# THE GRAPHIC-PHOTOGRAPHIC COMPUTER -- ASPECTS OF INTERPOLATION

By

Andrew Lippman

B.S.E.E. Massachusetts Institute of Technology
1971

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1978

Signature of Author _____
Department of Architecture
January 19, 1978

Certified by _____
Nicholas Negroponte
Associate Professor of Computer Graphics
Thesis Supervisor

Accepted by _____
Professor N. John Habraken
Departmental Committee for Graduate Students

The work reported herein was sponsored by the Office of Naval Research, Contract Number N00014-75-C-0460.

The Graphic-Photographic Computer --
Aspects of Interpolation

Table of Contents

The Graphic-Photographic Computer --
Aspects of Interpolation

By

Andrew Lippman

submitted to the Department of Architecture on January 19, 1978 in partial fulfillment of the requirements for the degree of Master of Science.

## Abstract

The Graphic-Photographic Computer is the first of a new generation of computer frame buffer displays and is designed to provide the tools for solution of the problems of (1) image dynamics and (2) conjoint image and graphic display. This thesis addresses these problems from the point of view of real-time output video processing. An interpolative approach is used to provide the capabilities for smooth scrolling, zooming and graphic generation. The operation of a procedural, table-driven interpolator is described and is simulated in software to demonstrate the results.

Thesis Supervisor: Nicholas Negroponte

Title: Associate Professor of Computer Graphics

# CHAPTER ONE

## The Graphic-Photographic Computer

### Background

The past few years have seen a resurgence of interest in the explicit frame buffer display. This particular method of portraying a computer generated image had been neglected for several years due to the primary constraint of the time: the high cost of the necessary memory. Instead, much effort was directed at perfecting alternative image generation techniques, to wit, calligraphic and scan conversion systems. However, currently, memory costs are dropping rapidly and show signs of continuing to do so, and competitive display technologies have reached their logical and architectural limitations. This has rendered the frame buffer once again a viable and reasonable solution to many display problems.

The extent to which calligraphic systems have been developed is best evidenced by the current Evans and Sutherland Picture System 2, a real time, three dimensional vector graphics display. This particular device is capable of displaying up to 20,000 inches of vectors per frame time (one thirtieth of a second) and of updating the endpoints of these at the rate of approximately 1000 vectors per frame. The update is accomplished by transforming the endpoints according to a

four by four matrix that denotes the positional change of the object or observer a three dimensional space, with perspective included. Also, depth cueing by intensity gradient is calculated and added to the display. However, the image consists of a "wire frame" representation of the scene: no surfaces are shown and no hidden lines are removed.*

While this may be taken as the state of the art in calligraphic display,** the system is costly, and processor intensive, requiring close interaction with a host (usually a PDP-11). Moreover, the specification of the monitor, which must be able to randomly sweep the requisite 20,000 inches of vector, is severe; high bandwidth deflection circuits are required and the vector drawing algorithm must take into account the specific time required to draw a vector of random length and with random orientation, yet still present that vector with a constant intensity. The use of a tri-color CRT is precluded.

Scan conversion systems have been brought to their highest stages of development in the realm of flight simulation.

---

*In a wire frame image, the issue of hidden lines is, in fact, real. In a more general case, they are usually regarded as a special case of hidden surfaces.

**Competitors, such as Adage and Vector General, offer similar vector display capabilities, often without the full three dimensional perspective computation.

General Electric, Lear, and Evans and Sutherland have produced systems that are capable of computing by this technique a complete three dimensional pictorial image as might be seen from the cockpit of an airplane, thirty times a second. However, these systems are unbelievably costly, running into the millions of dollars and requiring roomfuls of electronic equipment to support the display. Moreover, their design is highly optimized for each particular application, and they present an image with only limited resolution, and consisting of a sharply limited number of elements, usually less than 1000. In the case of the Evans and Sutherland systems developed for the Port Authority of New York and Lufthansa, much use is made of the fact that a large concentration of data lies in the plane of the earth, below which a typical simulated airplane or ship seldom travels and that the particular entity being displayed is immediately recognizable by its shape and form. Thus, little texture is added and only a certain number of elements are on screen at the same time.

Examples of current frame buffer systems are commonplace. They are manufactured by Ramktek, Genisco and Interpretation Systems, among others. They find their application in many varied fields all characterized by the requirement for high resolution, mostly static, gray scale or color pictures

coupled with low resolution graphics. Computer automated tomography, satellite imaging and image processing are perhaps the best examples. An important, though in some sense perpendicular application of frame buffers, is in broadcast television industry. They are used for time base correction, for still picture "freezes", and as special effects generators.

Most raster scan displays are made to be compatible with existing television technology, for several reasons. In the TV set, there exists a widespread, inexpensive medium for display. There are over 121 million television receivers in the country today, and the number is rapidly growing (IEEE, 1975). In addition, the technology to create a high resolution color reproduction tube has only recently become available, effectively limiting the use of a high bandwidth framestore to black and white representations.

## The Problems

The preceding argument has not been articulated to demonstrate that frame buffers offer a panacea for all current display ills, but rather to place their development into the proper historical and application perspective. The frame buffer will never be able to replicate the dynamics of the scan

conversion system,* nor will it be able to produce the line quality of the calligraphic display. Rather, it will be able to present densely populated graphic regions, characterized by a limited form of dynamics; fine color and textural gradations and photographic realism.

There are, however, problems associated with the realization of all of the above stated goals. These problems fall into two categorizations: those associated with the dynamics and those associated with the conjoint presentation of high quality graphics and images.

To clarify the dynamic display issues, transformations are divided into two classes: display dynamics and data dynamics. The term display dynamics refers to that class of manipulations in which the contents of the framestore are not modified, but the viewed image is. They are often effected by means of a large memory map and a configurable readout generator (window). The dynamic effect is the result of modifying the relationship between the memory and the screen. The Massachusetts Institute of Technology developed Spatial

---

*indeed, there is no need to buffer a frame, if it is to be displayed only once, then completely regenerated, as in the case of flight simulators. This will become a necessity only if reasonable techniques for using the frame-to-frame coherence of a sequence of images is developed. To date, this has not been done.

Data Management System is an example of this approach. An image with a user specifiable aspect ratio, density, and overall size is created in the memory, and a flexible video generator translates this into a standard, broadcast compatible television signal. The user is provided with controls to change the scale and position of the displayed data with all changes occurring within one frame time. By appropriate operation of the controls, somewhat smooth scrolling and zooming effects are perceived.

Data Dynamics refers to those manipulations wherein the contents of the framestore are changed, in real time, to effect the apparent motion. To date, this is a little studied area due primarily to the high cost of the computation hardware necessary to process the requisite one quarter of a million picture points in a single frame.

The problems of joint graphic and pictorial display lie in the essential difference between the two types of data: image data and graphic data.

In general, the underlying characteristic of images is that they are generated from originals of continuous, effectively infinite resolution. Thus, one problem of computer, indeed electrical, display of such data, is how to best code and

and replicate the rich detail of continuous tone, live scenes in a discrete medium. Broadcast television is an example of one solution. The method chosen is to vertically sample a horizontally bandlimited representation of the scene, as focused on the target area of the camera tube. This process is repeated thirty times a second to give the appearance of motion. Most computer systems sample the image in two dimensions, storing it either in a 512 square, or 256 square array, and quantize the data to a precision varying from four bits per sample (16 levels), to 24 (usually used for color). The effects of the sampling and bandlimiting and their application to digital displays will be discussed in chapter two.

In contrast to images, geometrically based graphics are inherently discrete. Their appearance is not predicated upon some underlying, continuously resolvable real-world scene. Rather, they are generated entirely within the discrete environment, from either totally artificial or simulation oriented criteria, and do not necessarily obey any of the laws of physics that constrain the real world scenes which form the basis for natural images. Therefore, any of the assumptions that govern the manipulation and compression techniques used on images fail on graphics. In the realm of graphics, lines may be perfectly sharp, circles precisely

round, and colors monochromatic; with images, this is rarely the case. Too, graphics are often limited in resolution and density relative to what the originator, usually human, deems desirable, or what is economically feasible to incorporate into a system.

Recently, computer graphics have taken on some of the attributes of pictures in that researchers are attempting to expand the limited lexicon of point and line representations to impart more realism and depth to computer generated displays. Some of this is due to the tailoring of computer representations to suit the limitations of the output medium itself, as in the case of television displays, wherein the techniques described in chapter two will demonstrate that this approach can create a subjectively better picture. Much of it is due to the direction of computer displays at a more widespread and demanding audience, as in the case of the flight simulators described previously. Programs that allow users to "animate" and "paint" are further examples of this. Graphic primitives now include points, and areas and textures; the resulting image is more a photographic representation than a schematic one.

## Approach

The Graphic-Photographic Computer may aptly be called the

first of the "new" generation of frame buffer displays in that it is designed at the outset to provide a basis for investigation and solution to the problems outlined above. It is characterized by the application of many new techniques learned from vast experience with commericially available frame buffers, and with the experience gained from the construction of one in particular, whose features are, as yet, unduplicated in the commercial domain. Two techniques stand out as unique: Interpolation and Feedback. These form the basis for the improved dynamic techniques and provide the tools for conjoint graphic and image display.

Interpolation, as used in this report, refers to the application of real time video rate processing to the output of the framestore. The genesis of interpolation as an imaging aid and its application to graphic displays is discussed in the following chapters. In addition, the specific proposed implementation, a procedural model, is analysed in detail in chapter three. It should be noted that video rate processing is no longer solely the domain of large, expensive mainframe computers, but is possible in a minicomputer based, interactive environment with the advancing sophistication of micro-electronics and large scale integration. The procedural model discussed in chapter three points the way for a simple and effective table based hardware implementation.

Feedback is a concept that follows directly from the application of many real time processing elements to the ouput of the framestore. The initial step is to develop the concept of the "window", or video readout element to a high degree, effectively imbedding it with some general purpose capabilities. With the addition of the interpolating elements, a powerful, real-time computing capability is created as an adjunct to the digital frame buffer. The feedback then is merely a path by which this processing may be used to rewrite the frame memory itself effecting modifications and dynamic effects and using it more than a simple "playback" device.

# CHAPTER TWO

## The Sampled Image and Interpolation

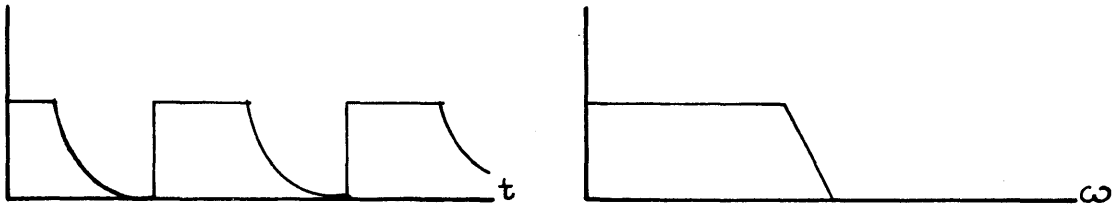
To better understand the use of interpolation techniques, it is first necessary to review the methods by which we represent a generalized sampled image on a digital display. The analysis in this chapter will proceed from a strictly image processing point of view, graphics being regarded for the moment as a special case of images. We will first review the techniques of sampling, and the associated discrete variable mathematics. The use of an interpolating filter will then be introduced as a device necessary when one wishes to change the sampling rate of a signal, and the sampling rate change will be interpreted as a linear filtering problem. The appropriate modifications to the approach to incorporate graphics into the domain will be introduced at the end of the chapter.

### Sampling

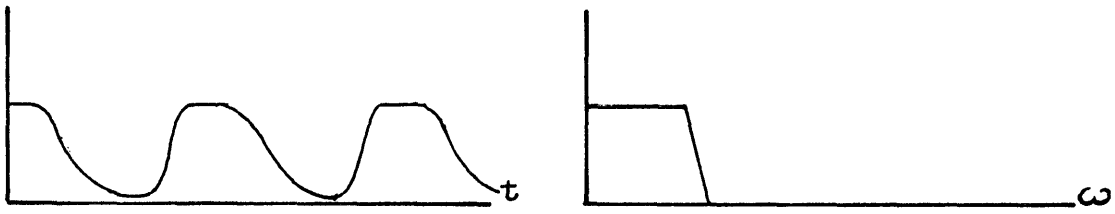
When a continuous signal is to be processed on a digital computer, two independent processes must first be carried out: the signal must be sampled, i.e., its values must be taken at certain periodic intervals; and it must be quantized, i.e., the value of specific samples must be mapped onto the limited range of numbers available to the

specific computer to be used. These processes are independent in that the existence of one does not imply the existence of the other, and their effects are quite different. Throughout this thesis, the effects of errors introduced by the quantization process will be ignored. The image generation process will result in a picture that has only limited intensity (or color) resolution, and thus the effects of quantization error in any earlier filtering step will be negligible if it is below that of the final display. Moreover, non-recursive implementations will be used exclusively, so that the stability of postulated filters will not be affected by any quantization errors.

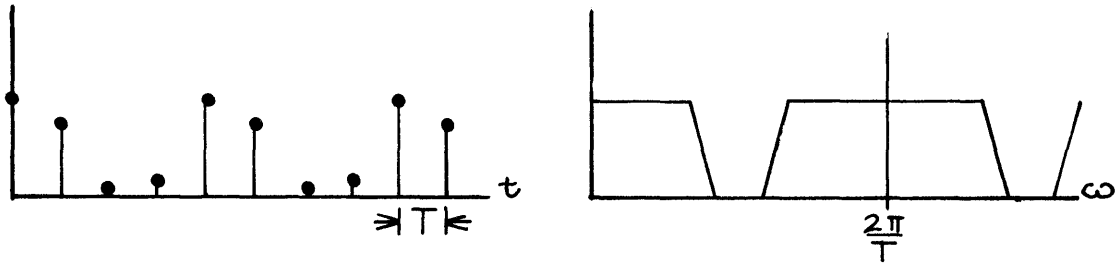
The sampling theorem states that a bandlimited signal that has a fourier transform may be recreated uniquely from its samples if the original sampling frequency is at least twice that of the highest frequency contained in the signal. The process by which this is done is illustrated in figure 2.1. The signal is first low-pass filtered so that it meets the bandlimited criteria; then it is sampled. To reproduce the signal, the samples (which are a sequence of impulses) is low-pass filtered again to insure that the ouput signal has no frequency components higher than the original signal. The two low-pass filters used are identical. In terms of the frequency domain representations illustrated on the




2.1a: Original




2.1b: Low-Pass Filtered




2.1c: Sampled

Figure 2.1: Sampling

right, it can be shown that the sampled signal and the original signal fourier transforms are related by the following formula:*

$$X(e^{jwt}) = \frac{1}{T} \sum_k \underline{X}(w+k\frac{2\pi}{T})$$

The output filter is thus an interpolation filter, and generates the values of the function between the sample points. Its impulse response is as shown:

$$\underline{x}(t) = \sum_k x(k) \frac{\sin\left[\frac{\pi}{T}(t-kT)\right]}{\frac{\pi}{T}(t-kT)}$$

Note that this impulse response is exactly that of an ideal low-pass filter, and since it is of infinite in extent in time, it is unrealizable. In practice, approximations are used. An ideal low-pass filter can be approximated to an arbitrary degree of accuracy, if a sufficiently long impulse response is allowed.

## Sampling Rate Changes

The process by which the sampling rate of a signal may be changed is illustrated in figure 2.2. The illustration shows an increase in the sampling rate. This has real use and

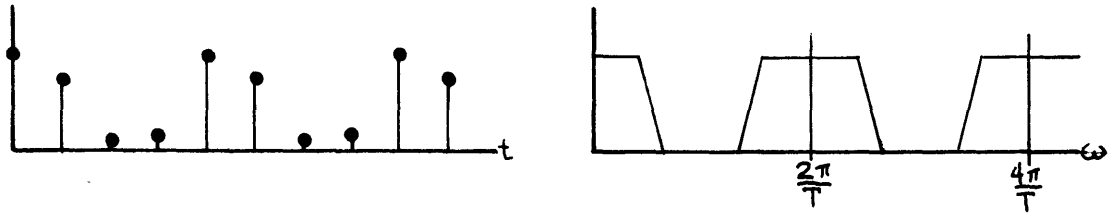
---

*$\underline{x}, \underline{X}$ are continuous time and frequency functions;
$x, X$ are discrete time and frequency functions;
T is the sampling period.

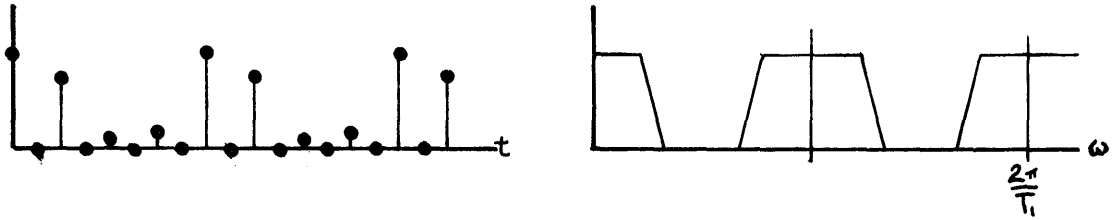
benefit in a digital image display, and is an example of scaling the image up, or zooming in. This technique is also useful when the sampling rate of the original image does not correspond to sampling grid of the display medium, as in the case where a 256 point square sampled image is to be reproduced on a system with a resolution of 512 points and 512 lines. It will be shown later in the chapter that an interpolative scaling technique can often result in a subjectively better representation of the iamge.

Referring to the illustration, the original sampled signal is shown at the top, with its spectrum on the right. When the sampling rate is increased, additional zero-valued samples are inserted, giving a spectrum that is now periodic with the period of the original sampled signal, as opposed to the new rate, as might be expected. This spurious periodicity is removed with an interpolation filter, which in effect, replaces the zero valued samples with values associated with the original signal at those points.

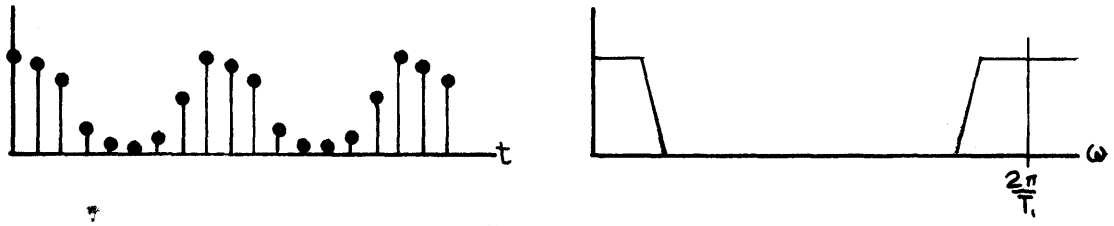
The case where the sampling rate is to be decreased is illustrated in figure 2.3. This is analogous in the imaging domain, to scaling the image down, or zooming out. In this case, either successive samples must be discarded, or




2.2a: Original Sampled Signal

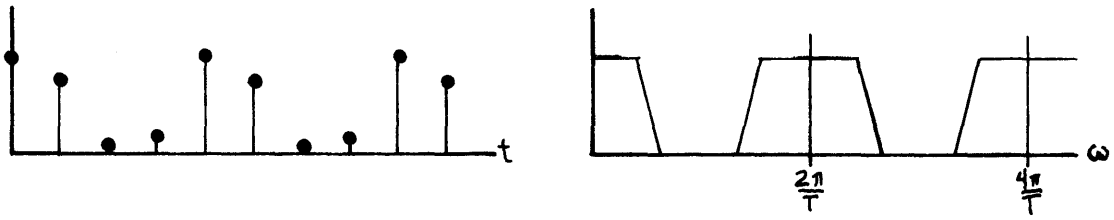



2.2b: New Rate, with Zero-Valued Samples Inserted

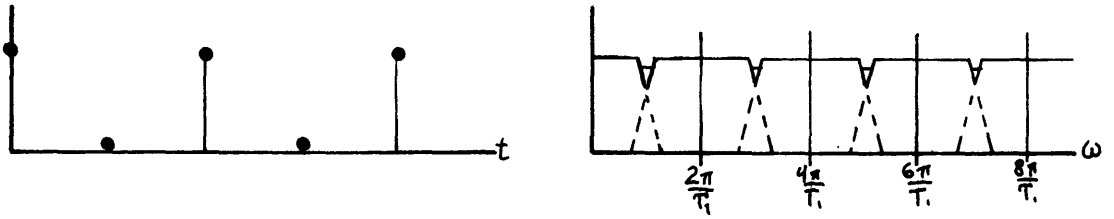



2.2c: Filtered Output

Figure 2.2: Increasing the Sampling Rate By 2




2.3a: Original Sampled Signal




2.3b: New Sampled Output

Figure 2.3: Decreasing the Sampling Rate by 2

successive samples must be arithmetically combined to create the final display.

Clearly, a requirement for reducing the sampling rate is that the original signal must satisfy the sampling theorem criteria for the new, lower sampling rate, i.e., there must be no frequencies higher than one-half the new sampling frequency. When samples are discarded, it is presumed that the required low pass filtering has been done first; the new signal then consists of samples of the original analogue signal. When samples are arithmetically combined to generate the new sequence, the filtering is done digitally at this step. In the figure, the result of undersampling is shown, with the attendant aliasing.

It has been shown (Rabiner and Schafer, 1973) that a finite impulse response (FIR) filter is often a desirable choice to use as the interpolator. FIR filters, while generally requiring longer impulse response durations and greater computation to accomplish the same bandlimiting action as infinite impulse response filters, are capable of linear phase, and thus can be a better approximation to the ideal interpolator. Their response may also be made arbitrarily close to that of the ideal interpolator if one is willing to accept the computation requirements.

When an FIR filter is used to effect the interpolation, it is often efficient to implement the convolution directly rather than by fast convolution techniques such as the Fast Fourier Transform. This is due, in part, to the fact that general convolution is faster when the impulse response of the filter is short and in part by the fact that much of the computation involves a multiplication by zero.

In general, to specify the design of such a filter, it is necessary to determine how many of the points in the sequence are to be used in the interpolation and then use any number of filter design programs to determine the exact impulse response and frequency response. If the interpolated values are to be exact at exact multiples of the original sampling rate, an odd filter length is appropriate. The length of the filter (N) as a function of the scaling (L), and the number of samples to be used (Q) is as follows:

$$N = QL-1$$

If a length greater than this is used, more samples will be used in some of the calculations; if it is less, fewer will occasionally be used. This is illustrated in figure 2.4 for a scale factor of four. To use only two of the original samples in the computation, the appropriate length is seven.

The result of using non-interpolative techniques to effect

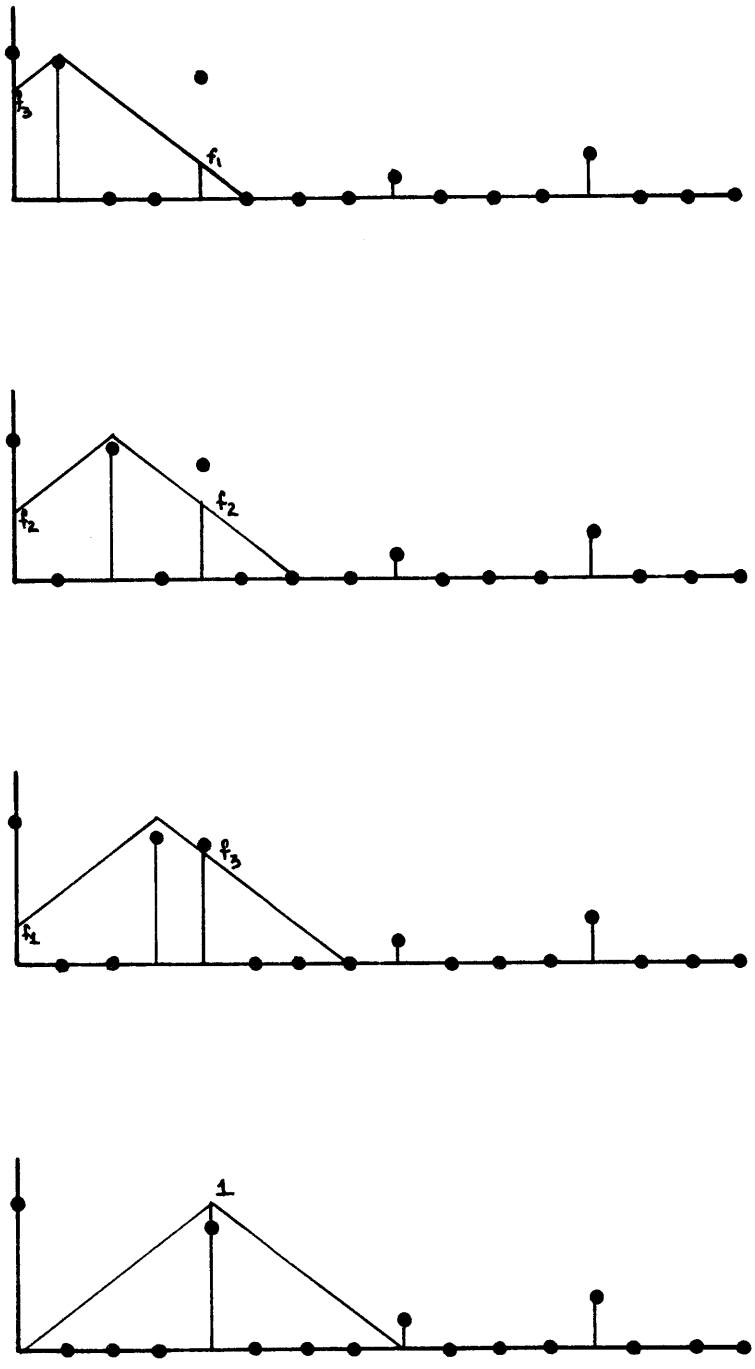



Figure 2.4: Interpolation

Sampling Rate Increase of Four

scale changes in a digital image display are illustrated in figure 2.5. The original image is scaled up by a factor of two. In the first attempt, added samples are repetitions of the previous sample. This is a slight modification of the technique of inserting zero-valued samples, and has a small low-pass effect. In the second attempt, a simple length three filter has been used. The most noticeable difference is in the area of the resolution wedges. The moire patterns are evidence of aliasing, and they are much more evident in the first scaled up version.

When scaling factors greater than two are used, the lack of filtering results in visible artifacts of the process in the image. This is due to the fact that there is a strong periodicity associated with the new "elemental" picture elements, that are now "n" times as large as they once were, where "n" is the scale factor. A scale factor of four results in an image that has the appearance of a mosaic, rather than a continuous picture.

When the image is to be scaled down, the filtering must occur before the scaling operation to insure that the old picture satisfies the sampling theorem requirements at the new sampling rate. Since natural scenes are rarely band-limited, excessive moire patterns appear if picture elements

2.5a: Picture Element Repetition

2.5b: Filtered Interpolation

Figure 2.5: 100% Scale-up

are merely skipped. (See figure 2.6.)

This approach is useful when the image is to be scaled by other than integral factors. It can be expanded to encompass rational scale factors by using both approaches in a cascade. For a scale change by a factor C = A/B, the sampling rate is first increased by a factor A, then decreased by a factor B, using the techniques described previously. In the direct application of this method, a filter is required only after the increase in the sampling rate to approximate the values of the analogue signal at these new points. The new sequence will then contain samples of the original signal. The entire procedure is illustrated in figure 2.7.

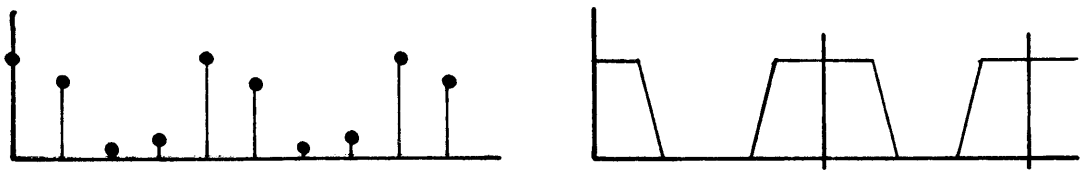
2.6a: Original High Resolution Image

2.6b: Unfiltered Reduction

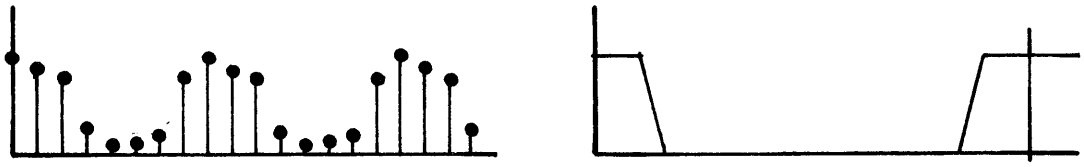
2.6c: Filtered Original

2.6d: Reduction After Filtering

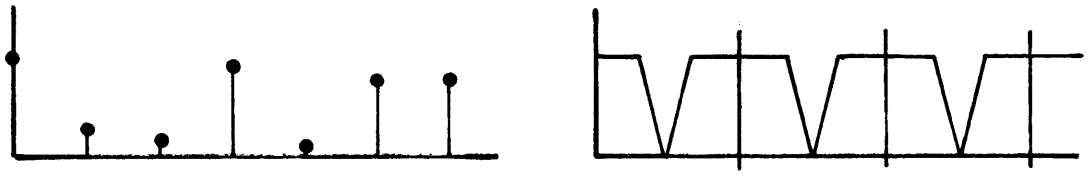
Figure 2.6: 100% Reduction



2.7a: Original Sampled Signal



2.7b: With Rate Increased by Two



2.7c: With Rate Decreased by Three

Figure 2.7: Integral Sampling Rate Change

# CHAPTER THREE

## The Procedural Model

### Introduction

The need to filter the image when a scale change is to be effected has been made abundantly clear by the arguments in chapter two. The aim of this report is to create a model by which specific hardware may be built to accomplish this task in real time, at video rates. The proposed interpolator will then be placed between the frame memory of a raster scan display system and the video digital-to-analogue converters. It will contain enough storage and arithmetic capability to perform the filtering and will generate a sequence of samples that will be the image data for a specific line of the display.

In many cases, the restriction of real time will severely limit the length of the impulse response of the filters to be used. This is considered an acceptable solution nonetheless, since the image will be an improvement over the non-filtered version. Moreover, the technique will be applied to the generation of scaling factors that are randomly selectable, as opposed to the normal integer scales that are available in commercial equipment. Also,

the interpolator may be used to effect many dynamic changes in the image in an interactive manner.

First, we will examine a method by which the three step process described at the end of chapter two may be accomplished in a hardware implementation. Next, several optimization steps will be applied, finally resulting in the procedural model.

## The Direct Approach

A two dimensional FIR filter may be described by a convolution matrix:*

$$\begin{matrix} h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & h_{32} \\ h_{13} & h_{23} & h_{33} \end{matrix} \qquad \text{(3 by 3 shown)}$$

For simplicity it will be presumed that the filter is separable, i.e., that it may be represented as the cascade of two one-dimensional filters, one oriented horizontally, the other vertically. In general, if this technique is chosen, symmetry will demand that they, in fact, be the

---

*For input $x(n,m)$; filter $h(n,m)$; and output $y(n,m)$, the convolution $y(n,m) = x(n,m) * h(n,m)$ is defined as:
$y(n,m) = \sum_{k} \sum_{l} x(k,l) h(k-n,l-m)$

same filter, but that is not a requirement for the analysis to follow.

Regarding for the moment, the horizontal separation, the filtering may be accomplished as shown schematically in figure 3.1. Firstly, the input sequence (display line) is read into a shift register with the appropriate locations skipped to introduce the zero-valued samples. Secondly, the horizontal filter kernel is convolved with the contents of the shift register. For a filter that uses exactly <u>two</u> of the original sample points in its interpolation, two multiply-add operations are required per point. At a display rate of 90 nanoseconds per point, this is well within the realizable computational limits of standard TTL or ECL integrated circuitry. The operation may be facilitated by the use of look-up tables to accomplish the multiplication. Since the image is usually stored to a precision of eight bits per point, and there will rarely be more than sixteen different values for the impulse response of the filter, a look-up table with $2^{12}$ entries is required. For an eight bit output, this can be realized with eight chips.

Finally, the video line is generated by reading out the required number of samples from the ouput shift register. For the complete two-dimensional convolution, the vertical

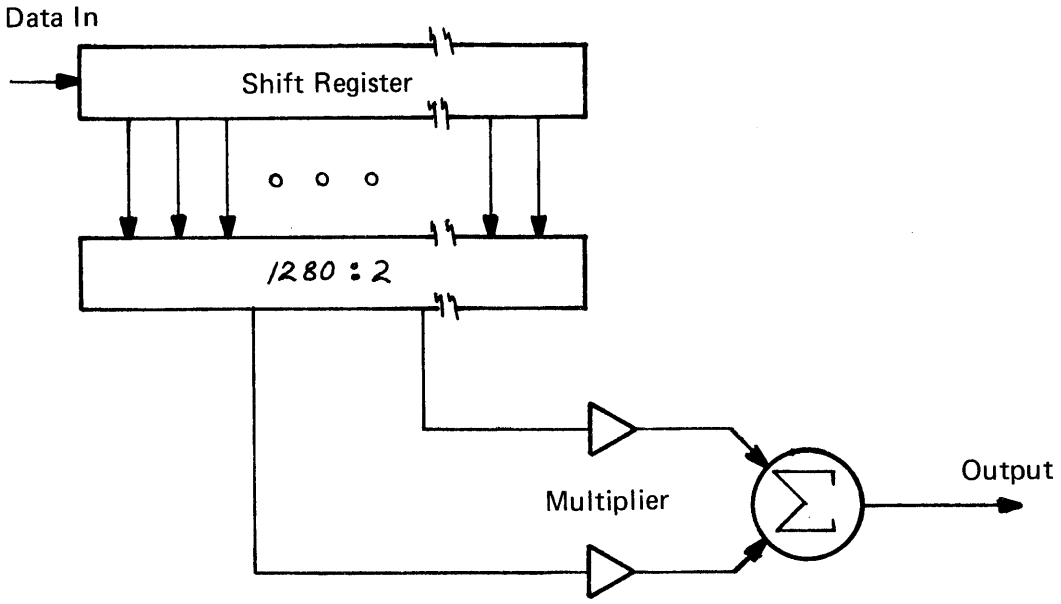



Figure 3.1: Horizontal Filtering

filter would be first passed over two shift registers, each containing successive lines of the original display data.

This will be denoted the direct method of scaling, and it is nothing more than a realization of the techniques presented in chapter two. There are two major limitations on its practicability: (1) storage is required in the main video data shift registers for the zeroes that must be inserted to effect the increase in sampling rate, and (2) calculated values are often discarded. For a scaleup of two, and a line length of 640 elements, the register must be 1280 elements long. For a scaleup of 25% to be effected by a primary increase in sampling rate of five, followed by a decrease by a factor of four, 5 X 640, or 3200 storage locations are required in the register. It is thus clear, that for anything but ratios that are representable as simple fractions, the storage in the output register quickly becomes excessive.

The second limitation lies in the fact that much computation is wasted calculating the value of samples that are to be discarded when the rate decrease is done. To accomplish the 25% scale up described, all 3200 samples must be generated in the time allocated for a single line of video display

(90 X 640 = 57.5 microseconds) even though only 640 will ultimately be used. Moreover, each calculation requires two multiply add operations, resulting in a time per multiplication of 18 nanoseconds. This time can be yet shorter if a larger scale up is required.

## The Procedural Method

The procedural approach addresses the two limitations of the direct method outlined previously. It results in an interpolation that includes no wasted operations, and uses little additional picture storage, yet still retains the ability to effect a large number of rational scaling operations. The interpolation will be performed with one set of operations to do both the increase in the sampling rate, the filtering and the final decrease. A separable filter will be assumed. Involved in the hardware implementation would be (1) three lines of video storage; (2) tables for the filter coefficients; (3) a single output register for the computed line. The analysis will proceed as in the previous section by first examining the operation of the horizontal filter, then the vertical. The combined operation will follow directly.

Looking more closely at the computation for a single picture

element, it is evident that regardless of the length of the filter, if two input picture elements are to be used in the computation of the output point, the two nearest elements are always used. In fact, the entire operation may be regarded as taking the weighted average of the two adjacent original elements where the weighting factors are the filter coefficients. Note that for simple linear interpolation, the weighting factors would be proportional to the effective distance between the input sample point and the output point. In the explanations that follow, linear interpolation will be the exemplar although an optimally designed filter will generally produce a more accurate interpolation.

Thus, to effect the 25% scaling described previously, figure 2.4 illustrates the operation of the horizontal filtering. To generate the first picture element, the first input point is multiplied by factor one, and the second by factor two. The results are summed. For general scaling, the factors will generally be different. To generate the second picture element, the second input point is multiplied by a new factor one, and the third by a new factor two. When the fourth element is required, it corresponds exactly to the fourth input point, and no calculations are necessary.

To implement this, a table is created with the following entries:

| position | left element # | right element # | left factor | right factor |
|---|---|---|---|---|

By advancing through the table by output picture element positions and performing the calculations as proceeding, no operations are wasted, and no extraneous storage is required. Rather, the convolution of the filter kernel has been broken up into a tabled series of operations that takes into account the multiplications by zero and the elimination of extra sample points. This table, moreover, is pre-calculated, and therefore trades execution time for generation time. It is presumed that a sequence of scaling operations is known in advance and successive tables can be created and stored in the host computer.

To implement the vertical section of the filter, the same technique is used with the following modifications: it is now necessary to have more than one line of output video available in a fast random access memory; generally two will be required. The effective kernel is passed over the data in these registers, from left to right, operating on two vertically aligned picture elements to generate the partial output sequence. In practice, both filtering operations would proceed jointly, in synchronism, and a single output point would be generated in one set of four multiply-adds.

Note that in transforming the vertical filter into a table driven procedure some simplifications over the horizontal procedure are possible. The horizontal filter control had available to it all of the input picture elements along a given horizontal line. It therefore had to address the ones it needed directly. The vertical filter has only the lines immediately above and immediately below in random access memory, and thus requires no addressing to determine which input points are to be operated upon. The complete table is as follows:

| Output Line # | Top Factor | Bottom Factor |
|---|---|---|

This table is indexed into by display line number, and contains a pair of factor entries: one for the line above, and one for the line below. The window has already presented to the interpolation hardware the final display lines in the correct order. The entire process is illustrated in figure 3.2.

## Optimizations

There are two optimizations that are included in the complete procedural implementation of the interpolator. The first

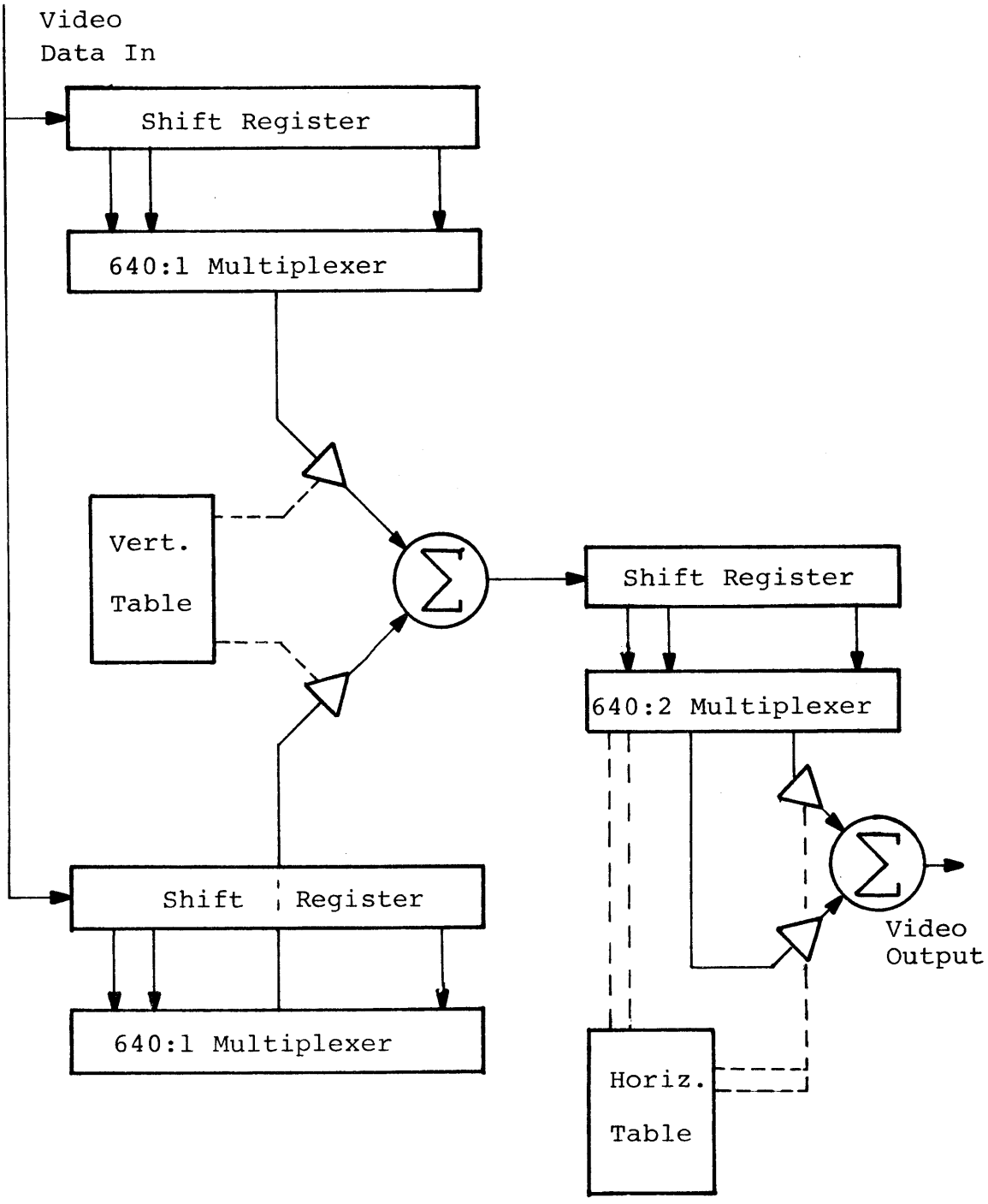



Figure 3.2: The Procedural Interpolator

involves the normalization normally required to allow integer arithmetic in an FIR filter, and the second makes use of the inherent symmetries of the process.

The first optimization assumes that a table look-up method is used to perform the multiplication. If an integer combinatorial multiplier is used, the result will have to be divided by a normalization factor to map it back to the allowable range of picture element values. This is the result of the fact that most filter coefficients are less than one. When a table is used, the table consists of several sets of input picture values, each modified by the multiplication factor. When a product is computed, a set is selected, and the appropriate picture element is used as an index into that set to directly produce the output value. Thus the normalization can be incorporated into the construction of the multiplication table entries and need not be done as a distinct step in the calculations.

The second optimization relies on the fact that many of the filter coefficients are the same for symmetric filters. For example, in the factor of four scale-up illustrated earlier, the required filter length was seven in each direction. However, since the filter is symmetric in four directions, only four of the values are different. Thus for an eight

bit precision image, the multiplication must have only $2^{10}$ entries for exact multiplications. When the scale-up required is eight, the table will contain $2^{11}$ entries. Thus, for a wide range of scaling factors, the length of the table required for the multiplication remains within reasonable bounds. For hardware implementation, memory elements with the required access time are currently available wherein 4K* bits are stored per chip. Thus a factor of sixteen scale-up could be implemented with only eight chips used for the multiplier.

Speed Requirements

To calculate the necessary speed of the individual circuit elements in the interpolation system, the following sample display parameters will be used:

| | | |
|---|---|---|
| Picture element time: | 90 | nanoseconds |
| Line time: | 63.5 | microseconds |
| Frame time: | 33.3 | milliseconds |

The parameters for the tables and arithmetic elements will be as follows:

| | | |
|---|---|---|
| Table access time: | 45 | nanoseconds |
| ALU Add time: | 25 | nanoseconds |

---

* K=1024

For the complete interpolation, four multiply-adds must be executed each picture element time. Since a table look-up and a combinatorial addition are required, and there is no addition for the first multiply, the net time for the entire calculation is 205 nanoseconds. This consists of four table access times, plus one additional summation after the fourth look-up. All other summation times are effectively masked by the memory access time in a pipelined structure. This implies that at least three copies of the table and three arithmetic-logic elements must be incorporated into the device. Less time will be required if the inherent symmetries of simple scale-ups are used and in cases where the original picture element is used directly with no computation. In the example case of the 25% scale-up, this case obtains 25% of the time, placing the computation within the realm of realizability with two copies of the logic.

To change the interpolation algorithm rapidly, as in the case of a continuous zoom or translation, it is necessary to reload all tables within one frame time. For the worst case, where the multiplication table contains $2^{12}$ entries, the total quantity of data that must be transmitted to the interpolator is as follows:

| | | |
|---|---|---|
| Multiplication table: | $2^{12}$ | bytes |
| Vertical factor table: | 480 | bytes |
| Horizontal factor table: | 2560 | bytes |
| | 7136 | bytes |

This results in a transfer rate of 214,000 bytes per second, well within the range of most available minicomputers.

Limitations

The particular implementation of the interpolator chosen was designed specifically for use in scaling and relocating digitally produced images on a raster scan display system. It was therefore determined that a small kernel could be implemented with only four multiply-adds per output image point. This is somewhat less than would be used in a general purpose three by three convolver and is a by-product of the fact that reasonable filtering can be accomplished by using only two of the original inputs in each direction.

The result of this is that the filter is somewhat limited in its application to standard one-to-one image filtering problems such as edge detection, general low-pass filtering, and unsharp masking. These applications would require that the system incorporate a feedback path and repeatedly operate on the input frame to produce a single output image. These applications are discussed in chapter four.

If the desire is to use the interpolator for general purpose filtering, the procedure tables and the arithmetic elements

must be duplicated to provide the necessary speed to perform the required nine operation pairs per element time. Currently, a simplified version of this device is available from the Comtal Corporation. It does the three by three convolution using a space invariant kernel in real time.

# CHAPTER FOUR

## Display Applications

Thus far, the use of the interpolation hardware has been limited to the filtering required to effect sampling rate changes in a display image. While this is one important use, and provides the capability for dynamic zooming and image scaling, the particular implementation described is far more general and permits many other dynamic manipulations. Some of these require simple additions to the interpolator, often as simple as extending the line registers to 1280 elements rather than 640. Others use double buffered techniques to allow the algorithm to switch on "frame line boundaries", and thus present a continuous image sequence with invisible table loading times. The two classes of dynamic manipulations will be discussed separately.

### Display Dynamics

In the case where a portion of a large, static data base is to be represented for display, the interpolator can help to "navigate" through that data base. By navigation is meant the selective viewing of parts of that image at user determined scales and locations on the screen. Two primitive operations are required of the frame buffer system

to provide this capability: scaling and relocation.

In the context of the interpolator, it has already been demonstrated that scaling functions can be performed. Relocation is equally simple, requiring a filter that operates as an "all-pass" filter with a specified phase, or delay. Thus for a one element shift in the displayed image, one would load the interpolation tables with the following algorithm:

1. output zero for the first element time.
2. output element N-1 for the $N^{th}$ element time.

In this case, the no arithmetic processing of the image data is required and the interpolator is used as a dynamic positioner.

A refinement of simple integral relocation is fractional relocation, wherein the picture is moved some non-integral number of display elements in a particular direction. This once again requires an all-pass filter, but the phase is no longer an integral number of samples. Thus a fractional shift filter would have to be designed and loaded into the tables.

Data Dynamics

Data Dynamics involves the use of the interpolator coupled

with a feedback path into the framestore memory for the computed data. The justification for this approach to image manipulation lies in the fact that much of sequential addressing and memory readout necessary for repetitive operations already exists in the display generation hardware and more efficient use of it is made by allowing it to perform some computation as well. Additionally, the interpolator contains the means by which computation may be based on local neighborhoods, a process that involves much address calculation and processing overhead when done in the main computer.

For this aspect of its use, the real time operation of the interpolator is secondary to the arithmetic and addressing functions that exist in it. In fact, one could postulate a "dual speed" version, where the data dynamic functions assumed control of the interpolation hardware for some specified number of frames and performed slower, more complex functions solely for the purpose of re-creating a new image in the main memory. While this might be interesting, it has not been incorporated into the design of this system primarily because the integrity of the display screen is seen to be of paramount importance for interactive graphics. It would be unacceptable to require that the image sequence be non-continuous or that the screen be blanked for some calculation period.

The main capability of the interpolator for data dynamics is its ability to be used to generate higher order filters than those associated with the real time operation. They can be either recursive, if the new image is fed directly back to its original memory locations or non-recursive, if the image is double buffered. Thus a more optimal interpolation filter could be designed that operated over several frames in sequence to produce the final output, using a filter whose length was proportional to the delay between initial and final image occurrence. The first pass through the system would re-write the image with the appropriate insertion of zero-value samples and each later pass would effectively convolve the image with a second order stage. The entire result would be a cascade of second order sections. This technique can be used to generate all one dimensional filters and separable two dimensional filters.

Non-linear operations could also be carried out. In this case, the multiplication tables are loaded with a set of "functional operands", and the new picture is a processed version of the old one. An example is to generate a single tone image from a continuous tone one, allowing the user to interactively set the black/white threshold. The tables are loaded with all black below the threshold, and all white above it. A second example is edge detection. A

non-linear approach could be used, wherein a kernel is passed through the data that detects local edges at various orientations, and fills in the new image appropriately.

For all non-linear uses of the interpolator, the key ingredient is the table look-up arithmetic. This permits a non-analytic function to be programmed.

## Graphical Applications

As mentioned in the introduction, computer graphics is now taking on many of the features and aspects that have long been the domain of images. Whereas once, jagged lines and flickering highlights were the norm, now they are unacceptable. Thus, graphic primitive generation algorithms are being modified to produce "image quality" output. Also, new primitives that are image oriented are being used. Examples of this are the use of areas and textured surfaces as primitive elements in a graphical picture. There are two main areas where the interpolator can be used to improve the quality of existing graphic generation programs: the automatic production of "de-jaggied" lines and vectors and the anti-aliasing of graphical entitites.

A jaggied line is the result of overlaying a line of finite

width onto a display grid composed of impulses. It may be regarded as the convolutional sampling of a line image with an aperture whose transmittance function is a spatial impulse. The result is that picture elements which lie over the point on the screen for which the computation is being done are written as black (or the particular color of the line), and those over which the line does not lie are written white (or left as background). The result is that a line at a small angle to the raster consists of several offset rows of horizontal picture elements (figure 4.1). Many new

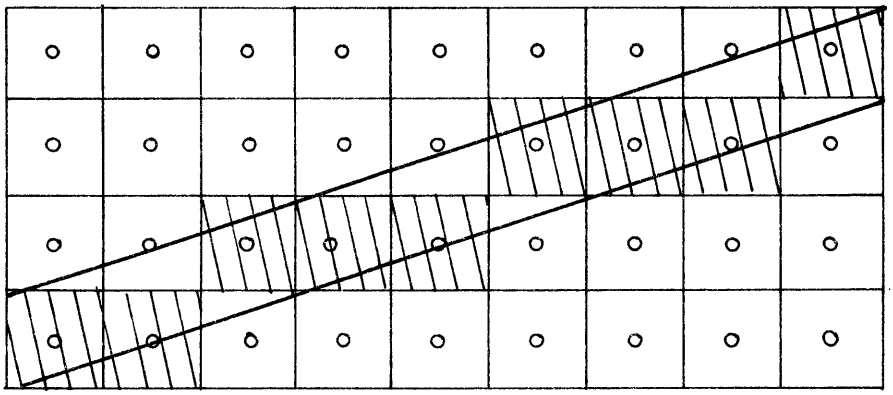


Figure 4.1: Jaggied Vector Generation

algorithms have been developed to improve this. Usually, a linear approach does not work, and a de-jaggying algorithm must be applied at the time of generation of the line. It

has been postulated that a linear filter could de-jaggy the lines by operating on the final display array, but that the filter length required would be as long as the longest line in the image.

A better approach implements the convolution of the image with a sample grid that is not impulses, but rather, the simulation of an aperture with some real, and spatially finite transmittance function. In the simplest, and hence most widely used, the screen is composed of squares, and the extent to which a given line covers the square is the extent to which that element is colored (figures 4.2, 4.3).

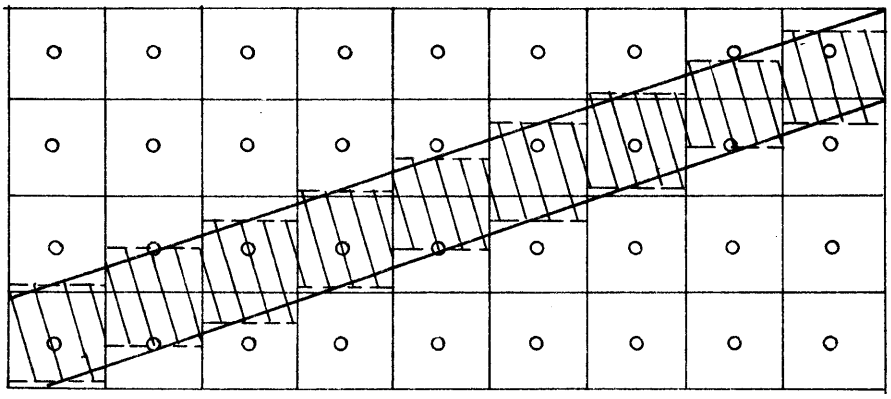


Figure 4.2: De-Jaggied Vector Generation

For simple narrow lines, this implies that a neighborhood of picture elements must be written in rapid sequence to

to draw the line. This is precisely the type of operation that is simple in the interpolator. Here, it is being used as a small, fast buffer for part of the display with the additional feature that when one picture element is called for, all the surrounding ones are available free of charge, or access time.

For a fully automatic de-jaggied vector algorithm to be implemented in the interpolation hardware, a few circuit modifications would be necessary. The tables would be re-defined to write into the line buffers the color of the line, as a function of the current screen value, and the amount of line overlap onto that particular picture element. A scan conversion vector algorithm that wrote the points in descending screen vertical position would allow the image to be continuous through the line generation time and effectively would make the drawing a background operation.

It should be noted that a jaggy line is merely a special case of an edge: Jaggies can appear at the boundaries of any figure in the display. Thus, to completely anti-alias a simple graphic image, all edges would have to be written into the frame memory via the interpolator.

One special case of de-jaggying stands out as unique: that

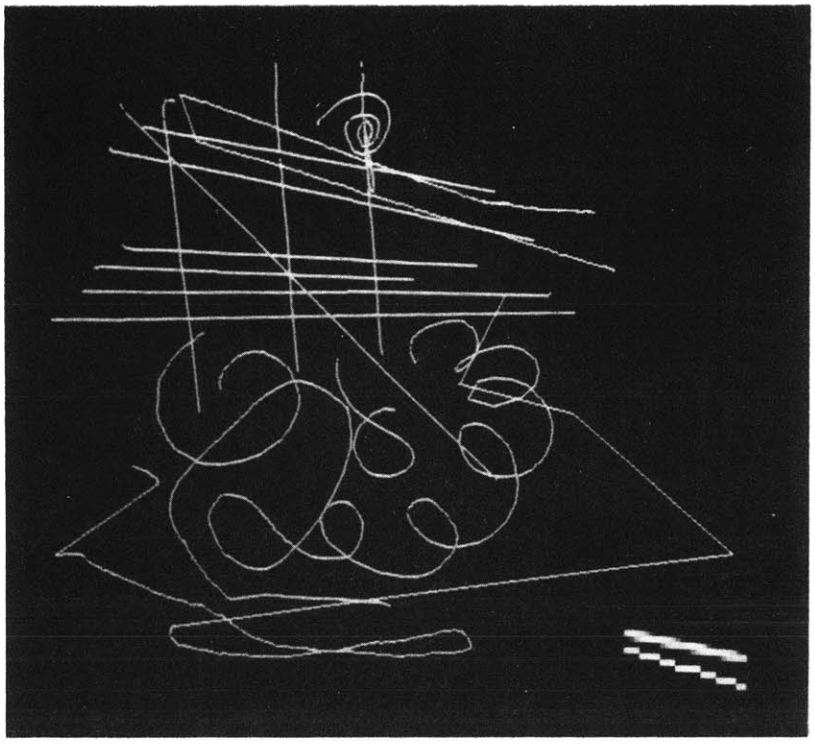
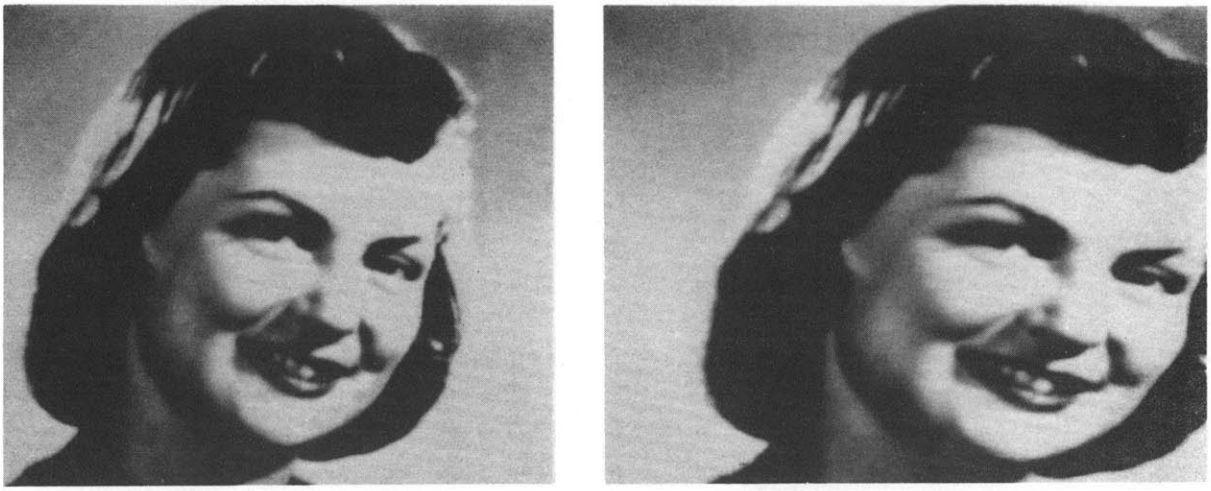
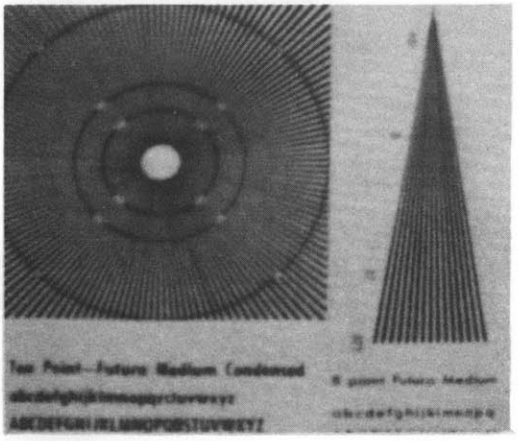


**Figure 4.3: Jaggy and De-jaggied Lines**

of textual display. For small characters, the length of the lines involved are, in fact, on the order of the length of the filter used in the interpolation. Thus the interpolator may be used directly to enhance the appearance of text on the screen. It may also be used to generate that text directly from read-only memory character generators that provide only one bit per point, and hence no gray levels. The interpolator will insert the required gray levels as it performs the filtering.





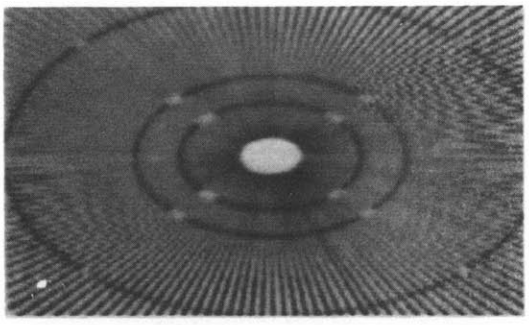




**Figure 4.4: Interpolative Horizontal Scaling**

## Appendix

The operation of the table-driven interpolator has been simulated in software at the Architecture Machine Group computation facility. The simulation consists of a program to execute each discrete, time-consuming step of the process, and a program to generate the table values for a scale change. It stores the time associated with each operation and each operation's execution time is specifiable separately. Most of the pictures included in this thesis were generated using this program.

## REFERENCES


Andrews, H.C. "Interactive, Real-time Adaptive Non-linear Convolution." Comtal Corp Image Processing application note, 1977.

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

Bolt, R.A. "Spatial Data Management, Interim Report." Report to the Defense Advanced Research Projects Agency, Contract number MDA903-77-C-0037, MIT, October, 1977.

Catmull, E. "A Subdivision Algorithm for Computer Display of Curved Surfaces." Unpublished Doctoral Thesis, University of Utah, 1974.

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

Crow, F.C. "The aliasing Problem in Computer Synthesized Shaded Images." Unpublished Doctoral Thesis, University of Utah, March, 1976.

EIA. "Electronic Industries Association Consumer Electronics Annual Review." 1975

Entwisle, J. "An Image Processing Approach to Computer Graphics." IEEE Conference on Computer Graphics, July, 1974.

Hu, J.V. and Rabiner, L.R. "Design Techniques for Two-Dimensional Digital Filters." IEEE Transactions of Audio and Electroacoustics, October, 1972.

Huang, T.S., Schreiber, W.F., Tretiak, O.J. "Image Processing." Proceedings of the IEEE, November, 1971.

Negroponte, N.P. "Return of the Sunday Painter." Future Impact of Computers and Information Processing. Edited by Michael Dertouzos and Joel Moses. 1977.

Negroponte, N.P. "Raster Scan Approaches to Computer Graphics." Computers and Graphics, 2, 1977, 179-193.

Oppenheim, A.V. and Schafer, R.W. Digital Signal Processing. Prentice-Hall, Inc. 1975.

Peled, A. and Bede, L. "A New Approach to the Realization of Nonrecursive Digital Filters." IEEE Transactions on Audio and Electroacoustics, December, 1973.

Rabiner, L.R. and Gold, B. Theory and Application of Digital Signal Processing. Prentice-Hall, Inc. 1975.

Rabiner, L.R. and Schafer, R.W. "A Digital Signal Processing Approach to Interpolation." Proceedings of the IEEE, June, 1973.

Acknowledgements

I would like to thank Professor Nicholas Negroponte for providing the time, incentive and encouragement necessary for the completion of this work. I would also like to thank the Office of Naval Research for their sponsorship. Additionally, Tom Boyle for his help in the programming; Dick Bolt, for his guidance and direction; Beth Luchner for her admirable job editing, decoding and producing this report.