Call McVickar re: Fiberoptics at Sikorsky

THE ADAGE GRAPHICS USER UPDATE

A publication of AGUS, the
Adage Graphics User's Society

SUMMER 1984

TABLE OF CONTENTS

VOL. I - ISSUE 2

LETTERS TO THE EDITOR

Please submit any letters to:

Keith L. McVicar
Sikorsky Aircraft, Dept. CAD/M
Stratford, CT 06602

CLASSIFIED

This section will be made availabe for anyone wishing to utilize it. Please submit classified information to the editor.

THE EDITOR'S PAGE

The Adage Graphics User-Update — Summer 1984

Keith L. McVicar
Sikorsky Aircraft, Dept. CAD/M
Stratford, CT 06602
203-386-5196

Welcome to the second issue of the AGUS newsletter. There has been a lot of interest in AGUS generated since the first issue, and I hope it will continue on the upswing.

A new user mailing list is being developed by Adage, so that we may reach out to some of you who are new to AGUS. If you know of anyone who wishes to be on the mailing list, please notify Adage, Inc., (they are maintaining the user list). There is a reply form at the end of this newsletter.

```
 SSSSSSS IIIIIII  GGGGGGG   GGGGGGG  RRRRRRRR   AAAAAAA  PPPPPPPP  H      H
S           I    G         G         R       R A       A P       P H      H
S           I    G         G         R       R A       A P       P H      H
 SSSSSS     I    G         G         RRRRRRRR  AAAAAAAAA PPPPPPPP  HHHHHHHH
       S    I    G   GGGGG G   GGGGG R     R   A       A P         H      H
       S    I    G       G G       G R      R  A       A P         H      H
SSSSSSS  IIIIIII  GGGGGGG   GGGGGGG  R       R A       A P         H      H
```

```
          ''  8888888      44
           ' 8       8   4  4
             8       8 44   4
              8888888  44444444
             8       8      4
             8       8      4
              8888888       4
```

Our annual AGUS meeting is scheduled for SIGGRAPH '84 this year, which is being held July 23-27 in Minneapolis. There is free shuttle service from the Minneapolis Auditoriam to the hotel. Please contact me as soon as possible if you are interested in giving a presentation.

The following Agenda is planned:

CONTINUED

AGUS MEETING AT SIGGRAPH '84

DATE: Tuesday, July 24, 1984
TIME: 3:30 P.M. - 8:00 P.M.
PLACE: AMFAC Hotel
30 South 7th Street
Minneapolis, MN 55402
Meeting Rooms 1 & 2
(A light buffet will be served)

AGENDA

3:00 P.M.- 5:00 P.M.	Business/Organizational Meeting Report-User Network
5:00 P.M.- 5:30 P.M.	Speaker: Nick England, Adage Topic: Videotape, question and answer session
5:30 P.M.- 6:00 P.M.	Speaker: Representative from Bell Laboratories Topic: New Compiler for Bit Slicer
6:00 P.M.- 6:30 P.M.	Speaker: Ben Dawson, MIT Topic: Image Processing, Slide Show
6:30 P.M.- 7:15 P.M.	Speaker: Richard Pferdner, Technology Service Topic: Flight Simulator, Videotape
7:15 P.M.- 7:30 P.M.	Closing Comments
7:30 P.M.- 8:00 P.M.	Social

NCGA '84 followup: There was an inpromptu AGUS meeting held at NCGA '84 for interested parties. Some fine suggestions were made by Richard Pferdner of Technology Service Corporation concerning the production of the newsletter. Many people out there have dial out lines on their mainframes, or own personal computers. If we can have a central dumping ground (say an account on Adage's VAX), transfer of information would be much more efficient. This could also include information concerning user problems and an electronic mail center between users. Anyone interested in participating in the setup of this effort, please let me know.

THE ADAGE FORUM

During the past several months, Adage has made a number of new product announcements. The purpose of this column is to inform the user community of recent announcements made by Adage for new produts and services. Below is a brief summary of these news releases.

- ADAGE INC. AND ADRA SYSTEMS INC., ANNOUNCE PRODUCT AGGREMENT

Adra Systems is to supply a high performance CAD/CAM system product to Adage Inc. There are no specifics available for this product at this time (announcement this summer), but it will be exclusively marketed and serviced by Adage. Adra Systems Inc. was founded in July 1983 to develop, manufacture and market low-cost, high-quality CAD systems for industrial use. Adra is located in Lowell, Massachusetts.

- ADAGE 3010 SYSTEM PACKAGING REDUCES COST

Priced around $40K, the Adage 3010 is a bundled package suited for work with the new Adage software packages SOLID-3000 and SEISMIC-3000. To make the product less expensive, new frame buffer memories are used (double buffered), and the unit is not as expandable as the Adage 3006. If these applications are what you are interested in, this gives you a reasonable startup price.

- ADAGE INC. SIGN UP MARKETING AGREEMENT WITH NIXDORF COMPUTER CORP.

Adage and Nixdorf have signed an agreement to facilitate the joint marketing of Adage 4250 workstations and Nixdorf 8890 series computers. This decision basically provides CADAM users (CATIA?) an alternative for running on IBM main-frame equipment. I have no information regarding the relative power of this machine compared to its IBM counterpart (i.e., same as 4341, 4361, etc.).

- ADAGE INC. ANOUNCES SOLID - 3000 DISPLAY SOFTWARE PACKAGE

SOLID 3000 is a FORTRAN callable surface rendering microcode package for the Adage 3000 diplay system. (Really folk's, I believe SOLID-3000 is more of a misnomer, SURFACE-3000 perhaps?) Unlike some systems on the market today, SOLID-3000 can accept direct surface input rather than polygon data (it will take that also). Surfaces may be based on B-SPLINE, BEZIER, CARDINAL-SPLINE or other basis functions of up to 8th order. As far as display functions provided, hidden surface shading, highlighting, transparency, anti-aliasing, and curved line display are just a small sample of the functions provided. The software also provides an easy interface to modeling programs. Since SOLID-3000 is a microcode package, it is very fast (I believe it was written with ICROSS).

NEW & NOTEWORTHY

The following is a summary of the survey which was distributed in the last AGUS newsletter, based on a response rate of 14 from the entire Adage user community. This may be due to the fact that the mailing list used in the last newsletter was out-dated and incomplete. Please return the form on the back of the newsletter to keep our list current.

NAME & ADDRESS	EQUIP	HOST/OS	APPLICATION	S/W EXCH.	AGUS FUNCTION
WILLIAM B COWAN Div. of Physics National Research Council of Canada Ottowa, Ontario K1A OR6 Canada 613-993-2504 watmath!watcgl! nvccgc!cowan	3000(1)	PDP-1123 RT-11	Psychophysics of vision	Yes	Communications for application exchange
DON MEAGHER Phoenix Data Sys. 80 Wolf Road Albany, NY 12205 518-459-6202(51)	3000(7)	VAX-11/ 750 VMS	Solids Modeling	No	Equipment news updates
BRUCE A. WALLACE R/Greenberg Assoc. 240 Madison Ave. New York, NY 10016 212-684-7886(42)	3000(1)	VAX-11/ 750 VMS	Entertainment graphics	None(yet)	SIGGRAPH mtg., List of users
ANDREW B. WATSON MS 239-2 NASA Ames Res. Ctr. Moffett Field, CA 94035 415-965-6290	3000(1)	PDP-1123 RT-11	Vision Research	No	Information Exchange
JOSEPH L. OLD MS 494 NASA/LRC Hampton, VA 23665 804-865-3457	3000(1)	VAX 11/ 780 VMS	Flight NAV Displays	No	Information Exchange (concepts, techniques, hardware)
MARK DIPPE 573 Evans Hall Computer Science Div., Dept. of EECS Univ. of CA Berkeley, CA 94270 415-642-0344	3000(1)	VAX 11 750 UNIX 4.2 BSD	Image Synth. (Painting & CAD too)	Maybe	

CONTINUED

NAME & ADDRESS	EQUIP.	HOST/OS	APPLICATION	S/W EXCH.	AGUS FUNCTION
STEVEN G. SATTERFIELD CAD & Interactive Graphics Group US Naval Academy Annapolis, MD 21402 301-267-4413 decvax!brl-bmd!usna!steve	3000(1)	VAX-11/780 UNIX 4.2 BSD	CAD	Not Yet	User Cooperation and Information Exchange
R.C. DURFEE E.P. TINNEL S.M. MARGLE G25,4500N P.O. Box X Oak Ridge, TN 37830 615-574-7449	3000(1)	MODCOMP IV (MAX IV)	Geographics Information Systems	No	
PAUL HUGHETT Dragoncraft P.O. Box 60 Palo Alto, CA 94302 415-327-7285	3000(1)	Symbolics LN-2 (LISP)	Computer Animation	No	User Communications
J.G. MIRBACH Lockheed Electronics Co. 1501 US HGWY 22 Plainfield, NJ 07061 201-757-1600	4250 6-CS 11-DS	Omega 480-2 VM	CADAM	No	
DOUGLAS S. KAY Joblove/Kay, Inc. 1545 N. Wilcox Av. Suite 201 Hollywood, CA 90028 213-461-6424	3000(1)	PDP-1123 RSX11M v3.2	Computer Animation for film	No	Info. Exchange (problems, techniques future equip.)

CONTINUED

NAME & ADDRESS	EQUIP.	HOST/OS	APPLICAITON	S/W EXCH.	AGUS FUNCTION
PENELOPE SHERRARD Ship Analytics Inc. North Stonington Professional Ctr. Rts. 2 and 184 N. Stonington, CT 06359 203-535-0990	3000 (18)	VAX-11/ 780 VMS	Training Simulators	No	H/W, S/W error exchange consulting personnel pool
KEITH L. MCVICAR Sikorsky Aircraft Dept. CAD/M N. Main Street Stratford, CT 06602 203-386-5196	3000 (1) 4250 (38) 4250R (3) 4370 (15) 4135 (11)	IBM 3081K MVS	Image Synth. (CAD) CADAM,CATIA other CAD	None Avail.	Information exchange bug fixes user coop.
NICK ENGLAND 531 Pylon Drive Raleigh, NC 27606 919-833-5401 decvax!duke!mcnc! ikonas!jne	3000 (2)	VAX-11/ 750 VMS	S/W Dev.	through Adage Inc.	Trade S/W opinions, helpful hints

In conclusion, it would be helpful if any other users out there would like to respond (how about some 4000 users!!!). Please call me or send information to above address. It seems that this group is turning into a 3000 users group (this may show the way of the future!). Communications is the most wanted function of this group, so I suggest that we try to arrange the aforementioned central data pool at the Adage VAX. A network of PC's might be nice too. Any suggestions on this would be appreciated. I do not believe that it is necessary at this time to have a software "library", however, this may become reasonable in the future.

MA1024 ADDRESSING PROTOCOL ON THE ADAGE 3000
(or, INFERNAL DEVICE EATS TWO WEEKS OF ENGINEERING TIME!)

Daniel H. Miller
Intermetrics, Inc.

The following information should be of interest to all MA1024 users (the MA1024 is the matrix multiplier on the 3000). There appears to be an undocumented requirement for addressing the MA1024, and if you don't know about it, you can waste a good bit of time searching for a bug in your "technically correct" program (believe me, I know!). In this article, I will discuss the requirement, and its consequences with respect to microcode programs, ICROSS and other high-level language programs, and debuggers.

MA1024 addressing requirement:

Simply, the UDR (upper data register) must be equal to 204 (octal) when a microcode WRSTRTMA instruction is issued to start the MA1024. In order to see why this is so, a brief review of the usage of the MA1024 is necessary. Assuming the MA1024 has been loaded with the standard Adage microcode, section 10 of the RDS 3000 Prog. Ref. Manual states:

The steps in using the MA1024 microcode routines is as follows:

- o Load the coefficient list or matrix into coefficient memory...
- o Load the points to be transformed into the SR-8 memory...
- o Load the MA1024 control registers...
- o Start the MA1024. Use a function code of 25 to store a 0 into address 20402%7.

The final step can be accomplished by issuing a WRSTRTMA instruction, which the BPS-32 Prog. Guide states "performs WRITE operation and starts execution of the MA1024". Thus, if you place the address of the last MA1024 register you desire to write in the MAR (memory address register), , and the desired value in the MDR (memory data register), you can then issue a WRSTRTMA to write the data and start the MA1024. In fact, WRSTRTMA apparently does two simultaneous writes: a "normal" write using the MAR and MDR as expected, and a "special" write (to 20402%7) to start the MA1024. One would expect that the address 20402%7 would be "hardwired" into this latter write, but this does not appear to be the case; the special write seems to use the current value of the UDR as the upper portion of the address, with only the lower portion being "hardwired". If you follow the steps exactly as given above, you will have no problem, because you will be writing to the MA1024 control registers just before you issue the WRSTRTMA, and the UDR will therefore contain the proper upper address. However, if you access the scratchpad after setting the MA1024 control registers and before the WRSTRTMA, you will have set the UDR to 202 (octal). The WRSTRTMA will do the normal write correctly, but the special write will not be able to access the register at 20402%7 and the MA1024 will not start. In general, after issuing a WRSTRTMA, programs wait for the MA1024 to finish before going on by looping on the CCMAAC (MA1024 busy) condition. If you have made the above error, the MA1024 never started so will not be busy, and the program will immediately break out of the loop.

CONTINUED

Microcode consequences:

Once you are aware of the above problem, it is easy to avoid errors in a microcode program by loading the MA1024 control registers just before the WRSTRTMA (i.e. set the stopbit earlier). If, for some reason, you need to do other operations just before the WRSTRTMA, you can protect yourself by resetting the UDR:

```
<load MA1024 control registers>
<other operations...>
LDUDR 204        ;force UDR = 204 (octal)
WRSTRTMA
```

ICROSS and other high-level language consequences:

ICROSS is an Intermetrics product that compiles and assembles a C subset into Adage machine code. The advantage of ICROSS, or any other high-level language you may be using, is that you do not have to worry about the details of the microcode. In this one instance, however, the advantage turns into a disadvantage; you do have to keep track of the UDR. The best universal solution is again to load the MA1024 control registers just before starting the MA1024. In ICROSS for example:

```
MA1024_CREG(0) = 0;           /* fill MA1024 ctl. reg. 0 */
MA1024_CREG(1) = 0x02000100; /* fill MA1024 ctl. reg. 1 */
STARTMA;                      /* Does a WRSTRTMA         */
```

The key is to make sure nothing occurs to produce unexpected microcode just before the WRSTRTMA. Subtle errors can creep in. For example:

```
for (i=0; i<=3; i++)
        MA1024_CREG(i) = PREDEFINED_REG(i);
STARTMA;
```

This seems to be correct at first glance. Note, however, that after each MA1024 control register is loaded, the generated microcode will obviously have to check the value of the loop counter i against the limits in the "for" statement. Suppose i is stored in the scratchpad - the microcode will have to fetch it to make the comparison, and will of course have to set the UDR to 202 to access the scratcpad! As a long term solution, it may be best to have any high-level language "start the MA1024" instruction compile into two microinstructions:

```
LDUDR 204;
WRSTRTMA;
```

Debugger consequences:

The behavior of the WRSTRTMA instruction should be kept in mind when using Adage's IKDEB, Intermetrics' I2DEB, or any pther debugger. Suppose you have the following correctly written microcode:

```
<load the MA1024 control registers>
WRSTRTMA;
```

and you desire to change the target/source offsets (MA1024 control register 1) for testing purposes. If you halt the program at the WRSTRTMA instruction, and manually reset the register, will the MA1024 perform correctly when the program is restarted? This actually depends on the order in which the registers were set in the statements above the WRSTRTMA instruction. If MA1024 control register 1 was the last register to be set, the MAR and the MDR will contain its address and the old data, and the normal write that the WRSTRTMA instruction

CONTINUED

Debugger consequences (cont.):
does will reset the register back to the old value! In effect, the MA1024 will appear to be ignoring your efforts to reset that register. In order to do your testing in cases like this (where you intend to override the last register load), you should either load the MDR with the desired new value (if your debugger allows this), or halt the program just before the last register load instruction, reset the program counter to the WRSTRTMA instruction, manually reset the register, and restart the program (the next to last MA1024 control register will be loaded twice). Note that in general, if you halt the program at some other unrelated location, reset the program counter to the WRSTRTMA instruction, manually load the MA1024 control registers, and restart the program, if the UDR does not happen to contain 204(octal), the MA1024 will not start.

Conclusions:

The ideal solution, of course, would be for Adage to modify the WRSTRTMA instruction execution such that the upper part of the address is also hardwired. Whether the instruction should also do a normal write is debatable. But until this fix is made (if ever), you can avoid much frustration and possible insanity if you remember:

1. A WRSTRTMA will not start the MA1024 unless the UDR = 204 (octal).
2. A WRSTRTMA writes the MDR to the MAR address.

Halftoning Algorithms Implemented on the Ikonas

Linda A. Adlum
Computer Aided Design and Interactive Graphics Group
U. S. Naval Academy
Annapolis, Md. 21402

(301) 267 - 4413
decvax!brl-bmd!usna!linda

April 16, 1984

The halftoning program was written to demonstrate several halftoning algorithms and, in addition, the capabilities of the Ikonas graphics system. The program was written in C and runs on a VAX 11/780 with the UNIX operating system.

Halftoning is a technique in which a picture of a number of varying intensitiesm whether color or black and white, is displayed on a device which is limited to the display of fewer intensities. The purpose of halftoning is to increase the visual resolution if the picture on such an output device.

The halftone program presently processes only black and white pictures. The picture is stored as a binary file of 256 or 512 scan lines. Each scan line is 256 or 512 bytes, respectively. A byte holds an intensity value from 0 (black) to 255 (white). Because an Ikonas pixel is made up of three bytes for rec green, and blue data, the program reads the monochrome byte and stores it as the red byte of the pixel. The green and blue bytes of the pixel are set to zero. A scan line displayed on the Ikonas is made up of 256 or 512 pixels depending on the resolution of the picture.

The Ikonas system is initialized with the channel crossbar switch (on the LUVO) set to full color. Displaying a pixel with only the red byte containing an intensity value and the green and blue bytes set to zero, produces a red tinted picture. This is because the input red, green and blue channels are sending data to the output red, green, and blue channels, respectively. Therefore, the program sets the channel crossbar switch to red pseudocolor. The monochrome data is sent not only to the red output channel, but also the green and blue output channels. The picture is displayed in black and white.

Four halftoning algorithms are available. The user may select the appropriate algorithm from a menu displayed at the user's terminal. After displaying the picture according to the chosen algorithm, the program returns the user to the menu for another selection or to quit the program.

The four halftoning algorithms are Simple Thresholding, Floyd and Steinberg error distribution, ordered dither, and 2 X 2 binary patterning for 256 X 256 pictures.

The Simple Thresholding algorithm displays the picture using only two intensity values. One value is the minimum (0 or black) and the other value is the maximum (255 or white). The user is prompted to enter a threshold intensity value between 0 and 255. The picture is scanned "on the fly" from top to bottom and left to right one scan line at a time. If the intensity value is greater than the threshold, white is displayed otherwise black is displayed. The algorithm produces errors; i.e. the details of a picture which are usually depicted through multiple intensity values are not shown because the algorithm is limited to the display of only black and white intensity values.

CONTINUED

Halftoning Algorithms Implemented on the Ikonas (cont.):

The Floyd and Steinberg error distribution algorithm corrects the loss of detail by computing the error for each pixel intensity and distributing the error to adjacent pixels. Three-eights of the error is added to the intensity value of the right adjacent pixel and the bottom adjacent pixel, and one-fourth of the error is added to the intensity value of the diagonally downward adjacent pixel. The user is prompted for a minimum intensity and a maximum intensity. For example, intensities entered may be black and white respectively. A threshold value is computed by adding the minimum and maximum and dividing by two. Scanning top to bottom, left to right, if the intensity value of the pixel under consideration is greater than the threshold, the maximum intensity is displayed. The error is computed to be the intensity less the given maximum; otherwise, the minimum intensity is displayed and the error is computed to be the intensity less the given minimum. This error is then distributed. The program stores all intensity values in an array. If memory resources are limited, instead of storing all intensity values, another method would be to store only two scan lines at all times; i.e., read ahead one scan line in addition to the current scan line being processed. This technique would supply the required adjacent pixel intensities which receive the error distribution.

The ordered dither matrix algorithm, also reads in all the intensity values into an array. In this case, storing the picture is not necessary and it may be scanned as read in. The user is prompted for the order of the dither matrix 2, 4, or 8. An ordered matrix larger than 8 is possible. The ordered matrix is filled using a recursive formula with an initial given 2X2 optimum matrix. The order of the maximum dither matrix may be made variable; however, the program presently contains a maximum ordered dither matrix of 8 stored in a static array. The ordered dither matrix represents a random error pattern. Before storing the intensity values, the values are scaled to the square of the user specified order. As an example, for a 8X8 matrix, values are scaled between 0 and 63. Again, the picture is scanned from top to bottom and left to right. The X,Y coordinate of the pixel under consideration is included in the computation of the row and column indexwhich is used to look up the dither value in the ordered dither matrix. If the scaled pixel intensity value under consideration is less than the dither value, black is displayed otherwise white is used. The algorithm produces a checkerboard pattern on the display. For an 8X8 dither matrix, 64 intensities are possible.

The patterning algoritm with a 2X2 pattern cell expands a 256 X 256 resolution picture to 512 X 512. The cell is a 2X2 matrix where each element of the matrix is considered a pixel. The intensity values are described using the cell. For a 2X2 cell, five intensity values are available.

The pixel patterns representing the five values are:

(1) all elements of the matrix are set to black;
(2) the (1,2) element white, rest black;
(3) the (1,2) and (2,1) elements white, rest black;
(4) the (2,2) element black, rest white;
(5) all elements of the matrix set to white.

CONTINUED

Halftoning Algorithms Implemented on the Ikonas (cont.):

It is important to choose patterns so as not to distort the pictures with incorrect lines. Note, pattern cells greater than 2X2 which provide additional intensity patterns are possible. Again, the picture is scanned "on the fly" from top to bottom. For each scan line read, two scan lines of 512 pixels are displayed. The 256 intensity levels are divided up into 5 intensity ranges corresponding to the above 5 pattern cells. Depending on what range the intensity falls in, a pattern is chosen and the cell is set accordingly. Note that a cell spans two scan lines, i.e. 256 cells are equivalent to two scan lines.

With the exception of the Simple Thresholding algorithm, these algorithms may also be implemented with color pictures. Additional information is contained in the reference given below. This program is available for distribution. Please submit a letter (on your letterhead) requesting the program along with a blank 9-track tape and return postage paid mailing label. The program will be returned in 1600 BPI UNIX 4.2 Tar format. Alternately, include your uucp address in your letter. The program will then be mailed to you via uucp.

REFERENCES:

Rogers, David F., Procedural Elements for Computer Graphics, McGraw-Hill Book Company, New York, 1984.

Foley, J. D. and Van Dam, A., Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, Reading, Massachusetts, 1982.

Newman, William M. and Sproull, Robert F., Principles of Interactive Computer Graphics, second edition, McGraw-Hill Book Company, New York, 1979.

FOR YOU!

M E M B E R S H I P U P D A T E

Please take a moment to return this form to facilitate our mailing list program in the handy, postage paid reply envelope enclosed for your convenience. Please pass this on to anyone else who may be interested. Thank You.

NAME _____

TITLE _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE _____ I AM A NEW MEMBER: YES ___ NO ___

TYPE OF ADAGE SYSTEM USED: 3000 _____ 4250 RASTER _____

4250 VECTOR _____ OTHER: (Explain) _____

FOR A FRIEND!

M E M B E R S H I P U P D A T E

Please take a moment to return this form to facilitate our mailing list program in the handy, postage paid reply envelope enclosed for your convenience. Please pass this on to anyone else who may be interested. Thank You.

NAME _____

TITLE _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE _____ I AM A NEW MEMBER: YES ___ NO ___

TYPE OF ADAGE SYSTEM USED: 3000 _____ 4250 RASTER _____

4250 VECTOR _____ OTHER: (Explain) _____