,T(43,43)/4.0/,T(43,44)/1.0/,
     *    T(44,43)/1.0/,T(44,44)/4.0/,T(44,45)/1.0/,
     *    T(45,44)/1.0/,T(45,45)/4.0/,T(45,46)/1.0/,
     *    T(46,45)/1.0/,T(46,46)/4.0/,T(46,47)/1.0/,
     *    T(47,46)/1.0/,T(47,47)/4.0/,T(47,48)/1.0/,
     *    T(48,47)/1.0/,T(48,48)/4.0/,T(48,49)/1.0/,
     *    T(49,48)/1.0/,T(49,49)/4.0/,T(49,50)/1.0/,
     *    T(50,49)/1.0/,T(50,50)/4.0/,T(50,51)/1.0/,
     *    T(51,50)/1.0/,T(51,51)/4.0/,T(51,52)/1.0/,
     *    T(52,51)/1.0/,T(52,52)/4.0/,T(52,53)/1.0/,
     *    T(53,52)/1.0/,T(53,53)/4.0/,T(53,54)/1.0/,
     *    T(54,53)/1.0/,T(54,54)/4.0/,T(54,55)/1.0/,
     *    T(55,54)/1.0/,T(55,55)/4.0/,T(55,56)/1.0/,
     *    T(56,55)/1.0/,T(56,56)/4.0/,T(56,57)/1.0/,
     *    T(57,56)/1.0/,T(57,57)/4.0/,T(57,58)/1.0/,
     *    T(58,57)/1.0/,T(58,58)/4.0/,T(58,59)/1.0/,
     *    T(59,58)/1.0/,T(59,59)/4.0/,T(59,60)/1.0/,
     *    T(60,59)/1.0/,T(60,60)/4.0/,T(60,61)/1.0/,
     *    T(61,60)/1.0/,T(61,61)/4.0/,T(61,62)/1.0/,
     *    T(62,61)/1.0/,T(62,62)/4.0/
      END
      END$
```

```
****** T=00000 IS ON LU 04

C*****************************************************
C*          Written by Ginny Kalb                    *
C*              Technology U. S. A.                  *
C*              P. O. Box 55333  Ph (301) 292-2592   *
C*              Oxon Hill Station                    *
C*              Washington D. C.    20022            *
C*        Rev.  29 Jan 1981                          *
C*****************************************************************
C*       This program is used to document Program EXPND          *
C*****************************************************************
C
01 Software Blueprint -- Level B Design of the Cubic Spline Expansion
   02 Program Modules
      03 Module Declaration

         #1 10 MAIN             main program
         #2    20 INVRS         matrix inversion
         #3    20 TREAD         read one row of input tape
         #4    20 SOLVE         solve matrix eqs for 2nd derivatives
         #5    20 GETCO         compute interpolation coefficients
         #6    20 ROW           interpolate across row of data points
         #7    20 F             evaluate interpolated value
         #8    20 DSPLAY        output interpolated row to screen
      03 Module Reference Structure
         <calling proc name>::=<called proc name list>

         #1 10 MAIN::=INVRS,DSPLAY,TREAD,SOLVE,ROW,GETCO,F
         #2    20 INVRS
         #3    20 TREAD
         #4    20 SOLVE
         #5    20 GETCO
         #6    20 ROW::=SOLVE,GETCO,F,DSPLAY
         #7    20 F
         #8    20 DSPLAY
   02 Data
      03 Data Declaration -- all names follow FORTRAN default data types

         Buffer
           DELTA, relative spacing between adjacent output pixels
           ICNWD, control word for EXEC calls to tape drive MT
           IDIM,  # pixels per side of input picture
           INDEX, pointer into output array KBUF
           IPIC,  output picture number
           ISECT, sector number for storing picture on disc
           ISHF,  bias count needed to center output picture row in
                  array KBUF
           ITK,   track number for storing picture on disc
           L,     line count for COMTAL or disc output control
           LDISK, # of extra rows on top & bottom of screen when
                  output picture is centered
           LIM,   # of extra rows on top & bottom of disc space
                  when output picture is centered
           LUD,   logical unit number of output disc
           LU3,   logical unit number of output screen
                  Initialize LU3 to 12
           MT,    logical unit number of input tape
           NPTS,  # of data points to be inserted between adjacent
```

```
                   input data pixels
                   Initialize NPTS to 3
              X,     position coordinate of interpolated pixel
              XSTEP, relative spacing between adjacent input pixels

            Array
              A(4),                 coefficients needed to interpolate between 2
                                    pixels in a row
              CCOEF(IDIM,4),        for each column, coefficients needed to
                                    interpolate between 2 pixels in that column
              DERIV(IDIM,IDIM-2),   for each column, 2nd derivatives at each
                                    input pixel (assuming 0 at first and last
                                    pixels)
              IBUF(IDIM,IDIM),      stores entire input picture
              IP(5),                stores RMPAR parameters -- only first one is
                                    used; IP = logical unit # of user's terminal
              JBUF(514),            output array for RAMTEK; first 2 words are
                                    predetermined, rest are data words
                                    Initialize JBUF(1) to octal 5001 and JBUF(2)
                                    to decimal 1024
              KBUF(512),            output array for COMTAL or disc
              S(IDIM-2),            2nd derivatives at each input pixel in a row
                                    (assuming 0 at first and last pixels)
              T(IDIM-2,IDIM-2)      inverse of matrix arising from the IDIM-2
                                    simultaneous equations for the 2nd derivativ
                                    for a cubic spline fit through IDIM equally
                                    spaced data points with 2nd derivative of 0
                                    at the endpoints
                                    Initialize T to 4 along the diagonal, 1
                                    off the diagonal, and 0 elsewhere
        03 Data Reference Structure
            <data structure type><data element name>::=<referencing proc's>

            Buffer
              DELTA  ::= MAIN,ROW
              ICNWD  ::= MAIN,TREAD
              IDIM   ::= MAIN,TREAD,SOLVE,GETCO,ROW
              INDEX  ::= MAIN,ROW
              IPIC   ::= MAIN,DSPLAY
              ISECT  ::= MAIN,ROW,DSPLAY
              ISHF   ::= MAIN,ROW
              ITK    ::= MAIN,ROW,DSPLAY
              L      ::= MAIN,ROW,DSPLAY
              LDISK  ::= MAIN
              LIM    ::= MAIN
              LUD    ::= MAIN,ROW
              LU3    ::= MAIN,DSPLAY
              MT     ::= MAIN
              NPTS   ::= MAIN,ROW
              X      ::= MAIN,ROW,F
              XSTEP  ::= MAIN,SOLVE,ROW,F

            Array
              A      ::= MAIN,GETCO,ROW,F
              CCOEF  ::= MAIN,GETCO
              DERIV  ::= SOLVE,GETCO
              IBUF   ::= MAIN,TREAD,SOLVE,GETCO
              IP     ::= MAIN,CHECK,TREAD
              JBUF   ::= DSPLAY
              KBUF   ::= MAIN,ROW
```

```
           S       ::= SOLVE,GETCO
           T       ::= INVRS,SOLVE
  02 Control
     03 Control Declaration

        Switch
          IOPT Of Status ("Y","N")    "save on disc" response
          NPRT Of Status ("RA","CO") output screen identifier
     03 Control Reference Structure
        <switch name>::=<where set>/<where tested>
        IOPT ::= MAIN/MAIN,ROW
        NPRT ::= MAIN/MAIN,DSPLAY
  02 Procedure Definition
     #1 Procedure MAIN

        Questions asked of user at run time:
        (1) "program defaults to RAMTEK screen on LU 12.
             Do you want to change either default? Y/N"
        If answer is "Y", (1a) "input CO for COMTAL or RA for RAMTEK"
                          (1b) "input LU #"
        (2) "specify expansion factor -- e.g. 2 will approximately doubl
             input picture which is 64x64. (or just RETURN and use defa
             of 4"
        (3) "specify input picture tape unit"
        (4) "do you want to save output picture? Y/N"
        If answer is "Y", (4a) "specify output disk LU and picture numbe
        At completion of expansion,
        (5) "do you want to expand the next picture? Y/N"
        Call RMPAR;            get LU # of user's terminal
        Write questions;       prompt user for specifics
        Read answers;
        Call INVRS;            invert matrix T
        Initialize L,ISHF,XSTEP, and ICNWD;
        If NPRT = "RA"
          Then issue reset command to RAMTEK;
        End If;
        Write leading rows of 0 to screen;
        If IOPT = "Y"
          Then write leading rows of 0 to disc;
        End If;
        Loop until entire input picture has been read
          Call TREAD;          read next row of input picture
        End Loop;

        Loop until all columns of input picture have been processed
          Call SOLVE;          get 2nd derivatives at column pixels
                               (0 at endpoints)
        End Loop;

        Embed 1st row of input picture in output array KBUF,
        leaving NPTS gaps in between for interpolated values;
        Call ROW;              fill in this row

        Loop until all input rows have been processed
          Loop until all input columns have been processed
            Call GETCO;        get coefficients to interpolate along this
                               column between old and current input rows
          End Loop;
          If NPTS is not 0 Then
            Loop until NPTS pixel positions have been processed
```

```
          X := X+DELTA;  put X at next pixel position
          Loop until all input columns have been processed
            Call F;      interpolate at X in this column
            Write new value in output array KBUF;
            Replace the data point in the input array IBUF at
             this column and old row by the new value;
          End Loop;
          Call ROW;      interpolate across the row just manufactured
          Move current row pointer into old row pointer;
        End Loop;
      End If;
      Embed current input row in output array KBUF,leaving
       NPTS gaps in between for interpolated values;
      Call ROW;          fill in this row
    End Loop;
    Write trailing rows of 0 to screen;
    If IOPT = "Y"
      Then write trailing rows of 0 to disc;
    End If;
    End MAIN;

 #2 Procedure INVRS

      This subroutine was obtained verbatim from NUMERICAL METHODS
    by Robert W. Hornbeck.
      This subroutine is only called at initialization to invert
    matrix T.

    Array
      A(IDIM-2,IDIM-2),  original matrix
      J(IDIM-2+21)       temporary storage
    Buffer
      M                  dimension of matrix to be inverted

    Calculate inverse of A;
    Store inverse in A;
    Return;
    End INVRS;
 #3 Procedure TREAD

    Array
      LBUF(IDIM/2)       stores one row of input picture

    Read next record from tape MT;
    If EOF encountered
      Then write "invalid # of records in picture file";
      Stop;
    End If;
    If # of words read < 32
      Then zero-fill rest of input array;
    End If;
    Repack each byte in LBUF into a word in IBUF, preserving
     the order of the bits;
    Shift each word left 2 bits to rescale data to 10 bits;
    End TREAD;
 #4 Procedure SOLVE

      Let g(x) be the cubic spline fit through an equally spaced
    line of pixels located at x ,x ,...,x  and with gray levels
                               1  2     N
```

```
G ,G ,...,G . Then the 2nd derivatives of g at the pixels
 1  2      N
satisfy the following equations:
  g"(x )+4*g"(x    )+g"(x    ) = 6*[(G    -2*G    +G )/(^x)**2]
      k          k+1       k+2          k+2    k+1   k

for k=1,2,...,N-2. Note: ^x = delta x.
There are N-2 equations in the N unknowns g"(x ),k=1,2,...,N.
                                               k
The additional constraints of 2nd derivative = 0 at the end-
points permit the solution of these equations. In matrix form,
the equations are:
  (4 1 0 ... 0)( g"(2) )
  (1 4 1 0 . 0)( g"(3) )
  (0 1 4 1 0..)( g"(4) ) = Y, Y(k) = 6*[(G    -2*G    +G )/(^x)**2]
  (..........1)(.......)                k+2     k+1   k
  (0 ... 0 1 4)(g"(N-1))

In this application, N = IDIM and T is the inverse of the
coefficient matrix.
Buffer
  I,                    0 or # of the row in input array IBUF
                         containing the desired pixels
  J                     0 or # of the column in input array IBUF
                         containing the desired pixels
Array
  Y(IDIM-2)             temporary storage for expressions on right
                         hand side of equations for g
If I is non-zero        row interpolation
  Then compute Y;       note G  = IBUF(I,K) and ^x = XSTEP
       S := T*Y;              k
       Return;
  Else compute Y;       column interpolation
                        note G  = IBUF(K,J) and ^x = XSTEP
                              k
       DERIV(J,.) = T*Y;
       Return;
End If;
End SOLVE;
#5 Procedure GETCO
g" has been determined at the pixel values along either a row
or column of the picture and stored. This determines g(x) on
each span (between consecutive input pixels) because each cubic
can be written as a function of g":
  between pixels at x  and x   (span k),

                     k      k+1
  Note: ^x = delta x
  g(x) = g"(x )/6*[(x    -x)**3/^x - ^x*(x    -x)] +
             k        k+1                 k+1

          g"(x    )/6*[(x-x )**3 /^x - ^x*(x-x )] +
              k+1          k                  k

          G *[(x    -x)/^x] + G    *[(x-x )/^x]
           k     k+1            k+1      k
Buffer
  I,                    either 0 or # of row in input array IBUF
                         containing the desired pixels
  J,                    either 0 or # of column in input array IBUF
```

```
                                 containing the desired pixels
       N                         span #
    If I is non-zero
       Then                      row interpolation
         If N > 1                note g" is 0 at x  and x
                                                  1       IDIM
           Then A(1) := S(N-1)/6;
           Else A(1) := 0;
         End If;
         If N < IDIM-1
           Then A(2) := S(N)/6;
           Else A(2) := 0;
         End If;
         A(3) := IBUF(I,N);
         A(4) := IBUF(I,N+1);
         Return;

       Else                      column interpolation
         If N > 1                note g" is 0 at x  and x
                                                  1       IDIM
           Then CCOEF(J,1) := DERIV(J,N-1)/6;
           Else CCOEF(J,1) := 0;
         End If;
         If N < IDIM-1
           Then CCOEF(J,2) := DERIV(J,N)/6;
           Else CCOEF(J,2) := 0;
         End If;
         CCOEF(J,3) := IBUF(N,J);
         CCOEF(J,4) := IBUF(N+1,J);
         Return;
    End If;
    End GETCO;
#6 Procedure ROW

    NPTS points must be inserted between each pair of pixels in
    the designated row of input array IBUF. The position values
    assigned to these pixels are:

      Kth pixel has x coordinate (K-1)*XSTEP.
    This allows NPTS pixels to be inserted at subintervals DELTA
    in each XSTEP interval.
    Buffer
      I                          # of row in input array IBUF containing
                                  the desired pixels
    If NPTS non-zero             bypass processing if 0 expansion is opted
      Then
        Call SOLVE;              get 2nd derivatives for cubic spline fit
                                  across this row
        Loop until last pair of pixels processed
          Call GETCO;            get the 4 coefficients which specify the
                                  cubic through current pair of pixels
          Loop until NPTS points have been inserted
            X := X+DELTA;
            Call F;              evaluate cubic at X
            Store in output array KBUF;
          End Loop;
        End Loop;
    End If;
    Call DSPLAY;                 output new row to screen
    If IOPT = "N"
```

```
            Then Return;
            Else Store row on disc LUD;
                 Return;
          End If;
          End ROW;
       #7 Procedure F
          evaluate the cubic interpolation polynomial
          Buffer
            X,                    position of new pixels to be interpolated
            I,                    span #
            DX1,                  temporary storage of distance between new
                                   pixel and left endpoint of span
            DX2,                  temporary storage of distance between right
                                   endpoint of span and new pixel
            H                     temporary storage of XSTEP
          Array
            A(4)                  weighting coefficients for this span
          DX1 := X-(I-1)*H;
          DX2 := H-DX1;
          F := A(1)*DX2*[DX2**2/H - H] +
               A(2)*DX1*[DX1**2/H - H] +
               A(3)*DX2/H + A(4)*DX1/H;
          Return;
          End F;
       #8 Procedure DSPLAY

          Compute track and sector for next row;
          If NPRT = "RA"
            Then output JBUF to RAMTEK;
            Else output KBUF to COMTAL;
          End If;
          Return;
          End DSPLAY;
```

```
****** T=00000 IS ON LU 04

C*****************************************************
C*          Written by Ginny Kalb                    *
C*               Technology U. S. A.                 *
C*               P. O. Box 55333  Ph (301) 292-2592  *
C*               Oxon Hill Station                   *
C*               Washington D. C.   20022            *
C*       Rev.  29 Jan 1981                           *
C*******************************************************************
C*       This program is used to explain how to use Program EXPND. *
C*******************************************************************
C
                    User's Guide to Cubic Spline Expansion
                              Program EXPND
  This program expands a 64x64 pixel image by row and column cubic spline
interpolation. The expansion factor is selectable, ranging from 1 (no ex-
pansion) to 8 (output image is 505x505 pixels). The output picture is
displayed on either a RAMTEK or COMTAL screen and optionally stored on dis
The user options are specified at run time in response to prompts issued
by the program. The input must be on tape and the tape must be positioned
by the user to the desired picture file prior to program execution.
  It is possible to speed up program execution if the user has a data tap
with consecutive images all of which are to be expanded by the same facto
because then the initialization of the algorithm can be skipped.
  Note: the input tape is not rewound at the end of the program.
  Sample computer-user dialogue:
C  NOTE program defaults to RAMTEK screen on LU 12.
   Do you want to change either default? Y/N

U  Y

C  .input CO for COMTAL or RA for RAMTEK

U  RA

C  .input LU #

U  16

C  .specify expansion factor -- e.g. 2 will double input picture which
    is 64x64 (or just RETURN and use the default of 4)

U  2

C  .specify input picture tape unit

U  7
delay while initialization is performed
C  .do you want to save output picture? Y/N

U  Y

C  .specify output disc LU and picture number

U  13,2
delay while expansion is performed
C  it is finished
   .do you want to expand the next picture? Y/N
```

```
U  Y
C  .do you want to save output picture? Y/N
             .
             .
             .
```

DISTRIBUTION LIST

12 copies	Director Walter Reed Army Institute of Research Walter Reed Army Medical Center ATTN: SGRD-UWZ-C Washington, DC 20307-5100
4 copies	Commander US Army Medical Research and Development Command ATTN: SGRD-RMS Fort Detrick, Frederick, Maryland 21701-5012
12 copies	Defense Technical Information Center (DTIC) ATTN: DTIC-DDAC Cameron Station Alexandria, VA 22304-6145
1 copy	Dean School of Medicine Uniformed Services University of the Health Sciences 4301 Jones Bridge Road Bethesda, MD 20814-4799
1 copy	Commandant Academy of Health Sciences, US Army ATTN: AHS-CDM Fort Sam Houston, TX 78234-6100

END

FILMED

7-85

DTIC