SGP Memo 9

DISCUSSION OF IMAGE DISPLAY SYSTEMS

Jim Torson
Version 4
November 30, 1981

1. INTRODUCTION

The purpose of this document is to discuss some aspects and features of image display systems that are currently commercially available. The items discussed are mainly those things which may be relevant to picking a display to be used on the "pipeline" system and which may not be immediately obvious. A complete discussion of all available features and options would be a much longer list.

Section 2 will discuss some of the things that can be done with image display systems that are currently available. This will concentrate on functions that the user of a system would see rather than on how the hardware works.

Section 3 will discuss some of the specific hardware features that are available in systems. This will describe how a given function can be accomplished in different ways by different hardware features or collections of features. There is often a need to consider the interactions amoung various features. A given feature considered by itself may not be very useful, but in combination with other features, it might be highly useful.

2. SOME FUNCTIONS THAT CAN BE PERFORMED BY IMAGE DISPLAY SYSTEMS

2.1. BLINK COMPARISION OF IMAGES

It seems that blink comparision of images would be primarily useful to us for the map display problem. This could be used to compare cleaned and dirty versions of a map. Or, it could be used for comparing two different frequencies. If more than two images can be blinked, it might be useful to show a sequence of maps which are different spectral line channels.

It has been suggested that blink comparision might be useful for comparing AA and CC visibility data.

2.2. HANDLING OF LARGE IMAGES

In older image display systems (such as the Comtal used by IMPS) the image stored in the refresh memory is the same size as the image that is displayed on the screen. Handling larger images can be accomplished by software. For example, when loading the refresh memory, the user can be asked to select the desired subsection of the input image to be loaded. Or, the large input image can be squeezed down to fit into the refresh memory by techniques such as averaging groups of pixels or loading only every N'th pixel. Some systems also allow you to scroll through a strip of the input image, e.g., as new lines are loaded into the top or bottom of the displayed image the existing lines are bumped down or up.

Some of the newer display systems allow storing an image that is larger than the displayed image. You can then dynamically select which piece of it you want displayed. Some systems can store up to a 2048x2048 8-bit image. The displayed window can range from about 512x512 to 1024x1024.

2.3. SIMULTANEOUS DISPLAY OF MULTIPLE VALUES FOR EACH PIXEL

In some cases we may want to have a displayed image which is showing two independent values for each pixel. Examples of this would be a display of visibility data which shows both amplitude and phase or display of map data which shows velocity and intensity from a spectral line observation. Some systems allow the three color guns of the CRT (red, green, blue or RGB) to be independently controlled. However, for our purposes, it would be more useful to be able to independently control the hue, intensity and perhaps saturation (HIS) of the displayed pixels. Some systems have this capability.

3. SOME HARDWARE FEATURES OF IMAGE DISPLAY SYSTEMS

3.1. USING THREE STORED IMAGES TO CONTROL RED, GREEN AND BLUE - "TRUE COLOR" MODE

A number of different vendors can supply systems in which three separate stored images independently control the red, green and blue color guns of the CRT. Most of these systems have a separate look up table (LUT) for each image or color. Some of these systems have a fixed correspondence between images and colors. However, some systems allow software selection of which image controls which color. Also, it may be possible to route one stored image to more than one color. In this case, the question arises as to whether there is a separate LUT corresponding to each image or whether there is a separate LUT corresponding to each color. Some systems may even allow more complex LUT configurations.

3.2. USING THREE STORED IMAGES TO CONTROL HUE, INTENSITY AND SATURATION

Only a single vendor offers systems in which three stored images control the hue, intensity and saturation of the displayed image. They do this by using a piece of hardware in the refresh data path which converts the hue, intensity and saturation values for each pixel into the red, green and blue values that drive the monitor. This capability would probably be much more useful to us than the direct control of RGB. However, one option might be to purchase an RGB system and then build our own box to do the conversion from stored HIS values to RGB.

Some systems may be flexible enough to be operated in either RGB mode or HIS mode. For example, the IIS has a very flexible LUT configuration. Associated with EACH color gun of the monitor is what they call a "pipeline processing channel." Among other things, each of these includes a LUT associated with EACH image refresh memory. The outputs of the LUT's are added together (selectively) and then put through an output LUT before going to the monitor. This system can clearly be operated in RGB mode. In principle, this system could operate in hue/intensity mode as follows. The three LUT's associated with the first image are loaded with the logarithms of the functions that would produce the desired range of hues. The three LUT's associated with the second image are loaded with logarithm functions and the output LUT's are loaded with antilogarithms. The hues determined by the first image are thus effectively multiplied by the second image. More thought about the details of this would be needed to determine if this would work satifactorily.

3.3. STORING MORE THAN 8 BITS PER PIXEL IN A SINGLE IMAGE

Some of our maps have a dynamic range that is greater than 8 bits. Thus, if the display refresh memory has only 8 bits per pixel, full examination of the data may require reloading the refresh memory with different scalings of map values into display pixel values. (Or, perhaps a logarithmic transfer function could be used when the image is loaded.) Storing more than 8 bits per pixel in the refresh memory would allow us to just change the LUT values to examine different parts of the dynamic range in the map data. This would be faster and nicer, but probably not necessary.

However, by loading appropriate values into the LUT, we could effectively divide up a single stored image into two separate images which control the hue and intensity. Doing this with an 8 bit per pixel system would probably not work very well because you would only have 4 bits (16 different levels) for each "sub-image". Perhaps a total of 12 bits per pixel would be needed so that you could use it as two 6 bit images (64 levels each) or as a 5 bit and a 7 bit image. In order for this to work well, the LUT would need to have at least 8 bits for specifying each color gun intensity. Most (but not all) systems have this. Also, using this technique would require that the number of input address bits to the LUT be as large as the total number of bits per pixel. For 12 bits this would mean that the LUT would have 4096 entries. This might make rapid modification of the LUT difficult and inefficient since we would be using a 4096 element LUT to accomplish what could be done by two separate 64 element LUT's.

Appropriate loading of the LUT can also effectively blink compare two "sub-images".

Some systems which store multiple 8-bit images may have features which allow two images to be combined to effectively give more than 8 bits of dynamic range in the stored image. One way this might be done is to allow arithmetic processing in which two 8-bit images are combined to form a 16-bit value. The processing could take the 16-bit value, scale it into 8 bits and store the result into a third image memory. A LUT with an 8 bit input would then be sufficient.

3.4. STORAGE OF A 1024x1024 IMAGE

Some systems allow storage of a 1024x1024 image with the entire image displayed on the screen. This would of course be nice for handling 1024x1024 images. However, without a zoom capability, the system would not be very good for handling a lot of 512x512 images.

If a 1024x1024 display system has zoom, then a 2x zoom will fill the screen with a 512x512 piece. The image would be a little "blocky" compared to a system that just displays a 512x512 image, but this would probably not be too important. If the roam location can be quickly changed, then this system could be used to blink compare two 512x512 images. However, proper blink comparision may require that the two images be displayed through different LUT transfer functions. A system that stores just a single 1024x1024 image would probably only have a single LUT, so it may be difficult to use it for blinking two 512x512 images. If a 1024x1024 refresh memory system is to be used for handling 512x512 images, it would be useful (although not essential) to be able to load partial lines in the refresh memory.

Some systems store a 1024x1024 image but only display a piece of it with ability to roam around with the display window. If the displayed piece is 512x512 and the roam can be quickly changed, then the system can be used for blink comparision of 512x512 images as can the previous system. However, some systems may display a piece that is bigger than 512x512, e.g., 640x480 or 880x704. These would not be very good for blinking 512x512 images unless the 512x512 pieces could be appropriately positioned on the screen with blanking of the areas outside the desired 512x512.

As described above, a system that stores a 1024x1024 image could probably be used as if it were storing multiple 512x512 images for display or maybe even for blink comparision. However, such a system could probably NOT be used as if it were storing multiple 512x512 images for independent control of RGB or HIS. This is due to the fact that at any given instant, only a single piece of the image is being read out for refresh. That is, there is only a single data path through a single LUT from the refresh memory to the monitor.

3.5. USING 4 512x512 STORED IMAGES AS A SINGLE 1024x1024 IMAGE

Some systems which are organized around storage of 512x512 images will allow using four images together as if they were a single stored 1024x1024 image. The displayed image will then effectively be a 512x512 window that can be roamed around on a stored 1024x1024 image. One system even allows using 16 512x512 images together as if a single 2048x2048 image were being stored.

An advantage of this organization would be that it would be more likely to have separate LUT's for each 512x512 to facilitate blinking. Also, such a system may be able to simultaneously read out more than one image from the refresh memory to allow independent control of RGB or HIS.

A disadvantage of this organization would be that it would probably not be possible to display an entire 1024x1024 image on the screen. Also, loading a 1024x1024 image into the refresh memory may be a little harder than in a system that stores a 1024x1024 image since the image may need to be loaded in pieces.

3.6. ZOOM AND ROAM

As mentioned above, zoom and roam might be useful for effectively breaking an image up into separate smaller pieces for blink comparision. Of course, the larger the zoom factor the more pieces you get. For example, a 1024x1024 stored image with a 512x512 display with a zoom factor of 4 would give you 64 pieces that are 128x128. This might be useful for displaying a sequence of spectral line maps as a "movie".

Some systems allow separate zoom factors in X and Y. This probably isn't very useful for maps, but it might be useful for displaying visibility data as an image where X is time and different Y values correspond to different correlators.

3.7. OTHER WAYS TO DISPLAY A SEQUENCE OF IMAGES

Another way to get a "movie" effect would be to store a sequence of images on a video disk rather than having them all stored in the refresh memory of the display system. Some displays also allow loading the refresh memory directly from the computer's disk or a digital disk directly connected to the display system. If the images can be loaded fast enough, this might also give a "movie" effect.

3.8. INTERLACED VS. NON-INTERLACED DISPLAY

Standard video systems use a 30 Hz interlaced refresh of the image. An advantage of using standard video refresh in our image display system would be that we would have the possibility of incorporating standard video devices into the system in the future. For example, a TV camera could look at an optical photograph and blend or blink this image with the image displayed from the refresh memory. Or, the images being displayed could be recorded on a video tape recorder. Or, it might be easier to interface a video disk to the system.

A disadvantage of using 30 Hz interlaced refresh is that the image has more flicker than a 60 Hz non-interlaced refresh. For fuzzy images like maps this is probably not a significant factor. However, for things like annotation and grid lines, the flicker becomes noticeable. For display of visibility data as an image, different scan lines would correspond to different correlators and thus adjacent scan lines may have very different intensities. In this case, the flicker may become a significant problem. However, if the system has zoom, you could turn on the zoom and have each stored pixel displayed with more than one scan line. (Separate X and Y zoom factors would probably be useful for this.) This may reduce or eliminate the flicker problem. Sorting the correlators into increasing baseline spacings may also reduce or eliminate the flicker problem. Or, putting time on the X axis and baseline on the Y axis may solve the problem. (We could probably experiment with the IIS to test these things.)

Some systems have other weird options, e.g., 40 Hz interlaced. Some systems even allow some degree of software control over the refresh format.

3.9. SIZE OF DISPLAYED IMAGE

NOTE: The following is believed to be accurate, but may be incorrect in a few details. "Standard video" refers to the U.S. standard. There are others which we won't consider here.

For standard video, a single refresh cycle has enough time for 525 scan lines. However, due to things like the vertical retrace, you only can see something like 480 or 483 lines on the screen. The number of pixels across a line is less well defined since it is just an analog signal that varies as the electron beam scans across the CRT. However, for standard video, the width:height ratio of the entire image is 4:3. Thus, with 480 lines, you will get square pixels on the screen if you have 640 pixels across a line.

Different display systems have a variety of displayed image sizes. One system that stores a 512x512 image maintains compatibility with standard video by displaying a 512x483 image with capability to roam up and down to see the few lines which won't fit on the screen. Square pixels are obtained by having the 512 only fill up part of the 640 "pixel locations" across the scan line.

Some systems display a 512x512 image. This is not standard video, but is nice since 512x512 is a common size for our images.

Some systems display 640x480 or 640x512. This might be nice since we could display all 512 columns in our images and then have 128 columns left over for annotation outside of the actual image data.

Some systems store a 512x512 image and display the entire 512x512 as a square centered in the standard video image. There is a separate character memory which is overlaid on the image. This allows displaying 25 lines of 80 characters within the total video image. Thus, there are 8 columns of characters that can be displayed on each side of the image data.

Systems that display larger images bump up against the limits of current CRT technology. Some systems display a full 1024x1024 image. However, these are only available with 30 Hz interlaced refresh. One system stores a 1024x1024 image but displays an 880x704 piece of it with a 60 Hz non-interlaced refresh. However, the monitor for doing this costs about $12,000.

3.10. MONITOR DRIVE SIGNALS

After the choice of interlaced vs. non-interlaced refresh has been made and after the number of scan lines and pixels on a scan line have been chosen, there still remains several options for the signals that are produced by the display controller which drive the monitor. A monitor needs four basic pieces of information in order to display the image on e screen. First, it needs horizontal and vertical sync pulses to tell t when to begin each horizontal scan line and when to begin each vertical scan at the top of the screen. Next, it needs the display signal which specifies the intensity of the image as the electron beam scans along each horizontal line. Three display signals are need for color monitors, one for each of the red, green and blue color guns. Finally, a blanking signal is needed to specify when the electron beams should be completely turned off for horizontal and vertical retrace. There are dozens of possible sync, display and color-encoding combinations! The following will consider only the schemes used in the United States. (A very readable discussion of these considerations (and others) can be found in Chapters 8 and 9 of the RASTER GRAPHICS HANDBOOK, a book produced by the Conrac Corporation, which is one of the main monitor manufacturers.)

Some monitors have separate inputs for the horizontal and vertical sync signals. This is called NON-COMBINED SYNC. Some monitors combine the horizontal and vertical sync in a single input. This is called COMPOSITE SYNC.

Some monitors have separate inputs for the display signal and the blanking signal. However, it appears that the blanking is usually combined with the display signal. Just as the horizontal and vertical sync pulses can be combined into a single sync signal, the display (including blanking) and sync information can be merged into a single MPOSITE DISPLAY SIGNAL (not to be confused with the term "composite nc", which refers to just the combined horizontal and vertical sync). For color monitors, the sync is usually put onto the green display signal. The EIA RS-170 standard refers to a monochrome composite display signal. However, the same standards are usually applied to each of the three colors in an RGB system. (RS-170 probably also implies standard video, i.e., 525 lines with a 30 Hz interlaced refresh.)

The three RGB signals can be combined into a single COLOR ENCODED SIGNAL. In the U.S., this is done according to the NTSC (National Television System Committee) color encoding technique. Other encoding techniques, e.g., PAL and SECAM, are used in other parts of the world.

The composite display signal (perhaps including color encoding) can be used to modulate an RF signal. This is the standard television signal.

Most if not all image display systems use separate RGB signals rather than a single color encoded signal because it gives a better quality color image. Also, there of course is no need to use an RF modulated signal. Most if not all image display systems use separate sync and display signals.

The three display systems currently at the VLA site seem to use three different sets of monitor drive signals. All of them use three parate cables for the RGB display signals. However, the COMTAL system es two separate cables for the horizontal and vertical sync.

The older IIS system uses a single cable for the composite sync signal (combined horizontal and vertical sync) plus a separate cable for blanking.

The newer IIS system uses a single cable for the composite sync signal.

3.11. LOCAL PROCESSING CAPABILITY

Some systems are organized to be rather "dumb." That is, they are strictly a peripheral device tied onto the host computer. A program in the host computer can write (and sometimes read) the refresh memory and LUT(s) as well as controlling the cursor and display modes. The system has no other local processing capabilities.

Some systems remain host controlled peripheral devices, but have a little more local processing capability, e.g., moving the cursor in response to an interaction device that is directly connected to the display system hardware.

Some systems can be operated as a host peripheral device but also can operate independent of the host. A keyboard is included and the stem can be put into a "local" mode in which typed commands can control st system functions, e.g. zoom and roam, without any use of the host. Some systems of this type can also operate as if they are an ordinary alphanumeric terminal connected to the host.

Some systems contain some sort of a microprocessor which can be programmed by the purchaser. This could be the case in some peripheral systems as well as systems which have a stricity local control mode or operation. This capability might be useful to us for such things as locally modifying the LUT in response to an interaction device. Other applications could be envisioned.

Some systems have even more stand-alone capability. We could of course get a DMA interface to our host computer. However, the display system itself contains a fairly powerful CPU. It may have the capability for locally connected devices such as tapes, disks or array processors. Such a system might be viewed as a commercial version of the IMPS system organization.

Some systems have a capability for doing various local processing operations on the images stored in the refresh memory. In the simplest cases, this might include simple arithmetic such as adding or subtracting two stored images and putting the result into a third image refresh memory. Some systems can do a wide range of other processing such as statistics calculation, convolution, etc., etc. These sorts of pabilities may be available in the strictly host peripheral systems as ll as in systems which have more stand-alone capability.