Constraint-Based Cinematic Editing

by
Benjamin Rubin

AB, Semiotics and Computer Science
Brown University, Providence, Rhode Island
1987

Submitted to the Media Arts and Sciences Section in partial fulfillment of the requirements of the degree of Master of Science in Visual Studies at the Massachusetts Institute of Technology

June 1989

Signature of Author: 5/12/89

Benjamin Rubin
Media Arts and Sciences Section
May 12, 1989

Certified by: 5/12/89

Glorianna Davenport
Assistant Professor of Media Technology
MIT Media Laboratory

Approved by:

Stephen A. Benton, Chairman
Department Committee on Graduate Students

Constraint-Based Cinematic Editing

by
Benjamin Rubin

Submitted to the Media Arts and Sciences Section
on May 12, 1989 in partial fulfillment of the requirements
of the degree of Master of Science in Visual Studies
at the Massachusetts Institute of Technology

Abstract

This work presents a computer-based motion-picture editing system capable of the automatic generation of narrative sequences which conform to simple constraints of character emphasis, length, pacing, and chronology. The cinematic content model used to generate the sequences is presented, and practical methods for gathering content data during movie production are proposed. The details of the system's implementation and the practical applications of its underlying principles to motion-picture production and multimedia computer systems are discussed. The written thesis is accompanied by a video tape which shows several edited sequences which were generated by the system and by a video disc containing the source material from which the sequences were formed.

Thesis Supervisor: Glorianna Davenport
Assistant Professor of Media Technology

TABLE OF CONTENTS

Acknowledgements

Many thanks are due to Marvin Minsky and Barry Malkin, who read and criticized the proposal from which this work developed. Craig McKay, Robin Romano and Mike Hawley provided me with countless pages of useful books and papers, and all helped to provide me with an "oral history" of their experiences with electronic editing technology.

Thanks to Alan Lasky, Kevin Landel, Paul Linhardt, and Joseph Vanderway, who formed the core of the film crew for "Rowes Wharf," and to Genevieve Johnson, Tom Hawk, and Nanette Rayman who were talented and tolerant in their starring roles.

Ben Davis provided crucial equipment when it was needed most. Tom Sgouros was subjected to many of these ideas long before they were thought through and patiently hashed them out with me. Sabrina Birner helped me get through it all with her cheerful smile and helping hands. Hal Birkeland and Jeff Johnson gave me supper-human technical support, without which this thesis might have been no more than a *proposal* for an automatic editing system! Thanks also to Stuart Cody and to Vicky Bippart — it could not have been done without you.

Ricky Leacock looked on bemusedly from across the Atlantic, but his eye and his heart continue to inspire and energize everyone who has ever been involved with the MIT Film/Video section; Ricky's vision is with me every time I look through a camera eyepiece. And finally, Glorianna Davenport was my teacher, my advisor, and my friend. She had confidence in me even when I didn't, and that was the biggest encouragement of all.

Dedicated to the memory of Gilles Bloch,
who I never had the chance to meet.
His ideas inspired and encouraged the
work presented here.

Introduction

The work presented here was originally motivated by the following problem: a large, multi-media case study on urban change in New Orleans[1] existed at MIT, and tools were needed so that students, faculty, city planning experts — people who were not media professionals — could both gain access to and manipulate the material for the purposes of study and presentation. The most powerful kind of tool would be one which could allow the user to make a query into the database, and in response would present an entirely new, edited video sequence consisting of clips drawn from throughout the material. Of utmost importance is that this new sequence should be cinematically coherent — rather than being just a group of clips spliced end-to-end, the new sequence should maintain spatial and temporal continuity so as not to jar the viewer away from the content of the material, so that the viewer's state of reverie[2] can be preserved. The hypothetical tool just described performs automatic, constraint-based editing on the material contained in the "New Orleans" case study, the constraints being specified by the database query.

The broad goal of this work is to formulate a content model for cinematic information that would give a computer enough information to determine, with precision and at a high level, what the content of every frame of a movie contains. Such a model would make possible three major kinds of applications:

1. Sophisticated editing tools which would be capable of suggesting possible next cuts and critiquing an edit in progress,

---

[1] The title of the case study, which was produced by MIT professor Glorianna Davenport, is "A City in Transition: New Orleans, 1983-1986." The material which makes up this case study includes three and a half hours of film (on video disc), still photographs, text documents, maps, and a database which links all this information together. See [DAVENPORT 1987] for a complete description of the project.

[2] The earliest use of the term "reverie" in this context should be attributed to theater critic Bill Alfred in a conversation with Glorianna Davenport, Ricky Leacock and Peter Selars in 1986.

2. Moviebanks, along the lines of the "New Orleans" project discussed above, capable of responding to user queries with coherent cinematic sequences illustrating relevant information, and

3. Interactive art and entertainment systems, games and installation pieces which use virtual editing technology and cinematic content knowledge to respond in sophisticated ways to user stimulus.

The more specific focus of the thesis is on the implementation of an automatic editing system, a critical component of each of the three kinds of applications just mentioned. To understand the automatic editing problem, it is useful to first examine some aspects of conventional cinematic editing. Chapter one begins with a brief overview of the function of editing in cinema, then goes on to describe the different technologies and methods currently employed by film and video editors, and the ways that specific editing tools affect the methodology employed by editors. Chapter Two reviews past research into the codification of cinematic information and the automation of the editing process, and Chapter Three contains a detailed description of the prototype of an automatic editing machine, the Constraint-Based Editor. Returning to some of the questions raised in the first chapter, Chapter Four discusses the results obtained using the system, and includes a theoretical discussion of the kind of editing we can expect a machine to be capable of in the future.

Chapter Five addresses the biggest problem with creating cinematic content models which is that an overwhelming amount of data needs to be associated with all source material. This chapter proposes a new methodology for movie making which distributes the burden of data collection throughout pre-production, shooting, and post-production.

Finally, the conclusion reviews the successes and failures of the experiment in automatic editing, and proposes alternative approaches to the problem which might yield even more successful systems in the near future.

Chapter One: Understanding the Editing Process

A CAPITAL DISTINCTION TO MAKE is that between *the story itself* (which is to say, quite simply, "what happens" when one lays out the scenario in

> chronological order) and another level which we can call narration, but which others call: telling, discourse, dramatic construction, etc., and which concerns *the way in which this story is told*. In particular, the way in which the events and the information of the story are brought to the awareness of the audience (ways of telling, information hidden, then revealed, utilization of time, of ellipsis, of emphasis, etc.), this art of narration can, by itself, lend interest to a predictable story. Inversely, a poor narration destroys the interest of a good story, as anyone who has ever tried to tell a joke which falls flat can attest.
>
> Michel Chion, *Ecrire un scénario*

Telling Stories

Without giving the matter much thought, one might assume that the editor of a dramatic feature film can have little influence on the finished product. The story has been determined by the script, the actors have been directed and played their roles for better or worse, and the images have already been composed and committed to film. And yet, given these basic elements as building blocks, *the story of the film is told through editing*.

Should we see the actor's eyes first, or should we see that he has a gun? Should we glimpse the woman taking aim at him from behind, or should she be revealed to us only after a shot has been fired and our hero turns around — too late? We may see something happen before we understand its meaning; we may witness an effect in one shot, its cause in the next, but not until four shots later are we shown enough information to make the connection between them. The withholding and revealing of relationships between characters, objects and surroundings is a sophisticated game of peek-a-boo which forms the basis of cinematic story telling.

All films, regardless of *genre*, are made up of sequences, purposeful successions of shots which have narrative properties. With each successive shot in a sequence, new information is presented, and *the narrative is advanced*. If a cinematic transition does not articulate anything new, does not advance the story, then it will confuse the audience and will seem out of place in the sequence.

Conventions exist in cinema which correspond to nothing in our experience of the physical world; only in movies can we accept that our point of view instantaneously jumps from one point in space and time to another.

We accept these these abrupt discontinuities — in fact, we usually don't even *notice* them — because we understand film as a fundamentally narrative medium, because we accept that a story is being related to us.

A frontal shot of a man sitting at a table and rolling up his sleeves is followed by a second shot of the same action, this time from the man's left side. Both shots clearly show the man's face and upper body; the background is dimly lit, so we see no more of his surroundings in the second shot than we did in the first. If there is no motivation for the transition from front to side other than to "break up the action," the cut between the shots will feel clumsy, and the sequence will stumble. However, if the second shot reveals a significant tattoo on the man's arm which was not visible in the first shot, then the same transition will probably pass unnoticed; now the film is *telling* the audience something, and even an awkward change in camera angle can be forgiven.

The work presented in these pages describes an attempt to create a machine[1] which draws on these principles of cinematic narration to automatically produce edited versions of a film. The films edited so far by the machine have had interesting and unexpected qualities. Although the editing algorithm used cannot possibly encompass the full complexity of the task of telling a story (and does not attempt to), the films it creates sometimes suggest a rich underlying dramatic composition. This fact raises issues about narrative voice and intentionality, and highlights the way meaning can be spontaneously created through the juxtaposition of images.

In the interest of maintaining suspense, these tantalizing issues will be set aside for now in favor of a more technical look at the editing process. The remainder of this chapter will be devoted to an overview of existing editing methods and tools. Later, as we discuss the results obtained from the

---

[1] The conglomeration of computer hardware and software, laser disc players, routing switchers, monitors, amplifiers, speakers, etc. which comprise the technological platform underlying this thesis is referred to collectively as an editing "machine." The term "machine" is used instead of the more generic "system" partly to acknowledge the mechanical nature of video disc-based editing, even when under computer control, and partly to distinguish that which makes up the machine from the higher-level applications programs which utilize the machine as a resource.

constraint-based editing system, we will return to the theoretical questions raised by the notion of machine-generated movies.

Technology and Process

The editor's task is essentially one of organization and reduction; starting with a disorganized mass of source material, the editor must choose and arrange those portions which, together, tell the story of the film. Whatever the editing medium, editors generally begin by viewing and logging the source material they receive, then dissembling the continuous footage into discreet chunks — shots — which are then organized in some way. Next comes the process of selection, wherein the editor chooses shots to form a *sequence*. After the sequence of shots has been assembled, fine adjustments are made at the edit points to achieve a sense of continuity across transitions.

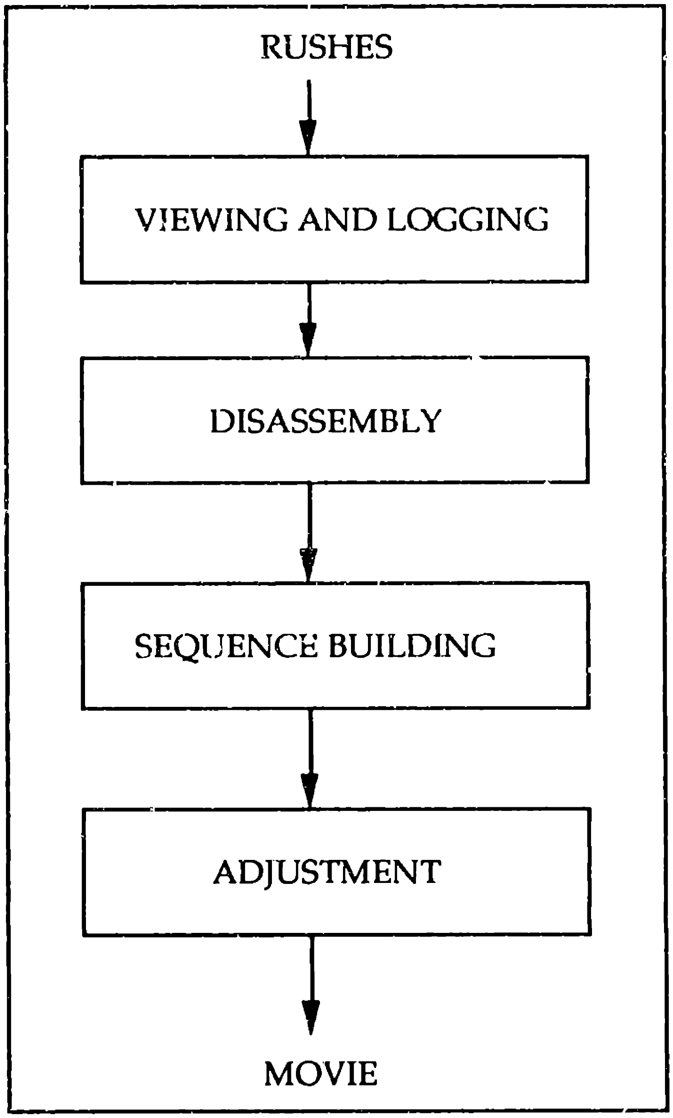


Figure 1: The Editing Process

Different editing technologies encourage different methodologies at each of these stages; if the tools for performing a particular task are uncomfortable or very expensive to use, the editor will be inclined to accomplish that phase as quickly as possible and not linger over decisions. If correcting a mistake results in degradation of the editor's work print (or tape), or requires re-editing or re-copying of work already done to satisfaction, then the editor will be disinclined to correct small mistakes. Next, we'll look at the existing editing technologies with an eye toward how each effects the editing process.

Film Editing

Film is a wonderfully physical medium. You can take a piece of film in your hands and stretch it out to get a sense of it's duration; you can hold it up to the light to see what the images look like. You can also get it accidently caught on the edge of your editing table, break it, scratch it, and get it stuck to your clothes. Film is the noble savage of audio-visual media. Its primitive simplicity is romanticized and ridiculed, yet film as an editing medium has qualities which the most modern computer editing systems (discussed at the then of this chapter) strive to emulate.

Editors will tend to spend time viewing and logging the rushes *before* they are taken apart, since this is easier to do when the film is still one contiguous strip[1]. The disassembly of the dailies (rushes) in film is then accomplished by literally cutting them up (with a knife) into short clips. These clips are then numbered and either hung in bins or rolled onto cores and stacked, one clip per core. Prior to assembling sequences, related clips are often grouped together by scene or subject content and laid out on tables or in bins. A rough cut is then assembled by retrieving clips one after another, trimming each at the head so that it will match the tail of the previous one.

The physical nature of film as a medium is both a blessing and a curse to editors. On the one hand, film makes possible a gradual methodology of editing. An editor can make successive passes over the rough-cut, trimming out awkward moments, tightening the action and fine-tuning the pacing until each sequence "works." The simple fact that splices can be taken apart and spliced back together makes some experimentation possible in the editing process.

But cutting and splicing film is tedious and physically demanding; the desire an editor might have to experiment with different ways of cutting a sequence can be inhibited by the time and effort necessary merely to cut and

---

[1] A viewing machine or projector must be re-threaded each time a different piece of film is to be viewed. While this is not a particularly difficult or time-consuming task, the dread of repeated threading serves as an incentive not to view clips on a whim once they have been disassembled.

splice the experimental version together, then to undo the experiment if it fails. The situation was even worse before the advent of tape splicing, as Ricky Leacock recalls:

> When I was young, you made "spit splices" — you'd spit on the film, scrape the emulsion off and put glue on it — and every time you remade a splice to change something you had to cut in a frame of black leader to compensate for the fact that some film had been lost. And so if you screened your final cut and it was full of blocks of black leader, it looked like hell. So you only made an edit when you were *very* sure that was the edit you wanted to make![1]

The basic technology of film editing — a viewing screen, motorized transport, film feed and take-up spools, and mechanical interlocking of picture with sound — has changed only superficial since the outset of the medium, and no technological changes have made any significant impact on the methodology used by film editors.[2]

Off-line Video Tape Editing[3]

Editing using video technology is physically less taxing than editing film; the film editor's unwieldy blades and sticky tape are replaced by electronic push-buttons, and counting film sprocket holes is obviated by digital frame-counters and machine-readable time code. And yet, few film editors opt to edit on video tape even though there may be cost advantages in doing so. This section will explain some of the problems inherent in tape editing.

The critical difference between film and videotape is that videotape can not be cut or spliced, only copied. The initial stage of disassembly and

---

[1] Ricky Leacock in a conversation with students recorded on May 5, 1989 by Stuart Cody.

[2] More modern, comfortable editing technology does not necessarily lead to better or faster editing, as Ricky Leacock recently observed: "When you had multiple reels on re-winds, working through this awful collection of sound readers and viewing machines, [editing] was extremely unpleasant, and you were aching — ouch! — trying to control the speed [of the film], and so you tended to make your decisions not only very definitely but very quickly. There's a tendency on a Steenbeck to *waft* the film this way and *waft* the film that way and you can spend the whole day and never make a decision!" (Ibid.)

[3] Please refer to the glossary for an explanation of the differences between on-line and off-line video editing.

reorganization of the material must therefore be done largely mentally and on paper rather than by physically moving clips of video around. Since most video productions involve many source tapes, and since winding and re-winding tapes is slow and tedious, an intermediate "pull" is often made prior to assembling sequences. The pull is a collection of promising shots gathered from many source tapes which in turn is used as the source tape for the actual assembly of a sequence.

Next comes the actual editing, in which segments can only be added sequentially onto the end of the piece being cut. If, upon completion of the first cut, the editor wants to shorten, lengthen, add or remove a shot in the the edited piece, s/he can only do so by re-cutting the entire piece from that point, or by re-recording edited portions of the piece onto yet another tape.

If the piece is being edited off-line with the intention of later conforming the original material to the resulting cut, then the problem of tape generation loss inherent in the method just described is not important, since the off-line cut will serve only as a guide for later work and not as a finished product. Yet even in this case, the process of transforming the first rough cut into the final version is awkward in video. Small changes necessitate extensive copying or recreation of work already done. A video project cannot evolve gracefully and continuously, as a film does, toward its final form.

"Non-Linear" Editing Systems

The most recent advance in editing technology has been the development of computer-based non-linear *virtual* editing systems. The concept underlying these systems is that editing is done not by copying or by physically cutting the source material, but through virtual random access. To "perform" an edit, a virtual editing system plays back the first shot through to its end, then instantly plays back the next shot with no audible or visible gap in between. Current systems accomplish this by maintaining multiple copies of all source material on computer-controllable playback machines. The technical issues of virtual editing are discussed in detail Appendix A, which provides a detailed description of the implementation of a virtual editing workstation.

Virtual editing systems maintain a computer list representation of the edit in progress. Instead of actually changing or copying the source material, an editor using a virtual system edits only the computer list. Changes made to the list are reviewed continuously through virtual previews of edits.

The goal computer editing systems was to combine the push-button ease of tape editing with quick access to material and the capacity for non-sequential editing. These systems are often described as word processors for film — allowing editors to cut, paste and rearrange the material they are editing, and removing them from the physical drudgery of cutting and splicing film.

The methodology used in virtual editing resembles the process of editing film more closely than editing video. Even though no changes can be made to source material, *virtual* clips can be defined and organized. One virtual editing system, the Montage, implements a notion of virtual bins in which related virtual clips can be "hung." Virtual editing systems all provide some method of defining clips, then retrieving them later and inserting them into sequences. As in film editing, virtual editing systems allow sequences to evolve gradually from rough cuts; unlike conventional video editing, workstations permit editors to trim or insert material in the middle of a piece without having to re-edit or dub the edited material which follows. Searching for source segments is practically instant, since the machine can access any point in the source material within a few seconds.

The most widely used virtual editing systems — Montage, EditDroid, Ediflex, and CMX 6000 — all use some form of database to store edit lists and to facilitate access to the source material. With the possibly exception of the Ediflex, none of them have any notion of a content model as distinct from the edit list management database. As we shall see, the lack of a sophisticated content model limits the potential of these systems.

Chapter Two: Representing Cinematic Content

Presented below is a survey of work relating to the development of coding systems for cinematic content data, one theoretical, one strictly technical, and one, Gilles Bloch's, which is the first to explicitly address the problem of automatic computer-generated editing.

Noël Burch's Taxonomy of Cinematic Transitions

Noël Burch's well known book on film theory, *Une Praxis du Cinéma* [BURCH 1968], was published years before computer editing technology was even on the horizon, yet his highly structured analysis of cinematic transitions has proved to be directly applicable to the design of automatic editing systems. Although Burch, in the introduction to the 1980 edition of Praxis, is harshly critical of his early book, citing its excessive formalism as evasive of the "real" issues of cinematic content, it is precisely the rigid structure of his approach that makes his work especially useful in the present context.

In the first chapter of his book, Burch outlines what amounts to a taxonomy or classification system for cinematic transitions in which he identifies five general classes of temporal relationships and three classes of spatial relationships which can exist between two shots.[1] These classes take into account concepts of spatial and temporal continuity, overlap, ellipsis, and disjuncture. They also address whether or not the time or space relationship across a cut is known or is measurable. For example, if in one shot shows a man putting on his coat and leaving his office, and then the next shot shows him exiting the building where he works, the time ellipsis between the two shots is known and measurable (it takes someone a minute or two to walk down the hall, take the elevator, etc). This type of temporal transition is in a

---

[1] The five classes of temporal articulations are temporal continuity, measurable ellipsis, indefinite ellipsis, measurable reversal, and indefinite reversal. The measurable ellipsis and reversal refer to cuts in which a well defined action (going up a flight of stairs, for example) is either abbreviated by ellipsis across a cut or is emphasized by a short, measurable time reversal such that part of the action is repeated. Indefinite ellipsis and reversal are the categories into which scene changes and flashbacks fall; when we see such a cut, we understand that the time in second shot is *different* from the time in the first, and we rely on the content of the film to tell us how much time has passed (or reversed) between the two. The three types of spatial articulation are spatial continuity (match cuts, in which some part of the space seen in the first shot is also visible in the second), proximate discontinuity (a cut to another part of the same space not seen in the previous shot), and complete discontinuity (a cut to another space entirely). See [BURCH 1968] for more details.

different class from one in which a shot of a man working cheerfully at a desk dissolves into another shot of the same man at the same desk, but now unshaven and ragged looking. Here, there has clearly been a passage of time, but exactly how many days, months or years have passed between the two is not explicit or knowable from the transition, but is a function of the viewer's understanding of the story at a deeper level.

Spatial transitions fall into similar categories; a cut from a shot of one part of a room to another part of the same room is clearly in a different class from a cut between, for example, a restaurant interior and a soccer field; in the first case, the spatial relationship between the two shots is either explicit (if there is visible overlap from one to the next) or can be inferred, whereas in the second case no spatial relationship is apparent.

Any formal language or notation of cinema, including systems for representing cinematic information on computers, should clearly be capable of expressing notions of time and space in such a way that temporal and spatial relationships between shots can be evaluated. At the very least, a formal cinematic representation should include enough information so that any shot transition could be placed into the temporal and spatial categories proposed by Burch. A notation which meets these requirements could form the basis of a sophisticated computer system for the analysis and generation of cinematic sequences.

The Constraint-Based Editor and its data representation for cinema do not yet achieve these goals. While I have managed to implement a useful subset of the kind of cinematic notation I have just described, the amount of data required to truly represent all the aspects of space and time remains daunting. What is missing from the Constraint-Based Editor in this regard has been omitted only because methods for acquiring enough useful space and time data do not yet exist. Chapter Five, the Proposal for a Data-Rich Production Pipeline, suggests some ways in which such information could be gathered continuously through the production process to make possible a more complete spatio-temporal model.

Script-Based Editing: Ediflex

Ediflex, the only commercial editing system to employ any form of content model, utilizes existing structures and conventions employed by the narrative film industry in order to provide editors with a primitive automatic editing tool. When footage is brought into the Ediflex system, it is reviewed by an assistant who logs each shot according to the lined script.[1] After the logging is complete, the Ediflex system contains a data representation of the footage broken down into shots, and, for each shot, the system maintains a set of marks which correspond to specific lines of dialog within the script.

The Ediflex then displays a visual map of the footage in which each shot is represented as a vertical line down the screen, along with reference numbers which correspond to lines of dialog from the lined script. Using the lined script as a guide, the editor makes a rough cut of the material by choosing points along the vertical lines on the screen; each point indicates a cut to a particular shot at a particular line in the script. Ediflex uses its knowledge of the correspondence between the footage and the lined script to create cut points between lines. This rough cut can then be viewed and, later, adjusted and changed until it is satisfactory. If the material has been logged accurately, the rough cut thus created will be a chronological representation of the script in which basic action and dialogue continuity is maintained across cuts.

While the Ediflex is a fast and powerful tool for cutting scripted narrative, it relies heavily on the chronology and structure of a piece as defined by the script. Ediflex does not maintain any data representation of the film's story or characters, nor does it's database contain any visual knowledge about individual shots, such as camera angle, motion, characters shown etc. Without a true cinematic content-model, [RUBIN 1989] any automatic

---

[1] The lined script shows what shots and takes exist of action and dialog at any point in the script. This information is indicated by numbered vertical lines drawn on the script. A sample lined-script is included in Appendix C.

structuring of the material beyond what is already pre-determined by the script, is impossible.

Gilles Bloch's Automatic Editing System

Aside from the present work, the only other attempt to create a truly automatic editing system was Gilles Bloch's *machine de montage pour l'Audiovisuel* (Audiovisual Editing Machine). [BLOCH 1986] Bloch, whose background was in artificial intelligence, developed a frame-based system for representing cinematic knowledge. Working at Sup Télécom in Paris, Bloch created a system which formed edit lists based on knowledge of plot content, then tested the lists for visual correctness based on editing rules and a database containing motion, position, angle, field of view and lighting information. When it discovered a "bad" cut, it would replace shots to resolve the problem, and repeat the process until it had eliminated poor cuts. In the creation of his underlying knowledge representation, Bloch too drew on Burch's structured analysis of cinematic transitions.

Unfortunately, Gilles Bloch died not long after completing his doctoral work on the *machine de montage*. At the time of this writing, I have been unable to obtain any videotapes which show the results of his system, and the written accounts of his work do not indicate to what extent he was actually successful at producing edited sequences. Nonetheless, his work stands as the major conceptual predecessor to the Constraint-Based Editor, and his thinking was important in shaping the direction of the work presented here.

Chapter Three: The Constraint-Based Editor

The terms "elastic cinema" and "re-configurable video"[1] suggest more than has ever truly been achieved through interactive video applications; the terms suggest a kind of movie that is malleable and plastic, that is changeable and fluid, a movie whose dimensions and qualities can be altered at will. The

---

[1] These terms have been used for over a decade at MIT to describe experiments in non-linear editing there. See [SASNETT 1986] and [DAVENPORT 1987] for details.

terms also suggest coherence; an "elastic" movie is one which can be stretched and twisted without being broken into fragments.

The Constraint-Based Editor provides a structure for a truly elastic movie, one in which the notions of time, of space, of narration can all be changed, and yet one which will always tell a story as a seamless, coherent whole with a beginning, a middle and an end.

Material: "Rowes Wharf Intrigue"

In order to provide a body of appropriate material with which to work, I produced a half-hour video disc, "Rowes Wharf Intrigue," a copy of which is included with this thesis. Original material was produced because existing rushes[†] already contained so many editing constraints that adding additional constraints would have been quite difficult. In particular, the presence of dialog places rigid limits on what material must be used, in what order the material must be placed, and where the cuts can fall, since any dialogue must make sense in the edited version. Also, dramatic features are typically produced according to a story board; enough coverage[†] is provided to create one version of any scene, but no more. In fact, in a ideal dramatic production, the director has made many of the editing decisions before shooting even begins, so that what is shot is exactly what is needed, and no more.

To provide material for the Constraint-Based Editor, the opposite approach was taken: as few decisions as possible were made about way the story would be told, and far more coverage was attempted than is normal in dramatic film production. The large amount of coverage greatly broadens the possibilities for multiple and varied tellings of the "Rowes Wharf" story. In addition, the "Rowes Wharf" script contains very little dialog, allowing maximum freedom in cutting and arranging the available shots.

[†] This thesis contains technical vocabulary from a variety of fields, including film and video production, computer science, and critical film theory. In the interest of not losing too many readers, I have included in the appendices a glossary of technical terms used. Words which are defined in the glossary are marked with a [†] the first time they appear in the text.

The script portrays a clandestine exchange of packages between a man and a woman at Rowes Wharf in Boston, an exchange which is watched by two mysterious observers and photographed by an inconspicuous grey-haired woman. The idea was to set up a recognizable situation — an intrigue scenario — and set forth some predictable actions within that scenario — a pursuit, an exchange, a voyeur taking pictures — so that both the content and the context of the sequence fit into pre-existing frames held by the audience; a generic situation was selected to remove as many story-telling constraints as possible from the film, again increasing the freedom with which the story can be told.

The original treatment for "Rowes Wharf," which also served as a screenplay, is presented in appendix B, which describes the process of conceiving and producing the video disc as an elastic narrative.

Logging and Annotation

After the "Rowes Wharf" video disc had been produced, an edit log was created[1] to provide the Constraint-Based Editor with a map of the material from which it could construct its data representation. The video disc was logged by breaking the disc down into *shots* (defined as segments of the disc from one continuous, uninterrupted take) and then by *actions* within shots. Actions were defined as segment within a shot in which some narrative event visibly occurs. Information about camera position, subject (character), angle of view,[2] and location was logged with each shot, and action, chronology and "narrative significance" information was logged with each

---

[1] At the time when the bulk of the material was originally logged, no logging program existed. Log information was entered by simply typing frame numbers and other information into a formatted ascii file. A primitive batch-style program was then written to read the data out of the ascii file and convert it to the standard database format used by all applications running on the editing workstation. Since then, I have created an interactive logging application that allows log information to be added directly to the database and then be edited, copied or deleted. All additions and modifications to the database since the initial data entry have been made using this application.

[2] The angle of view is a subjective measure of how much the camera can see, and is a function of the focal length of the camera's lense and the camera's distance with respect to the subject of a shot. Standard angles of view are the extreme close-up, the close-up, the medium shot, the wide shot, etc.

action. In addition, frame numbers corresponding to the segment in point, out point, and icon frame[1] were also logged.

A chronology of narrative events was developed during the logging process by inserting each action as it was logged into a time-line; when all the material had been logged, the actions were assembled on the time-line in chronological order, at which point each was assigned a relative chronology index. The chronology index is used by the Constraint-Based Editor to evaluate temporal precedence relationships between segments. The chronology indices are specified as real numbers (rather than integers) so that a new index can always be added between two existing indices.

The "narrative significance" indicator associated with each action is a subjective rating (assigned by the person logging the footage) of how important that action is to the advancement and comprehension of the narrative. This rating is one of several factors used later by the edit list generator to determine which shots to keep and which to drop when trying to meet a set of editing constraints.

The Database

The result of the logging process was to create several parallel sets of objects (shots, actions, and segments), each with a particular set of data fields associated with it. Figure 2 shows the relationship between shots, segments, and actions. An action is a unique description of a single occurrence situated chronologically within the fictional world of the film. Segments are *instances* of actions; if the same action was filmed more than once (as is common in narrative cinema), then there will be at least one segment associated with that action for each shot it occurs in. In addition, the same action might be logged more than once within the same shot to yield, for instance, a long and a short version of the same take of a particular action. By definition, a segment

[1] Icon frames are selected by the editor (or in this case, the person logging the material) to provide visual mnemonics by which video segments can be remembered. These icon frames are used by various applications on the workstation to provide graphic representations of segments, groups of segments and edited lists of segments.

cannot traverse the boundary between two shots; every segment is wholly contained between the in and the out frames of a single shot.

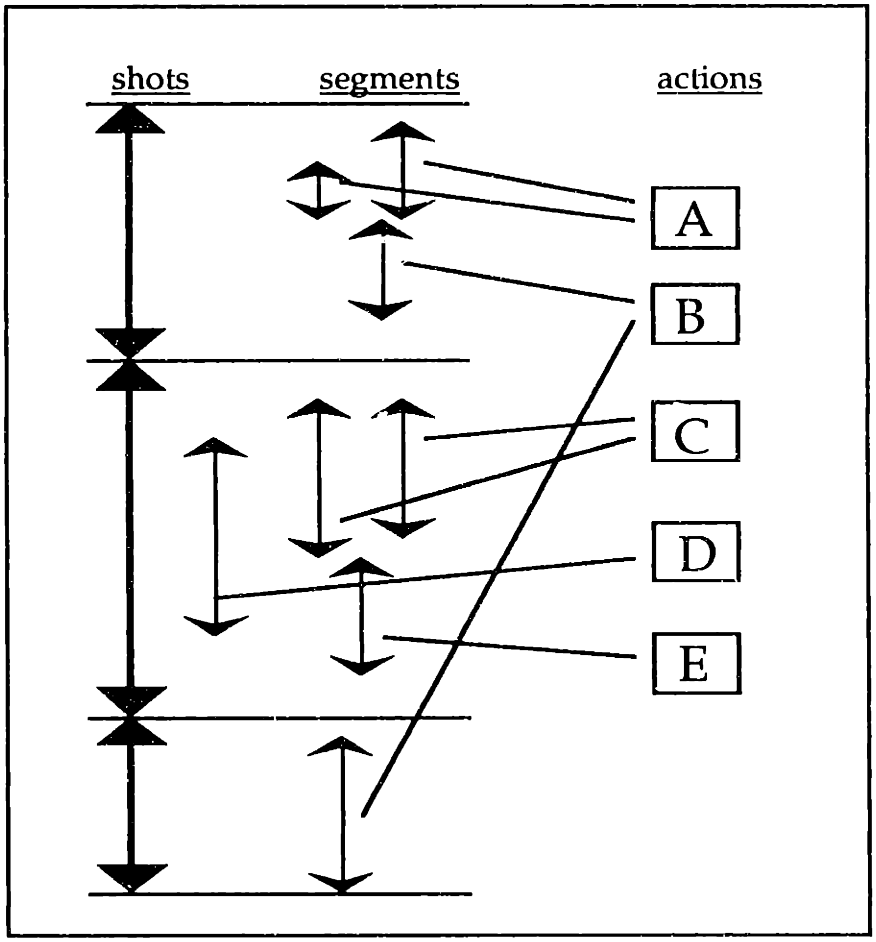


Figure 2: Shots, Segments and Actions

As we can see in figure 2, there is no clear hierarchical relationship between the three kinds of data objects. There are, however, certain rules: any given segment is contained in exactly one shot[1] and is associated with exactly one action. A shot can contain any number of segments, and the time defined by those segments may overlap. An action may be associated with any number of segments in any number of shots.

The following table describes the information that is associated with each kind of data object:

---

1 "Contained in" is defined as follows: segment A is contained in shot B if and only if the in point of B precedes the in point of A, and the out point of A precedes the in point of B.

shot	description (label) characters in shot camera position (3 vectors plus field of view)* interior/exterior* angle of view composition* location character point of view (if any) in frame, out frame icon frame
Segment	description (label) parent shot action portrayed motion vector* gaze vector* object of gaze* in frame, out frame icon frame
Action	description (label) starting chronology reference ending chronology reference* narrative significance

* Items marked with an asterisk are not yet used by the Constraint-Based Editor, but were provided in the data structure to support future enhancements and other automatic editing applications.

Table 1: Fields Associated with Data Objects

Most of these fields are either self-explanatory or have already been discussed, but a few of them do need some explanation. The camera position is defined in absolute 3-space with respect to a three-dimensional model of the location. The camera data is specified as it is in computer graphics: 3 vectors define the position (the "from" vector) and the orientation (the "at" and "up" vectors) of the camera,[1] and the field of view is determined by the focal length of the lens. The difficulty of obtaining both absolute position information and an accurate 3-dimensional model of the "Rowes Wharf" location explain why camera position data was not used in the Constraint-Based Editor! If it existed, such data would have been extremely useful in

[1] See [FOLEY 1982], pp 279 - 283 for a full explanation of mathematical representation of camera. The camera definition used here (3 vectors plus field of view) is adequate as long as the film plane remains perpendicular to the plane defined by the "at" and "up" vectors.

helping the edit-list generator determine whether two shots can be put together without spatial discontinuity. The section below entitled "Creating a Data-Rich Production Pipeline" proposes a model for cinematic production in which just such data can be collected.

Another data field needing explanation is "composition." Not yet used by the constraint based editor, the "composition" field is intended to refer to one of a group of icons which depict common arrangements of major visual elements within the film frame. Even this kind of rudimentary knowledge of the visual composition of a shot would be of tremendous value to the edit list generator, providing it with some clues as to screen direction, perspective etc.

Finally, the gaze and motion vectors are defines as vectors *with respect to the plane of the viewing screen*. These vectors both exist to give the edit-list generator additional information about screen direction, but the gaze vector (a vector indicating what direction the character on the screen is looking) was created in the hope of actually helping the program to develop the narrative space for the viewer. Noël Burch writes:

> A second way in which a film-maker can describe off-screen space involves having a character look off-screen. In *Nana*, an entire sequence or some part of it...frequently starts with a close-up or a relatively tight shot of one character addressing another who is off-screen. Sometimes the gaze of the character is so intense, so fraught with meaning, that the character off-screen (and therefore the imaginary space he occupies) becomes as important as, if not more important than, the person who is visible in the actual screen space. [BURCH 1968], p 20.

Gilles Bloch [BLOCH 1986] is to be credited with the idea of defining both screen motion and gaze as vectors relative to the viewing screen; this representation greatly reduces the difficulty of using motion and gaze information to extract information about screen direction.

The database described above has been implemented on the video workstation using CTree™, a commercial database product for microcomputers. MIT undergraduate Jeff Johnson developed a high-level programming interface to CTree™ to ease the task of data manipulation.

The Edit List Generator

The heart of the whole Constraint-Based Editor, the edit-list generator is actually a very small program. It operates by taking the totality of all the segments, sorting them chronologically, then progressively dropping segments out until all the parameters that were specified have been met.

The first pass over the material is made by a part of the program called the *butcher*. The butcher's job is to eliminate repeated actions; proceeding down the list of actions chronologically, the butcher chooses one segment to represent each action. This first choice of a segment is based on any content constraints (such as location, character emphasis etc.), combined with the pacing constraint (if slow pacing was specified, then longer segments are favored and vice versa). Given a number of segments which are all associated with the same action, the butcher rates each one as to how well it satisfies the constraints, then picks the segment with the best rating. If two or more segments are "tied" for the best rating, the tie is broken randomly.

Once an ordered list of unique segments has been selected, the list will passed to the *tailor*, the part of the program which cuts the edit list to size in order to satisfy the length constraint. Using the "narrative significance" rating, the tailor will eliminate the least important actions until the resulting list is shorter than the specified length constraint. Deleted segments will then be added back in to bring the final list to within a tolerance threshold of the desired length.

The resulting list is then passed to the *doctor*, a simple grammar-checker which uses character and angle of view information in order to spot and cure potentially poor shot transitions; for example, if two adjacent medium shots of the same character are found, the teacher will replace one of them with a segment of the same action from a different shot, or, if no such segment is available, will try to find an appropriate cutaway.

The corrected list is then sent back to the tailor for any minor alterations which might now be necessary, then from the tailor back to the teacher to make sure that no poor transitions were introduced by the tailor, and so on until both the teacher and the tailor pass the list through with no changes. Next, the output resulting from this program will be presented and analyzed.

Chapter Four: Results

For me, no technical accomplishment has ever provoked the kind of thrill I experienced when watching the very first computer-generated version of the "Rowes Wharf" movie. The first time I got the program to work was right after I had spent several days painstakingly trying to edit "Rowes Wharf" using conventional video tape. Suddenly, I found myself watching a whole new version of the movie, containing some 90 edits, which had been created in a few seconds simply by running the program! Having just spend an entire weekend laboring over no more than a dozen edits, I was intrigued and even perplexed to realize that a few of the edits made almost at random by the machine (the program was still quite primitive at this point) were better or more suggestive than edits I had made at the same point in the movie scenario.

Since then, the program has become more sophisticated, but at the same time the novelty has worn off, and I have begun to see the system's inadequacies more clearly. In this section I will present a sample sequence that was created by the Constraint-Based Editor, and I will address some of the consistent problems I have identified with the edits.

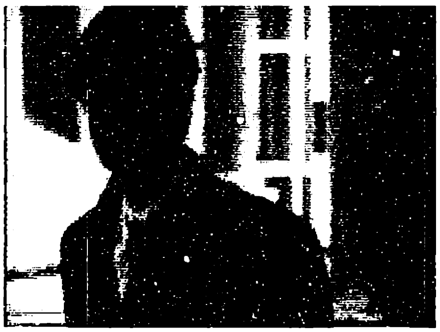
The videotape which accompanies this thesis contains three different versions of "Rowes Wharf," each generated by the system and recorded directly from the system's output. The parameters used to generate each sequence are as follows:

sequence one:	length = 3:00 minutes,
sequence two:	length = 2:00 minutes, emphasize the character "Tom," start the sequence when Tom looks up from the bench,
sequence three:	length = 2:00 minutes, emphasize the character "Genevieve," quick pace.

These three sequences represent the range of quality found in the sequences produced by the program; the first two are reasonable, and contain both coherent moments and some obvious discontinuities. The third is rougher, showing the difficulties the system has with both spatial continuity and story coherence.

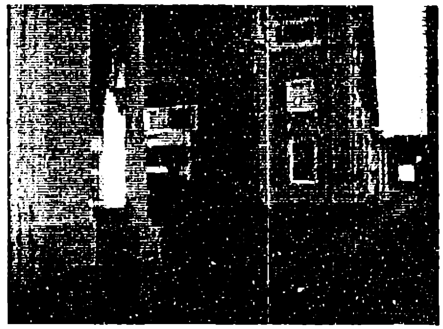
Sequence one is the longest of the three, and contains examples of most of the problems which tend to occur in these sequences. The next few pages contain still frames taken from sequence one, one frame for each shot that occurs in the sequences. The analysis which follows will refer to these images. If the reader has access to the videotape, I recommend that s/he watch this sequence now in order to better understand the context of the analysis. Also, reading the story outline in appendix B will help clarify the discussion which follows.


frame 1


frame 2


frame 3


frame 4


frame 5


frame 6

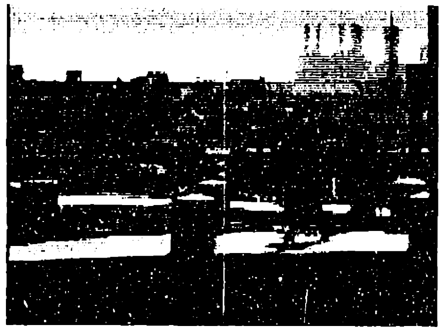

frame 7

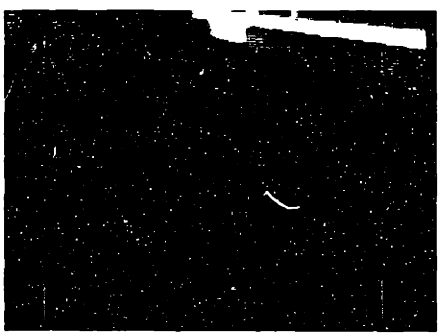

frame 8

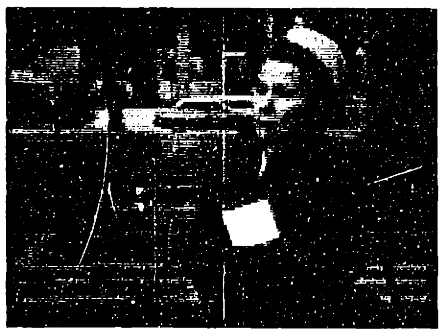

frame 9

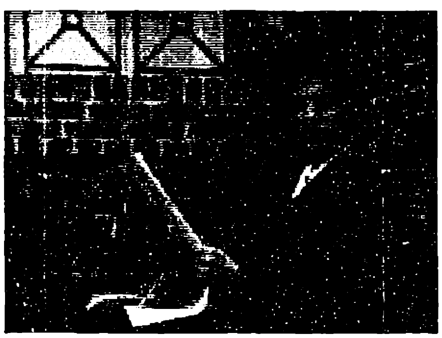

frame 13

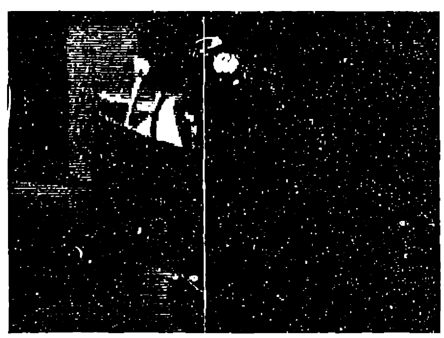

frame 10


frame 14


frame 11


frame 15


frame 12

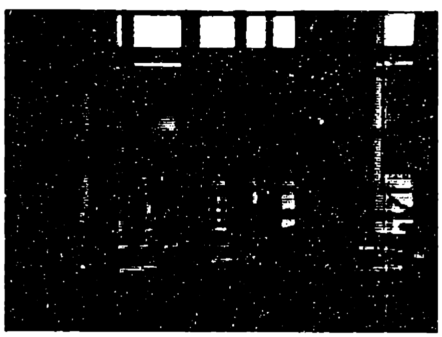

frame 16

frame 17

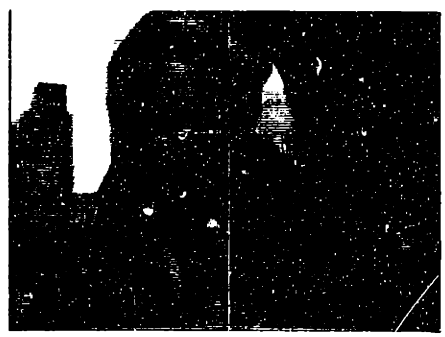
frame 21

frame 18

frame 22

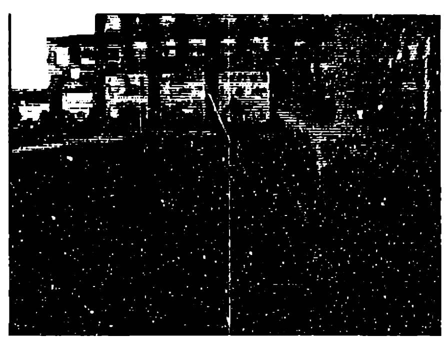


frame 19



frame 23

frame 20



frame 24

Space and Time Across a Cut

The first transition in the sequence is quite disorienting. We see a woman, Genevieve, disembark from a boat, then suddenly we see a man, Tom, walking across a plaza. There is no visible connection between them until Genevieve appears in the background at the end of the second shot; nothing gives us any clue as to the spatial relationship between shot 1 and shot 2, and Genevieve's appearance late in shot 2 seems unlikely, since we just saw her getting off a boat which is now nowhere in sight.

The transition between the first two shots points to one of the most basic deficiencies in the way the program puts shots together: as shots are deleted to reduce the sequence to a reasonable length (three minutes), spatial and temporal coherence are lost. In this case, the five minutes of diegetic† (story) time that it takes for the characters to disembark from the boat and traverse the space between the dock and the plaza has been condensed into three five-second shots which render both the space and the events completely incomprehensible.

Other spatio-temporal discontinuities in the sequence are more complex, since they do not occur at a single transition point (as in the cut from shot 1 to shot 2), but between shots separated by one or more intermediate shots. For example, in shot 10 Tom emerges from the revolving door of a Hotel. Five seconds later, in shot 12, he is seated on a bench. Five seconds after that, in shot 14, he is walking out on a pier. While we might be able to accept some compression of action within a space we understand, this sequence of three shots in three different places completely confuses us, since in this case we have no way to connect the three locations. The spatial discontinuity created by these three shots also leads to temporal disorientation; since we have no idea where these three spaces are in relation to one another, we cannot infer how long it should have taken Tom to get from one to the next, and we lose our ability to understand the passage of time in the sequence.

Gaze and Point of View

There begin to be hints of articulated narrative in the small sequence formed by shots 4, 5 and 6, which effectively suggests that the spies are watching Tom: shot 4 shows the spies watching something or someone intently, shot 5 shows Tom receding, then disappearing from view, and shot 6 appears to show the spies reacting to loosing sight of their quarry. Shot 4, which shows two people *watching*, has a powerful effect on our expectations for shot 5; we now expect to see the object of the gaze from the previous shot. Shot 5, a wide shot across the plaza, presents a completely plausible rendition of what is being seen by the spies. The enunciation of shot 5 as a point-of-view (POV) shot from the spies vantage point is strengthened when we cut from shot 5 back to the spies in shot 6, where their actions (a glance at each other, then a quick exit) can be easily interpreted if we believe that they have lost sight of Tom and must leave to find a new view point.

The cut from shot 9 to shot 10 stands in contrast to the last example; this cut is intended to show Genevieve reacting to seeing Tom emerge from the revolving doors of the hotel. However, in spite of Genevieve's strong gaze out of frame in shot 9, it is impossible to interpret shot 10 as her point of view, since shot 10 is from a low angle and from a vantage point only a few feet from Tom. Even if shot 10 were a more plausible POV shot for Genevieve, it is unlikely that two shots would be enough to clearly set up Tom as the object of Genevieve's gaze. The three shots from the previous example (the gaze, the object, the reaction) seem to represent the minimum number of shots necessary to establish unambiguously the relationship between watcher and watched.

Shots 17 through 24, in which Genevieve secretly photographs the meeting between Tom and the second woman, Nanette, is moderately effective in depicting the Genevieve's efforts to spy on the other two. As in the other examples, there is no single shot which shows the spatial relationship between the watcher (Genevieve) and the watched, but as Genevieve prepares her camera, the two cuts to the long shot of Tom and Nanette (shots 19 and 21) strongly suggest that view as Genevieve's POV.

In the first and last examples here, the gaze becomes an effective way of implicitly structuring the narrative space for the viewer. In the first example, although we never see an establishing shot, the sequence of three suggests that the spies are at the edge of the plaza watching Tom from behind. In the last example, although the precise nature of the space is less clearly implied, we understand at least that there is a direct line of sight between Genevieve and the rotunda where Tom and Nanette are meeting.

Telling the Story

Although the program takes into account a notion of shot priority and tries to preserve individual shots which portray important plot events, there is no notion in the current representation of the film's plot which takes into account interdependencies between shots. To make this three-minute version, the program cut out the series of shots at the beginning of the story which shows Genevieve spying on Tom with the mirror. In the script, the mirror scene establishes the relationship between Genevieve and Tom — that Genevieve is spying on Tom. Without this knowledge, we have almost no basis to interpret the early part of the sequence as a pursuit. Had we seen the mirror scene, we might, for example, correctly presume that Genevieve is watching Tom in shot 9; without having seen it, we have no idea who or what she is looking at.

The program needs to contain a better representation of the story and the interdependency of its elements. The current representation, in which plot actions are only classified according to chronology and subjective importance, is clearly inadequate to the complex task of story telling. If the story being told were better understood as a collection of meaningful, interrelated events in which there are causes and effects, the ability of the program to create a meaningful narrative would be greatly enhanced. In addition, if the story begins to make sense, many of the temporal and spatial continuity problems would be alleviated, since the passage of time and motion through space would be motivated by the unfolding narrative.

Chapter Five: Proposal for a Data-Rich Production Pipeline

Any of the automatic editing applications suggested in the introduction — from sophisticated tools for movie editors to movie banks to interactive art and entertainment systems — will need content models of the cinematic information they operate on. To be truly effective, all these systems need a representation of *what is in every movie frame*. As movie banks and other interactive delivery becomes not only feasible but also profitable, and as the post production process of *all* movies becomes increasingly computerized, the production apparatus of cinema will need to be re-tooled so that content data is generated and collected at every stage of production.

Our experience producing "Rowes Wharf" illustrated the difficulty of collecting content data while shooting; even though it was specifically produced for the computational environment of the Constraint-Based Editor, "Rowes Wharf" ended up having only slightly more content information associated with it than a conventionally produced narrative film. Much of the data we collected was too disorganized to be easily integrated into the computational environment. Subsequent experience creating the content model of "Rowes" further illustrated how expensive it is (in terms of time) to generate usable content data after the fact. In particular, the task of converting into computer data the information we had laboriously collected on subject and camera positions proved to be too overwhelming a task to manage, and this information ultimately had to be discarded.

The reason for these failures was not so much inadequate planning or resources as the simple lack of an integrated information framework and data collection tools for media production. The goal of this chapter is to propose some realistic means by which movie content information can be collected through the production process from a piece's conception through final editing. Methods and tools are proposed to more fully utilize the data that is already collected during conventional movie production and to collect the additional data which will be needed for sophisticated cinematic content models.

What is Content Data?

Much of the information that is used by the Constraint-Based Editor is present in the film editor's logs presented in appendix D. Looking at the lined script and the editor's log, a film editor can infer a great deal about a shot: the characters in view, the camera angle and position, point of view, location, the action taking place, and the position of the shot within the chronology of the screenplay. All of these are encoded in hand-drawn lines and squiggles, numeric codes and telegraphic shorthand which are standardized within the film industry. These logs are used not only during the editing process but during the pre-production and shooting as well, where they help to identify gaps in coverage, places where extra takes or additional shots are needed to insure continuity in the edited scene. *This is all content data, but it does not exist in a form which can be interpreted by a computer program.*

Active and Passive Data Collection

The gathering of the information which ends up in the editor's logs represents active and conscious collection of content data. Where are the points in the conventional production process where active information gathering already occurs, and how can we enhance these activities so that usable computer data could be produced at the same time as conventional paper records?

Existing Data-Gathering Roles

On the set of a conventional dramatic production, the continuity person is uniquely responsible for the collection and maintenance of the information (including the director's comments) about each scene, shot and take; this information eventually ends up in the editor's log. Other active data gathering and reporting functions are performed by camera and sound assistants, who must log rolls of audio tape and exposed film so that they can be tracked though post production and be correlated with each other with the shot, scene and take information from the continuity logs.

More active data input is done by the editing assistants, who apply edge-codes and/or time code to the material for tracking, retrieval and reference. These assistants must also correlate these reference codes with the shots and takes as they were numbered during production (see the edge-code log reproduced in appendix D).

A Model for Information Flow

In order for the creation of a content model to be economically feasible in general movie production, we need to gather the large amount of information needed without creating extra work for the production team. We want to make full use of the data which is already being collected actively, and we also want to have devices which gather information passively. Therefore, new types of data collection tools are needed: appropriate logging systems for the production and post-production environments, and passive data generators and recorders for the field. Image processing can yield still more knowledge about frame content, motion, etc. These new tools will be discussed in detail below.

In the scheme for data flow shown in figure 3, the various data gathering instruments each report to a central database, which in turn provides information to the director, editor, and others concerned with the production as it progresses, and also interacts with the editing system, interactive delivery systems, and other tools which rely on a content model. The important notion here is that of a generalized, central database maintaining an abstract content model of the movie.

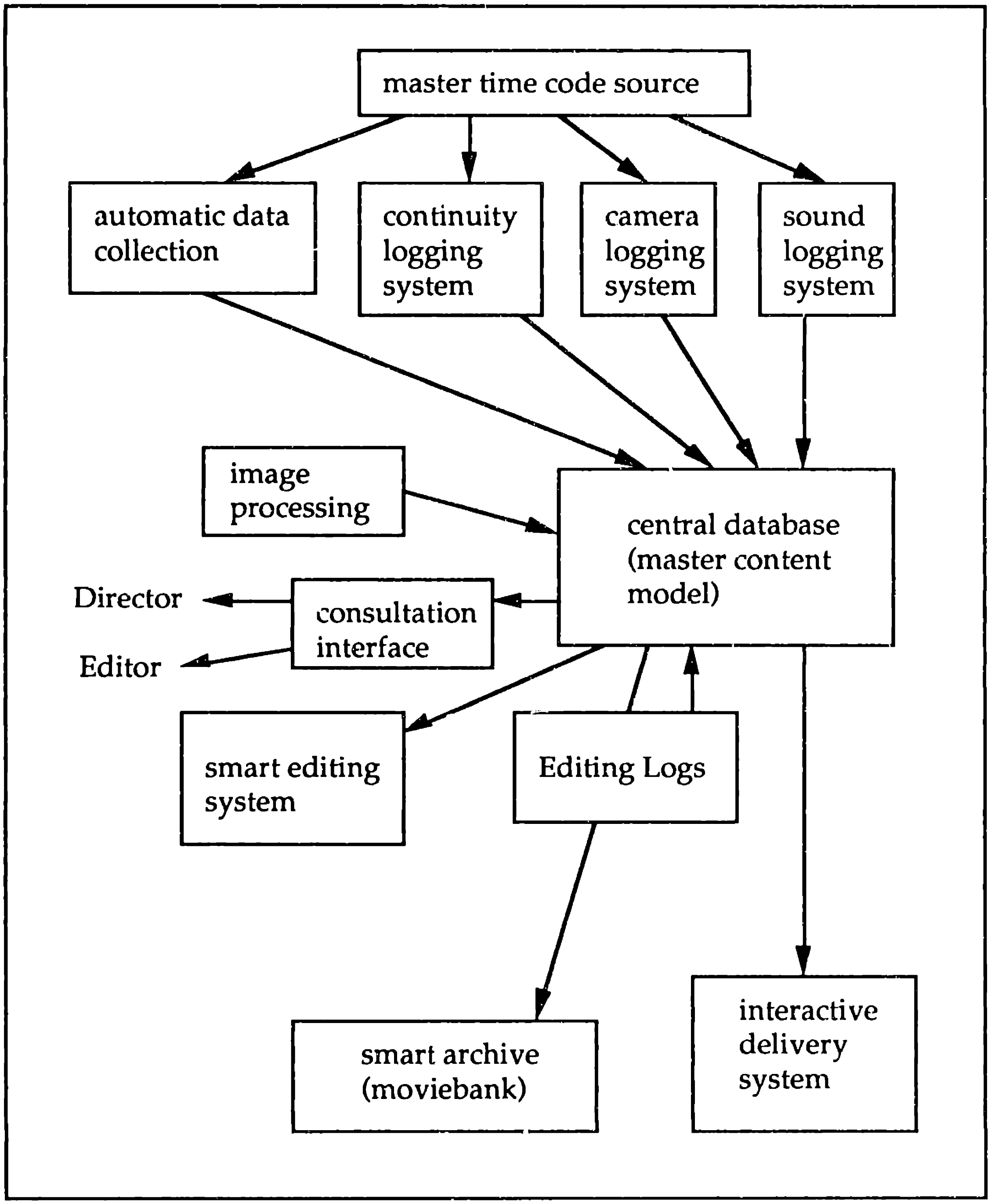


Figure 3 Information Flow in a Data-Rich Production Environment

As important as the notion of a centralized database is the need for standardized interchange formats for cinematic content data. Just as the CMX edit list format has become the *de facto* standard for transporting edit decision lists from one editing system to another, we need standard data interchange formats for shot information, camera data, script and story information. Interchange standards will allow for communication between specialized hardware and software tools made by different manufacturers, each suited to a particular aspect of cinematic data collection or processing.

Logging Tools

What are the appropriate tools for this model? Clearly, the continuity person and the various assistants should use some form of computer to perform their normal logging activities. Although at present, small, battery-operated portable computers are still slightly exotic and somewhat unreliable, that certainly will not be the case for long; laptop computers are already a common sight on movie sets, and so their increased use in the future as tools for logging and logistical organization is all but a foregone conclusion.

Nothing more exotic than a Macintosh running Hypercard could greatly enhance the usefulness of the conventional log information collected during production and editing. Hypercard (or any similar program) can be used to emulate familiar log sheets used by the continuity person and the various assistants, yet maintains an internal data representation which is well organized and can be downloaded trivially to a master database. Here, the need for standard cinematic data interchange formats is particularly well illustrated, since a continuity person will probably want to own his or her logging computer, and should be free to choose from a variety of machines and software packages without concern for what kind of database system is being used on any particular production.

The use of universal timecode, whereby each logging computer has access to the same timecode which is being striped onto the sound and the picture, would speed up the logging process for the camera and sound assistants by automatically recording accurate time-code with each log entry, and would be helpful down the line for correlation and tracking of sound and picture rolls, edge codes, etc. Distribution of common timecode could be accomplished using a device like the Aäton Escort, a small, battery operated master clock which can be carried from one device to another to initialize everything with the same timecode at the start of a shooting day. This way, each computer (or timecode generator) independently maintains the same timecode without the need for physical wires connecting them.

Passive Spatial Data Collection Systems

Accurate cinematic spatial information — camera position and direction, subject position, dimensions, motion, and orientation — can allow a computer system to determine precisely the content and composition of any movie frame. If a program knows where the camera is, where it is pointing, and what it is pointing at, it can then calculate a great deal about the contents of the frame; it can analyze screen direction properties, locate characters and possibly even determine the compositional function of each character within the shot. The presence of enough good spatial data would open up new realms for motion-control special effects in situations where they can not normally be used (hand-held camera, for example). Editing applications using the content model would also benefit, since precise evaluations of the spatial relationships between shots (*a la* Burch) could now be made computationally, allowing software programs to determine whether a space is being presented coherently. We could imagine the cinematic equivalent of a "spell checker," which could use spatial data to find and bring to the attention of the editor cuts which might be disorienting or have screen direction problems (since, using spatial data, screen direction can be computed).

Unfortunately, currently available methods for gathering spatial information are tremendously costly, requiring special computer controlled camera mounts as well as lots of hand-measuring of subject-to-film-plane distance and endless drawing of diagrams. No standard format exists for recording this information and passing it on to other phases of production. What we need is a standard for "spacecode," spatial information analogous to timecode, along with devices which can generate and record it. Spacecode "tracks" could be generated for the camera and for objects of interest in the frame.

To generate spatial data for the camera, we would like to see a small, passive device, mounted on the camera and calibrated with respect to the film plane and the nodal point of the lens. Such a device should generate camera position, orientation, lens data (focal length, aperture, focus), and transport information (film/tape speed, shutter speed), and output digital information can be recorded (along with corresponding timecode) on a conventional storage medium. Such a device might calculate its position and orientation using low-power radio or laser sources located at fixed points on the set.

While generating spacecode for a camera could be accomplished using existing technology, tracking the position of filmed subjects is much more problematic; while a single point in space (along with a two orientation vectors and lens information) is sufficient to completely describe a camera, much more information is needed to describe the whereabouts of an articulated figure (such as a person) in a scene. A device such as Mike Bove's Range Camera [BOVE 1989], which creates a depth map of a scene rather than a two-dimensional image, may eventually become a practical solution to this problem. If even a low-resolution depth-map could be recorded in addition to the normal film or video image, the prospect of deriving spatial data for characters and objects in a scene would become much more realistic.

Additional Information From Image-Processing

Image processing offers a different approach to the content data collection problem. Mike Hawley [HAWLEY 87] proposes ways in which some spatial data, along with other content information, can be derived from frame-to-frame color differences. By examining the differences between successive frames over time, scene changes, camera motion, and object motion can be detected, simple objects can be identified and tracked, and shots from the same camera position can be identified.

Motion information could be used to augment data gathered in other ways -- if a depth map had been created using a range camera, motion information gathered from image-processing techniques could then be correlated with the depth map to yield specific data on character gestures and actions. If there is motion in part of the frame, and we know from the depth map and camera data that character X's hand is in that part of the frame, then we know that character X has moved his hand. In addition, if we know from the scrip that at this moment, character X draws a gun, we can then identify specific frames where the gesture of drawing the gun begins and ends. Ultimately, it is information with the depth and detail of the example just described that is needed to maximize the potential of computers as smart editing tools.

Conclusion

This document attempts to synthesize a year of diverse work in which I produced a short movie, designed a computerized video editing system, and wrote software to perform "constraint-based" editing on the movie I produced. I also interviewed movies editors and studied cinematic theory to better understand the cinematic editing process, the process I was trying to facilitate and emulate computationally. In these pages I have done my best to draw all this work together by focusing on the Constraint-Based Editing program, its data representation, and the technical aspects of its design.

Movie editing is a subtle and exacting craft; as practiced by some, it is an art. In trying to design a program which make rudimentary editing decisions, it is not my intention to suggest that the creative aspects of editing could ever be performed by computers. Instead, my hope is that the underlying concepts of the Constraint-Based Editor will lead to computer systems which can enhance and assist the creative film editor. Existing non-linear editing workstations are just beginning to provide tools which actually increase the creative freedom of editors. By making editing computers as smart as possible, we might begin to see editing systems which offer real assistance, suggestions, and even criticism to movie editors.

In addition, intelligent editing computers, combined with adequate content models, can create more engrossing cinematic experiences in information retrieval situations (like archives and libraries) where no human editor can be present.

Appendix A: The Video Workstation

First, a clarification: the Constraint-Based Editor as such refers explicitly to a single (and relatively small) computer program. The program runs in an extensive software environment which allows it access to a movie content database, and which provides high-level methods for interacting with the video hardware. All the software runs on a hardware configuration which makes real-time, seamless, computer-controlled editing possible. The program, the supplemental software, and the hardware are the star, supporting cast, and crew of this thesis. This chapter will deal with the hardware and support software which form the technological underpinnings of this thesis.

The Technical Problem of Seamless Virtual Editing

Seamless playback refers to the ability of the computer to play back contiguous segments of video and audio, one after another, with no visible "seams" (video glitches, static, black frame or audio clicks) between them. Any of these artifacts destroys all hope of maintaining the effect of continuity between shots, making it impossible to judge whether a particular edit (or a whole sequence) "works" cinematically. Actually achieving seamless playback was the major technological problem that had to be solved before the conceptual work could be undertaken.

Ideally, a computer editing system should be able to access instantly any piece of material in its work space. If that were possible, then previews and editing could be performed trivially: at the end of one segment, the machine would instantly jump to the next within the vertical interval†, and the edit would have been performed invisibly. Unfortunately, the only current technology capable of such instant access is digital memory, and even the largest storage devices are only capable of holding a few minutes of video.

Video disc technology does not permit instant random access, but it comes close; any frame from one-half hour of video on a disc can be located in less than two seconds (faster using more advanced players). Seamless cuts can be made by using multiple video disc players, each containing an identical copy of the source material being edited.

To achieve instant previews or assemblies, the workstation functions not unlike a radio disc jockey; utilizing multiple copies of the same source material, the workstation cues up playback machines and then starts the first shot rolling. Shortly before the end of the first shot, the second shot will be started on another playback machine, and then the workstation will trigger a video and/or audio switch from the first to the second. As soon as the switch occurs, the first playback machine is free and is cued to a shot further down the edit decision list. Software schedules the movements of the disc player heads, accessing each player in a round-robin fashion. Using video disc technology, the workstation provides a hardware and software platform which simulates true random access, providing real-time previewing and editing capabilities.

The Hardware Platform: The Media Lab Video Workstation

The basic hardware needs for disc-based seamless editing consists of several video disc players, a vertical-interval audio/video switcher, and a computer capable of real-time control of these devices. The computer on which the Constraint-Based Editor runs is a Project Athena Visual Workstation, which consists of a DEC MicroVax II with 12 serial ports and a Parallax video/graphics display board.[1] The MicroVax runs Unix 4.2, which is a not a real-time operating system. In spite of this drawback, however, we have developed software (described in the next section) which is capable of performing seamless video playback with nearly frame-accurate results.[2]

---

[1] The Parallax board allows live video to be digitized in real-time and displayed on the work station's color monitor. Using Project Athena's video extensions to the X Window System, digitized video from the Parallax board can be displayed and moved around the screen inside an X window.

[2] Under normal load conditions (1 or 2 users on the system, and no computationally intensive activity aside from seamless editing), approximately 80 percent of the edits performed by the system are frame accurate, and the remainder are off by plus or minus one frame.

Connected to six of the MicroVax's serial ports are DEC VDP-50 video disc players. These are modified Phillips players which are RS232 controlled and have a maximum edge-to-edge seek time of about 3 seconds. Six disc of these players containing copies of the same disc are sufficient to play most edit lists seamlessly, including edits in which sound and picture are not cut together. While these players are capable of performing all standard video disc functions (play, search, scan etc.), they are not able to play sound at speeds other than normal play speed (30 frames/second), which is a grave disadvantage for editing applications.

An Akai DP-2000 programmable patch bay/switcher, which is a serial controllable 16x16 audio and video routing and distribution switcher, is the last major equipment component in the system. This device switches video invisibly in the vertical blanking interval, and switches audio noiselessly. It also allows the controlling computer to route video and audio independently of each other, an absolute requirement in order to provide independent editing of sound and picture, and a capability not found in most general-purpose routing switchers.

The six players and the switcher all receive external video sync from a common source, insuring that the signals originating from the players are all in phase and that switching occurs during the vertical retrace.

Figure 4 shows a block diagram of the system, and figure 5 shows the video inter-connections between the major hardware components.

THE MIT VIDEO WORKSTATION

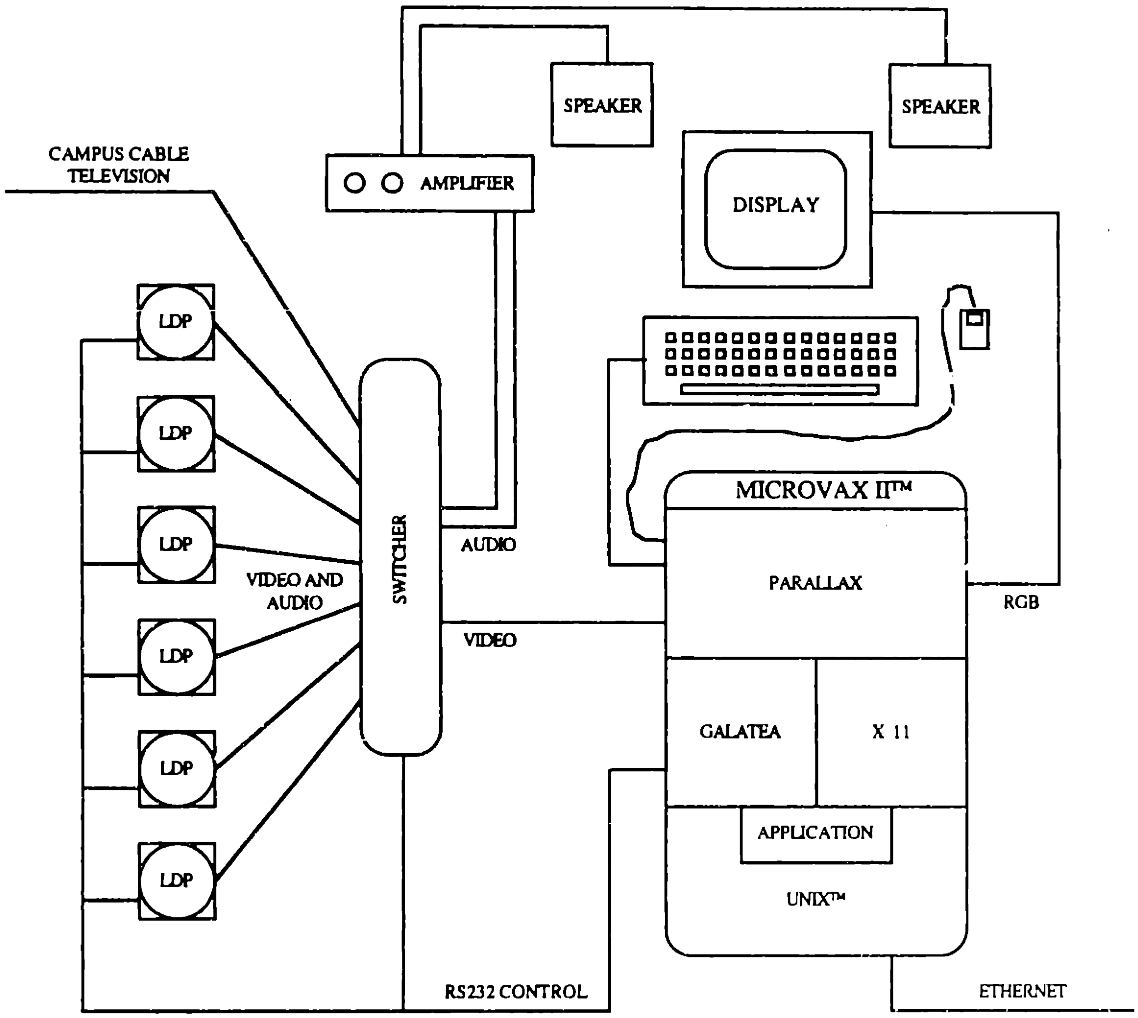


Figure 3: Block diagram of video workstation hardware

VIDEO INTERCONNECTIONS

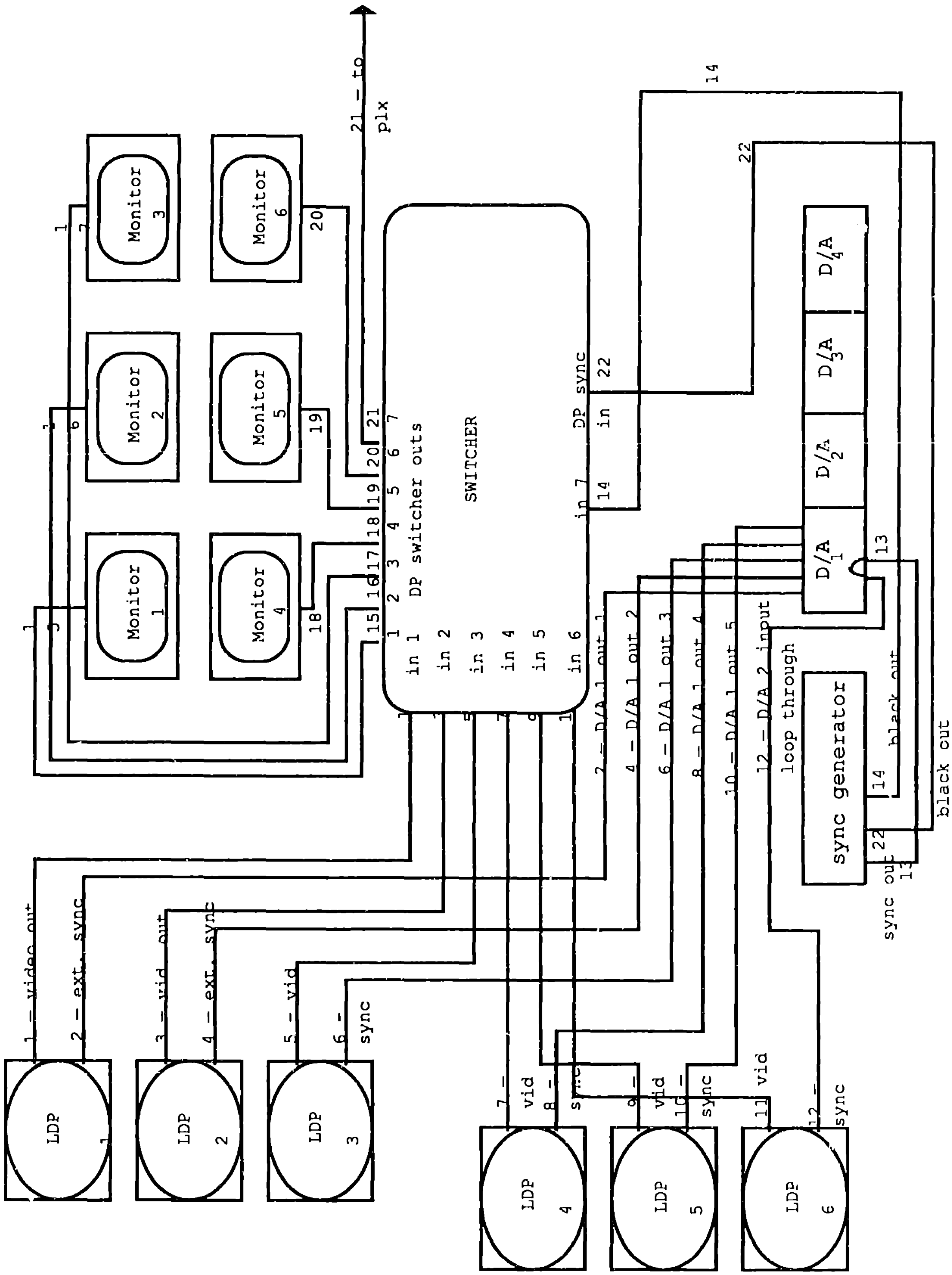


Figure 5: Video wiring diagram

Software Platform: The Galatea Video Device Control Server

A software platform exists on the workstation to provide high-level methods for video application programs to communicate with the video hardware. Rather than sending control codes directly to the disc players and switchers, application programs can issue more abstract instructions to the system. For example, in order to play a video segment (with audio) on the workstation display, the following steps are necessary: first determine which of the available players contains the segment, then query those players to find the one whose head is closest to the starting point of the segment, then issue a command to that player to search to the starting point, then issue separate commands to the switcher to route video and audio from that player to the workstation display, then issue the "play" command to the player, then wait while the player plays the segment, and then, finally, stop the player.

Instead of issuing all of those commands explicitly, an application program needs a way to tell the system "play 'Rowes Wharf,' frames 2000 to 3000 with picture and sound." The system should then handle the details of communicating with all the devices involved. The notion of an abstract means of communication becomes even more essential when an application needs to play back complex seamless edit lists, rather than a single segment as described above.

All the software developed for the video workstation utilizes our video device control server† called Galatea.[1][APPLEBAUM 1989] Galatea provides a level of indirection between application programs (Galatea's "clients") and the video hardware. The intermediate software layer represented by Galatea provides the level of abstraction just discussed, and it also allows the disc players, switcher and other video resources to be shared by multiple client applications and to be accessed transparently over MIT's Ethernet.[2]

---

[1] Galatea was developed by Dan Applebaum for MIT Film/Video, Project Athena, and the Visible Language Workshop. I added extensions to Galatea to permit seamless playback of edit lists, and Hal Birkeland made modifications to those extensions.

[2] Ethernet is a local-area network (LAN) used by MIT to connect hundreds of computers on campus.

Client[†] applications issue abstract commands to Galatea, which translates those commands into all the necessary device-specific instructions which it sends out through the serial ports. To play back seamless edit lists, a client down-loads to Galatea a program which abstractly describes the edited sequence. Galatea takes this abstract description and converts it to a real-time list of device commands which are adjusted for pre-roll, start-up delays, etc.. Using the Unix system clock as a time reference, Galatea then executes the device commands at the appropriate times, and the resulting seamless edit list is played back on the workstation monitor.

As mentioned earlier, the workstation does not use a real-time operating system, which means that the accuracy of seamless playback is affected by the load on the machine when the list is played back, and that absolute accuracy can never be guaranteed. Future hardware and software development plans include creating an independent and truly real-time server which runs on a separate CPU. Running the Galatea server on a dedicated outboard processor will result in improved real-time performance, and will obviate the current necessity that the host workstation have eight or more serial ports.

The Galatea Seamless Extensions

The set of seamless Galatea extensions were designed with an assumption in mind that different kinds of applications would want to maintain different internal representations of edit lists. For instance, a traditional video sequence editing application might use a very different internal structure to represent an edit in progress from an archival sequence retrieval program; the editing application might use a complex, object oriented tree structure, while the archival program might use a simple linear representation.

The seamless extension provides a generic, unstructured representation for edit decision lists; when an application wants to play an edit list, it first converts its own internal representation of the list to this generic representation, then down-loads the generic list to Galatea, where it is compiled and executed.

The structure of the generic list is such that edit decision lists of arbitrary complexity can be represented, with sound and picture cuts occurring independently. The generic list consists of a list of segment definitions each containing the following information: source volume name, list start address, source start address, shot duration, and a list of one or more switch events, each accompanied by a channel selection bit-mask and an address indicating where the switch should occur with respect to the list. The bit-mask determines what combination of video and audio channels the switch applies to. Figure 6 shows a sample generic edit list:

```
source vol:  "Rowes Wharf"
start addr:  0
source addr: 23324
duration:    125
switch addr: 0      mask: LRV

source vol:  "Rowes Wharf"
start addr:  125
source addr: 23137
duration:    250
switch addr: 125    mask: LRV
switch addr: 270    mask: V

source vol:  "Rowes Wharf"
start addr:  240
source addr: 44328
duration:    30
switch addr: 150    mask: V

source vol:  "Rowes Wharf"
start addr:  350
source addr: 42883
duration:    100
switch addr: 350    mask: LR
switch addr: 375    mask: V
```

Figure 6: Generic Edit List

The edit list above represents a 450 frame (15 second) sequence which consists of a simple first shot, followed by a second shot, followed by a one-second picture-only cut to a third shot, then a picture-only cut back to the second shot, followed by an "L" cut (sound first, then picture) to the last shot.

To form the generic list, no knowledge is needed about the number, status, characteristics or volumes contained in the disc-players attached to the system. When this list is down-loaded to the server, it is compiled to form a real-time, device-specific command list. The server has access to an up-to-date table of disc players showing all player characteristics and current status information, and so is able to assign each of the segments above to a player and also calculate pre-roll times and make timing adjustments to compensate for that player's particular performance characteristics. Figure 7 shows a compiled version of the generic list from figure 6:

frame	device	command
-100	VDP-1	search 23314
-100	VDP-2	search 23127
-10	VDP-1	forward 30 fps
0	switcher	CH-1 LRV
115	VDP-2	forward 30 fps
125	switcher	CH-2 LRV
125	VDP-1	search 44318
230	VDP-1	forward 30 fps
240	switcher	CH-1 V
270	switcher	CH-2 V
280	VDP-1	search 42873
340	VDP-1	forward 30 fps
350	switcher	CH-1 LR
375	switcher	CH-1 V
375	VDP-2	stop
450	VDP-1	stop

Figure 7: Real-Time Command List

The list above is based on a system in which two players, VDP-1 and VDP-2, contain the volume named "Rowes Wharf." The negative frame addresses at the top of the first column represent events which occur during the 100-frame setup time prior to running the list.

While the generic representation provides a completely adequate way to represent any edit list, it has two deficiencies which make it a poor data representation for list manipulation (shot insertion, change and deletion) within applications programs. First, the generic representation uses absolute addresses to represent the time (since the start of the list) that events occur. This means that if, for example, the first shot on the list were lengthened, the remaining start and switch addresses, which are absolute with respect to the list, would have to be adjusted ("rippled") all the way down the list to compensate for the frames added at the beginning. Second, the generic representation is rather opaque; even for someone who understands the list format, the way the switch commands are grouped with segments makes it difficult to simply look at the list and understand the way the edit will flow.

The non-intuitive nature of the generic list representation also impedes the task of writing programs which manipulate the list structures.

Because of these drawbacks, the generic list representation is not intended to be used directly by application programs. Instead, the strategy is that different applications will use different internal list representations which meet the particular requirements of each, and when ready to run a list, an application will transform its own representation into the generic representation, whereupon it can be down-loaded to the server. Currently under development is a powerful, general-purpose list representation suitable for editing applications. This new representation will utilize some form of relative list addressing so that segments can be easily added, deleted and changed without any rippling of subsequent addresses. The new representation will also be easy to transform into the generic representation for the server.

Appendix B: An Elastic Narrative, "Rowes Wharf Intrigue"

The Original Treatment

The following is the original treatment which formed the basis for the shooting script of "Rowes Wharf Intrigue." Some details were changed or omitted for expediency during production, but the basic story of the material on the video disc remains quite similar to that presented below:

Rowes Wharf Intrigue

by Ben Rubin

A water shuttle filled with well dressed commuters approaches the Rowes Wharf terminal. A woman passenger, dressed more like a tourist than a commuter, opens her compact to touch up her lipstick. In her mirror, she spies on a man carrying a square, black case. He does not notice her. As the boat approaches the dock, she pulls out a small camera and snaps a picture of Rowes Wharf.

The boat docks, and all the passengers disembark. The man looks at his watch, then wanders out toward the main plaza at a leisurely pace.

The woman is the last to disembark. She starts slowly toward the plaza, stopping to light a cigarette and glance up surreptitiously at the man, who by now is well toward the plaza. The man wanders toward the arch, and the woman follows at a distance.

The plaza is populated by a lone sax player, a kid throwing a ball to a dog, a jogger warming up and stretching, and two pamphleteers with clipboards soliciting money. A steady stream of commuters course from the shuttle terminal, around the plaza, and out through the arch. Mournful bursts from the sax player echo through the plaza together with the kid's yells, the dog's bark, and boat sounds from the water.

As the man nears the arch, he is approached by one of the pamphleteers. They speak briefly, both smile, and the man gives five dollars for the pamphleteers cause, then disappears into the hotel lobby entrance which is inside the archway.

The woman pulls out a green Michelin tourist's guidebook to Boston and admires the architecture, looking up at the arch. She takes another picture.

The jogger finishes warming up and jogs off toward the residential peer. The kid's ball has fallen into the water, and he and the dog are down on the dock trying to fish it out with a pole.

The man emerges from the hotel lobby with a newspaper. He looks at his watch again, then strolls down a flight of steps to the lower level of the plaza and seats himself on a bench from which he has a good view of the rotunda, the dock, and the water.

Behind him, the woman walks down a ramp and sits on the bench around the corner from him. She also has a good view of the rotunda, the dock and the water, but she cannot see the man. On the bench to her left a young couple is kissing.

She pulls a high-tech looking camera with a long lens out of her bag and places it on the bench next to her, covering it quickly with her scarf. She pulls out her guidebook and begins to read.

The jogger has circled around the residential wharf, and now jogs past the couple and woman on the bench and on towards the terminal docks.

Out on the water, a small launch with two men in it approaches the dock. One of the men scoops the kid's ball out of the water and tosses it to him. The kid is delighted.

The man on the bench sees the boat come in, looks at his watch, and puts down his paper. He watches as one of the men in the launch disembarks. Carrying a thin, black, zippered portfolio, the man from the launch makes his way up from the dock, around the plaza and over to the rotunda. Inside the rotunda he stops, looks at his watch, lights a cigarette and admires the domed ceiling.

The man on the bench rises and begins walking toward the rotunda, case in hand. As he approaches the rotunda, the woman on the bench slowly puts down her guidebook and reaches down to uncover her camera.

The pamphleteers have by now made their way down to the lower level of the plaza. Passing by the kissing couple, one approaches the woman and, joking, poses with his clipboard to have his picture taken.

She takes several pictures as the men in the rotunda shake hands and exchange cases, then packs up brusquely and heads up the stairs and out toward the arch.

The man from the launch, now carrying the square case, strides back down the ramp to the dock, boards the launch, and casts off. The launch heads out to sea.

The other man heads back to the shuttle terminal with the thin portfolio. He boards a waiting shuttle which departs almost immediately.

The kid is now throwing his ball around with the pamphleteers and the dog.

The water shuttle heads out across the harbor as a jet ascends from Logan Airport.

Appendix C: Guide to the Accompanying Video Tape

The accompanying videotape contains three sample sequences generated by the Constraint-Based Editor. These sequecnes were recorded directly from the output of the system.

The paramaters used to generate each sequence are as follows:

sequence one: length = 3:00 minutes

sequence two: length = 2:00 minutes, empasize the character "Tom," start the sequence when Tom looks up from the bench

sequence three: length = 2:00 minutes, emphasize the character "Genevieve," quick pace

Appendix D: A Film Editor's Log Sheets

A useful way to understand the methodology of any complex human process is to look at the paper work used by those involved. The following pages show examples of the kind paper work used by film editors to help them track material through the editing process. The ways that shots and takes are cross-referenced through the three types of documents — the log sheet, the lined script and the edge-code log — help illuminate the mental organization of the film editor.

The lined script is just that — copy of the shooting script with vertical lines indicating where shots begin and end with respect to the script. Where there are squiggles in a line, it indicates that the character who is speaking or acting at that moment in the script is not visible in that shot. The lined script is used by the editor to determine at a glance what his/her options are at any point in the script.

The log sheet lists shots and takes by number along with comments by the editor and director. A descriptive shorthand is used to indicate the type of shot (classified roughly by angle of view) and the subject.

Finally, the edge-code log allows the editor's assistant to trace and retrieve footage corresponding to shots the editor might want.

These log sheets from the forthcoming feature, "Miami Blues" were provided courtesy of film editor Craig McKay; the script is by Corey B. Yugler.

Lined Script

44 CONTINUED: 32 44

SUSIE

27 Junior, you want me to fix those pork chops now?

Hoke comes to life.

HOKE

28 I'll put 'em back in for pork chops.

Junior turns to stare daggers at Susie, but she is already on the move.

SUSIE

29 Pork chops comin'up.

---

45 SUSIE'S DINING ROOM-LATER 45

Hoke, his jacket removed, his teeth replaced, is on his fourth Polar as he eats pork chops, peas, honey rolls, and mashed potatoes. Junior has his entire forearm surrounding his plate as he shovel the food into his mouth. He cannot believe Hoke is still here. Hoke grabs another roll.

HOKE

Herman, your fiancee is the best cook in South Florida.

Susie hears this as she returns from the kitchen with the rest of the potatoes.

SUSIE

More taters? Junior?

Hoke holds up his plate. she serves him and gives more to Junior.

HOKE

So, Herman, where'd you do your time?

JUNIOR

Time? What do you mean?

HOKE

The way you're guarding that food. Like some other con could take it away.

JUNIOR

I was brought up in foster homes. I never ate no dessert till the eighth grade.

Lined Script (continued)

45 45A 45B 45C 45D 45E 45F

45 CONTINUED: 33

45

HOKE

Damn. I got a daughter in the eighth grade. I send half my paycheck to her orthodontist.

JUNIOR

She's got your teeth.

HOKE

The joint's about the only place you got the time to work out to get a grip like that.

JUNIOR

I was an aerobics instructor.

HOKE

Shoote the shit out of that theory. Any more Polars?

SUSIE

Junior got the last one. I could run out and get some.

Susie goes to get up but both Hoke and Junior reach to stop her.

HOKE

Beer's gone. I'm gone.

JUNIOR

Hey, your kidding.

Both Junior and Hoke stand up.

HOKE

But I got to get that pork chop recipe first.

Susie is delighted to share her secrets. Junior stomps into the kitchen. Hoke is aware of his impatience and is deliberately overstaying his welcome.

SUSIE

Oh, that ain't nothing. I'll give it to you if you want but you don't have to write it down. Its so simple, all is is, is just take the pork chops and cook them in a pan...

Editor's Log

TITLE: MIAMI BLUES
EDITOR: CRAIG MCKAY

SCRIPT: COREY B. YUGLER
PAGE: D-32.1

SCENE	TK.	SND.	TIME	CODE	SHOT DESCRIPTION
45		SYNC			W.S. OVER HOKE-JUNIOR AND SUSIE INCLUDES JUNIOR IN B.G. WITH GUN INCLUDES SC. #46
#A125	1		.10		-NG.
	2		2.21		-COMP, FLARE WITH RISE, GOOD ENERGY
	3		2.23	AH 3019	-GOOD
	(4)		2.30	' 3238	-GOOD, LINE #9 FLUB TK. #4 FIRST GOOD FOR JR. IN B.G.
#A126	5		.11		-NG.
	6		.21		-INC. LINE #2
#A127	(7)	SR#77	2.30	3474	-GOOD 16MM 10/20/88
45A		SYNC			MED. JUNIOR-TILT UP ON RISE
#A127	1		1.45		-COMP
	(2)		1.37	AH 4013	-GOOD
	3		.10		-NG.
	(4)	SR#78	1.54	4169	-GOOD 35MM 10/20/88
45B		SYNC			MED. HOKE TILT ON RISE
#A127	(1)		1.56	AH4349	-GOOD
#A128	2		-		-NG.
	(3)	SR#78	1.58	4536	-GOOD 35MM 10/20/88
45C		SYNC			CLOSE-UP HOKE-RISES OUT OF FRAME
#A128	(1)	SR#78	1.52	AH5012	-COMP, LINE #7 FLUB
#A129	2	SR#79	.17		-LOST W/ MAG PROBLEM
	(3)		1.54	5186	-GOOD 50MM 10/20/88

Editor's Log (continued)

TITLE: MIAMI BLUES
EDITOR: CRAIG MCKAY

SCRIPT: COREY B. YUGLER
PAGE: D-32.2

SCENE	TK.	SND.	TIME CODE		SHOT DESCRIPTION
45D		SYNC			CLOSE-UP JUNIOR-RISES OUT OF FRAME
#A130	1		1.28		-INC. TO LINE #8
	(2)		1.43	AH5366	-GOOD
	(3)		1.43	5529	-GOOD, SND. TINKLY
	(4)		1.41	5694	-GOOD
		SR#79			50MM 10/20/88
45E		SYNC			MED. SUSIE WALKS IN & SITS-MED. 2 SH/ SUSIE AND HOKE
#A129	1		.05		-NG.
	(2)		2.35	AH6012	-COMP
	(3)		2.33	6256	-GOOD
		SR#80			35MM 10/20/88
45F		SYNC			CLOSE-UP SUSIE SITS INTO FR.-TITE 2 SH/ HOKE AND SUSIE
#A129	(1)		2.12	AH7012	-COMP, HEAVY FOOT STOMP FROM HOKES BOOTS
#A131	(2)		2.08	7222	-GOOD
		SR#80			50MM 10/20/88

Edge-Code Log

10/20 DR64

ROLL NO	EDGE NUMBERS	CODE NUMBERS	SC	TK	SND	DESCRIPTION
A127	G2X 28046-28201	AH 4013-4168	45A	2	78	M/S Jr
↓	28227-28406	4169-4348		4	↓	
	28407-28593	4349-4535	45B	1		M/S Hoke
A128	G13X 03888-04080	4536-4729		3		

10/20 DR65

ROLL NO	EDGE NUMBERS	CODE NUMBERS	SC	TK	SND	DESCRIPTION
A128	G13X 04082-04254	AH 5012-5185	45C	1	78	ECU Hoke
130	G2X 23588-23767	5186-5365		3	79	
↓	23910-24072	5366-5528	45D	2	↓	ECU Jr.
	24073-24237	5529-5693		3		
	24238-24406	5694-5862		4		

10/20 DR66

ROLL NO	EDGE NUMBERS	CODE NUMBERS	SC	TK	SND	DESCRIPTION
A129	G2X 19665-19907	AH 6012-6255	45E	2	80	MCU Susie
"	19908-20145	6256-6492		3	"	

10/20 DR67

ROLL NO	EDGE NUMBERS	CODE NUMBERS	SC	TK	SND	DESCRIPTION
A129	G2X 20157-20362	AH 7012-7218	45F	1	80	ECU Susie
A131	G13X 97275-97476	7222-7424		2	↓	
↓	97477-97630	7425-7473	46	3		M/S on Jr w/ gun
	97631-97673	7474-7515		4		
	97674-97718	7516-7560	46A	1	81	MCU on Jr. w/ gun
	97719-97771	7561-7613		2	↓	
	97772-97834	7614-7676	46B	1		Jrs POV
	97835-97897	7677-7739		2		
	97898-97967	7740-7809		3		

Appendix E: Glossary of Technical Terms

bin: A bin is a container specially designed for hanging clips of motion picture film. Film is hung on numbered hooks above the bin, and a record is kept of what clips are hung on each hook so that clips can be easily retrieved.

client/server: In a multi-tasking computer environment (such as Unix™), a client program is one which utilizes a *server* program to perform some complex task. The client and server run as independent processes with some communication link for data to flow between them. The client/server model is useful in any situation where multiple programs need to access a common resource, such as a shared database or a peripheral device.

core: A core is a small plastic spool around which motion picture film can be wound. Cores are less expensive and less bulky than actual reels, and are normally used to store the elements of a film during the editing and post-production process.

coverage: The *coverage* of a narrative film scene refers to the collection of overlapping shots which portray the same action from different angles. For instance, when filming two people having a conversation, the coverage might include one shot of person A's face, one shot of person B's face, and one showing the two of them together.

dailies: Dailies are positive prints struck from the negative after a day of shooting.

diegesis: The diegesis of a film is the fictional universe in which the film takes place. Diegetic space and time refer to space and time within the fictional universe of the diegesis.

match-cut: A match cut is any cinematic transition in which continuity of space and/or motion is preserved.

off-line/on-line editing: In video post-production, off line editing refers to cutting done with a work-dub, often with visible time-code inserted into the picture area for reference. Off line editing is generally though of as preparation for on-line editing, which is the final phase of picture and sound editing. Audio and visual special effects, titles, etc. are all added to a piece during on-line editing. If a piece has first been edited off-line, then it will often be re-edited from the original source tapes in accordance with the off-line cut.

pro-filmic space: The pro-filmic space is that subset of the diegetic space of a film which is actually *seen* on the screen.

rushes: See *dailies*.

timecode: Timecode consists of machine-readable absolute time reference information which can be associated with motion picture and audio material. Editors use timecode for reference, logging, organization and tracking of material, and on-line editing systems use timecode to control automated cuing of video and audio tape. Several systems exist in common usage, among them SMPTE timecode, a system of analog pulses which can be recorded on any audio track, VITC (Vertical Interval Time Code), which modulates the timecode information into the video picture, and Aäton ClearTime Code, which uses tiny LEDs to expose visible timecode along the edge of motion-picture film so that it can be read by either a person or a machine.

vertical interval: Also called the vertical retrace interval, the term refers to the time interval when the electron beam inside the cathode-ray tube (in a television monitor or camera) has completed scanning a field and is in transit to its starting point, where it will begin to trace the next field. Switching between video sources which are gen-locked (in phase) must occur in the vertical interval in order to appear smooth (glitch-free) on the screen.

Bibliography

Applebaum, D. "The Galatea Network Video Device Control System," short paper, MIT Media Laboratory, 1989.

Aumont, J., Bergala, A., Marie, M. and Vermet, M. *l'Esthétique du film.* Editions Fernand Nathan, Paris, 1983.

Barbour, J. *The Execution of a Video Editing Controller.* S.B. Thesis in Electrical Engineering and Computer Science, MIT, 1987.

Backer, D. "Personalized Visualization: Interactive Cinema for Training," Architecture Machine Group, MIT

Backer, D. and Lippman, A. "Future Interactive Graphics: Personal Video," Architecture Machine Group, MIT. Presented to the NCGA Conference, Baltimore, June, 1981.

Bazin, A. *Qu'est-ce que le cinéma?* Les Editions du Cerf, Paris, 1958.

Beachamp, D. *A Database Representation of Motion Picture Material.* S.B. Thesis in Electrical Engineering, MIT, 1987.

Blake, W. *Demonstration Software for an Experimental Video Workstation.* S.B. Thesis in Electrical Engineering and Computer Science, MIT, 1986.

Bloch, G. *Elements d'une machine de montage pour l'Audiovisuel.* Doctoral thesis, Ecole Nationale Superieure des Telecommunications, Paris, 1986.

Bloch, G. "From Concepts to Film Sequences." Short Paper, Yale University Department of Computer Science, Artificial Intelligence Lab, 1987.

Bordwell, D. and Thompson, K. *Film Art.* Knopf, New York, 1986.

Bove, V. M. "Synthetic Movies Derived from Multi Dimensional Image Sensors." PhD Thesis, MIT Media Laboratory, Movies of the Future Group, April, 1989.

Burch, N. *Une praxis du cinéma.* Gallimard, Paris, 1968.

Chion, M. *Ecrire un scénario.* Cahiers du cinéma / I.N.A., 1985.

Davenport, G. "Software Considerations for Multimedia Video Projects," Film/Video Section, MIT Media Laboratory. Paper presented to the X11 Video Extension Technical Meeting at Bellcore, June 1988.

Davenport, G. "New Orleans in Transition, 1983-1986: The Interactive Delivery of a Cinematic Case Study," Film/Video Section, MIT Media Laboratory. Paper presented to the International Congress for Design Planning and Theory, Boston, August, 1987.

DroidWorks, "Editor's Guide," user's manual for the EditDroid editing system, 1986.

Eisenstein, S. *Film Form.* Harcourt, Brace, and Company, New York, 1949.

Eisenstein, S. *The Film Sense.* Harcourt, Brace, and Company, New York, 1942.

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

Gow, G. *Suspense in the Cinema.* Castle Books, New York, 1968.

Hawley, M. "Indexing Movies," short paper, MIT Media Lab, Movies of the Future Group, 1987.

Lippman, A. *Movie Manuals: Personalized Cinema as an Instructional Partner.* Architecture Machine Group, MIT, Proposal to the Office of Naval Research, January, 1981.

Lehnert, W. "Plot Units and Narrative Summarization," in *Cognitive Science,* 4:(293-331), 1981.

Levitt, D. and Davenport, G. *Symbolic Description of Movie Media.* Media Laboratory, MIT, Proposal to the National Science Foundation, 1988.

Negroponte, N. "The Impact of Optical Videodiscs on Filmmaking," Architecture Machine Group, MIT, 1979.

Minsky, M. *The Society of Mind.* Simon and Schuster, New York, 1985.

Parent, K.T. *Development and Examination of a Movie Edit Code.* S.B. Thesis in Physics, MIT, 1987.

Pepe, L. *A Digital Icon Representation for Movie Editing and Analysis.* S.B. Thesis in Electrical Engineering and Computer Science, MIT, 1988.

Register, D., Sherbon, J., and Spangler, L. "Automatic Logging of Edit Information on Laservision Videodiscs by the Spectra System™," Spectra Image, Inc, Burbank, California. Presented at the 129th SMPTE Technical Conference in Los Angeles, November, 1987.

Reisz, K. and Millar, G. *The Technique of Film Editing.* Focal Press, Boston, 1968.

Rosenblum, R. and Karen, R. *When the Shooting Stops, the Cutting Begins.* Da Capo Press, Inc., New York, 1979

Rubin, B. and Davenport, G. "Structured Content Modeling for Cinematic Information," position paper for SIGCHI workshop on video as a research and design tool. To be published in SIGCHI conference proceedings, 1989.

Rubin, B. "Montage: a Proposal for an Interactive Videodisc Editing System to Teach Concepts of Cinematic Narrativity and Continuity." Short paper, Brown University Department of Computer Science, 1986. (No association with the Montage Picture Processor® by the Montage Group Ltd.)

Sasnett, R. *Reconfigurable Video.* M.S.V.S. Thesis in Architecture, MIT, 1986.

Sayles, J. *Thinking in Pictures.* Houghton Mifflin, Boston, 1987.

Schroeder, C. *Computerized Film Directing.* S.B. Thesis in Electrical Engineering and Computer Science, MIT, 1987.

Schuler, C. L. "The Montage: A New Approach to Editing Feature Films," *SMPTE Journal,* vol. 95, number 8:(811 - 813), August, 1986.

Tindell, J.M. *Picture and Sound Editing on Optical Media.* S.B. Thesis in Computer Science, MIT, 1986.

Tucker, A. *Applied Combinatorics.* John Wiley & Sons, New York, 1980.

Weynand, D. *Computerized Videotape Editing.* Weynand Associates, Los Angeles, 1983.

Wilson, W. "An Analysis of the Montage and Ediflex Off-line Editing Systems with an Emphasis on their Suitability for Editing an Episodic Action-Adventure Television Series," report produced under contract from Paramount Pictures by Barking Spider Productions, Los Angeles, March 1988.