Stills from Movies:
Personal Photography and the Personal Computer

by

Leo Hourvitz

B.S.E., University of Michigan
1982

Submitted to the Department of Architecture
in partial fulfillment of
the requirements for
the degree of

Master of Science in Visual Studies

at the

Massachusetts Institute of Technology

February 1985

copyright (c) Massachusetts Institute of Technology 1984

Signature of author ....
Leo Hourvitz
Department of Architecture
Friday, 31 August 1984

Certified by ....
Professor Andrew Lippman
Associate Professor of Media Technology
Thesis Supervisor

Accepted by ....
Professor Nicholas Negroponte
Chairman, Departmental Committee for Graduate Students

Rotch
MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
FEB 22 1985
LIBRARIES

Stills from Movies: Personal Photography and the Personal Computer

by
Leo Hourvitz

Submitted to the Department of Architecture on August 31, 1984
in partial fulfillment of the requirements for the degree of
Master of Science in Visual Studies.

Abstract

A system for the capture and combination of video images is proposed as a prototype for a future personal computer that allows free manipulation and intermixing of information from the computer, video, and photography domains. The benefits in power, flexibility, and applicability to each area from combining with the others are explored.

The history and development of the three media contributing to the work is briefly examined, as are the techniques usuable in moving between and combining them. The "Moviecliptool" system is described. Moviecliptool's purpose is to allow the generation of high quality stills from a recorded video sequence; that is, moving from the video to the photography domain via the computer. It allows the user to capture a sequence of frames from conventional video equipment into the computer. The frames are brought up on an interactive display where the user can select among them, view them in detail, and, more importantly, begin to combine them to his taste. Moviecliptool provides tools for image enhancements to try to alleviate the noise problems inherent in the video medium, as well as interactive tools for manipulating the content of the images, either by direct modification or by selective combination of the images. The environment supporting moviecliptool is extensively described, as it has helped shape the design of the system.

Future developments and directions for similar systems are discussed. The appearance in the commercial world of relevant systems is seen as heralding a promising future for tools which transcend media boundaries.

Thesis Supervisor: Andrew Lippman
Title: Associate Professor of Media Technology

The work reported herein was supported by a grant from Coleco Industries, Inc.

Table of Contents

Chapter One

Introduction

This thesis is about the converging of three technologies, all of which are already well known to the world at large. The triple worlds of photography, personal computers, and home video are not new to our society. However, many of the possibilities of their combination have not and are not being realized in equipment available outside of research laboratories. The proposition that this thesis attempts to show is that it is not only beneficial to bring together these technologies, opening up new possibilities in the creation of images, but that there is much to be gained by doing so. It is only the lack of attention to the potential of this overlap that has prevented commercial systems for this work from being available at the consumer level.

The barriers preventing this are not technical in nature. At the Architecture Machine Group, as in a recent hobbyist article [Tymony 84], the techniques for building VCR/computer interfaces are well known. The necessary technologies of control, frame grabbing, and video output are known and in many cases in use. Yet, they are generally available only as professional units or as hobbyist projects. We are just beginning to see the appearance of consumer level video/computer units, such as the Koala Technologies video digitizer, made with everyman rather than the professional in mind.

The combination of the videotape deck, the personal computer, and photographic equipment into a single system offers more than the simple bringing together of these items into one place. Making the video deck serve as a still camera, and making the computer a manipulator of images as well as textual data, opens up a vista of operations with both that were not reasonable to do before. As will be discussed later in the thesis, when using Moviecliptool, the taking of a picture does not stop with the recording of the seqeunce, but continues indefinitely. Although the images may have been recorded already, the selection and improvement operations remain. These activities correspond to the careful selection and printing procedures currently used by professional photographers, and bring the computer into the photo album, as printer, archivist, and display device.

With the continuing decline in the cost of these technologies, the accompanying decrease in the cost of the necessary computer capabilities to drive them, and the overwhelming amount of video equipment owned by consumers, the appearance of a system such as Moviecliptool seems to be only a matter of time. This thesis hopes to both help point the way towards and speed the arrival of that time.

Chapter Two

Forms of Visual Expression

Since the days of cave paintings, men have used the creation of imagery as a way of recording what they have seen or expressing their thoughts regarding it. In the past hundred years, however, there have also come to be ways of recording what has happened without placing an interpretation upon it; of being an objective recorder. Naturally, creative individuals have and will find ways to use these supposedley objective media as tools for creative expression.

2.1 Photography

Figure 2-1: Early field photographer, c. 1865
(All figures in section 2.1 are from [Upton 81])

> Students today photograph in many styles from intensely personal to cool and restrained. They make portraits, landscapes, technical photographs, psychological studies, abstractions, picture postcards, and more. In short, they work in all areas in which photography has already been used, and they will undoubtedley open up new ones as well. [Upton 81]

The current medium of photography is, as the above passage suggests, a rich field usable for fielding visions of the world. Not only the large body of straight photographers, who make their photographic art by careful selection of the piece of the world they wish to survey, but the less traditional workers in the photographic realm, to whom devloper and fixer are just a new kind of paintbrush and chalk, have helped to move photography to the art it is regarded as today. But even as photographic exhibits move alongside those of painting of fine arts museums, photography retains much of its older character as a portraiture and news medium. With the exception of *The Wall Street Journal*, no newspaper would think of putting out an issue devoid of photographs. And for sheer volume, nothing can match the friends, family and fun pictures taken in the thousands by amateur photographers (i.e., everyone) the world over.

2.1.1 History

Figure 2-2: One of Daguerre's first photographs

The first photographs are attributed to Louis Jacques Mande Daguerre, who announced in 1839 his Daguerreotype process. This used silver-plated copper sheets, and although a Daguerreotype took several minutes to expose, the result was a detailed and rich image. They also age well, such that one can still see and appreciate Daguerreotypes.

Daguerreotypes became an instant success with the public, not only in Daguerre's France but across the world as well. By 1853 three million per year were being produced in the United States [Upton 81]. Their primary use was for portraiture. Whereas up to this time only the very wealthy could afford to have portraits painted, Daguerreotypes opened the idea of a permanent portrait up to the millions.

> "Even when exposure times were long and having one's portrait meant sitting for several minutes with eyes watering, trying not to blink or move, people flocked to portrait studios to have their likenesses struck by 'the sacred radiance of the Sun.'" [Upton 81]

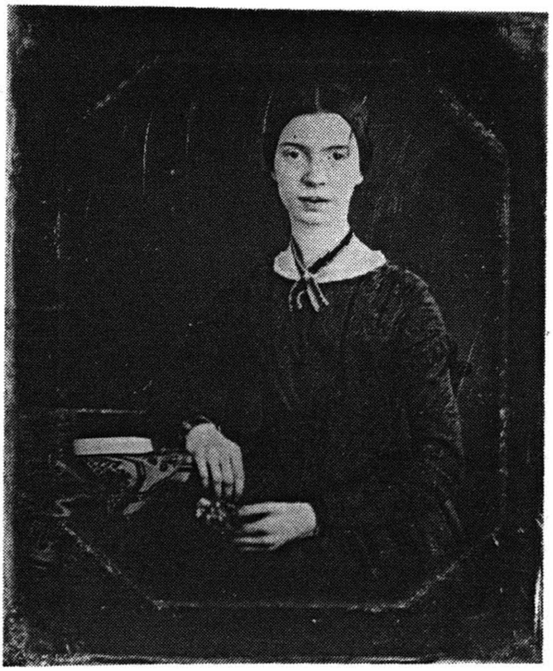


Figure 2-3: Emily Dickinson at seventeen

Unfortunately, Daguerreotypes also had several serious disadvantages. They were not a positive/negative process, so additonal prints could not be made, they were poisonous to process, and the metal plates were not viewable at all angles. These shortcomings gave birth to the wet plate process, which used coated pieces of glass as the negative. This wet plate process, introduced in 1851, was far more light-sensitive than the Daguerreotype but required that the glass negative be created, exposed, developed, fixed, and washed while still wet. This meant that a photographer had to have a fair amount of equipment with him wherever he might want to take a picture. Despite this, this was the process which was used to take the famous Matthew Brady pictures of the Civil War -- a significant first in the use of photography as a news medium.

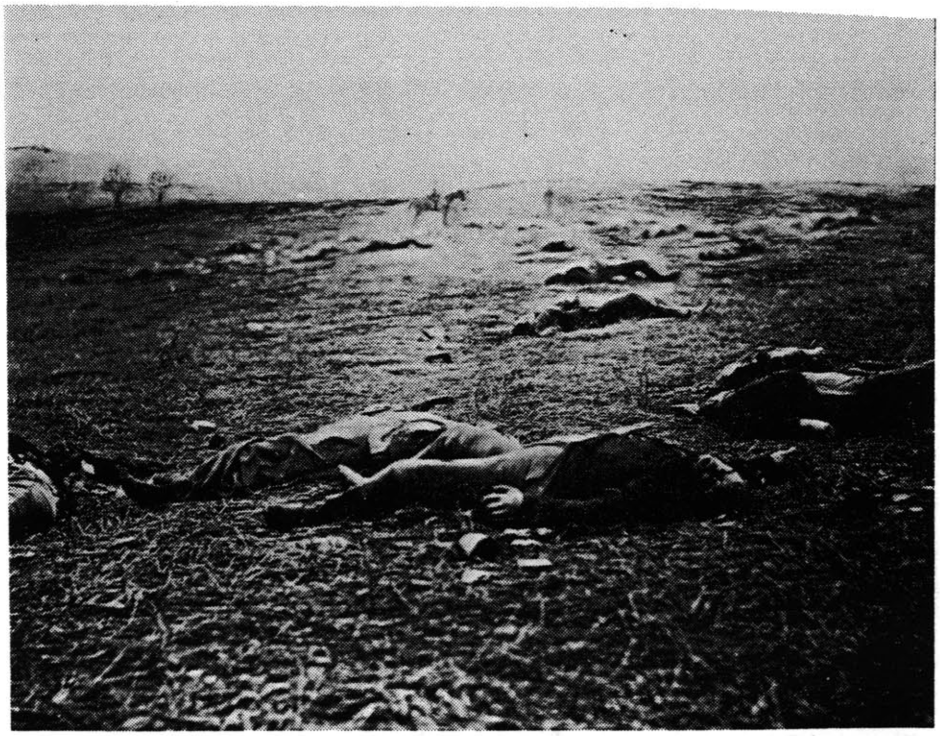


Figure 2-4: *A Harvest of Death*, Gettysburg, July 1863

With the wet plate process in their trunk, photographers could finally venture out of the portrait studio. Although portraiture remained the most popular use of photography, photographers were now able to document and comment on other parts of the world. The Crimean War and the Civil War marked the bringing home of images of war to the civilian population for the first time, and was in its time as provoking as televisions' placement of Vietnam into American living rooms was to be a hundred years later.

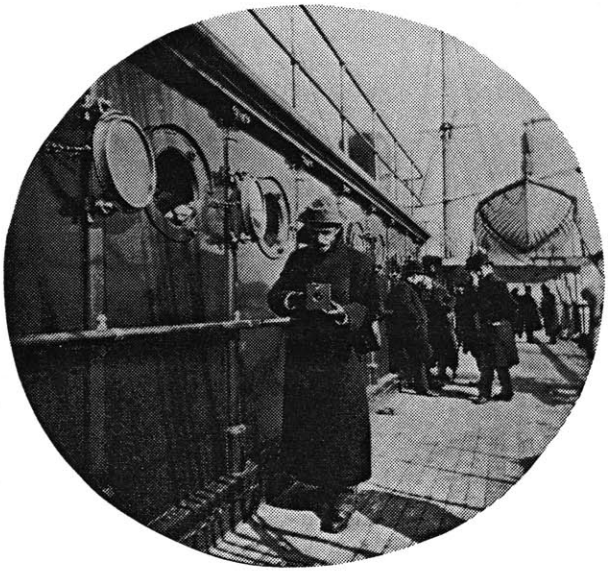


Figure 2-5:
George Eastman
and his Kodak

By the 1880's everyone had seen photography, but allowing everyone to practice it came about only after the introduction of products like the Kodak roll film camera, produced by George Eastman in 1888. The Kodak needed no adjustments to use, and the camera with exposed film inside would be sent back to Kodak for processing. The amateur boom in cameras began; people began to regard photography as something which everyone could own. A form of recreation, recording and creativity for everyman had arrived.

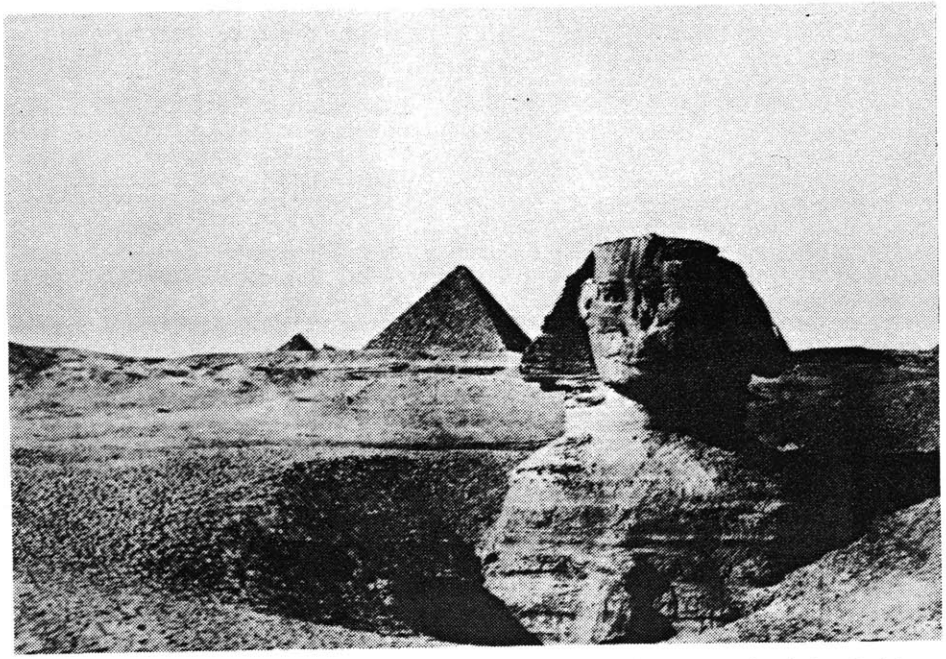
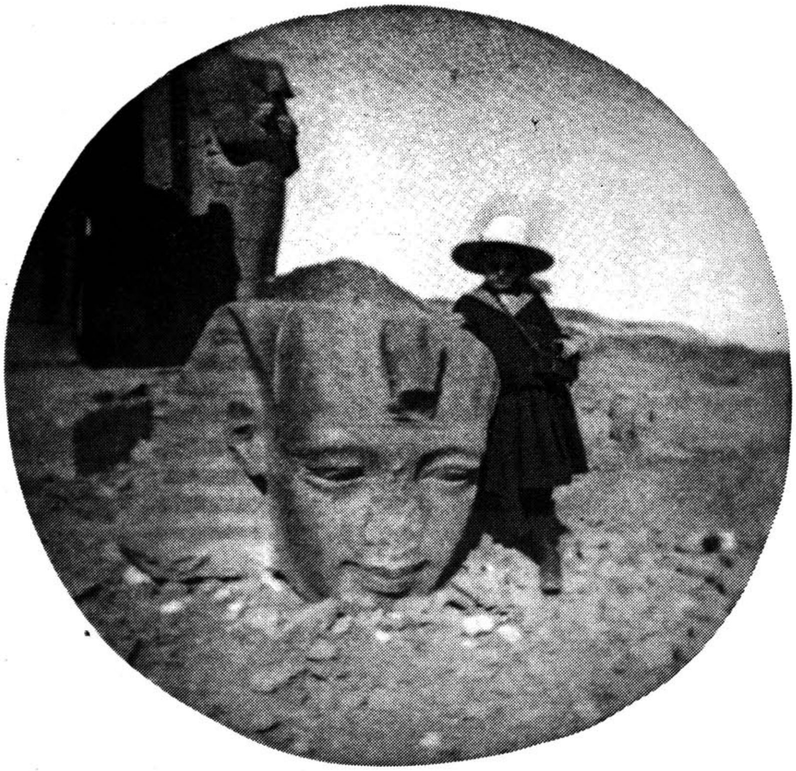


Figure 2-6: An early wet plate photograph of the Sphinx

2.1.2 Evolution

Photography has continued evolving towards new purposes. The power of photography to capture events was not fully realized until two forces of publicity came to harness it. These were the newspaper and the picture magazine. The newspapers delighted in photography once the halftone process became available, and they could bring to their readers more information and more of the spirit of an event than words alone had been capable of. And in the picture

Figure 2-7: constrasts with the arrival of tourist photography below (and the arrival of the Kodak around her neck)



magazines, photography found a literary outlet. In the pages of *Life* and *Look* photography was able to tell the story with more emphasis than the accompanying text, which served merely to string together the pictures. This purpose of visually enhancing the news function was to be overshadowed by video in the post-World-War-II era, but photography blazed the way in the fleshing out of verbal reports.

As the new century brought in concerns that had not prevailed in the preceding hundred years, photographers began to find ways to use photographs to express social, political, and artistic concerns. August Sanders (see Figure 2-9 below) docuemnted a generation of Germans, as the photographers of the

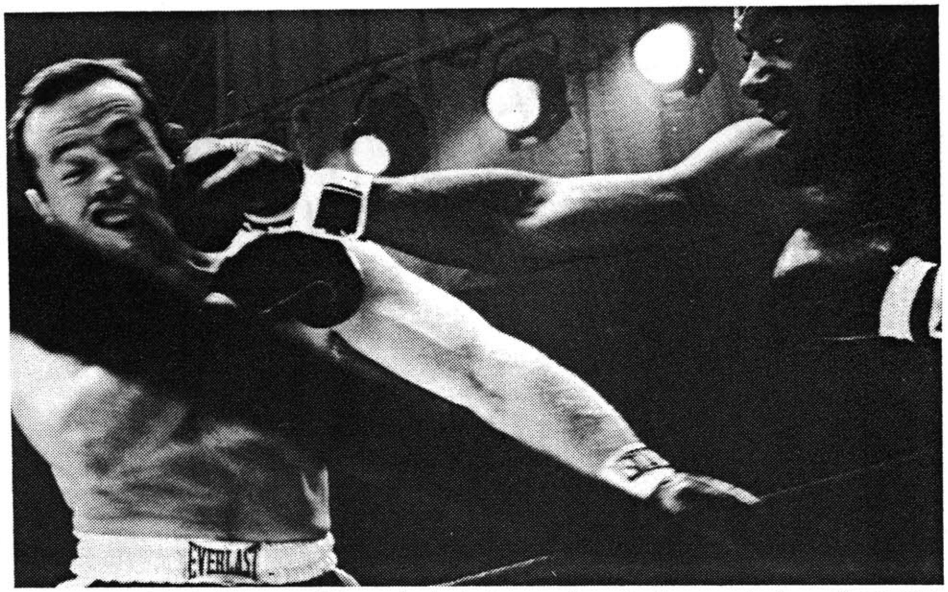


Figure 2-8: The tradition of photojournalism is the capturing of events, as in this shot of the Patterson-Johansson fight

American government brought back visual proof of the suffering of a farmland. Alfred Steiglitz showed us the voyage from Europe.

Others pushed the frontiers of photography in different directions. The Bauhaus movement used all techniques available to get a visual effect, and photography proved a powerful one. Later, after the war, the straight photography movement led by photographers such as Paul Strand and Minor White (see Figure 2-11) unfolded as photography was finally accepted as art and folded into the circles of artistic endeavors.

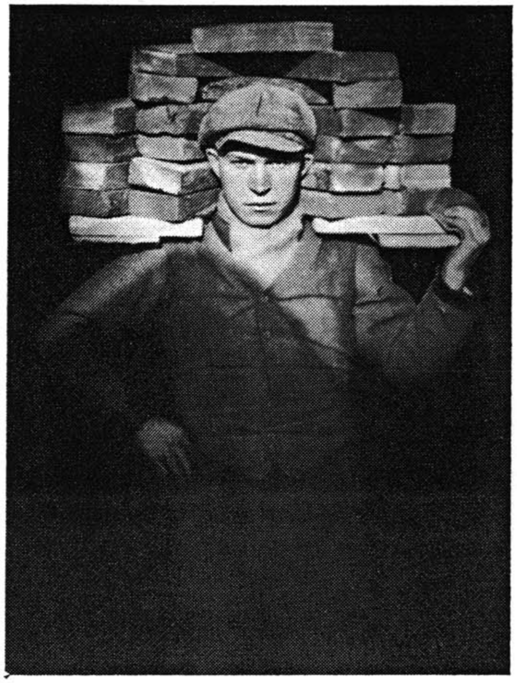


Figure 2-9: *Laborer*, August Sander

2.2 Video

The history of video was the largely the history of broadcast television until the introduction of the Sony Portapak unit in 1968. The Portapak was a portable videotape recorder and handheld camera, and it opened a new field of nonprofessional video. The videotape became an accesory to the independent filmmakers' 16mm camera.

The sale of video recorders for home use began with the Portapak, but video units did not immediately become popular, as prices were still quite high. As costs declined, volume increased, however, and in 1979 over a million videocassette recorders were sold in the US.

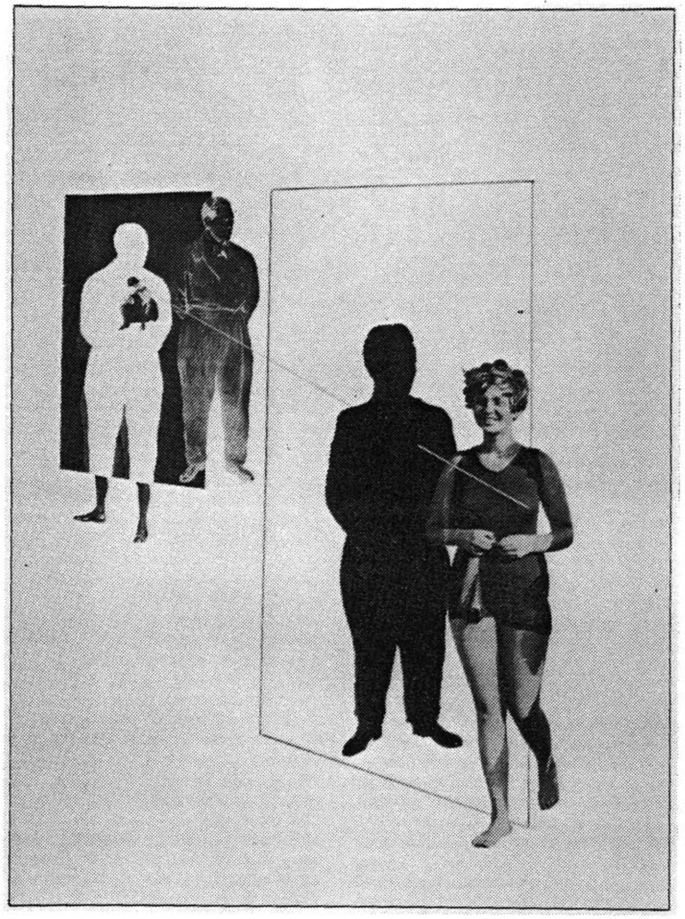


Figure 2-10: Laszlo Moholy-Nagy's *Jealousy*. Nagy was in the midst of the Bauhaus movement

2.3 Computer Imagery

> "The personal computer can be regarded as the newest example of human mediums of communication." [Kay 77]

Computer imagery dates from Ivan Sutherland's Sketchpad [Sutherland 63]. Using a computer-driven vector display, Sketchpad allowed a user to create a simple composition of

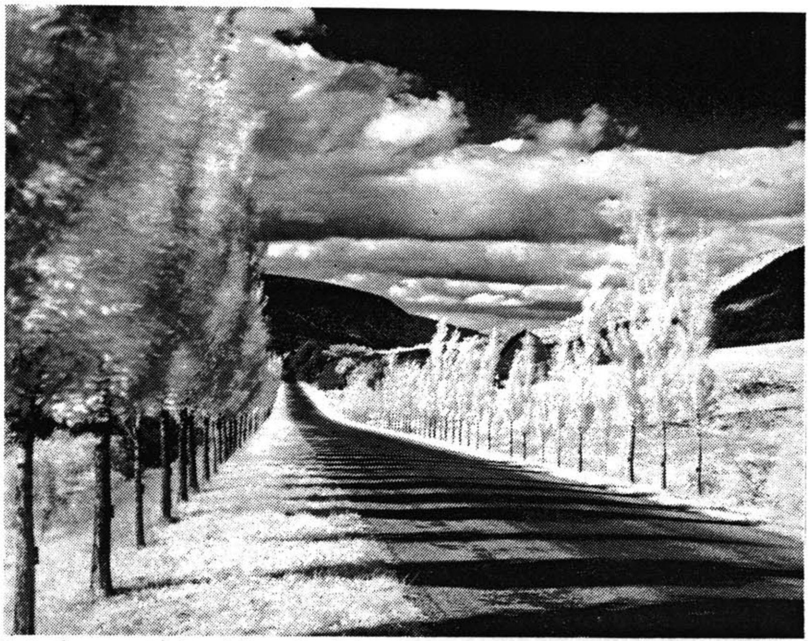


Figure 2-11: An example of the straight photographic style of Minor White

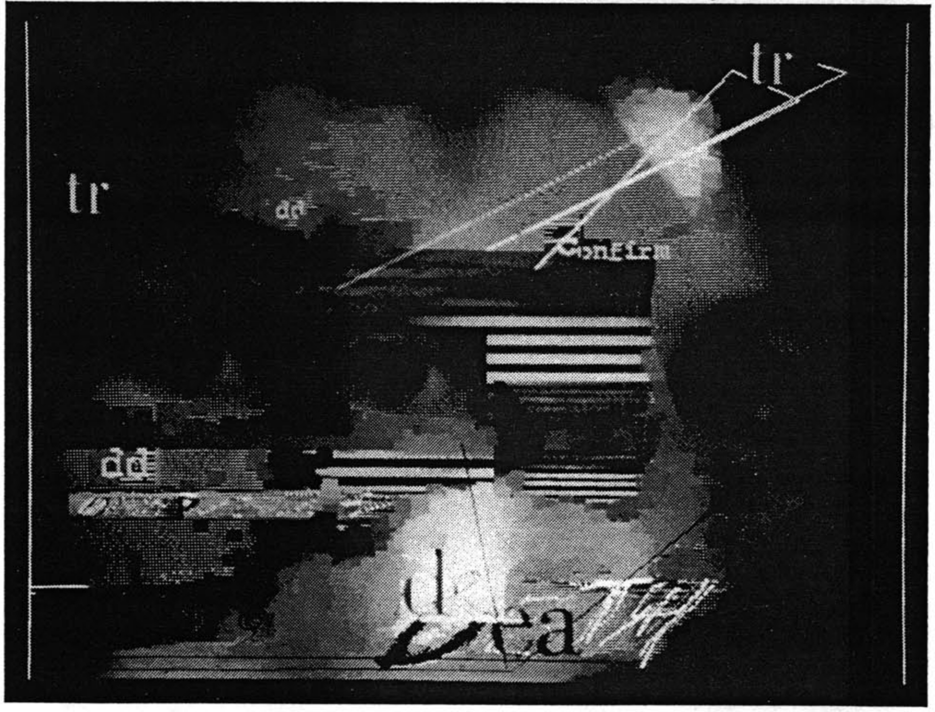


Figure 2-12: *untitled*, Joel Slayton

lines, circles, and rectangles. Although it did not have the peripheral devices or the sophisticated commands of some of today's systems for freehand sketching, it defined the basic tools of a new medium.

2.3.1 The Early Applications: Data Representation and Mechanical Drawing

As computer graphics devices became more readily available, they were been used as were the computers of the time: in a non-interactive, data-processing style. The earliest commonly available graphics output device was the plotter. It was well-suited for doing drawings such as simple bar charts and plots, and engineering drawings. Computer-controlled plotters are significantly more accurate than draftsmen, and often are much faster. Because of these characteristics, early computer graphics was devoted to applications such as plotting functions for engineering and scientific work, and preliminary versions of computer-aided design (see figures 2-13 and 2-14).

One of the goals of early computer graphics was frequently to provide an easy-to-understand interpretation of the tremendous databases on the computer. Plotting was a way to do this, as were various applications of computer mapping (Figure 2-15).

2.3.2 Simulation of Reality

Efforts, particularly in computer-aided design, were directed towards enabling the computer to simulate reality. Computers

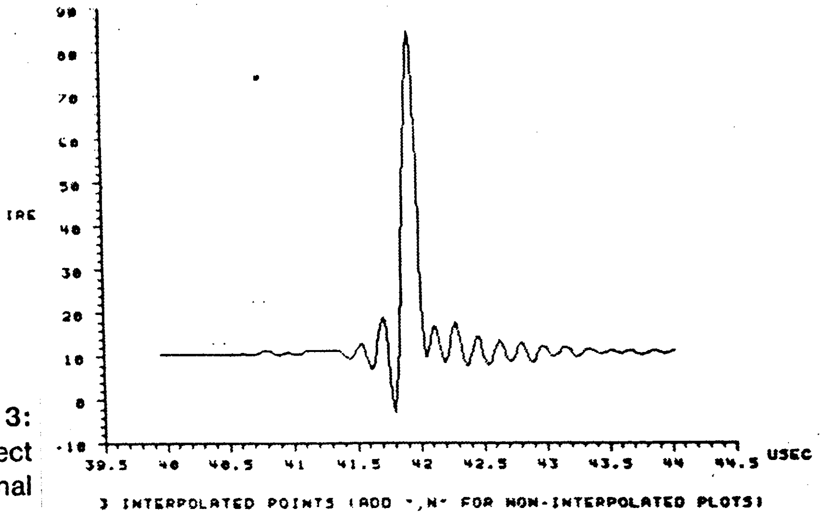


Figure 2-13: A plot of the ringing effect in a video signal

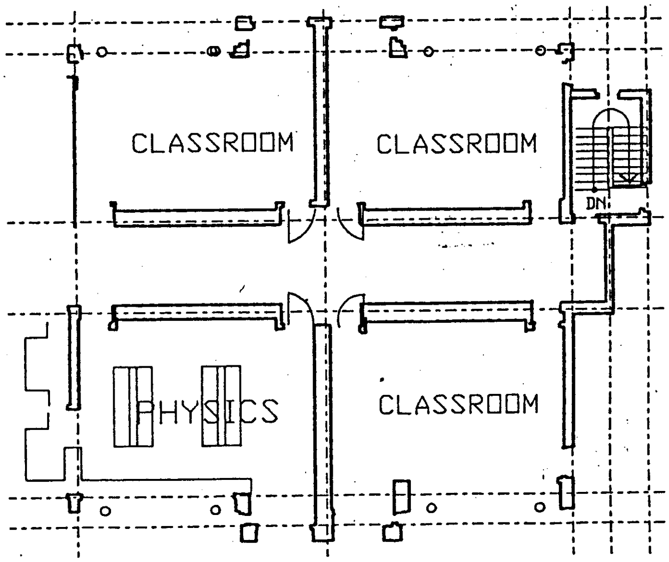


Figure 2-14: An example of two-dimensional computer-aided design

are often used for simulation, and in the graphics area this became an effort to produce a picture of simulated reality. Ranging from simple three-dimensional line drawings (figure 2-16) to the near-perfect pictures from current movies such as The Last Starfighter(figure 2-17), the struggle for simulated reality has been one of the continuing areas of research in

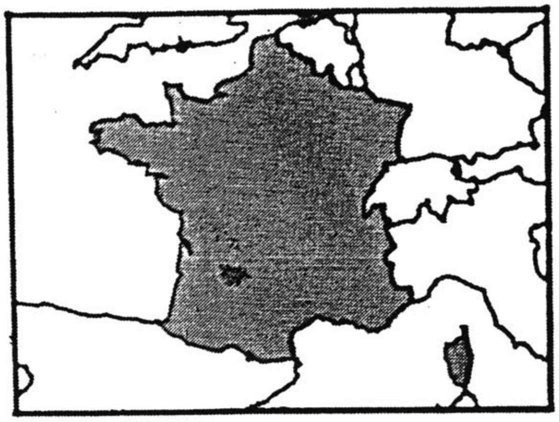


Figure 2-15: France, from a computerized longitude/latitude database

computer graphics. Often, the goal has been to replace photography and optical effects with computer-generated pictures. Jim Blinn used image synthesis to provide the world, well before the event ever occurred, movies of American spacecraft passing by Jupiter and Saturn. The television industry has used the wide variety of 'special effects' easily possible with computers to generate flashy logo sequences, as have several movies. In The Last Starfighter, the computer-generated images replaced what would have been a carefully crafted scene of model photography. As computers become able to synthesize scenes that successfully mimic more and more situations, they will become 'photographers' of many unusual situations.

2.3.3 User Interface Graphics

Another area where computer images have been important is in providing the user interface to general computer systems. In this case, the graphics is not necessarily the purpose of the

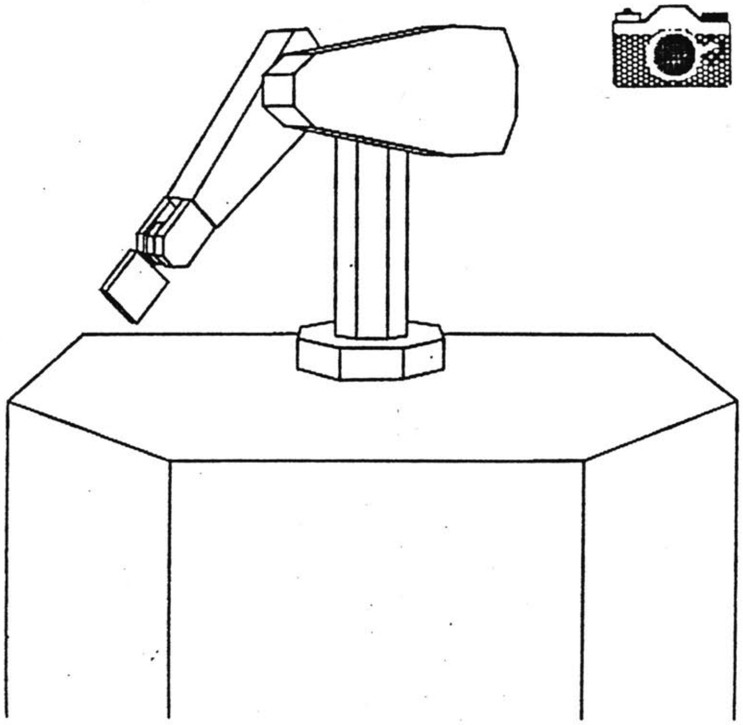


Figure 2-16: A simulated robot arm poses for a portrait

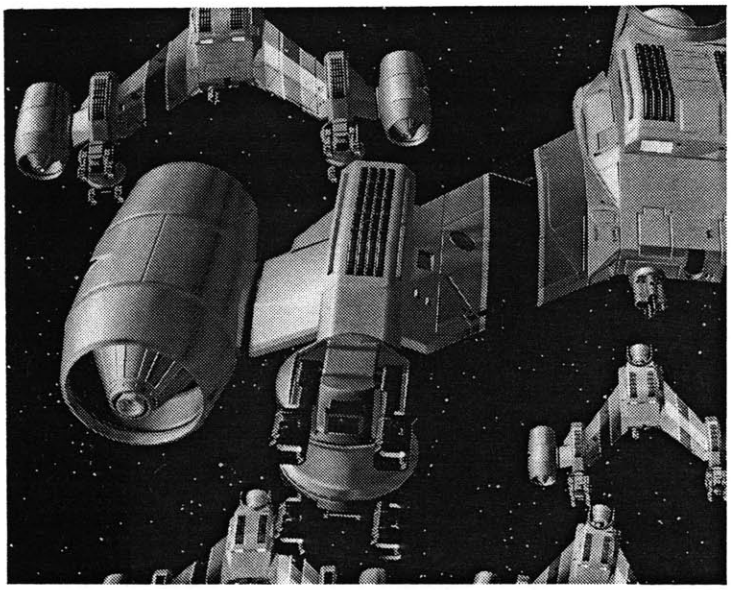
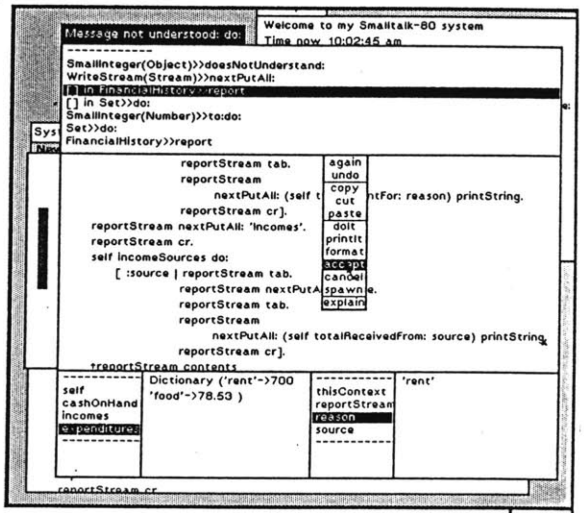


Figure 2-17: A still from The Last Starfighter

computer system in use, but is the means through which the user addresses the computer. The first comprehensive system for this is the justly famous Xerox Alto, with its Smalltalk langugage [Flegal 81], [Goldberg 83], [Ingalls 81], [Kay 77](Figure 2-18). In Smalltalk, everything in the computer system has a screen appearance and is manipulated through the screen. Textual commands are used, but the act of guiding the system heavily relies on changes in appearance of the items being manipulated. The ideas of Smalltalk have been widely adopted, and are examined in more detail in Section 3.3.



Welcome to my Smalltalk-80 system
Time now 10:02:45 am

Message not understood: do:

```
------------
SmallInteger(Object)>>doesNotUnderstand:
WriteStream(Stream)>>nextPutAll:
[] in FinancialHistory>>report
[] in Set>>do:
SmallInteger(Number)>>to:do:
Set>>do:
FinancialHistory>>report
```

```
                reportStream tab.
                reportStream
                      nextPutAll: (self t            ntFor: reason) printString.
                reportStream cr].
    reportStream nextPutAll: 'Incomes'.
    reportStream cr.
    self incomeSources do:
        [ :source | reportStream tab.
                reportStream nextPutA           e.
                reportStream tab.
                reportStream
                      nextPutAll: (self totalReceivedFrom: source) printString.
                reportStream cr].
    ↑reportStream contents
```

```
----------
self
cashOnHand
incomes
expenditures
----------
```

Dictionary ('rent'->700
'food'->78.53)

```
----------
thisContext
reportStream
reason
source
----------
```

'rent'

reportStream cr

Figure 2-18: A screen from the Smalltalk-80 system [Goldberg 83]

2.3.4 Image Processing

Image processing refers to a field that includes techniques for compressing, reducing the noise in, and understanding real world images. Movieccliptool shares the goal of noise reduction, and borrows many of the techniques of image processing for its own ends.

A central concept to image processing is the idea of spatial frequency. Spatial frequency regards the successive pixels in a row or column of an image as representing samples in a signal, and analyzes the frequencies in this signal. The fast fourier transform, discovered in 1965, and related techniques, have made it practical to actually find the spatial frequencies present in an image, and use that information for various applications. In [Burt 83], the concept is used to combine images in a manner appropriate to their frequency content.

Image processing researchers have found many techniques for realizing gains in the quality of imagery. The most straightforward of these include operations such as averaging, filtering, or otherwise combining pixels that are either adjacent in one image or in the same place in several. In [Huang 81] a number of examples of linear filtering of image sequences are presented, as well as some nonlinear techniques such as finding the median of a series of images. This type of image processing is reflected in the linear operators in Moviecliptool, discussed in chapter five.

The omnipresence of television in the western world has produced a great deal of work regarding various characteristics of the broadcast signal. [Coll 76] discusses the frequency and time-varying characteristics of the broadcast signal. Techniques such as those in [Netravali 78] have been found for reducing the noise in a color signal, although the particular one cited depends mostly on the high correlation between the chrominance and luiminance information in a signal.

Image processing researchers have used other techniques, such as those in [Hausler 72] and [Amato 84] to produce improvements both in purely optical properties, such as depth of field, and in the 'look' of a video project. These are discussed in the next chapter.

Chapter Three

The Bases of Movieclipool

3.1 Computer-Video Symbiosis

> Fades, dissolves, zooms, and other grammatical transitions of film are also available. This ... indicates the ability of computer graphics to easily absorb many imaging techniques proven effective by earlier media. [Richards 83]

The computer has already earned a place in the video editing suite. In this section, what it is doing there, as well as what the other functions it may have relative to video, are examined.

3.1.1 Computerized Event Marking

Computers have started in the video area as they first have in many fields, by eliminating the obviously boring and repititive tasks involved with the work. In the case of video, computers have entered the editing suite as bookkeepers for the detailed task of editing together a video segment. The necessity of maintaining a careful list of the changes, and selectively changing it to produce the best possible edit, has provided a number of companies opportunities to produce equipment which uses simple computer equipment[1]to control the video devices used in editing.

[1] not that the prices of this equipment have reflected the trivial nature of the computer equipment

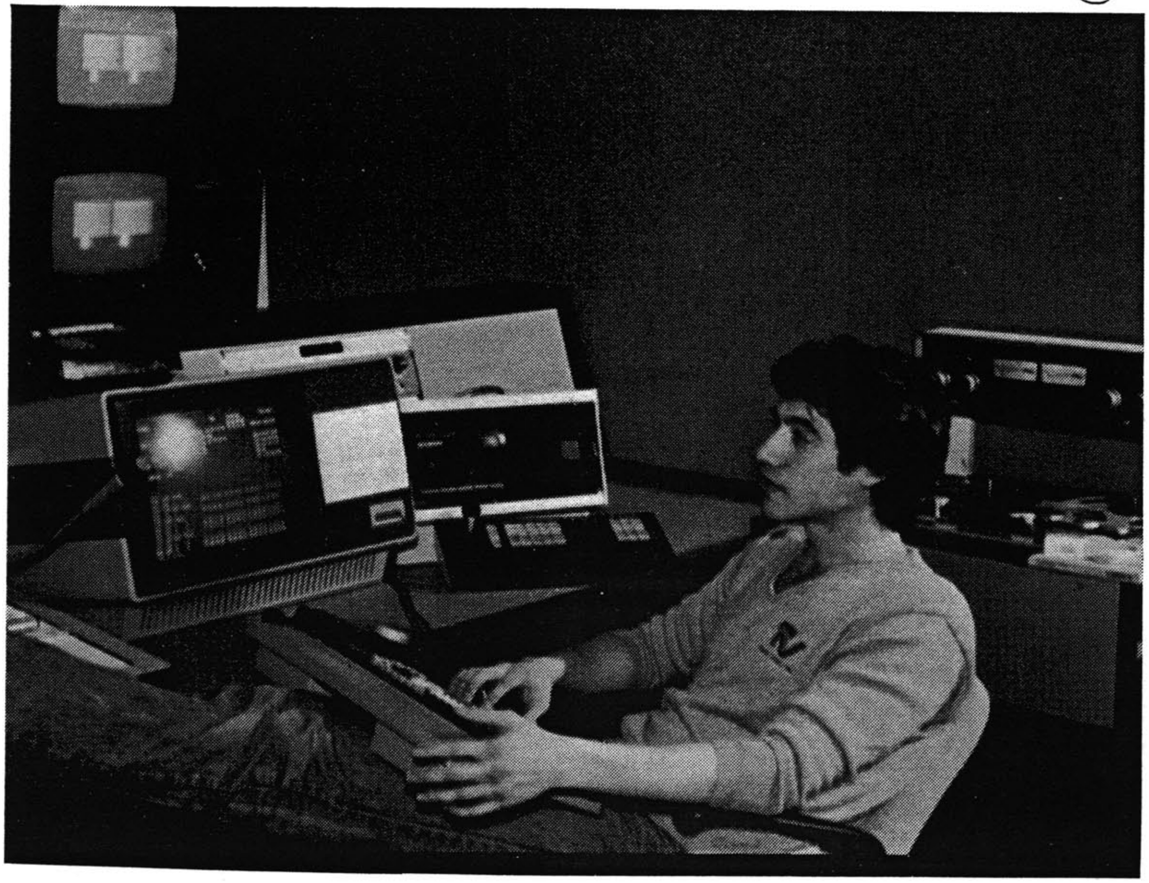


Figure 3-1: Using a CMX computerized editing system

The development of this equipment for the professional market has caused several techniques to come into widespread use. Because maintaining a detailed record of edits necessitates identifying individual frames on both the source and resulting tapes accurately, a scheme for labelling videotape frames known as SMPTE time code has been used. This writes a specified sequence of digital data into the audio track of a videotape, enabling appropriately designed equipment to identify the current frame. Variants of this technique are used to record this information in the video track iteself. The fact that it is possible to directly identify video frames will be important to Moviecliptool, and its use will be outlined in full.

There are possiblities for other applications of the computer as a bookkeeper besides the simple-minded event list building used in editing. One can imagine a computer that was present at the time the video images were created as being used for recording any number of items, such as when each image when taken, where the camera was and where it was pointed (even if only relative to where it was pointed afterwards), and the lighting and environmental conditions under which the images were taken. This 'intelligent camera' could record the information with the images for future use. For instance, a camera that recorded the lighting conditions on the tape could help a post-processing stage tremendously; knowing that a certain scene was shot in a dark room might tell a smart image processor how to better eliminate noise from that image.

In the context of Moviecliptool, we can see the intelligent camera recording images that the user is especially interested in; and, in fact, we can see the camera actively changing the way it takes images in response to what it knows about the environment. If the camera has decided the environment is contrasty, it might vary its exposure to capture all the information, relying on a Moviecliptool at home to reassemble these images into one that captures in a single shot the information recorded over a series.

3.1.2 Postprocessing of video images

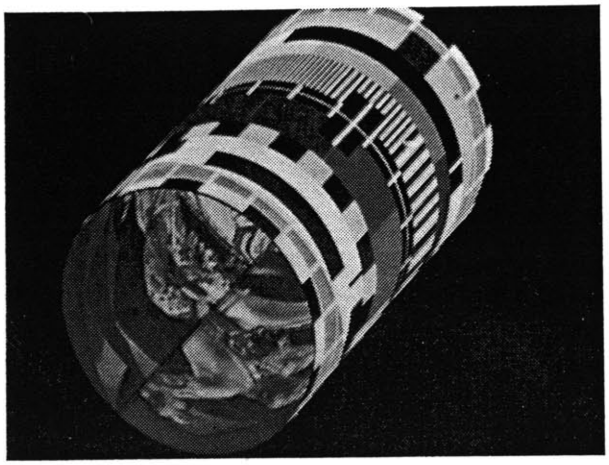


Figure 3-2: An example of the special effects that can be generated in real time from a video signal by an MCI/Quantel Mirage

Computers have been used in creating various desired special effects in video post-production, and recently in live boradcasts as well. The types of special effects generated are often borrowed either from the effects generated by analog video switchers, or from the effects used in the computer image synthesis fields. The application of these effects in broadcast TV has resulted in another set of effects being added to the producers' bag of tricks. These systems, even more so than the computerized editing suites, have been the province of extremely expensive professional systems[2].

3.1.3 Creative Image Processing

Some amount of work has been done recently in improving the quality of a video work by applying computer image

[2] The MCI/Quantel Mirage is priced at around $300,000.

processing to it [Amato 84]. The only instance of this known to the author used several fairly straightforward image processing techniques to reduce the noise in a project that had been shot in video, but which was to be distributed as a 35mm movie. This is the closest in concept to what Movieсliptool has tried to achieve. The result of the work was very satisfying to the producers, who had shot in video for its versatility and price. The digital processing was also used to perform scene-by-scene color and brightness correction. More work in this area will undoubtedley occur, as the increasing use of video and the increasing ability of computers to quickly modify entire images makes it practical.

3.2 Computer-Photography Symbiosis

> A number of people have begun using the computer as an extension of their work in photography and electronic imaging. [Furlong 83]

By putting computers to work in darkrooms, in printing apparatus, and someday in the camera itself, there are a number of possibilities available for easier or better photographic results. In this section some of these possibilities are reviewed.

3.2.1 Exposure Control and Color Correction

When equipped with appropriate sensing instruments, computers have already proven themselves quite capable of making technician-level decisons about the various steps in a conventional photographic process. The most obvious example of this is the automatic picture processing and printing machinery used at roadside photo processors across the country. The computer in these machines looks at the negatives to determine overall density, color balance, and other properties, and adjusts the printing variables accordingly. While these machines are no match for a skilled photographer doing custom printing, they provide a basic level of competance in printing.

3.2.2 Depth of Focus Improvements

As described in [Hausler 72], computers can be used to recover information present in a picture that is not available via other means. Because of the ability of the computer to move the information between various domains (in this case, between the spatial and frequency ones), it can sometimes provide operations such as the depth of focus procedure which are totally unavailable without the computer.

In this case, what Hausler did was to use a great deal of knowledge about the context in which the picture was taken to improve the depth of focus for an image. By first degrading the image through depth to obtain an image which was equally bad at all depths, then performing coherent filtering on this image, Hausler was able to reconstruct an image which simply had more depth of field than the original. He specifically notes in his paper the advantages of performing this technique electronically rather than optically (as he first did it).

3.2.3 The Digital Darkroom

The ability of computers to do automatic compensation for exposure differences by controlling photogrpahic equipment has demonstrated that they can be made to understand the printing process, and the application of digital image processing has demonstrated that the digital image form offers more potential techniques than any other form. This combination has led several people to suggest the concept of the digital darkroom. In the digital darkroom, a printer does not get out his paper, his polycontrast filters, and his safelight, but rather his computer, his framebuffer, and his photographic software. There is little reason to suggest that a computer could not take over the functions of a darkroom, often providing a functionally superior process. Rather than dealing with different papers and the processing steps necessary to get different textures, the photographic artist simply selects a different texture and has it applied to his image. To vary contrast, simple look-up table techniques are applied. The dodging and burning techniques of photography are implemented with painting and combination operations.

This concept of the computerized darkroom lends much to

Moveicliptool. Moviecliptool is, in a sense, an electronic darkroom for video rather than photographic images. This is a more achievable goal in some sense, because the largest obstacle to the electronic darkroom is simply image quality. Computer imagery does not yet offer the resolution necessary to produce images of quality comparable to photography. Even at the 4000 line resolution used in production of images for motion pictures, a computer image does not allow itself to be blown up, projected on a screen, and still not show traces of its computer origin. However, by restricting itself to the video domain, Moviecliptool accepts as a necessary evil the 525-line resolution of the standard video environment.

3.2.4 The Electronic Still Camera

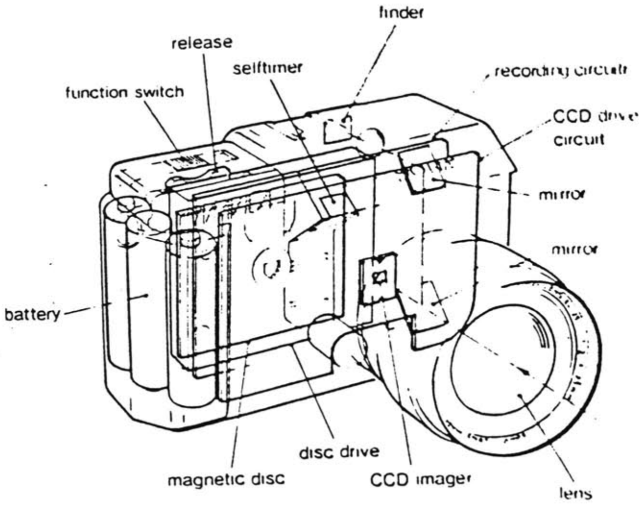


Figure 3-3: A cutaway drawing of Sony's prototype electronic still camera [Kihara83]

Another avenue that is suggested by the electronic darkroom is to do away with film in the capturing as well as printing of images. In this concept, a camera becomes a portable computer, reading the image with some sort of solid-state, digital-readout sensor, and recording the image in digital form on a portable, incredibly dense medium.

A device very close to this concept is the Sony Mavica prototype [Kihara 83], where images are recorded in analog form on a magnetic disk, for later playback and conversion to video on a specially-designed playback unit. The Mavica prototype exhibits both the positive and negative features of the electronic camera. It provides a reusable, quickly taken, and immediately available still image, of at least television resolution, from a camera package that is comparable in size and weight to conventional 35mm cameras. It's output can be displayed on standard television equipment, or printed on Sony's prototype Mavigraph printer. However, the Mavica's video resolution image does not compare to the images available in even the smallest format film and the price of a Mavica system (the camera and the Mavigraph printer) was projected to be around $1300. The Mavica was announced by Sony amidst much fanfare in late 1981, and its appearance on the general Japanese market was predicted for late 1983. As of this time, no actual products have been shipped, leading many to believe that the predicted improvements in resolution and costs did not materialize.

3.3 Graphical Information Systems

Many computer researchers have tried, with varying degrees of success, to use the graphics capability of computers to produce a system allowing users to manage their information, find out about some system, or write their program by manipulating and/or viewing graphic entities. A number of the important systems in this class are surveyed in this section.

3.3.1 Spatial Data Management System

SDMS [Bolt 79] was an attempt to organize information[3]in a spatial manner, by laying it out in a two-dimensional "Dataland" and allowing the user to roam at will throughout the created world. Upon seeing something of interest, the user could zoom in on it to view it in greater detail. Dataland also included some items that were more than informational, such as the phone, which could be dialed via a touch-sensitive screen.

Although SDMS remains a landmark in allowing graphical views of data, and was notable in using a fundamentally graphical layout to organize data, it had limitations in the power it offered a user. Little or no provision for modifying the displayed data was provided, making it a wonderful

[3] Specifically, that information stored in the computer, although SDMS did integrate types of information, such as pictures of project members, that were not considered the normal domain of computers

database browser but useless for editing. It did use color and a number of unusual input devices successfully, as well as showing the potential of multiple-screen systems for allowing easy switching between various levels of viewing the system.

3.3.2 Rocky's Boots

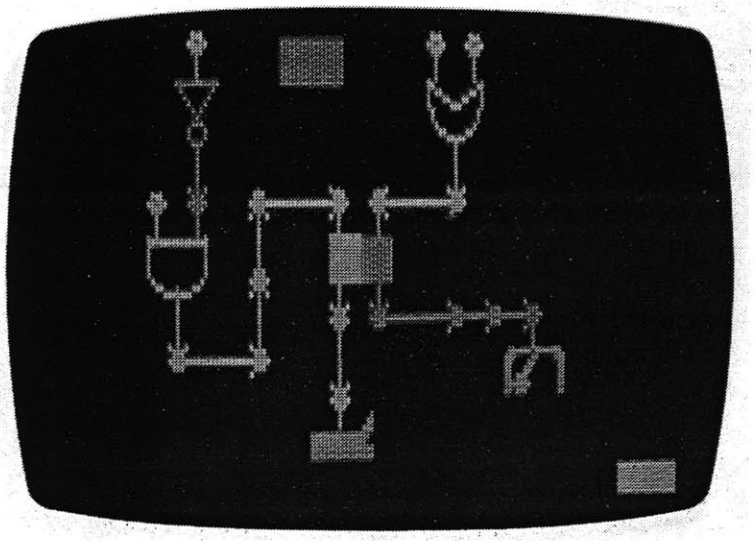


Figure 3-4: The logic gates of Rocky's Boots show the presence or absence of current by the color of the affected gates

Rocky's Boots [Piestrup 84] is an educational program which allows children to experiment with various systems. Piestrup refers to it as in instance of a 'Builder' program.

> "A builder is a program with real-time, animated graphics, with which a user can put parts together to make something new. Nothing in text could simulate a builder program, with its functional graphics. Its purpose is to learn by doing in an exploratory environment." [Piestrup 84]

Rocky's Boots is organized as a series of rooms. The most well-known of these allows children (or adults!) to build up a logic circuit and animates the effects of the changes the user makes. The builder program is simulating a reality of a kind with its rules, but in this case the idea is simulation for the sake of understanding, rather than producing the appearance of reality, as in image synthesis. The overriding concern of a builder program is to provide a reassuring world on the screen that can be manipulated by the user in a consistent manner. Rocky's Boots is also very significant in that it is not a research system. Rocky's Boots runs on popular, cheap home computers and was developed as a product to be sold.

3.3.3 Smalltalk

Smalltalk was implemented in conjunction with the Alto personal computer at the Learning Research Group of Xerox's Palo Alto Research Center. The Smalltalk language and Alto computer were responsible for publicizing the concept of using a bit-mapped monochrome screen and a mouse to provide the user interface for a computer. The Learning Research Group was led by and inspired by the ideas of Alan Kay, whose concept of the Dynabook described the computer they sought to approximate [Kay 77]. He wanted a personal assistant, a machine that could go everywhere with its owner, suggest solutions, suit itself to the foibles of its user, and provide the services that people associated with computers. "...the computer medium is active; it can respond to queries and can even engage the user in a two-way conversation." [Kay 77] The conversational abilities of the computer may still be distant, but some of the advances made in trying to evoke this vision have affected computing tremendously.

In the Alto/Smalltalk system, an effort was made to use representations of objects that the user had encountered in life as the objects he manipulated in the computer. For instance, the screen looked like pieces of paper, and to rearrange them the user picked them up and moved them (with the mouse) rather than invoke some special command for motion.

Equally innovative was the Smalltalk language itself, which continues to generate considerable interest in the computing community as a new topic, more than ten years after its first implementation. In Smalltalk, there are not program and subroutines, as in conventional computer languages. Rather, the programmer in Smalltalk defined objects. These objects receive and send messages between each other. A bank account, to use the classic example from [Goldberg 83], would receive a message that a withdrawl of $200 had been made, and would thereafter remember its new, lower balance. This message-passing-objects scheme is particularly appropriate for networked programs, as programs on separate processors must necessarily pass a limited amount of information between them.

Besides adopting the mouse and screen paradigm of the Alto, Moviecliptool tries to provide metaphors for its objects and manipulations, much as the pieces of paper and desktop of the Alto used. The idea of selecting source images from a strip of film was developed to try and provide as many cues to the user as possible regarding the nature of his objects; in this case, that they are a series of time-sequential images.

3.3.4 Logo

Seymour Papert has been working with children and computers, and has sought new ways to make computers easier for children, including pre-literate ones, to effectively use. As Papert repeatedley points out in [Papert 80], although we regard learning computer programming as a difficult, analytical task which can be performed only by trained people after a difficult learning period, our perception of this may well be based more on how we learn these tasks than the taks themselves.

> "Two fundamental ideas run through this book. The first is that it is possible to design computers so that learning to communicate with them can be a natural process, more like learning French by living in France than like trying to learn it through the unnatural process of American foreign-language instruction in classrooms."

Papert tried to approximate this natural process by portraying the computer as a being in the child's world which 'talks mathematics’ in the way that a child in France is surrounded by beings who speak French.

> "... The idea of ’talking mathematics’ to a computer can be generalized to a view of learning mathematics in ’Mathland’; that is to say, in a context which is to learning mathematics what living in France is to learning French." [Papert 80]

In Moviecliptool it is believed that many kinds of useful image processing operations are neither conceptually complicated nor hard to visualize, given a good place in which to learn them. Moviecliptool seeks to ’speak images’ to everyone in the same way that Logo tries to ’speak mathematics’ to the child.

Chapter Four

The computer environment of Moviecliptool

In this chapter, the computer environment supporting the Moviecliptool system is described. The Sun workstation network in operation at the Architecture Machine Group is unique in its approach to distributed interactive and real-time devices, distributed true color framebuffers, and common picture systems.

4.1 Screen and Mouse

> Within the history of collage and multi-screen video and film, these window frames are unique. They are user directed viewports into ever-receding depths or ever-expanding horizons of information. [Richards 83]

The Sun computers on which the Moviecliptool system is implemented are one of the current generation of personal workstations on the market. These systems are characterized not only by 32-bit processors and a reliance on local area networks, but by their use of monochrome, high-resolution screens and mice as input devices. This combination is certainly a strong trend in current computing devices.

With these resources available, Moviecliptool was written to try

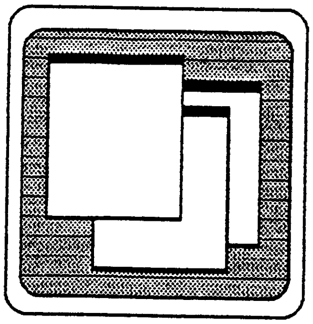


Figure 4-1:
The Sun monochrome screen
allows overlapping windows
on a gray background

and take full advantage of them. The Sun monochrome screen runs a window system, which enables different Unix processes to be running inside screen frames that are manipulated much like overlapping sheets of paper. A typical screen display with several generic windows might appear like Figure 4-1. These windows can each be running independent programs, which may or may not be interacting behind the scenes. The Moviecliptool system opens up one or more windows of its own on the Sun monochrome screen, as well as using the color display screen associated with each of the Sun workstations at the Architecture Machine(see Section 4.2, below). Whenever the cursor is inside this screen, Moviecliptool interacts with the user. If the user wishes to do something else while running Moviecliptool, however, he may bring some other window to the top and work in it, while Moviecliptool continues running.

The Sun window system has a well-defined convention for use of the mouse (Figure 4-2). The Sun mouse is a three-button optical mouse, meaning that it lives on a pad specially designed to enable it to track the mouse motion. The motion of the

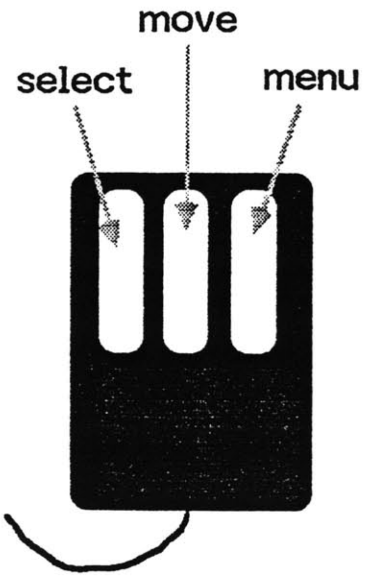


Figure 4-2: The button convention for the mouse

mouse on the pad produces motion of a cursor on the monochrome screen. The three buttons of the mouse are used for defined purposes in the Sun window system itself. The left button is used as the selection button. When some item on the screen is to be indicated, the left button is generally used for this purpose. The middle button is used to pick up some screen entity and move it to a new position, where it is dropped by letting up the button. The right button is used to cause a list of available commands, referred to as a pop-up menu, to appear wherever the cursor is currently pointing. The meaning of all buttons is dependent on the location of the cursor; the window that the cursor is over, and what part of the window it is over, determine what program interprets the button pushes.

The Sun mouse is currently the only input device available, though work is underway to provide a driver for an Elographics touch-sensitive screen.

4.2 Video and Screens

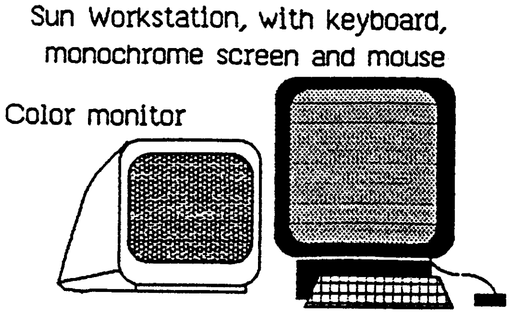
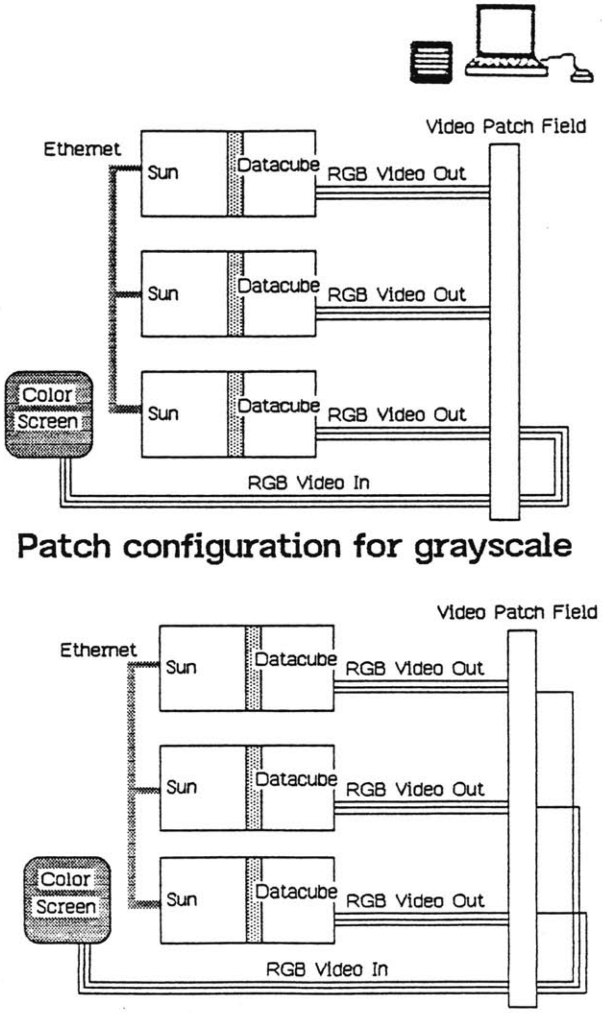


Figure 4-3: An Architecture Machine Group Sun configuration

Each of the Suns at the Architecture Machine Group is equippped with both the Sun-supplied monochrome screen, and a color monitor. Inside our Suns we have installed Datacube color framebuffers. Each Datacube is capable of holding a 768x484, 8 bits per pixel picture. The Datacubes and the monitors are all cabled through a central video patch panel, along with all the other video equipment in the laboratory. This makes it easy to set up any of the necessary patches for a particular video configuration.

There are several ways in which Moviecliptool uses the video environment. When Moviecliptool is being run as a black-and-white system, the color output of the Datacube in the workstation where the user is sitting is brought to the color monitor next to that computer. When Moviecliptool is being



Patch configuration for grayscale

Patch configuration for true color

Figure 4-4: Patching for Moviecliptool

run in full color mode, however, a slightly more complicated video patch is set up. The red output of one Sun's framebuffer, the green output of another, and the blue output of a third are connected to the one monitor at the user's station. This enables that monitor to display a full 24-bit image, even though there is no single 24-bit framebuffer available.

4.3 The Networked Computer Environment

The Suns are connected together over a high-speed local area network in addition to the video network discussed above. This network provides the disk storage to all of the workstations from a single disk-equipped node, as well as enabling the use of remote machines in running any single program.

Software on our Suns ([Carson 84a], [Carson 84b], [Hourvitz 84a]) enables device interface routines to operate across the network in a transparent manner. The program simply calls interface routines for a device, and these calls translated into network transmissions that look for a machine with the appropriate device connected. That machine then accepts the device connection, and forwards commands between the user's program on its machine and the device being controlled. The videotape deck controller, the time code reader, and the video switcher used in Moviecliptool are all interfaced in this manner. Even when the device being used is actually connected to the machine where the request originates, the network connection is used to provide the device interface, so that the same programs can be transported to any machine on the network. Figure 4-5 shows the hardware/software configuration when the snapshot program, discribed in Section 5.2, is running.

An important addition to the devices supported in this manner

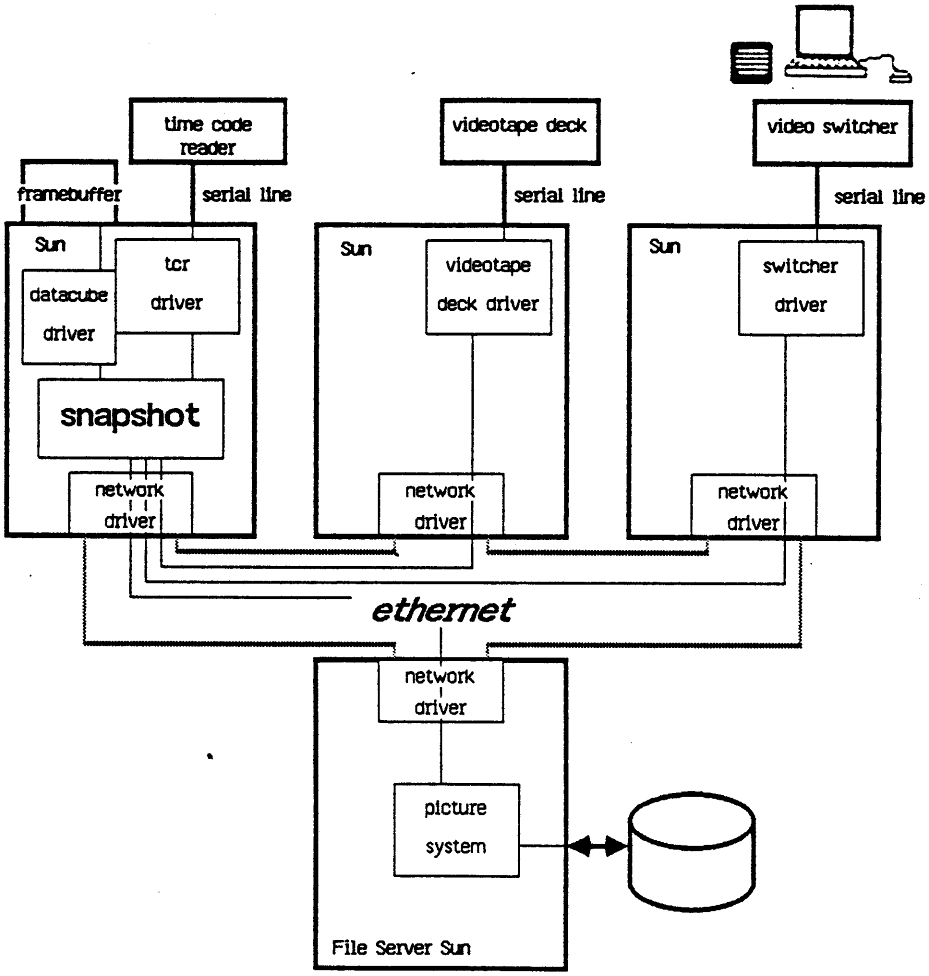


Figure 4-5: The software/hardware configuration when the snapshot program is running

is the color framebuffers themselves [Hourvitz 84b]. Besides the serial interface devices mentioned above, the network is used to allow processes on a remote machine to access the framebuffer on any node. Although transfers of large amounts of data across the network can result in appreciable delays, the general performance of framebuffers in this mode is quite acceptable. A common operation well-suited to this environment is loading a 24 bit picture into the three framebuffers. Since the pictures are all stored on the file server node, the process of loading a 24 bit picture consists of sending a message to each machine with a framebuffer to fetch from the file server and load the appropriate color pass of the image. In this case, practically no data passes between the user and the framebuffer servers, and any user suddenly has available a true color framebuffer.

In addition to using the remote framebuffers, Moviecliptool also distributes its computation across several machines. When run to process 24 bit images, Moviecliptool will only run the parts of the computation relevant to the user interface, and one of the three color passes, on the user's workstation. It will start remote processes[4]which execute in parallel to the main process, but on other workstations, to compute the other two color passes of the images. Moviecliptool actually builds its data structures over time on all three nodes, so that only the 'load the result' command is transmitted when the user actually instructs Moviecliptool to compute an image.

[4] This remote execution, as well as the device driver and framebuffer connections, are supported by the mechanisms in [Carson 84c]

Chapter Five

Movieclipttool
A graphical image manipulation system

> "**still:** A single photograph taken with an ordinary camera, as opposed to a series of pictures taken with a motion picture camera. More specifically, a still is a glossy photograph blown up from a frame of a motion picture (action still) or taken by a still photographer on the motion picture set for the purpose of promotion or publicity." [Katz 79]

Moviecliptool's purpose is to allow people who have made a videotape of some scene to make a still from that videotape, hopefully making from the whole of the tape a better still than simply taking one of the frames from the tape. Unlike **The Film Encyclopedia**, from which the above definition was taken, Moviecliptool does not concern itself with the reason for production of the still. It is built with the knowledge that there are many who have the video equipment, and that there are many who produce still photographs, and it seeks to help them come together. The steps it goes through to do this are outlined in detail below.

5.1 Image Capture

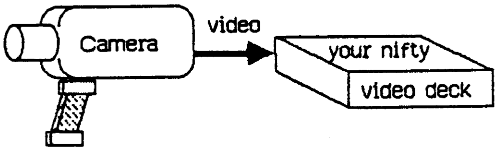


Figure 5-1 *Part I: In the field*

Moviecliptool takes as input a video sequence from any standard video source. Thus, the user begins the process of taking a still with Moviecliptool by making a movie of the scene with a video camera. In particular, a home video camera is the intended input to Moviecliptool. Our test sequences were done with a consumer JVC camera, and recorded on a $400 consumer video deck. Although these components were new and represent 1984 state-of-the-art in comsumer video, they are the type of equipment that is now in people's homes across the U.S. and the rest of the western world.

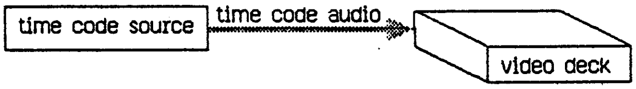


Figure 5-2 *Part II: First post-processing step*

In the Moviecliptool system, after sequences were shot they were transferred to 3/4" videotape and SMPTE time code was dubbed into the audio track. The transfer to 3/4" enabled a computer-controllable videotape deck to be used for the subsequent digitization. This transfer could be avoided if a controllable, portable deck were available; tests showed that digitizing directly from our Hitachi 1/2" deck was entirely feasible.

5.2 Image Digitization

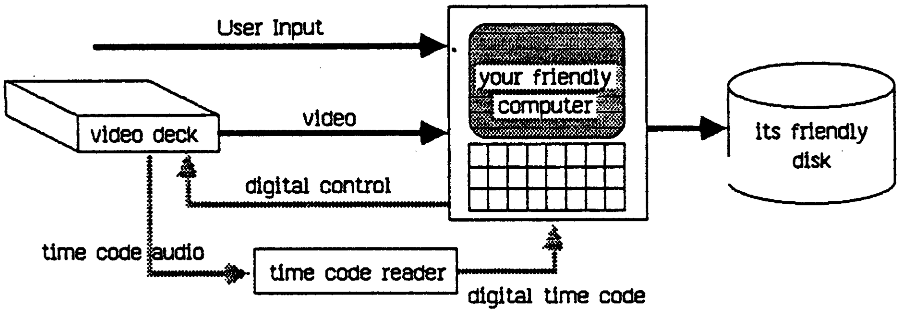


Figure 5-3

Part III: Second post-processing step

The transition from the video domain to the computer domain is accomplished by running a program called **snapshot**. To run **snapshot**, the 3/4" dub is in the computer-controlled videotape deck and the user sits where both a monitor showing the JVC videotape deck [JVC 81] and his workstation's display screen are visible. **Snapshot** plays the tape, and waits for the user to select the frame that is near the desired still. When the user has selcted this frame, **snapshot** rewinds the tape, grabs a rough black-and-white still of the selected frame, and asks for confirmation. If this was (near) the intended frame, **snapshot** will digitize a sequence around that frame without further user attention.

The size of the sequence can be varied. Small segments were typically used for test sequences, due to space limitations.

[5] **Snapshot** operates by decoding the videotape's NTSC signal into R, G, and B signals, selecting these one at a time through a video switcher, and grabbing them with the frame grabber built into the framebuffers. **Snapshot** grabs all the red passes from the sequence first, then the green and blue passes. Currently, because there is no provision in our vidoetape controller for controlling the shuttle[6]feature of the deck, the tape is rewound to its beginning before grabbing each color of each frame. The tape is then played at normal speed, and the computer watches for the SMPTE time code of the desired frame to come up on the time code reader. When the desired time code comes up on the reader, the computer grabs the current input frame with the datacube framegrabber. The grabbed frame is then stored under a name reflecting the sequence name, the frame number, and the color.

A background task called **mclp – process** then resamples and

[5] The problem of storing the digitized pictures is a central issue of Moviecliptool. A single frame from videotape is digitized at 768x512 resolution, and 8 bits are grabbed of red, green, and blue. This adds up to approximately 1 Megabyte per frame. With current storage technologies, this is manageable on the workstation-sized machines used to implement Moviecliptool. In the experimental field, see [Kihara 83], 50 television-resolution images have been packed onto a 3.5 inch disc. Another generation of storage technologies must pass, though, before it will be practical to discuss personal computers with enough secondary storage to grab a sequence of frames.

[6] In a shuttle operations, also called visual search, the videotape deck is put into a pause mode, where a single frame appears on the screen continuously. Then the tape is advanced or rewound at a speed independent of the real-time speed of tape. Typical shuttle controls allow for ranges up to 20x normal speed, and as slow as still frame

halftones the images in preparation for the Moviecliptool program. The images are stored on the disk attached to the network disk server at the Architecture Machine. The pictures are accessible to any of the machines on the net, and are stored in the standard picture system.

5.3 Image Manipulation

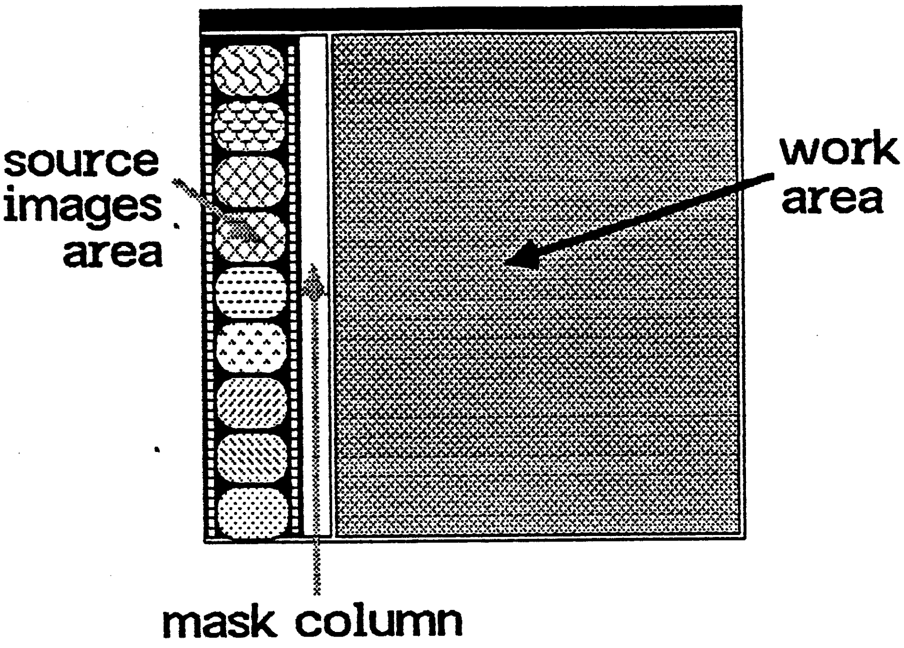


Figure 5-4: A view of the Moviecliptool window

The Moviecliptool program itself is a window-oriented graphical information system for dealing with the images captured by the above process. Moviecliptool uses two displays, the primary one of which is on the Sun monochrome screen. This is a window in the Sun's window system, and it appears as in Figure 5-4.

5.3.1 The Source Images Area

The image sequence digitized by **snapshot** is shown in the source images area. The current source image is always highlighted. It can be changed by selecting a different one with the selection button on the mouse. When the cursor is in the source images area, a set of commands for affecting the source images is available via the menu button.

The *last strip* and *next strip* commands will scroll the viewable strip of source images backwards or forwards, allowing a longer series of images than will fit in the screen to be viewed.

The *load image* command loads the current source image into the display screen. The *paint mask* command allows a mask to be painted over this image (see the **key** operator on page 63).

5.3.2 The Work Area

The work area, the large grey area on the right, is where still frames are created from the source frames. A new frame is created by picking the *create image* command from the work area menu (again, available by the menu button). This creates a new image, which appears as in Figure 5-5. The bitmap representation of the resulting image will eventually be filled into the middle area of the image icon, although it initially appears black. The operation indicator along the left side of the image is used to show what type of operation will be used to create the image. It's a question mark initially, indicating

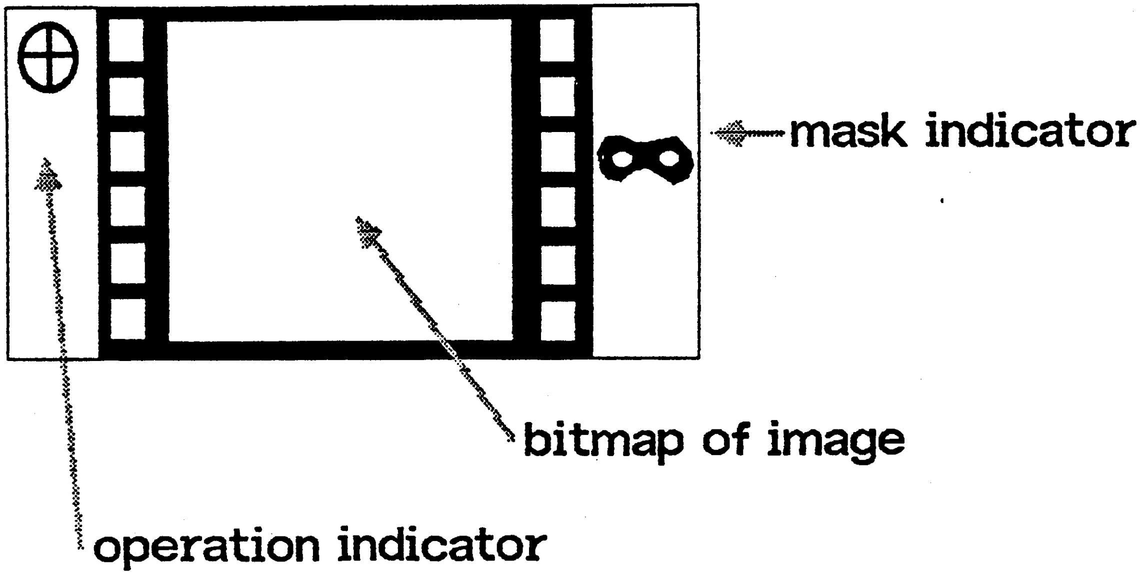


Figure 5-5: An image as it appears in the work area

that this hasn't been set. If a mask is painted over this image at a later time, a mask indicator will appear at the right (see page 63).

One of the images in the work area is the current image, which is generally the last image operated on (for instance, a newly created image is made the current image). The commands on the work area menu manipulate the current image.

The *set op type* command is usually used immediately after an image has been created. The image represented will be defined by applying some operation to some of the source images. The *set op type* command tells Moviecliptool what operation to apply, such as the average operator, which produces its image by averaging together all of the input images. The list of operators in the next section details what each of the available operators in Moviecliptool does to produce its image from its inputs.

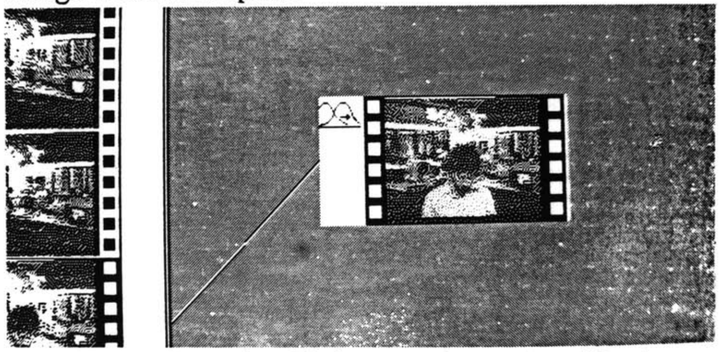


Figure 5-6: A link from the third source image to a filter image

When the operation type has been set for an image, its input images can be specified. A Moviecliptool user specifies input images for an image by moving to an image in the source area, depressing the select button, and dragging out a line. When the select button is let up over an image in the work area, the source image is an input to the work image, and a link is drawn on the screen to reflect this.

Not only source images, but other work area images can be used as inputs to an image. This allows trees of operations to be built up in the work area. Because this can result in a confusing work area, the move button of the mouse allows the user to rearrange the work area at will.

5.3.3 Viewing Images

Once an image has been created, its operation specified, and its inputs linked to it, it can be produced. The usual first step is to take a quick look at it with the *preview image* command. This

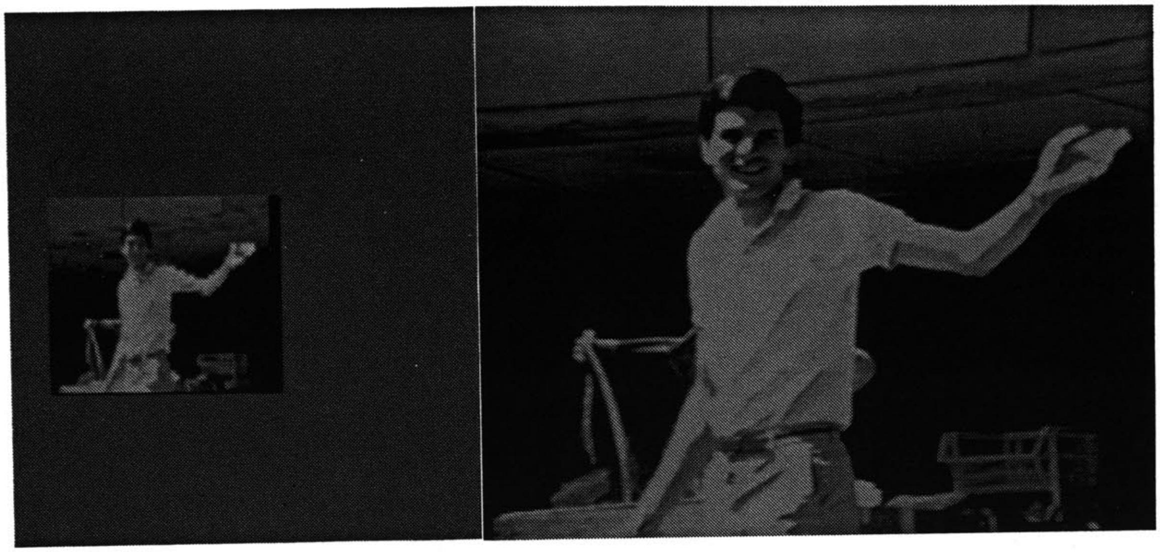


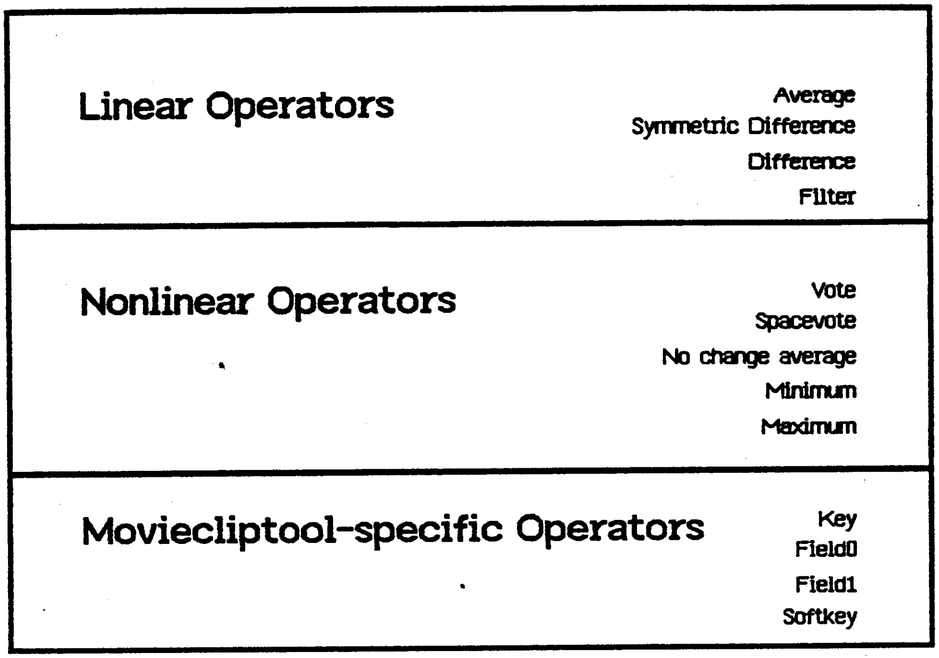
Figure 5-7: A preview of an image, on the left, and a finished image

loads the bitmap of the image onto the monochrome screen and a low-resolution version of the image onto the display screen. Although this image always looks bad, it is useful in finding out whether the operation worked at all. Previewing is very fast; it takes only five seconds or so for a simple two-frame average.

If the preview looked good, the image itself may be put on the display screen by invoking the *load image* command. If the result of this looks good, the still has been made, and the *quit* command can be put to good use, leaving the final image on the display screen for recording. For images that don't turn out so well, Moviecliptool provides the *delete image* command, which removes the current image from further consideration.

5.4 Moviecliptool Operators

The operators available in Moviecliptool are discussed below. Although in the future it is hoped to have more inituitive operators, many of the current operators are ones that would be familiar to image processing workers.



Linear Operators	Average Symmetric Difference Difference Filter
Nonlinear Operators	Vote Spacevote No change average Minimum Maximum
Moviecliptool-specific Operators	Key Field0 Field1 Softkey

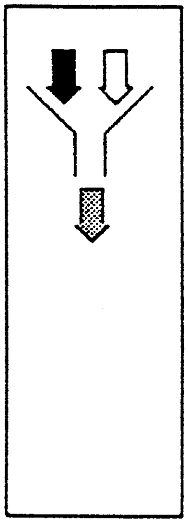
Figure 5-8: The three types of operators

The first two groups of operators are both derived from image processing concepts. The linear operators combine images in manners suggested by image processing, and include such operators as the average operator and a generalized three-dimensional filter.

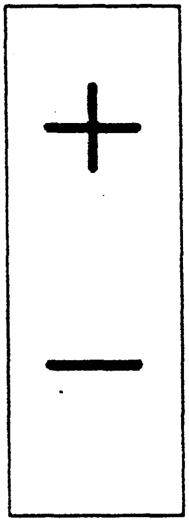
The nonlinear operators produce less predictable effects by combining their images in a manner dependent upon those images. The No change average operator, for instance, averages its inputs, except where they show significant change.

The third group of operators are those that are entirely unique to the Moviecliptool environment. The field operators are necessitated by the video nature of Moviecliptool's input, while the key operators use the interactivity of Moviecliptool to provide personalized images.

5.4.1 Linear Operators

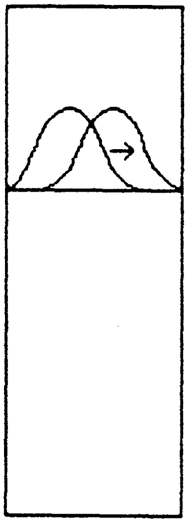


Average: The average operator can accept any number of input images. At each pixel in the image, it takes the average of all the input images, constituting a basic low-pass filter. Averaging also has a very basic use in cleaning up a type of noise common in video images. Video images are recorded on consumer videotape in NTSC video format. This means that the color information is carried on a subcarrier frequency in the video signal. Although this should be entirely transparent in the RGB video image that comes out of the decoder, often some traces of the subcarrier frequency can be seen in the decoded images. Color subcarrier has a property that can help eliminate this, however, which is that it is of the opposite in magnitude for any two consecutive frames. This means that averaging any even number of frames should help the image by cancelling out any residual subcarrier. Unfortunately, this ideal property is only true when the image in the two frames is identical; when there is no motion from frame to frame. It still results in some improvement in image quality.



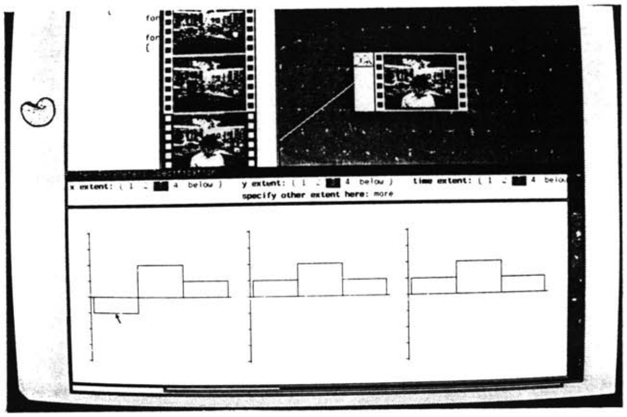
Symmetric Difference: The symdif operator computes the absolute value of the difference between two pictures. This is a standard image processing technique, and although no outstanding image improvements are known that always use it, it is useful in examining a set of frames and finding what is going on in them. One observation on symdif is that the result of taking the difference picture of two RGB images can be very counterintuitive!

Difference: The Difference operator functions like the symmmetric difference operator, except that it subtracts one image from another, clipping to zero.



Filter: The filter operator provides a generalized three-dimensional filter operation. In order to specify this filter, another window appears on the screen when the user selects filter (See Figure 5-9).

In this window, three one-dimensional profiles are shown. These are the cross-sections of the filter kernal for the x, y, and t dimensions. The user can change the size of the filter in each of these dimensions by selecting a choice from the list in the top strip of the Filter window, and can change the values of each of the elements of the kernal by clicking in the filter display subwindow. The value of each element can range from -4 to +4. By setting these values, a user can provide a range of high- or low- pass filters in any of the three dimensions. The



x extent: (1 . ■ 4 below) y extent: (1 . ■ 4 below) time extent: (1 . ■ 4 belo

specify other extent here: more

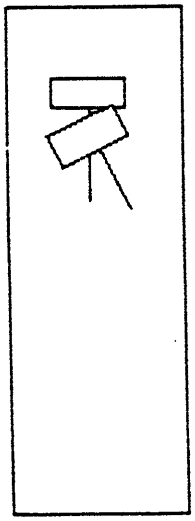
Figure 5-9: The filter specification window in action

three profiles of the kernal are combined when the filter operator is actually run, and the resulting filter kernal automatically normalized; the user need not worry about keeping his filter normalized when specifying it.

A real disadvantage of the filter operator is that it is exceedingly slow. A 3x3x5 filter kernal requires a hundred or more operations to compute a pixel, as well as requiring reference to a large array of data, and can take 15 minutes to run. However, the near-complete range of filters that can be specified under this operator make it easily the most powerful of the available image tools.

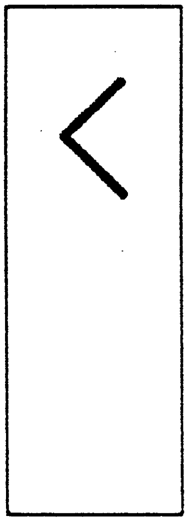
5.4.2 Nonlinear operators

No change average: This operator looks at the intensities for a given pixel in all of its inputs and, if they are close together, averages them. Where the intensities differ significantly, it simply outputs the value from the middle input frame. This operator is designed to get around one of the problems of the average operator. Although averaging tends to remove noise from an image, it also tends to blur the actual picture in the process. The No change average operator operates on the assumption that the items of interest in an image are moving, and thus change significantly from frame to frame, while the background objects stay relatively stable. This technique was suggested to the author by reading [Amato 84].



Vote: The vote operator is used for dealing with problems such as a dropout in one frame of an image. In this case, one frame has a glitch in it that is totally different than the frames around it. Although this could be de-emphasized by a simple average or low-pass filter operation, the vote operator will get rid of it entirely.

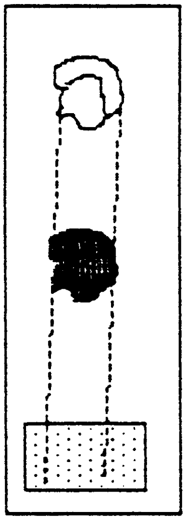
The vote operator, as with most operators, looks at a particular pixel and gets the intensity of that pixel in each of its input images. It then finds the average and standard deviation of these pixel intensities. Then, it goes back and finds the average of all the pixels that are within one standard deviation of the mean. This can also be described as averaging all the pixels that are near the mean. It will totally reject dropouts such as those described above, but it has the problem of clipping off the edges of moving objects in a scene. Like several other operators, the best solution is to use it in conjunction with a key, applying it only where necessary.



Space Vote: The space vote operator was implemented as an alternate sort of high-pass filter. It takes a spatial area (currently 4x4) and applies the vote algorithm to the pixels in that area to produce the output pixel. If one thinks of this as sliding over a smooth edge, it should tend to sharpen the edge. In practice, it has produced indifferent results.

Min and Max: These operators simply take the minimum or maximum values of their input images at each pixel.

5.4.3 Moviecliptool-specific Operators



Key: The key operator is the compositing operation; it takes parts of two different source images and combines into the output image. The key operator works in conjunction with the *paint mask* command.

When the user selects the *paint mask* command in either the source images or work area, the current image in that area is loaded into the display screen. Then the user's mouse becomes a cursor on the display screen. The painting shows up as a red overlay on the image, as shown in Figure 5-10.

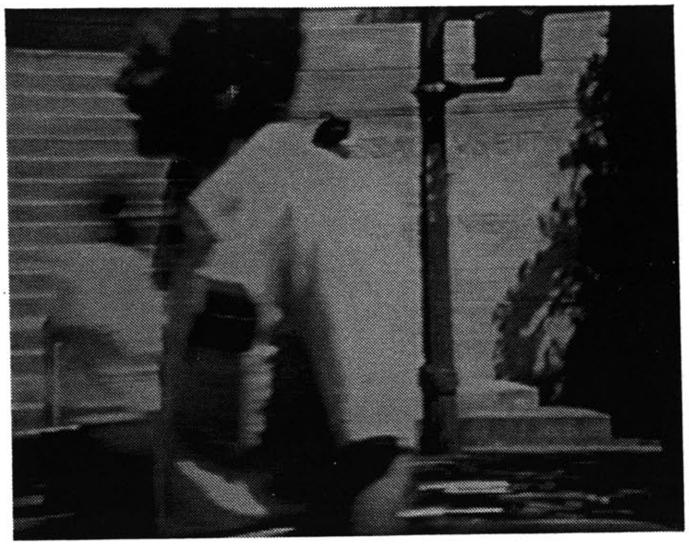


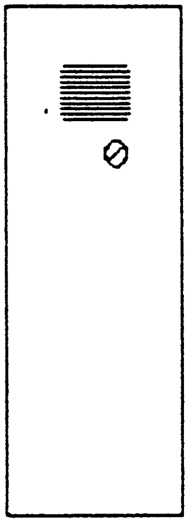
Figure 5-10: Painting a mask over an image on the display screen

When painting, the user has three tools available. The normal tool is the paintbrush, which simply lays down a small area of mask wherever the left (select) button of the mouse is depressed. The user can switch to either of the other modes by selecting them from a small menu that appears on the monochrome screen when the menu button of the mouse is depressed. When using the eraser, the mask under the cursor is erased. When using the fill tool, a seed fill is done from wherever the select button is depressed. A typical use of the mask is to indicate some area of the image to be removed. So, the user paints around the edge with the painter, cleans up this edge with the eraser, and fills the area with fill.

The other two items on the menu available when painting indicate that the mask being painted should be saved with this image, or the mask painting should be aborted. After a mask has been painted to go with an image, the mask icon appears. This icon is always just to the right of the image over which it was painted, which puts it in the mask column for source images.

The key operator, then, takes three inputs. The middle input is the picture whose mask should be used to divide the image area into foreground and background. Wherever the mask linked to the mask input is, the resulting image will be foreground. Everywhere else it will have background. The two other inputs are the images to use as the foreground and background.

Using the Key operator in conjunction with other operators provides the ability to produce a large variety of images from the source tape.



Soft Key: Because the key operator tends to leave artifacts at the edges of the mask, the soft key operator tries to perform the same function as key, but also smooths the transitions between the two images. Currently, the soft key operator performs an unsophisticated blend, but it could be improved by the use of published techniques for careful soft composition, such as in [Burt 83].

Field0: A problem commonly encountered with video images is that a video signal does not acutally scan the lines of an images sequentially, but interlaces two fields, each of which contains every other line in the image (see [Conrac 80] for more information on interlace). Because of this, moving items in a scene will be in different positions in each field, and will have a very disconcerting jittering appearance when the frame is viewed.

In order to enable the user to deal with the two fields of an image separately, Moveicliptool provides the *Field0* and *field1* operators. These operators take one frame as their input, and produce an image consisting of the lines in that field, each repeated twice. Although this discards half of the information in the original image, it produces a much more cogent image of moving objects. It also improves the entire image when the camera itself was in motion. In practice, the *field* operators are used on almost every sequence, because of their usefulness in stopping action.

5.5 Production of final image

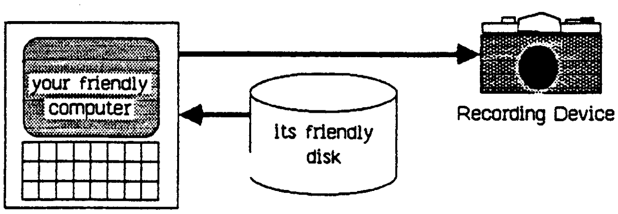


Figure 5-11: The last step in moving from video to photography

As a result of using Moviecliptool, a high-quality single image can be produced on the computer. However, the final conversion into a photograph in the traditional sense remains to be achieved. Moviecliptool itself does not directly address this problem, because there are a number of readily available commercial products for creating a color slide or print from a framebuffer image. The Matrix and Dunn cameras which have been used in computer graphics for a number of years typically make excellent reproductions of the video output of a framebuffer on 35mm or larger formats. These devices are still costly, being in the neighborhood of $10,000, however various manufacturers now make low-end cameras for less deep-pocketed users. Videoprint makes a $3000 4x5 camera which has met with some success, and Polaroid just introduced a $1500 35mm recording system. Given this array of available products, Moviecliptool leaves off having put the desired image in the framebuffer. The user is assumed to be able to push the expose button on the camera himself!

5.6 Extensibility

One of the important features of Moveicliptool is the ease with which new operators may be added to the system. The list of operators given above may not agree with what is visible in any pictures of Moviecliptool, because the process of adding and changing operators continues indefintely. Moviecliptool has been structured so as to allow this easy addition of operators. This section discusses what an operator in Moviecliptool is defined as, how someone goes about adding a new operator to the system, and the advantages and shortcomings of this method.

Operator definition

An operator in Moviecliptool is defined as a routine which can accept an array of input images and produce an output image from them. The routine which does this is the operation routine; operators may optionally have another procedure associated with them, the set operator procedure. This procedure is called by Moviecliptool whenever the user sets a work area image to use that operator. In addition to these routines, the writer of an operator must specify an icon for the operator, a string to go in the operations menu, and the number of inputs this operator takes. This last can be either a fixed number (as in the key operator, which takes 3 inputs) a constant specifying that any number of inputs are acceptable (as in the average operator, which can average as many input images as the user cares to give it).

When the user selects an operator for a particular work area image, the set operator procedure is called with pointers to the data structure for the work image being set and the definition of the operator. Thus, the set operator procedure can modify the definition of its operator whenever it is called. The filter set operator procedure uses this to set the required number of inputs to be the number of inputs the user specified for the filter. In effect, the filter set operator procedure redefines the filter operator to be the filter last specified by the user. The pointer to the work image data structure is used to fill in the parameter block pointer in the structure. Each work area image can have an arbitrary parameter block associated with it, which is used to pass information from the set operator procedure to the operation procedure itself. In filter images, this contains the dimension and filter kernal of the filter to be used.

As can be inferred from the fact that the filter set operator procedure opens up its own window on the monochrome screen, there are few limits on what can be done in a set operation procedure. This is due in part to the fact that set operator procedures are in fact run in a different process than the main part of Moviecliptool. This is necessary to allow set operator procedures to have their own windows, as one process cannot have more than a single window under the Sun window system.

The operation procedure itself is invoked whenever the user previews or loads an image which requires that operator to be run. The operation procedure is called with the following parameters.

Number of input images
Horizontal size of image
Vertical size of image
Array of pointers to input images
Array of pointers to the masks associated with inputs
Pointer to the parameter block of the output image
Pointer to area in which to store the output image

As can be seen from the above list, the issue of the resolution of the images is fairly irrelevant to the operation procedure. It never knows whether it is processing a preview or load image; the resolution is dealt with by the Moviecliptool program. Although it should never receive an invalid number of inputs, most operation procedures check this for validity anyway. The input data pointers point to raster scan ordered input images; the masks are also stored in that order, but with 8 pixels of mask packed to a byte. The operation procedure should produce its output in raster scan order into the area specified. Note that because Moviecliptool is running on a virtual memory system, all operations are simply performed in memory. This alleviates the need for complicated rolling buffer schemes necessary in other cases. Since the workstations supporting Moviecliptool are configured with a minimum of 2 megabytes of memory, this use of memory does not seem unreasonable.

These features have made it very straightforward to add new operators to Moviecliptool. The author can typically add simple operators in less than five minutes of editing, followed by a recompilation. The facilities provided, particularly the fact that operators do not have to know about the linking of images or their resolution, has proven to make the operation procedures themselves among the smallest parts of the program. The many needed structures have been carefully organized and documented to make the operation writers' job easy.

There are, however, still several deficiencies in the operations structure. Although all the changes in the source code needed are easily done, it is still necessary for a operation writer to go into the source code of Moviecliptool and add his operation to several arrays. An automatic tool for the building of the array declarations, etc., needed should have been provided. Additionally, the set operator mechanism is not is general as it should be. There are no trivial methods for asking the user for a simple parameter to an operator, such as the threshold of no change for the no change average operator. Getting even this simple piece of information from the user requires the separate process and separate window as is used for specifying an entire filter. In some regards, this lack of simple input procedures is a deificiency of the window system; some systems [Apple Computer 84] provide these kind of window input routines at the system level.

In summary, it is easy to write new operators for Moviecliptool. The prodcedure does require recompiling the Moviecliptool, due to the linking nature of Unix, but this is also automated with the Unix *make* facility [Sun Microsystems 83a], and so any number of people can add facility to Moviecliptool at will.

5.7 Future Work

Moviecliptool is far from complete. In this section, some necessary improvements and directions are discussed, as well as the directions the Moviecliptool points for future systems.

5.7.1 Moviecliptool extensions

Unfortunately, Moviecliptool as presented here requires that its user have a fair understanding of image processing techniques, and the problems of the initial video medium. Particularly if such a system were to be introduced at the consumer level, this need for specialized knowledge must be eliminated. In the Moviecliptool project, the goal was to find generally useful operators, that could be combined into canned routines for use by non-technical people. The project must be regarded as fundamentally incomplete, because these combinations of the techniques already in the system have not been identified. It is not necessarily true that there is, or need be, a universal 'image crispener' operator, but there should be tools which have an effect more inituitive than that of a '3x3x3 low-pass filter'.

The Moviecliptool structure is waiting for these operators to exist, and can easily support them. The set operator procedure for such a meta-operator would function as a macro, to expand itself into a series of the more primitive operators currently in place. An simple example of this type of macro operator that is frequently useful is the combination of a *field* operator with a 1x2x1 *filter* operator. This selects a field and interpolates the missing lines from the lines of that field, and is a significant improvement for most images.

The other direction for Moviecliptool to move in is back towards the camera. The possiblities of having the computer active at the taking of the picture are not sufficiently explored. If the computer were there watching the user take the picture, instead of finding out about the picture only later, what else could it do to make it a better picture? Might not the computer adjust the lighting, the focus, the aperture, to bring out some desired property? This idea remains for a future project.

5.7.2 As a prototype for the integrated video computer

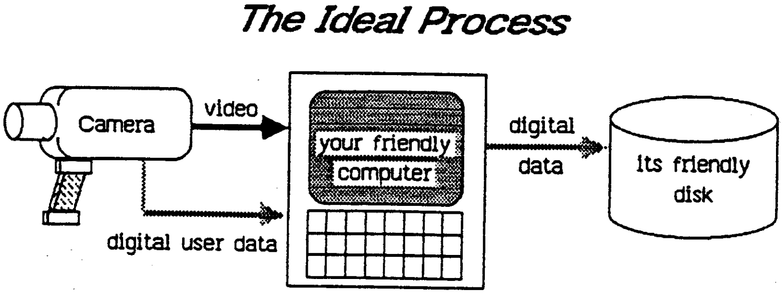


Figure 5-12: The ideal video computer system: directly to the computer

In bringing together equipment such as the videotape deck, the SMPTE time code reader, the frame grabber, and the camera recorder, Moviecliptool may have pointed towards the integrated video system/personal computer of the future. It is already possible in the professional video world to get a family of computerized equipment that interfaces with each other and works together for the purposes of professional video editing (See Figure 5-14). However, the possbilities are even greater than the simple interconnection of discrete equipment. A personal computer with sufficient power, storage, portability, and digitization hardware could someday replace the video deck entirely (Figure 5-12). The image coming from the camera, which could be either a conventional video camera, or a CCD array scanner, or some other optical sensing device, would be processed by the computer and stored directly into permanent storage. The computer display would take the place of monitor, viewfinder, and terminal. This vision is not as distant as it seems; the realities of opticai storage are already bringing the possiblities for portable storage of the required magnitude into view. Some systems for electronic (though analog) recording of still images have already been realized in prototype form [Kihara 83]. It is less clear whether the needed improvements in computers, batteries, and above all software to make the portable video computer a consumer reailty will be available this decade.

Even if the computer is rescricted to remaining home while the user ventures out with his camera and video deck to procure the movies, there is a great deal of improvement that could be realized (Figure 5-13). The provision of a simple time code generation mechanism in video decks and the corresponding reader (either in the deck or in the attached computer) plus a

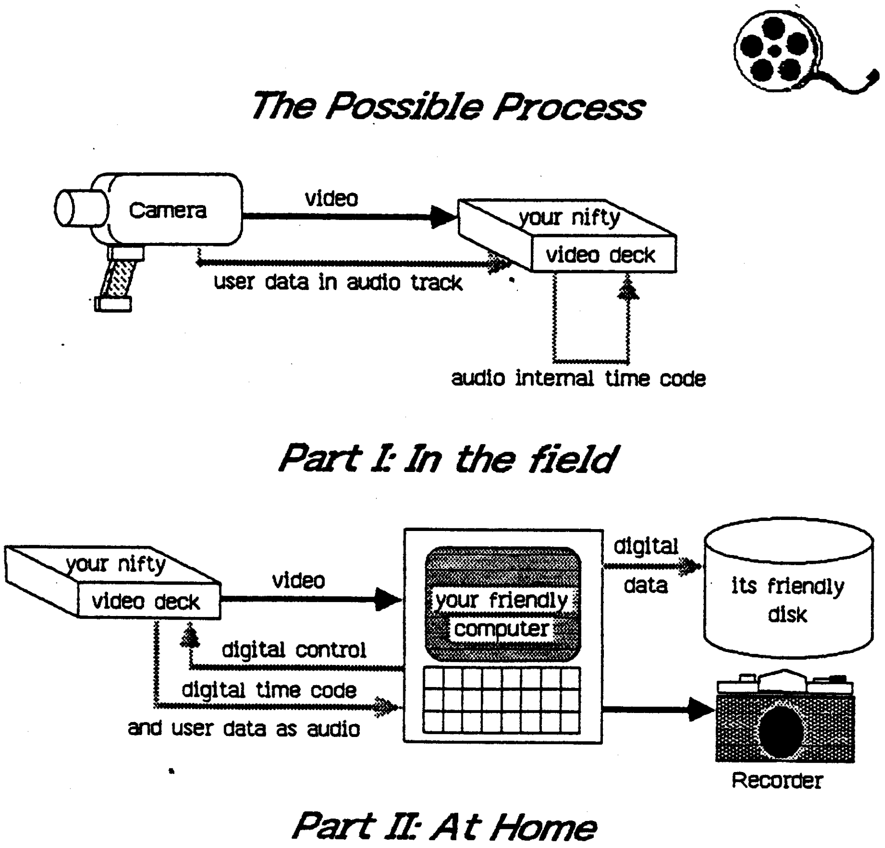
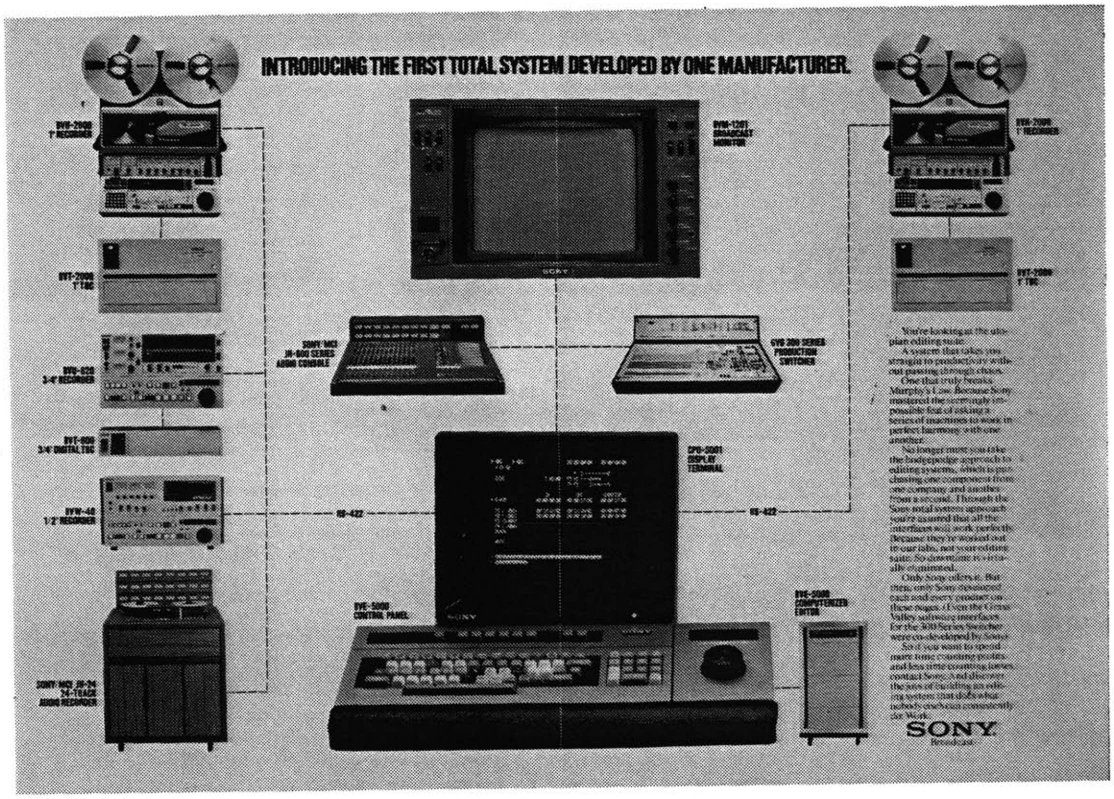


Figure 5-13: An immediately achievable video computer

general control ability for the video equipment would result in not only all the ablities of a Moviecliptool, but other abilities of the ideal system outlined above. Improving a movie as a movie, either by the kind of movie processing dealt with in [Amato 84], being a home editor, reassembling video sequences on the fly, integrating video sequences into other computer-driven material, or moving *from* the photography *to* the video domains are all areas were the integrated video computer idea applies.

The system need not be a single component, or even a single system. All that is necessary is that the ability for the devices to communicate, and the ability for the computer to control the various peripherals, be present. This is well within current technology, and the near future should introduce some of these abilities to us soon.


INTRODUCING THE FIRST TOTAL SYSTEM DEVELOPED BY ONE MANUFACTURER.
BVH-2000 1" RECORDER
BVT-2000 1" TBC
BVU-820 3/4" RECORDER
BVT-800 3/4" DIGITAL TBC
BVW-40 1/2" RECORDER
SONY/MCI JH-24 24-TRACK AUDIO RECORDER
BVM-1201 BROADCAST MONITOR
SONY/MCI JH-600 SERIES AUDIO CONSOLE
GVG 300 SERIES PRODUCTION SWITCHER
CPD-5001 DISPLAY TERMINAL
RS-422
RS-422
BVE-5000 CONTROL PANEL
BVE-5000 COMPUTERIZED EDITOR
BVH-2000 1" RECORDER
BVT-2000 1" TBC
SONY
Broadcast

You're looking at the utopian editing suite.

A system that takes you straight to productivity without passing through chaos.

One that truly breaks Murphy's Law. Because Sony mastered the seemingly impossible feat of asking a series of machines to work in perfect harmony with one another.

No longer must you take the hodgepodge approach to editing systems, which is purchasing one component from one company and another from a second. Through the Sony total system approach you're assured that all the interfaces will work perfectly. Because they're worked out in our labs, not your editing suite. So downtime is virtually eliminated.

Only Sony offers it. But then, only Sony developed each and every product on these pages. (Even the Grass Valley software interfaces for the 300 Series Switcher were co-developed by Sony.)

So if you want to spend more time counting profits and less time counting losses, contact Sony. And discover the joys of building an editing system that does what nobody else has consistently [illegible] Work.

Figure 5-14: An excerpt from a Sony ad. They have the right idea

Acknowledgements

First and foremost goes thanks to all the members of the Architecture Machine Group. Arcmac has been a place that engaged my attention, sharpened my senses of what research is about, and provided me with a new set of friends for the last two years.

Walter Bender is the person without whom Arcmac could never function. He has consistently set higher standards than I can meet.

Ellen Robinson, Janette Noss and Alice Markunas have done much to make my life of the past two years a better thing. To them, my thanks.

Stephanie Houde gets credit not only for much of the imagery in these pages, but for giving me in a short time a better perspective on other approaches to the world.

Lastly, credit is due to my father. His enthusiasm from afar is a constant I have always been able to depend upon, and have often needed.

[ACM 82] ACM-SIGGRAPH.
Siggraph '82 Art Show.
Program Guide to Art Show at Siggraph '82 Conference. New York, 1982.

[ACM 83] ACM-SIGGRAPH.
Siggraph '83 Exhibition of Computer Art.
Program Guide to Art Show at Siggraph '83 Conference. New York, 1983.

[Amato 84] Mia Amato.
Video as film: Transferring in dramatic film texture to a movie shot in video.
Video Systems :26-33, January, 1984.

[Apple Computer 84]
Inside Macintosh
March 1984 edition, Apple Computer, Inc., 20525 Mariani Ave., Cupertino, California 95014, 1984.

[Baudelaire and Stone 80]
Patrick Baudelaire and Maureen Stone.
Techniques for Interactive Graphics.
Computer Graphics 14(3):314-320, July, 1980.

[Beach 83] Richard J. Beach and Maureen Stone.
Graphical Style -- Towards High Quality Illustrations.
Computer Graphics 17(3):127-136, July, 1983.

[Bolt 79] Dick Bolt.
Spatial Data-Management.
Technical Report, MIT Architecture Machine Group, March, 1979.

[Bolt 80] Richard A. Bolt.
"Put-That-There": Voice and Gesture at the Graphics Interface.
Computer Graphics 14(3):262-270, July, 1980.

[Burt 83] Peter J. Burt and Edward H. Adelson.
A Multiresolution Spline With Application to Image Mosaics.
ACM Transactions on Graphics 2(4):217-236, October, 1983.

[Carson 84a] Ken Carson.
Grass Valley video switcher driver documentation
Architecture Machine Group, MIT, 1984.
Online documentation on the AMG Sun system.

[Carson 84b] Ken Carson.
Skotel Time Code Reader driver documentation
Architecture Machine Group, MIT, 1984.
Online documentation on the AMG Sun system.

[Carson 84c] Ken Carson.
Amgserver network services dispatcher doumentation
Architecture Machine Group, MIT, 1984.
Online documentation on the AMG Sun system.

[Cohen 81] Charles Cohen.
Experimental Camera records color slides on small disk for display on TV set.
Electronics :76, September, 1981.

[Coll 76] D.C. Coll and G.K. Choma.
Image Activity Characteristics in Broadcast Television.
IEEE Transactions on Communications COM-24(10):1201-1206, October, 1976.

[Conrac 80] Conrac Division, Conrac Corporation.
Raster Graphics Handbook.
Conrac Corporation, 600 North Rimsdale Ave., Covina, California 91722, 1980.
An excellent handbook on the video aspects of computer graphics.

[Daniels 84] Bruce Daniels.
The Architecture of the Lisa Personal Computer.
Proceedings of the IEEE 72(3):331-341, March, 1984.

[Datacube, Inc. 84]
Datacube, Inc.
VG/AF-123 Video Acquisition and Display Board
May, 1984 edition, 04 Dearborn Rd., Peabody, MA 01960, 1984.

[Flegal 81]
Bob Flegal.
ToolBox: A Smalltalk Illustration System.
Byte 6(8):369, August, 1981.

[Furlong 83]
Lucinda Furlong.
Siggraph '83 Exhibition of Computer Art.
ACM-SIGGRAPH, New York, 1983, Chapter *Essays:* Artist and Technologists: The Computer as an Imaging Tool.

[Futrelle 74]
Robert P. Futrelle.
GALATEA: Interactive Graphics for the Analysis of Moving Images.
In *Information Processing 1974.* IFIPS, North-Holland Publishing Company, 1974.

[Futrelle 75]
R.P. Futrelle and M.J. Potel.
The System Design for Galatea, An Interactive Real-time Computer Graphics System for Movie and Video Analysis.
Computer and Graphics 1(1):115-121, May, 1975.

[Glinert 84]
Ephraim P. Glinert and Steven L. Tanimoto.
PICT: Experiments in the Design of Interactive, Graphical Programming Environments.
Technical Report FR-35, Department of Computer Science, University of Washington, Seattle, Washington 98195, 1984.

[Goldberg 83]
Adele Goldberg and David Robson.
Smalltalk-80 The Language and its Implementation.
Addison-Wesley, Reading, Massachusetts, 1983.

[Guibas 82]
Leo J. Guibas and Jorge Stolfi.
A Language for Bitmap Manipulations.
ACM Transactions on Graphics 1(3):191-214, July, 1982.

[Hanhardt 77] John G. Hanhardt.
The TV Book.
Workman Publishing, New York, 1977, pages 33-34, Chapter Video Art.

[Hausler 72] G. Hausler.
A Method to Increase the Depth of Focus by Two Step Image Processing.
Optics Communications 6(1):38-42, September, 1972.

[Heckbert 82] Paul S. Heckbert.
Color Image Quantization for Frame Buffer Display.
Computer Graphics 16(3):297-308, July, 1982.

[Herot 80] Herot et. al.
A Prototype Spatial Data Management System.
Computer Graphics 14(3):63-70, July, 1980.

[Hourvitz 84a] Leo Hourvitz.
JVC 8200 controller driver documentation
Architecture Machine Group, MIT, 1984.
Online documentation on the AMG Sun system.

[Hourvitz 84b] Leo Hourvitz et al.
Datacube board driver documentation
Architecture Machine Group, MIT, 1984.
Online documentation on the AMG Sun system.

[Huang 81] T.S. Huang and Y.P. Hsu.
Image Sequence Analysis.
Springer-Verlag, Berlin, 1981, pages 289-309, Chapter 4, Image Sequence Enhancement.

[Ingalls 81] Daniel H. H. Ingalls.
Design Principles Behind Smalltalk.
Byte 6(8):286, August, 1981.

[JVC 81] *JVC Service Manual Model CR-8200U Video Recorder*
Revised edition, 41 Slater Drive, Elmwood Park, New Jersey 07407, 1981.

[Katz 79] Ephraim Katz.
The Film Encyclopedia.
G.P. Putnam's Sons, New York, 1979.

[Kay 77] Alan C. Kay.
Microelectronics and the Personal Computer.
Scientific American 237(3):230-244, September, 1977.

[Kihara 83] Kihara Tamagawa.
Electronic Still Camera.
Journal of Applied Photographic Engineering 9:159-163, September, 1983.

[Minsky 84] Margaret R. Minsky.
Manipulating Simulated Objects with Real-world Gestures Using a Force and Position Sensitive Screen.
Computer Graphics 18(3):195-204, July, 1984.
Describes work done at the Atari Cambridge Lab.

[Myers 83] Brad A. Myers.
INCENSE: A System for Displaying Data Structures.
Computer Graphics 17(3):115-126, July, 1983.

[Negroponte 81] Nicholas Negroponte.
Media Room.
Proceedings of the SID 22(2):109-113, February, 1981.

[Netravali 78] Arun N. Netravali.
Noise Removal from Chrominance Components of a Color Television Signal.
IEEE Transaction on Communications COM-26(8):1318-1321, August, 1978.

[NHK 83] NHK Culture Center/OMNI.
Exhibition of Computer Graphics.
NHK Culture Center/OMNI, 1983.
This is the program guide to the 1983 Japanese Computer Art Show.

[Onosko 84] Tim Onosko.
Japan and Technology: A Nation Looks to the Future.
Creative Computing 10(8):82-90, August, 1984.

[Papert 80] Seymour Papert.
Mindstorms: Children, Computers, and Powerful Ideas.
Basic Books, Inc., New York, 1980.

[Pease 71] R.F.W. Pease and J.O. Limb.
Exchange of Temporal and Spatial Resolution in Television Coding.
Bell System Technical Journal 50:191-200, January, 1971.

[Piestrup 84] Ann Piestrup.
Game Sets and Builders.
Byte 9(6):215-219, June, 1984.

[Potel 76] M.J. Potel and R.E. Sayre.
Interacting with the GALATEA Film Analysis System.
Computer Graphics 10(2):52-59, July, 1976.

[Richards 83] Catherine Richards.
Siggraph '83 Exhibition of Computer Art.
ACM-SIGGRAPH, New York, 1983, Chapter *Essays:*Mapping a Sensibility: Computer Imaging.

[Schmandt 81] Christopher Schmandt.
The Intelligent Ear: A Graphical Interface to Digital Audio.
In *Proceedings, 1981 International Conference on Cybernetics and Society,* pages 393-398. IEEE, 1981.

[Schmandt 84a] Chris Schmandt and Barry Arons.
A Conversational Telephone Messaging System.
In *Digest of Technical Papers.* International Conference on Consumer Electronics, June, 1984.

[Schmandt 84b] Christopher Schmandt and Barry Arons.
Phone Slave: A Graphical Telecommunications Interface.
In *Digest of Technical Papers.* SID International Symposium, July, 1984.

[Skotel 84] Skotel Corporation.
Skotel SMPTE/EBU Time Code Reader Model TCR-80 Instruction Manual
6.0 edition, 1445 Boulevard Provencher, Brossard, Quebec, Canada J4W 1Z3, 1984.

[Sun Microsystems 83a]
Sun Microsystems.
Beginner's Guide to the Sun Workstation
Revision C of 1 November 1983 edition, 2550 Garcia Ave., Mountain View, California 94043, 1983.

[Sun Microsystems 83b]
Sun Microsystems.
Programmer's Reference Manual for the Sun Window System
Revision C of 1 November 1983 edition, 2550 Garcia Ave., Mountain View, California 94043, 1983.

[Sun Microsystems 83c]
Sun Microsystems, Samuel J. Leffler, Robert S. Fabry, and William Joy.
Interprocess Communiations Primer
2550 Garcia Ave., Mountain View, California 94043, 1983.

[Sutherland 63] Ivan Sutherland.
Sketchpad: A Man-Machine Graphical Communication System.
Technical Report TR-296, Massachusetts Institute of Technology Lincoln Lab, September, 1963.

[Tymony 84] Cy Tymony.
Computer Control of a Video Recorder.
Byte 9(7):179-183, July, 1984.

[Upton 81] Barbara Upton and John Upton.
Photography.
Little, Brown and Company, Boston, 1981.

[Xerox Learning Research Group 81]
Xerox Learning Research Group.
The Smalltalk-80 System.
Byte 6(8):36, August, 1981.

[Youngblood 82] Gene Youngblood.
Siggraph '82 Computer Art Show Guide.
ACM-SIGGRAPH, New York, 1982, Chapter *Essays:*Toward Autonomous Reality Communities.