

# Whole ALTO World Newsletter

## Technology and Tools

---

XEROX

March 31, 1978

### GENERAL NOTES

**MAINTENANCE NOTES** - A new subsection is being added to the Newsletter to circulate knowledge of specific hardware problems, solutions, and maintenance techniques. It's located between the **HARDWARE** and **SOFTWARE** subsections under **TOOLS**. While many of the items will be directed to the people that maintain the hardware, some will be of general interest to Alto users, so don't just skip over this section.

**MESA** - The information on MESA in the last Newsletter needs some clarification. Although MESA as a language is still evolving and should not be used outside SDD for tightly scheduled projects, it is a reasonably robust and complete system and will be highly compatible from the Alto to successor machines. As such it should be seriously considered for both short and long term projects. Remember, the WAW coordinator, Frank Ludolph, is the MESA contact for non-SDD, non-PARC users.

### TOOLS

#### HARDWARE

**SEQUOIA BUILD** - This is the last opportunity to indicate your interest in purchasing a Sequoia from the proposed October build. Contact Sam Losh at XEOS now, Intelnet 8\*844-2501.

**HUSHING THE TRIDENT** - Jensen Engineering will custom build quiet boxes for the T80 and T300 drives. Externally the units will look identical. The prototype will be "Alto" grey but the production color(s) is not yet certain. Preliminary price estimates are \$450 to \$500 for the T80 box and \$350 to \$400 for the T300 box. (The T80 box costs more because a platform is necessary to raise it to the same height as the T300.) If interested please send a message to Barbara <BAIRD>; quantity orders will lower the price. *Don't wait for an "iffy" second build.*

**BUILDING REGULATIONS AND THE ETHERNET** - The 1978 National Fire Code states that electrical wiring within the environmental airspace must be within conduit. In a building with a drop ceiling, the space above the ceiling is part of the environmental air space if that space is used for air return by the airconditioning system. If both intake and exhaust ducting is used, the space above the ceiling is not considered a part of the environmental airspace. Specifically exempted is low-voltage communication cable provided the cable is approved for the purpose, i.e. it does not contribute to combustion. To be approved the cable must meet both smoke and flammability standards. Common polyethylene insulated coaxial cable is not acceptable but cable having a non-contaminating vinyl sheath has been approved.

## MAINTENANCE NOTES

**DIABLO 31 WRITE-HEAD CURRENT** - Because Diablo intended the Model 31 to be run at a slightly lower recording density than used in the Alto, the write-head current beyond track 128 may be insufficient for reliable operation. Diablo advises that the write-head current can be raised by altering the value of resistor H-64 on the J10 board, starting with 1K $\Omega$  and reducing if necessary. Resistor F-63, part of the same voltage dividing network, should not be cut as this drives the associated transistors to full on, altering their characteristics and causing splatter (the reason for the current cut in the first place).

**HARDWARE CHECKOUT BY THE USER** - There are several diagnostic programs that users can run to verify that the pieces of their Alto are running properly: DMT, CRTTEST, KEYTEST, and MADTEST. These are available from boot servers such as Gateways. To execute them, boot over the ethernet (boot while depressing the BS and quote keys) or type the 'netexec' command to the Executive. At this point the NetExec will appear on the screen. Type the name of the diagnostic and you're off and running. (Enter a '?' to list the boot files that can be called by the NetExec; the diagnostics listed above should appear. *Do not use DISKTEST. It's intended only for maintainers, requires documentation to use, and could overwrite a readied disk.*) CRTTEST and KEYTEST are used to checkout the workstation and are very simple to use (see below). DMT and MADTEST checkout the Alto itself and will be described next month.

CRTTEST draws parallel vertical and horizontal lines. The thing to look at is the sharpness of the lines (are they fuzzy?) and the shape of the boxes circumscribed by the lines. The boxes should be square, not tall or wide or diamond shaped (romboid). There will be a little distortion at the corners so don't worry about that. Depress the space bar (or any other key for that matter) and the lines will be redrawn with a different spacing; there are three different spacings. To quit, boot.

KEYTEST tests the action of the keyboard, keyset, and mouse. When the program begins, these items will be drawn on the screen (you may have to move the mouse to find it's diagram). It was recently enhanced to display either the Alto I or Alto II keyboard as appropriate. Depress each key, one at a time; the corresponding key on the display should turn black. If it stays white, or more than one key turns black, there is a problem. The latter is particularly a problem with the Alto II keyboard. To quit, boot.

## SOFTWARE

In general, the subsystems, packages, and documentation indicated here will be available from your local IVY server under the directories <Alto> and <AltoDocs>. If they are not available, or if you are in doubt as to the version, they may be retrieved from [MAXC] (same directories). Files stored under other directories are on [MAXC] unless otherwise indicated, e.g. [XEOS].

**NEW RELEASE: AISshow.run** - This new addition to the AIS (Array of Intensity Samples) system was written by Paul Roetling. As the name implies, the program displays AIS picture files on the Alto screen. If the image is larger than the screen, the image will be demagnified using a nearest neighbor algorithm. It operates on either 1 bit/pixel and 8 bit/pixel images; the 8 bit/pixel images will be displayed using a Floyd halftone routine. Demagnified 1 bit/pixel images may show substantial Moire patterns. Retrieve [WRC]<AIS>Subsystems>AISshow.run. The documentation, <AIS>Memos>AISshow.memo is appended to the Newsletter.

## ReReleases - Subsystems

**AISdump** - An error which printed some incorrect values for windows of odd length was fixed. The new version, 1.1, is available from [WRC]<AIS>Subsystems>AISdump.run.

**COPYDISK** - The "[thisHost]device" bug has been corrected and the default value of WRITEPROTECT is now TRUE. This subsystem is available from boot servers.

**DDS** - This new version conforms to the new time standard. Retrieve <Alto>DDS.run.

**DMT** - This new version conforms to the new time standard. Retrieve <Alto>DMT.run.

**EXECUTIVE** - This new version conforms to the new time standard, fixes a few bugs, and has some enhancements including the ability to load CHAT, FTP, Scavenger, and NetExec over the ethernet, a FILESTAT command to report file attributes, and a SETTIME that obtains the time over the ethernet (delete SetTime.run). Use <Alto>NewOS.cm to update this subsystem as well as the new OS/15 and FTP. The documentation, <AltoDocs>Executive.tty, has been revised.

**FTP** - The 21 March 78 version fixes more file date bugs and contains new features including a typescript of the user window, new commands (Open, Close, and Compare) in the command line, and improved command line error handling. This subsystem will be updated when installing the new operating system. See the revised documentation, <AltoDocs>FTP.tty.

**KEYTEST** - This diagnostic, available from boot servers, has been enhanced to display the correct keyboard, i.e. Alto I or Alto II. Retrieve <Alto>KeyTest.run only if boot servers, e.g. gateways, are not on your ethernet.

**LISTSYMS** - This new version conforms to the new time standard. Retrieve <Alto>ListSyms.run. The documentation, <AltoDocs>ListSyms.run has been revised.

**MICRO/MICROD** - A primary reason for the rerelease is to alleviate a space constraint which prevented certain diagnostics from assembling. Also the filename extension, .MC, is now defaulted rather than forced. Retrieve <Alto>MICRO.run. The changes document, <AltoDocs>MICRO.tty has been updated.

**MU** - A bug which failed to give an error message when a semicolon was left off the end of a predefinition was corrected, the filename extension now defaults to .Mu, and the listing file contains constants sorted by value as well as by address. Retrieve <Alto>MU.run and the revised documentation, <AltoDocs>MU.tty.

**NETEXEC** - This boot file has been enhanced to poll boot servers for the boot files they can supply, and has some new user command functions such as PROBE and HOST. No .run file need be retrieved. The new documentation is available on <AltoDocs>NetExec.tty.

**OPERATING SYSTEM** - This version fixes bugs in the file date code which caused the FTP update command to fail. Retrieve and execute <Alto>NewOS.cm. Verify that there are at least 300 free pages on your disk before executing the command file. This will also update FTP and Executive.

**PEEK** - This new version conforms to the new time standard. Retrieve <Alto>PEEK.run.

# Whole ALTO World Newsletter

4

**PRESSEEDIT** - The experimental version reported last month has been officially released. It enables the merging of one page graphics onto a text document at any location on the document page, so now it's an easy matter to edit text after inserting graphics (re-edit the Bravo file, make a Press version, and remerge). It also fixes bugs concerning Private Data Labels and very complex pages. Retrieve <Alto>PressEdit.run and the new documentation, <AltoDocs>PressEdit.tty.

**PROM** - The nature of the changes are unknown to me. Retrieve <Alto>PROM.run.

**READPRESS** - The new version properly recognizes Press file elements that have just come into use. Retrieve <Alto>ReadPress.run.

**TRIEX** - This release incorporates the new time standard. Do not use it under OSs prior to version 14. Retrieve <Alto>Triex.run and updated documentation, <AltoDocs>TFS.tty.

**TFU** - This release incorporates the new time standard. Do not use it under OSs prior to version 14. Retrieve <Alto>TFU.run and updated documentation, <AltoDocs>TFS.tty.

## ReReleases - Packages

**PUPPACKAGE** - The new version contains changes to the BSP code that improve performance when communicating through Gateways and also fixes several bugs. Several of the modules have been broken into smaller pieces to permit substantial overlaying. The documentation, <AltoDocs>PupPackage.tty, has been updated.

**TIME** - The new version conforms to the new time standard. If you are responsible for subsystems that deal with time in any way, you are requested to begin converting such subsystems to use the new software. The backward compatibility measures that have been implemented in OS 14 will cease to work correctly on April 30, 1978, so it is desirable that the revised subsystems be released well before that time. Announcements of new releases should include a warning that they will not work under pre-OS 14 versions. Load <Alto>Time.dm. The packages is documented in <AltoDocs>Time.tty; the new time standard is described in <AltoDocs>AltoTime.Bravo.

**TFS** - This release contains the modifications necessary to conform to the new Alto time standard. It will run only under OS 14/15. Also, the microcode source file has been broken up to facilitate combining it with other microcode, however the microcode binary is unchanged. Load <Alto>TFS.dm. Updated documentation is available on <AltoDocs>TFS.tty.

## TECHNOLOGY

Most users are familiar with the printing of text and graphic information but there is a third type, imaginal. Continuous tone pictures (e.g. photographs) are of this third type. The problem of printing continuous tone images is essentially that of representing many shades of grey with only black ink on white paper. A solution of the printing industry was the halftone, a pattern (usually regular) of black dots which vary in size and, possibly, shape. Pictures in the newspaper are an example. Two papers are included, one which describes the halftone process and a second that presents the resolving limits of the eye, why we see groups of black dots as multi-grey tone images.

The first, *PRESS, Halftones, and You* by Joe Maleson, is something of a tutorial on halftones and the halftone process as implemented in Press. The Press specific information applies to Press 1. Press 2 has a different halftone screen pattern and no longer uses error distribution. The new screen pattern, similar to that used in the printing industry, has dramatically improved image quality. Joe has written a memo describing the halftone screen being used in Press 2. Since the electronic form does not contain the image and graphic information, contact Frank Ludolph by message, <Ludolph>, or phone, 8\*923-4356, for a copy.

There is an interesting problem in the printing of papers that demonstrate methods of image representation, namely that of accurately representing the image. It is the appearance, not the content of the image that is important. Since imaginal data can currently be printed at Xerox only on low speed, low volume printers and because of distribution volume, the Newsletter contains a copy made on the 4000 copier (with resultant image degradation). It is appended at the very end rather than in its usual place following this introduction. An original (from the Slot-3100) is being sent to each site for posting.

The second paper, by Paul Roetling, *Visual Performance and Image Coding*, presents data on the eye's ability to resolve what it sees and develops guidelines for determining how much information must be captured and retained in digital form to meet the eye's requirements. Continuous tone images contain a huge amount of data; retaining more than necessary severely impacts storage requirements and processing times.

Paul has written several other papers of a tutorial nature that contain continuous tone images. Check the library for the published versions because the image quality suffers in reproduction. The first describes several methods of representing continuous tone images, the second is a specific method for improving the appearance of halftones, and the third a discussion of factors affecting halftone image quality.

*Binary Approximation of Continuous Tone Images*, Photographic Science and Engineering, Vol. 21, No. 2, March/April 1977.

*Halftone Method with Edge Enhancement and Moire Suppression*, Journal of the Optical Society of America, Vol. 66, No. 10, October 1976.

*Analysis of Detail and Spurious Signals in Halftone Images*, Journal of Applied Photographic Engineering, Vol. 3, No. 1, Winter 1977.

Paul's paper, which originally appeared in SPIE/OSA Vol. 74 (1976) Image Processing, had to be rekeyed and the graphs redrawn for inclusion in the Newsletter. The editor accepts sole responsibility for its appearance and accuracy.

The Whole Alto World Newsletter is a monthly publication for Xerox employees that use the Alto. It is not to be shown to non-Xerox people. Copies are available on [MAXC]<AltoDocs>WAWnews.press or may be obtained from the editor, Frank Ludolph, XEOS, by messaging <Ludolph> or calling Intelnet 8\*923-4356.

To	Distribution	From	Paul G. Roetling Mgr., Image Processing Area W128/22037
Subject	New Program AISshow.run Version 1.0	Date	March 7, 1978

A new ALTO program, AISshow.run Version 1.0, is available from the WRC ivy system under <AIS>SUBSYSTEMS>AISshow.run. As the name implies, this program is intended to allow a person to examine AIS picture files by presenting them on the ALTO screen.

### **Program Operation**

When you run the program, the first screen display will come up as a menu similar to that used in AISmagnify and AISdump (if you're not familiar with using the menu, look in the documentation for AISmagnify). All that is needed is the picture file name, the window can be defaulted. Selecting start will remove the menu from the screen and start showing the picture file. If the picture window selected is larger than will fit on the screen, the image is demagnified (by a nearest neighbor routine) so that the complete window shows on the screen. If the selected window fits on the screen, it is shown full size without magnification, and a notation appears above the image stating that the window is shown full size. As with magnify and dump, the program operates on either eight bit or one bit per pixel files. If the file is eight bits per pixel, a Floyd halftone routine is used to show the image. One bit-per-pixel images are shown without halftoning. The photometry sense is noted from the attributes and is used to show the image in the "correct" sense. A cautionary note should be added here. Demagnification of bit-per-pixel halftone images may cause substantial moire patterns. Do not assume that the moires are really in your picture unless the picture has the notation that it is being shown at full size, in which case you are seeing the correct image.

### **Display control**

As the image is in the process of being displayed on the screen, if you realize you do not want that image, or already have enough to see what you wish to see, typing a control-S will terminate additional display to the screen, leaving the portion of the picture which it has already displayed. In its current form, it sometimes takes a long time to produce a large image in reduced form, thus this is a short cut.

Once the picture is completely displayed on the screen, (or stopped by control-S), a set of instructions will appear above the picture on the screen. These are various key-in possibilities:

I - will invert the sense of the screen. Use this if your impression of "correct" is different from that in the photometry section of the file. You can tell when you are opposite to the normal photometry because the rest of the screen will be black instead of white.

Carriage return - will remove the picture and return to the menu.

Q - will quit the program without returning to the menu.

Escape - will reset the window parameters to full picture, remove the old display and start displaying the complete picture without returning to the menu.

M - is used to show the area selected by the mouse (see below).

### Selection by mouse

If, while the image is displayed, you point the cursor to one corner of an area you wish to see, depress the left mouse button, move the cursor to the diagonally opposite corner of the area you wish to see displayed, and release the left mouse button, a box will appear surrounding the area you have just selected. If you wish to change the selection, simply repeat the process and a new box will appear after you release the left button. If you try to select an illegal area, for example, outside the area shown, no box will appear and any previous one will be erased. If there is a box on the screen at the time M is keyed, the area outlined by the box is used to define the new window both for the display and for writing into the window parameters in the menu.

In our limited use of this program, we have found this last feature to be very useful. A file name is selected, and the full picture is shown initially. We then use the mouse to select the area we wish to see, key an M to see the region, repeat the process to window down further, until we get exactly the area we want. We then use carriage return to return to the menu and copy down the window parameters for use in other programs as for example, magnify.

The primary purpose of this program is to allow us to experiment with user interaction both for image display and window selection. Thus, we expect that portions of the program will be revised frequently as we conduct experiments. We would also appreciate comments from users on their reactions to program features. Please contact either Keith Knox or myself with comments.

PGR/sm

c: JEBollman  
TMHolladay  
JEstinehour  
LBHolt  
DEDamouth  
WMReilich  
LDMailloux  
KTKnox

# VISUAL PERFORMANCE AND IMAGE CODING

Paul G. Roetling

**ABSTRACT:** Sample spacing and quantization levels are usually chosen for digitizing images such that the eye should not see degradations due to either process. Sample spacing is chosen based on the resolution (or high frequency) limit of the eye and quantization is based on perception of low contrast differences at lower frequencies. This process results in about 8 bit/pixel, 20 pixel/mm digitization, but, being based on two different visual limits, the total number of bits is an overestimate of the information perceived by the eye. The visual MTF can be interpreted in terms of perceptible levels as a function of spatial frequency. We show by this interpretation that the total information perceived by the eye is much less than 8 bits times the number of pixels. We consider the classic halftone as an image coding process, yielding 1 bit/pixel. This approach indicates that halftones approximate the proper distribution of levels as a function of spatial frequency; therefore we have a possible explanation of why halftone images retain most of the visual quality of the original.

## I. INTRODUCTION

In this paper we consider the problem of how visual performance characteristics can be related to the average number of bits per pixel (picture element) in a sampled and quantized image. If we establish an average number of useful bits per pixel in the sense that only these are used by the eye, we then have a number against which to compare the efficiency of various image coding or bandwidth reduction schemes for cases where system performance is related to the visually perceived image quality.

It is well-known that visual performance is a function of spatial frequency. That is, at high spatial frequencies we do not see as well as at lower spatial frequencies and, without magnification, we can see no detail at spatial frequencies above about 10 line pairs per millimeter. By examining how visual performance varies as a function of spatial frequency, we should be able to establish guidelines for image coding experiments as to which pictorial information is useful to the eye.

In the following sections we will examine the selection of sampling interval and quantization levels based on visual performance, thereby establishing design guidelines and a useful number of bits per pixel as a function of sampling interval. We then examine a simple halftone binary-image code and a similar code applied to multilevel images and estimate the performance of these simple codes on the basis of established limits.

## II. SAMPLING AND QUANTIZATION CHOICES

To bound our problem we first assume that the images will be examined at unity magnification at normal reading distance. We lose no generality by this assumption, since all results can be scaled by the magnification. We assume ideal sampling, that is, we ignore sampling aperture effects and display spot (Kell factor<sup>1</sup>) effects. We can then assume that if we sample at an interval  $\Delta$  we can adequately represent pictorial information to a spatial frequency  $(2\Delta)^{-1}$ .

A simple example serves to show how we use visual performance limits in everyday estimates of coding efficiency. Let us consider the case where the image is sampled 40 times per millimeter. By our assumptions, and the fact that the eye does not see detail beyond 10 line pairs per millimeter, we lose no visible information by taking alternate samples (or perhaps averaging every pair of samples). We have therefore reduced the total number of



samples in the two dimensional image by a factor of four without losing perceived image quality. Most of us would argue that this four-to-one bandwidth reduction should have been obvious, because the information eliminated by the bandwidth reduction was information not being used by the visual system. Thus, we tend to judge the bandwidth reduction in terms of the visually useful information in the original.

To count the useful bits in a picture, we would certainly limit the sampling interval to approximately 20 per millimeter. Similarly, we would limit the quantization levels to about 8 bits (256 levels) since we know that the eye cannot perceive more levels in the output picture. Nevertheless, if we were to multiply 8 bits per pixel by the number of pixels (20 per millimeter sampling interval) we get a considerable overestimate of the number of useful bits in the image. We have failed to take into account how visual performance varies with spatial frequency. In different words, we have established the sampling interval based on the high frequency performance of the eye and the quantization levels based on the low frequency performance of the eye, without taking into account the translation from one to the other.

#### A VISUAL MTF

We describe an improved approximation to the determination of visually useful bits of information by basing ourselves on known psychophysical data on visual performance in terms of the modulation transfer function (MTF). Text books, such as Cornsweet's<sup>2</sup>, consider various sources of such data. Dooley<sup>3</sup> gives a convenient functional form which approximately fits most of the vision data. His fitted curve, in terms of modulation transfer function (MTF), is given by

$$\text{MTF} = 5.05 (e^{-0.138f}) (1 - e^{-0.1f}). \quad (1)$$

This curve has been normalized and  $f$  is spatial frequency in cycles per degree. The peak of the curve represents a just detectable modulation of 0.005. The psychophysical data to which this curve was fitted were measurements of the just detectable modulation of a sine wave as a function of the spatial frequency of that pattern. It should be noted that we do not use the data in MTF form, but rather we determine directly the detectable contrast, i.e., we use the original form of the experimental data.

We now assume, for our approximation, that at every spatial frequency we should quantize levels such that the just detectable modulation represents one quantization step. If we further assume that, for all sine waves, the average luminance was mid-grey, then the modulation can be written

$$M = \frac{\text{MAX} - \text{MIN}}{\text{MAX} + \text{MIN}} = \frac{\text{detectable difference}}{\text{total range}} \quad (2)$$

In other words, the number of intervals is the reciprocal of the just detectable modulation, and the number of levels we must represent is the number of intervals plus one. That is, we can write the number of detectable levels as a function of spatial frequency by taking out the normalization from the given MTF curve and adding one, or

$$\# \text{ of Levels} = 1010(e^{-0.138(5\nu)}) (1 - e^{-0.1(5\nu)}) + 1 \quad (3)$$

To convert from the cycles per degree measurement common to psychologists to the cycles per millimeter more common in image processing, we have used a conversion factor of one cycle per millimeter equals five cycles per degree. This curve has a peak at a spatial frequency of approximately one cycle per millimeter. There is evidence<sup>3</sup> that threshold measurements are not a good measure of visual performance for all contrasts at frequencies below one cycle per millimeter. To obtain a conservative estimate, we have used Eq. (3) only above one cycle per millimeter, using its maximum value below that point. This curve is shown in Fig. 1.

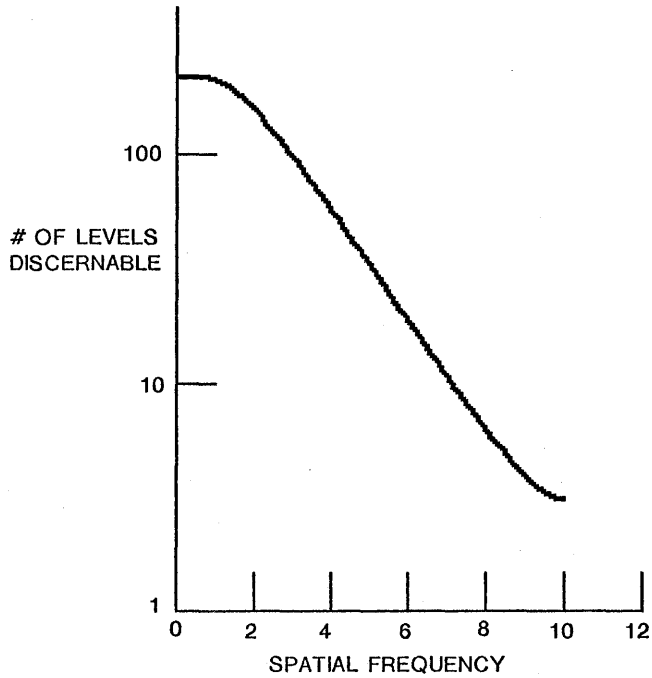


Fig. 1. Visual Performance Limits

*B. VISUALLY USEFUL INFORMATION*

We can now express visually useful information in a sampled and quantized picture as an average number bits per pixel, by utilizing the visual performance curve in Fig. 1. We take an image area  $L$  by  $L$ , sampled at an interval  $\Delta$  by  $\Delta$ , as shown in Fig. 2a.

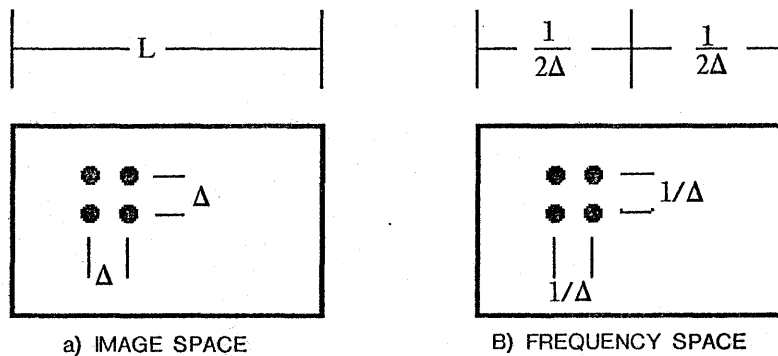


Fig. 2. Image and Frequency Sampling

The visual performance data is described in spatial frequency space. We therefore convert to the image transform space as shown in Fig. 2b. The defined transform space has a frequency range  $\pm 1/2\Delta$ , with frequency samples spaced at intervals  $1/L$ . In each space, the total number of samples is the same, that is,  $n^2$  equals  $(L/\Delta)^2$ . To arrive at an average number of bits per pixel, we evaluate the total number of useful bits and divide by the total number of pixels. In frequency space, we arrive at the total number of useful bits by

multiplying the number of bits per frequency sample ( $\log_2[\text{\#LEVELS}]$ ) by the number of samples per unit frequency interval ( $L^2$ ), times the frequency interval, and integrating over the frequency range. These operations are combined and expressed as

$$\text{\#bits/pixel} = \frac{1}{n^2} \int \int \{\log_2[\text{\#LEVELS}(\mu, \nu)]\} (L^2) d\mu d\nu \quad (4)$$

where  $\mu, \nu$  are spatial frequencies. Putting together the above relations between constants and the fact that the visual performance data has approximate circular symmetry we can rewrite the expression as

$$\text{\#bits/pixel} = 2\pi\Delta^2 \int_0^{\infty} \nu \text{MAX}\{\log_2[\text{\#LEVELS}(\nu)]\} \nu d\nu \quad (5)$$

Finally, for any high enough sample rate, we can perform a numerical integral approximating Eq. (5) which will give

$$\text{\#bits/pixel} = 2\pi\Delta^2(177.5) \quad (6)$$

If we now insert  $\Delta$  as approximately 20 samples per millimeter we find that the average visually useful information in an image is approximately 2.8 bits per pixel, substantially below the 8 bits per pixel which we would have estimated had we not taken into account the fall-off of visual performance with spatial frequency.

It is also of interest to note that if we decrease the sampling interval, that is sample more often, the useful information per pixel drops with the square of the sampling interval. We can therefore also calculate a sampling interval at which the average information per pixel would be approximately 1 bit. Substitution in Eq. (6) yields 1 bit/pixel at a sample interval of 33.4 per mm. This result states that, at that sampling interval, an efficient binary code might be found which could represent the image at no loss of visual quality. We therefore consider next simple binary image codes.

### III. HALFTONE CODES

One simple form of binary image code has been used for somewhat over a century<sup>4</sup> by the graphic arts industry. The binary images are referred to as halftones, and are used to give the appearance of grey while printing only full black and white. Such images are generated by combining a non-image related pattern (called the halftone screen) with the pictorial data by addition or multiplication. The combination is then subjected to a threshold to turn the continuous tone to a binary image. This process has been applied to sampled imagery by many authors (for example, see Ref. 5) and the image detail for given screen patterns has been described.<sup>6</sup>

We now ignore the design of the halftone screen to consider what optimal encoding might achieve. The halftone process is essentially one of trading off grey scale for texture. That is, a combination of black/white spots is generated which, when averaged over some area, give the illusion of various shades of grey. Again considering ideal sampling, we examine how a given periodic structure can be represented. Clearly, we must have available at least one sample for every half wavelength of the pattern to be represented. Fig. 3 shows a small region of the picture with two regions identified, each a half wavelength of the desired sample in length. To represent the sample of minimum modulation achievable at this spatial frequency, one bit must change between these two areas of image. This is equivalent to asking how many different levels can be represented by turning on different numbers of bits within the image area whose length is one-half wavelength for the desired spatial frequency. This can easily be seen to be

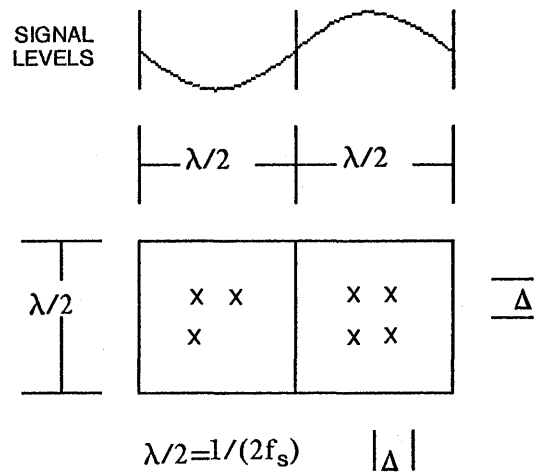


Fig. 3. Image Areas Available to Encode A One Grey Level Change

$$\# \text{ of LEVELS} = (1/(2F_s))^2 (1/\Delta^2) + 1, \quad (7)$$

where  $f_s$  is the object frequency, and  $\Delta$  the sample interval. It is convenient to immediately generalize this result by asking what would happen if each pixel could take on more than black or white values. If each pixel is described by  $m$  bits, rather than 1, each element now adds  $2^m - 1$  additional non-black values. Thus the same type of coding applied to a multilevel image yields the possibility of representing additional grey levels, given by

$$\# \text{ of LEVELS} = ((2^m - 1)/(2f_s \Delta))^2 + 1. \quad (8)$$

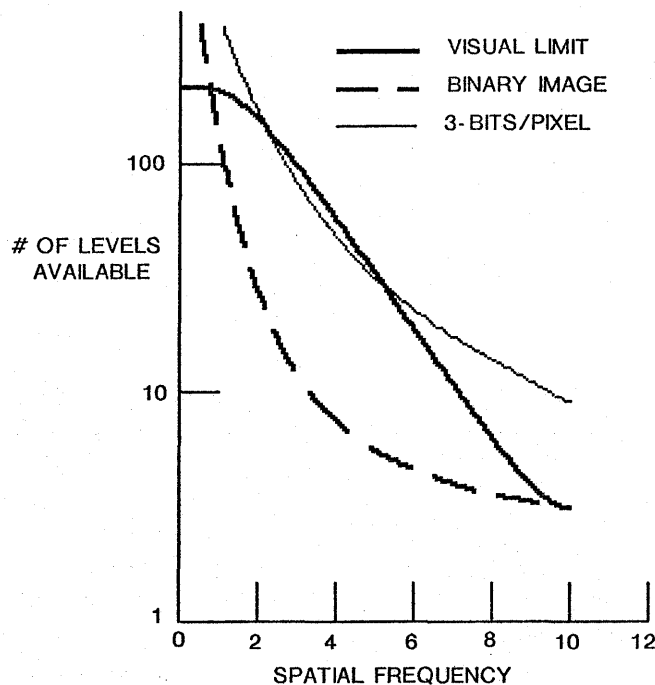


Fig. 4. Texture Code Limits

Fig. 4 compares the original curve from Fig. 1 with the number of levels achievable as a function of spatial frequency for the binary case with a sample rate of 20 per millimeter and for the case of 3 bits per pixel at the same sample rate.

It is interesting to note that the 3 bit per pixel curve indicates that a texture type code should be able to represent almost all information visible to the eye. Since we found that the actual useful information averages approximately 2.8 bits per pixel at this sample interval, we have an indication that the texture type code could be reasonably efficient. In the binary case, at this sample interval, we see that the 1 bit per pixel code falls short of that needed to avoid visible degradation of the image. The manner in which the curve falls with spatial frequency, however, gives an indication of why halftoned images look as good as they do, since the curve shapes are somewhat similar.

## IV. CONCLUSION

We have described an approach in which visual data for modulation transfer function of the eye can be utilized to determine the useful information in an image. At a sample interval of 20 samples per millimeter, we have found that the visually useful information corresponds to approximately 2.8 bits per pixel. The shape of the visual performance curve indicates that more levels need to be represented at lower spatial frequencies and less levels at higher spatial frequencies. Thus, it has been shown that halftone or texture codes, although simple, represent image information in a manner which tends to be compatible with the characteristic of the visual system.

## V. ACKNOWLEDGEMENTS

The author wishes to express his gratitude to his co-workers for many useful discussions which have helped to clarify the concepts described in this paper. In particular, D. Kermisch and K. Knox provided many helpful suggestions.

## REFERENCES

- <sup>1</sup>Costigan, D. M., *Fax*, Chilton Book Co., Philadelphia, Pennsylvania 1971, p. 127.
- <sup>2</sup>Cornsweet, T. N., *Visual Perception*, Academic Press, New York, New York 1971, p. 330-342.
- <sup>3</sup>Dooley, R.P., "Predicting Brightness Appearance at Edges Using Linear and Non-Linear Visual Describing Functions", presented at SPSE Annual Meeting, May 14, 1975, Denver, Colorado.
- <sup>4</sup>*Pocket Pal*, Tenth Edition, International Paper Company, New York, New York, 1970, p. 17.
- <sup>5</sup>Klensch, R. J. "Electrically Generated Halftone Pictures," *RCA Review* p. 517-533 (September 1970).
- <sup>6</sup>Kermisch, D. and Roetling, P.G., "Fourier Spectrum of Halftone Images," *Journ. Opt. Soc. Am.* 65: p. 716-723 (1975).