L

A PERSONALIZED, INTERACTIVE MOVIE MANUAL

by

John Carlos Correa

Submitted in Partial Fulfillment

of the Requirements for the

Degree of Bachelor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1981

Signature of Author.................................
                  Department Electrical Engineering and
                  Computer Science, May 21, 1981

Certified by.......................................
                  Andrew Lippman, Thesis Supervisor
                  Assistant Professor of Media Technology

Accepted by.......................................
                  Chairperson,
                  Departmental Committee on Theses

A PERSONALIZED, INTERACTIVE MOVIE MANUAL

by

John Carlos Correa

Thesis Supervisor

Andrew Lippman, Assistant Professor of Media Technology

May 1981

Abstract

A television screen with a computer interface is an
interactive workspace. It allows a human to talk to a computer
to elicit information as progressions of images, sound, and text
played from an optical videodisc or generated by computer. The
workspace can be thought of as a learning tool: an interactive
movie manual. The manual is as random access a device as a book.
The user requests information through a table of contents,
interacting at any time to modify the movie along a number of
axes. By interacting with the system the user becomes a filmaker
and a computer programmer, editing an educational film
on-the-fly, suited to his particular interests and curiosity.

With time the movie manual grows to adapt to the individual
user. He may modify the material in the manual, reauthoring it
for future presentation. As a book falls open to a worn page,
the movie manual generalizes from previous interaction, chosing
material it guesses the user will prefer; as opposed to a
standard default. Different algorithms for the selection and
presentation of information may be implemented, depending upon a
preferred theory of learning or film making.

A movie manual is a synthesis of ideas from educational
cinema, artificial intelligence, and computer aided instruction
into a new and promising topic of research.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

Masculine pronouns are used in this paper by convention.
The reader should assume that they apply to both genders.

INTRODUCTION

THEORY

Material on the videodisc is described in a movie manual
database. A topic overview and a learning theory are implicit in
its structure. The database is a network of semantic nodes, each
node corresponding to a single concept, connections being
informational links between concepts. The concept corresponding
to a node can be illustrated in a number of ways, each
illustration is called a sequence. Detailed information on which
sequences are available for a node and where to find related
material on the videodisc is stored within the node.

By traversing the semantic net a progression of ideas is
developed. Criteria for specifying or modifying the algorithm
that choses nodes and sequences forms the basis for
personalizing the presentation of cinematic material.

Interaction entails making information and controls
concerning relevant nodes and sequences available to the viewer.
He may then alter his path through the material or manipulate it
in time.

PERSONALIZATION

Attributes are included in the movie manual database so
that the selection of material may be personalized. Nodes in

addition to being more or less sequential are classified in two ways. One attribute specifies an experience threshold. For example, a movie about replacing the heating element in a toaster may include nodes which explain three steps. The first time a viewer watches the movie, the movie manual presents all three steps. The second time, it assumes that the user knows enough to shut off the power, skipping that node all together or just mentioning a quick reminder. In subsequent movies the manual omits the section telling the user to remove the cover.
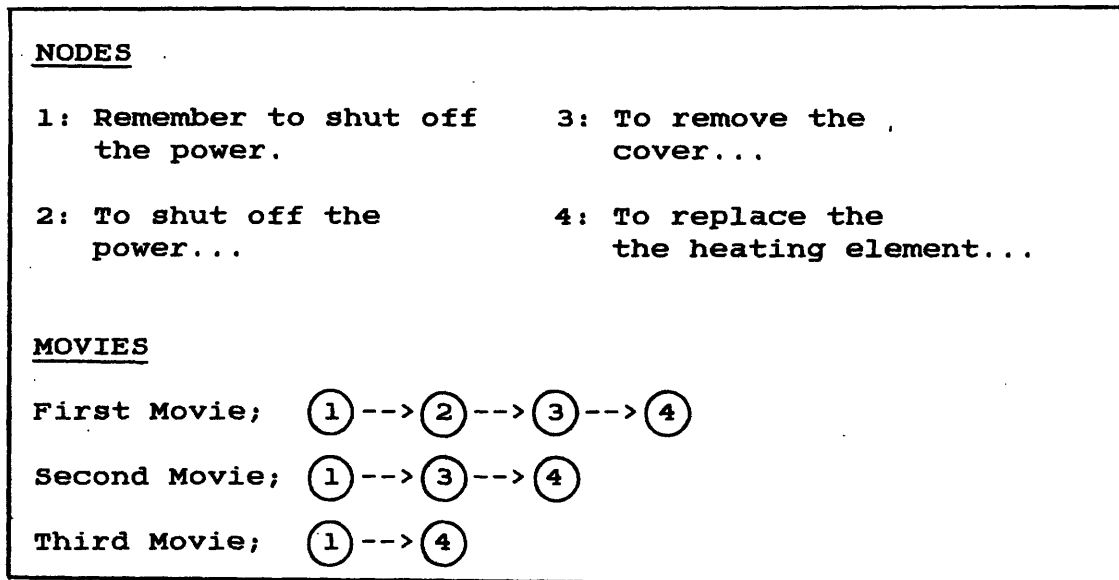

figure 1

The other attribute describes the content of the node in terms of how theoretical it is. One set of nodes may simply explain step by step how to remove and replace the heating element. Another set explains the function of the element and how it relates to the parts surrounding it, showing the theory and giving the user the freedom to infer the specifics. The

experience threshold and the theory level are stored by the computer for each user.

Sequences are classified according to their presentational style. Styles include live footage, animation, diagrams, closeups, registered cutaways, and dynamic computer generated overlays. Overlays are helpful in illustrating logic levels in a circuit or some other dynamic phenomena in a static framework. For a given node two or more styles are available. For example one style, say animation, includes the Pink Panther explaining how to perform a task. Whereas another style is a series of registered cutaways, each one showing an assembly with one more step performed- parts are peeled away one at a time.

Since a viewer may prefer one style over another, both the style and the content of a movie are personalized. The viewer's preference is stored by the computer and used in an algorithm to determine which sequences to show. One possible algorithm is a weighted sum of the user's preference and a random number, assuming that it's best to present a variety of material but generally weighted in the direction of the user's preference. Description of nodes, sequences, and user preferences can be expanded to allow for more complex algorithms, if such descriptors prove practical and benificial. Styles may be expanded to include alternate ways of explaining concepts. Also, altering selection algorithms may prove to be an interesting way to study the effectiveness of different types of

educational movies.

Preferences are at first set at random to encourage interaction. The user then adjusts these levels to appropriate values. The user description stored in the computer is not a convergent model. Approximations of user preferences are not successively closer, ultimately causing the computer to make a perfect movie for the individual. This is so because human interest and curiosity are non-deterministic, they change unpredictably. Hence interaction is always available. The system is then exciting and interesting to use, drawing the user actively into learning because of its immediacy and responsiveness. Learning becomes a participative process with a wide variety of raw material to explore; presented in a stimulating fashion.

The movie manual is not intended to replace a human teacher. There are aspects of interpersonal communication, important to teaching, that no machine can ever simulate. However as a reference, or in the absence of a human teacher, the movie manual certainly serves a purpose.

Since the movie manual is intended to grow with the user it is his private possession: as a notebook or diary belongs to a person. It is envisioned as part of a tool box- an interface between the human, a knowledge base, and the equipment to be fixed. The movie manual's power as a learning system is related to its controlled variability and ultimately, to its personal knowledge of the user. It will come to know his skills and

abilities, albeit quantized in some rudimentary way. This knowledge of the user is intended only for use in the movie manual. Subverting the user model for any other purpose, without the user's permission, would be a misuse of this technology and a disservice to the student.

In sum, by classifying material on the videodisc and by compiling statistics about individual users, the movie manual can tailor its content and delivery to the student. It takes cues from the student as a teacher would. The manual makes guesses about how to pace its delivery based on interaction with the student and on knowledge of how much the student has seen of the material in the system.

Altering the content of the movie manual is another aspect of personalization. At present, a user can write notations on the screen to become part of his manual. In the future he should be able to define new nodes and reconstruct the movie manual database. Updating the manual will be a function of using it. The movie manual will become a read/write system where the student learns from the movie manual and it learns from the student.

INTERACTION

Interaction can modify the movie along four axes: content, style, subject, and time. The movie may be modified at any point- it is interactive. For this reason maintaining continuity is important. If the screen went dark every time the user interacted the distance between the user and the movie would be increased. Cutting to a closeup should be as easy as asking or pushing a button while watching the movie. If it involved a break in continuity the movie manual would be more of a branched learning system where interaction would be unnatural, even distracting.

Content and style may be altered by modifying the personalization parameters. In addition they are immediately changeable when appropriate. For instance on some nodes, sequences can be requested which show a closeup, a diagram, or an exploded view. It is also possible to select nodes about the tool currently being used or to ask for more detailed information- an expansion of the current node.

With respect to time the user can pause the movie or play it in slow motion forwards or backwards at any point. The power of the manual as a simulator is enhanced with this ability. The user can watch a part functioning or breaking in slow motion, reversing and pausing the player to observe specific details. One advantage of the videodisc is that the part might normally be expensive or inaccessible. Simulation on a videodisc involves

shooting a movie of the part once. The alternative is to create
an interactive knowledge-based system to simulate and display
functioning and failure modes as a flight simulator would- an
expensive proposition. Even if such a system were required the
videodisc could significantly reduce its complexity. Display
need not be computed in real time since it would be stored on
the videodisc. The user can then cut between live footage of the
part breaking and a computer-animated finite element analysis of
the failure while slowing, pausing, and reversing the action.

Another aspect of interaction in time allows the viewer to
alter his location in the movie. Three functions are neccessary:
"quit", "replay", and "skip". "Quit" aborts the present movie
and allows the viewer to chose a new one. "Repeat" repeats the
current sequence. This allows the viewer to watch the same
sequence over again if he misses something. "Skip" either skips
forward or backward some number of sequences in the movie. This
gives the viewer the ability to review large parts of the movie
with new content and style parameters so that the new
presentation will different from the last. If a student does not
understand something it is better to provide an alternate
explanation than to repeat the old one. Finally the user can
move forward in the movie in case it becomes tedious,
extrapolating with the present parameters.

These controls raise the issue of one's sense of place
within the movie. It may be useful to provide a map or overview

of the entire movie manual, or parts of it, and to indicate position within that map. A graphical display of the semantic net is one possibility; perhaps a thumb index on the side of the screen or a storyboard. These questions merit future investigation.

The subject of a movie is presently chosen via a table of contents and remains fixed for the duration of the movie. Because of this the database is fairly simple to construct; it contains little knowledge of what topic a node covers. If the database were expanded to an expert knowledge base or if each node had a set of relational keywords, the movie manual would be able to respond to queries. Then it becomes a "conversational" movie which can intelligently answer questions. This is the direction future research on movie manuals should follow.

The man-machine interface can be constructed a number of ways. Currently, graphical displays are presented on a touch sensitive television screen. Buttons are overlayed on the videodisc image by a computer controlled television special effects generator. They are always present and labelled but not always activated. When a button becomes enabled it is filled in with color. The central portion of the button is made clear when touched to provide feedback to the viewer. Since buttons are labelled with text their function is easily comprehended. It takes only a few minutes to learn how to use the movie manual. An advantage of the touch sensitive screen as an interface is that it draws the user into the movie manual, touching things

and being actively involved with the presentation. However with one's hands full of tools and covered with grease, touch sensitive screens may not prove to be entirely practical. They also lack the depth and flexibility neccessary for wide bandwith communication between the viewer and the movie manual. Speech recognition may prove to be a better solution. However a nice aspect of visual interacton is that it avoids the natural language recognition problem. Graphical display is powerful in that it allows pointing. For instance, the computer can graphically display an assembly. As the viewer looks or points at one part it grows in size and detail until he requests a movie which is presented at the corresponding level of detail. This could be achieved with a touch sensitive screen, a three dimensional sensor, or an eye tracker as a pointer. An interactive work table has also been suggested that projects the video image as well as senses, through shape recognition, objects and tools placed upon it. Methods of communicating with the movie manual have yet to be fully explored.

In addition to making movies the movie manual allows the user to create or modify them. Notations, written or spoken, can be added to the movie. It should be possible to review the footage on the videodisc and author new movies, modifying the movie manual database. One method of accomplishing this involves presenting a key frame for each node to the viewer. A key frame is one frame that graphically embodies the concept of a node.

The viewer creates a movie by touching interesting key frames and arranging them into a storyboard on the screen. This problem is related to the mapping question.

Key frames can also be presented sequentially to give the user a quick glimpse of the material in the movie manual. They can be shown from the table of contents as subchapter headings or independantly as an index. A user may peruse through them like flipping through pages in a book, stopping at one and requesting the full sequence.

Finally the movie manual can function as a spatially oriented information retrieval system. For instance, touching a part on the screen may allow the viewer to order the part or find its stock number. In addition text can be be revealed to provide in depth information about the part. Items can be located by maintaining a database describing their position on the screen. Then they can become ports through which the movie can be redirected.

IMPLEMENTATION

HARDWARE

Video is played from one videodisc player while a second videodisc player searches to the next sequence to be played. A computer controlled special effects generator is used to cut from player to player in an A-B roll arrangement. In this manner visual continuity is always preserved- the screen never goes black. Both videodiscs contain the same material. The special effects generator is also used to key computer graphics from a Ramtek frame buffer over the video for the table of contents and for generation of buttons. Videodiscs have two audio channels each of which may contain different audio, one for sync sound- another for wild sound. Separate languages may also be placed on each track. Hence a stereo sound system is useful. A transparent touch panel is used as an interface to the Perkin-Elmer 3220 processor, a 32 bit minicomputer, which is being used as a development system. Future movie manual systems might include speech input and output systems or more elaborate sensing devices.

TABLE OF CONTENTS

To watch a movie the viewer sits in front of a touch sensitive television screen and selects a topic from a table of

contents. The movie manual currently uses discs about bicycle repair. Hence the table of contents is a picture of a bicycle. When parts of the bicycle that correspond to an available movie are touched they light up in red. There may be more than one movie for a particular part, i.e. assembly and disassembly. Text is displayed that tells the name of the part and the name of the movies about that part. The user can get a quick overview of movie topics, without the visual distraction of the text changing, by rubbing his finger over the screen. To find the name of the part or to see what particular movies are available the viewer can lift his finger from the screen or just tap the part. Touching the text will start the movie.
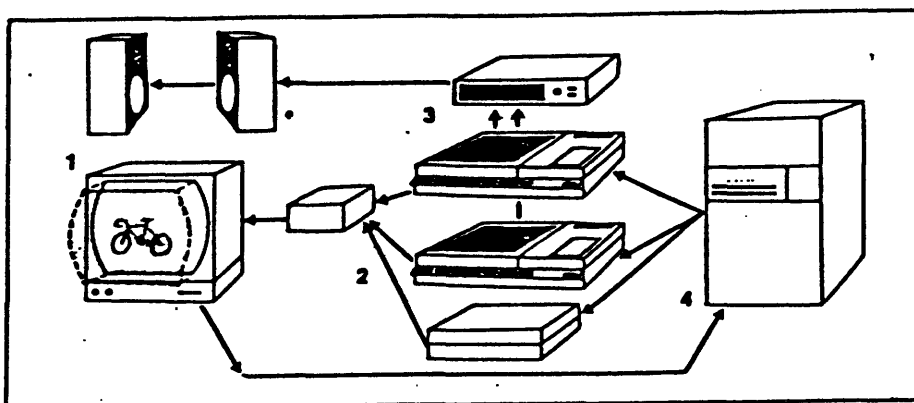
HARDWARE CONFIGURATION

figure 2

(1) User station with monitor, transparent touch panel, and speakers.
(2) Video generation: two disc players, Ramtek graphic display, video switcher and key.
(3) Sound generation: audio amplifier, and discs.
(4) System management: Perkin-Elmer 3220 processor.

PLAYER

There are three concurrent processes operating when the movie manual runs; the input process, the movie player, and the videodisc players. The input process handles user input and signals interaction requests, through a standard interface, to the movie player. It also updates the user data when new experience, theory, or style parameters are entered by the user. Finally, when signalled by the movie player and given the currently playing sequence, the input process will read the movie manual database to see what options are available. These options consist of alternate sequences or nodes that the user may want to play. The input process makes the user aware of these options.

The movie player makes the movie. It guesses what sequence to play next, queues that sequence on the videodisc player that is not playing, and then cuts to the next sequence when the current sequence is done. Queuing is neccessary because the worst case search time on the videodisc is three seconds, enough to interrupt the continuity of the movie with annoying pauses. The movie player continues alternating between videodiscs until the movie is done or until an asynchronous interaction signal from the input processor interrupts it. The movie player processes the interaction request and then returns to playing the movie.

The code that guesses what to play next attempts to

personalize the content of the movie. It reads the movie manual
database to find out what nodes and sequences are available and
then reads the user data to get parameters for deciding which
sequence to play. After chosing, the movie player updates the
user data to to keep track of how many times the user has seen
each node and sequence.

MOVIE PLAYER

```
on interaction begin;
     call interaction_handler;
end;

do while movie_is_not_finished;
     guess what to play next and update user data;
     queue next sequence;
     wait for this sequence to finish;
     play next sequence;
end;
```

The input process and the player run in the main frame.
These two processes can be combined into one if processing the
input does not require much computation in parallel with playing
the movie. The input can be handled while the movie player is
waiting for the current sequence to finish. The player is often
idle since the videodisc player takes care of most of the
details once a request has been passed. Also, input signals can
pause the movie, halting further movie processing while input is
being processed. Since reading the touch sensitive screen and
determining which button was hit requires little computation it
is not actually neccessary at this point to have separate
processes. However future versions of the movie manual, with
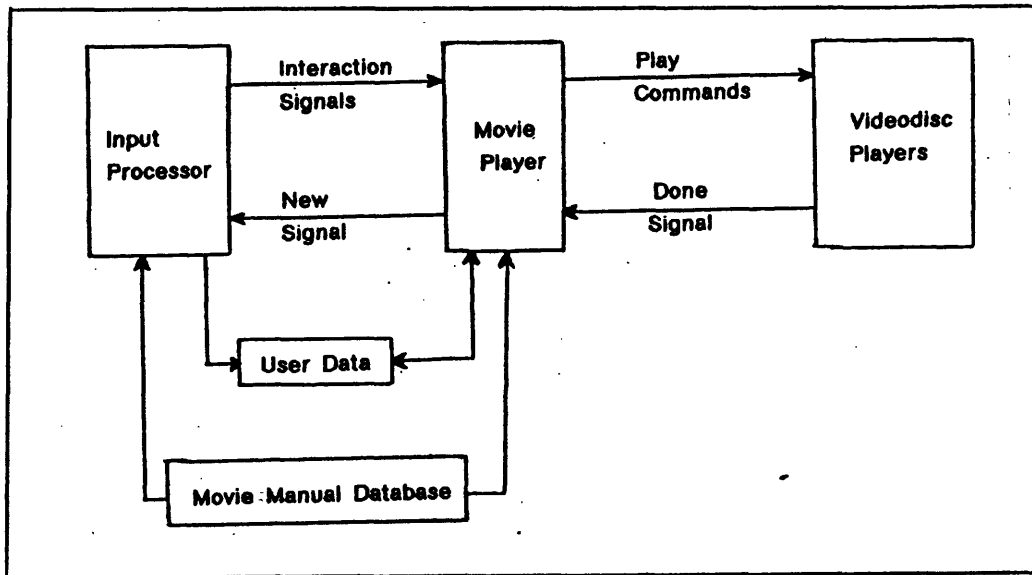more sophisticated interaction, will require dual processes.

MOVIE PLAYER SYSTEM

figure 3

The videodisc player contains a microprocessor. Programs and requests may be passed from the mainframe. The videodisc micro will search to a given frame, it will play at normal speed, or go slow forward or slow reverse from any specified frame to another. Finally the videodisc player stacks commands and will signal the mainframe that it is done when the command stack is emptied.

The videodiscs and switcher are interfaced to the mainframe as one. There is a visible disc- the one currently punched up on the switcher, and an invisible disc- the other disc. When given a play command the videodisc interface will check to see if the frame to play from (or one frame in either direction) is queued up on either player. If it is then it will cut to that disc and

play. The queue routine checks that the next frame does not start where the first one ends. Thus if one sequence ends where the next one starts, the movie player will not needlessly queue or switch discs.

Since the videodisc players only play sound at the normal speed, 30 frames per second, sound is played from the invisible disc when the visible disc plays at any other speed or when the sequence to play does not contain synchronous sound. The search request is stacked on the invisible player so that when it is done playing the sound it will do the search. In addition to being played from the invisible disc, sound could be played from an auxilliary sound generation device.

The movie player can queue and play a number of different types of sequences. A "sync" sequence is played at normal speed with or without sound. The sound may be synchronous or wild. A "slow" sequence can be played at a specified slow speed forwards or backwards with or without sound. "Slow" and "sync" sequences are shown for the duration of the sound and video. If the video ends first, the player will still frame on the last frame while the sound continues to play. Otherwise the sequence ends when the video ends. A "still" sequence will show a still frame with or without sound. If the still has no sound it will be shown for a specified number of seconds otherwise it is shown while the sound is playing.

Alternate modes of viewing material on the videodisc are neccessary to expand the viewing time of the 54,000 frames of raw material. Although 54,000 frames are a great number of still frames, they compromise only half an hour of video. Since presenting dynamic material is one of the advantages of the videodisc it is nice to present frames at rates slower than 30 per second and still create the perception of motion. There are a few ways to do this. Two of them prove to be practical. In addition computer generated animation can be keyed over a still frame from the videodisc to create a dynamic image.

The first method is pixellation. Pixellation involves cutting back and forth quickly between two or three frames to

create the illusion of motion. The human perceptual system integrates the two frames and sees a blurred image of the sparsly-sampled motion. To display a pixellated image convienently the frames should be adjacent on the videodisc. None of the presently available discs contain pixellated material.

Another method involves playing back and forth over the same material: "chow-chow" mode. (Named after the Purina Cat Chow television commercial which used the same technique for a humorous effect). There are motions that are repetitive, such as squeezing brake levers or unscrewing a bolt. By playing the same footage of a repetition forwards and backwards a few times the illusion of a temporal progression is created, it appears that the bolt is being removed. "Chow-chow" mode may start at the first or the last frame in a sequence, and sound over is possible. If no sound is requested the action is shown for a specified number of seconds. When queuing a "chow-chow" sequence a program is down-loaded into the microprocessor in the videodisc player. The play command runs the program whichs steps the player forwards and backwards through the sequence. The micro instruction cycle time is slow enough that the sequence plays in slow motion; the effect appears to work better this way. There are other types of repetitive motions but they cannot be played forwards and backwards, they must play through like a film loop. This would require cutting back and forth from disc

to disc for each repetition. Because of the overhead involved

these types of repetitive motions are not practical to show

using the videodisc.

CONCLUSIONS

The movie manual and the human form a symbiotic relationship- they become partners in repair. The manual presents a personalized tutorial which can be modified immediately as the viewer's interest changes. Then the system may be driven by specific questions as they are developed. It provides information through a convenient interface so that a human can creatively use his intuition to develop a model or hypothesis based on his own information, the manual's knowledge, and observations of the real world. As an interface to the real world the manual can be part of an intelligent toolbox, incorporating features of automated testing and diagnostics. Finally the manual can serve as a simulator to provide a specific model or to allow the viewer to test a hypothesis by manipulating that model.

APPENDIX

DESCRIPTION OF MOVIE PLAYER CODE

Type movie to run the movie manual. This calls the bound segment movie. The top level routine in movie is mainmovie which parses the command line arguments. If there are no arguments mainmovie specifies the command line argument syntax. The first argument which is required is the name of the movie manual database. The other three arguments are optional. A prerecorded movie may be specified. Also the movie manual can be requested to record a movie. The movie is always recorded in a scratch file so that the player knows where to go when skipping backwards. Record mode simply renames the scratch file. Finally debug mode can be specified which enables the display of debugging information.

Mainmovie calls player which plays the movie. Player contains the player loop and interaction handlers. Initially player queues the first and second sequences. Then it enters the play loop. The play loop first calls play, playing the first sequence. Then it calls guessqueue, guessing what to play next. After guessing it calls queue, queuing the next sequence. Finally player calls wait, waiting for the current sequence to finish. Player continues in this loop until the movie ends. Then it drops out of the loop and waits for the final sequence to finish. An interaction signal will interrupt the loop. The interaction will be processed and the loop will be resumed. When

- 25 -

the final sequence is done or when break is hit, quit is called. Quit aborts and detaches the players and the switcher.

Play determines whether the sequence is sync, still, slow, or chow-chow and then calls playsync, playstill, playslow, or playrpt.

Queue checks whether the sequence to queue is a chow-chow mode. If so queue calls rptqueue and returns. Otherwise if the sequence to queue is not the first sequence queue checks to see if it begins within one frame of the end of the last sequence. If so nothing is queued. Also queue checks to see if the first sequence is being prequeued, if so the sequence is queued on the visible player instead of the invisible player.

Wait waits for one or both discs to finish depending on how many are playing. When the discs are done wait turns off the audio and restores the the videodiscs players to a known state.

DESCRIPTION OF MOVIE EDITOR CODE

The movie editor is a full screen editor. The screen is divided into two windows, the left and right. The left window displays local information about the current node and sequence being edited. The right window display global information about the node and the database, command descriptions, or special displays. At any level in the editor one may type a "?" and information about relevant commands will be displayed in the right window. All commands are two characters long. No carriage return is neccessary.

When the user enters play mode, a stack of nodes and sequences can be specified. The movie player can be called from play mode to play the stack as if it were a prerecorded movie.

Type medit to call the bound segment medit. Mainedit is the top level routine. Mainedit reads the database name which is a command line argument. If a new database is specified databaseinit is called to initialize the database and nodeinit is called to initialize the first node. Output and outrite are called to display information on the left and right of the screen. Commandloop is called which reads in editor commands and executes them.

REFERENCES

[1] Bolt, Richard A., "Spatial Data Management". MIT
      Architecture Machine Group, Cambridge MA.

[2] Daynes, Rod and Sharon Wheaton, "A Proposed Part Task
      Trainer for Flight Training". Proceedings. 1980 MIDCON
      Professional Program Session #10; Applications of
      Interactive Television and the Videodisc.

[3] Gregory, R.L., The Intelligent Eye. McGraw Hill, New York,
      1970.

[4] Hochberg, Julian and Virginia Brooks, "Perception of Motion
      Pictures". Handbook of Perception, Volume X. Academic
      Press 1978.

[5] Lippman, Andrew, "Movie-Maps: An Application of the Optical
      Videodisc to Computer Graphics", Proceedings. SIGGRAPH
      '80, Seventh Annual Conference of Computer Graphics
      and Interactive Techniques, 1980, Seattle, WA.

[6] Lippman, Andrew, "Movie Manuals: Personalized Cinema as an
      Instructional Partner". MIT Architecture Machine Group,
      Cambridge MA. ONR Proposal, 1980.

[7] Lorenz, Konrad Z., "Analogy as a Source of Knowledge".
      Science, July 14, 1974.

[8] Mayer, Ralph J., "Personalized Movies", Master's Thesis. MIT
      Architecture Machine Group, Cambridge MA. June 1977.

[9] Mohl, Robert, "The Interactive Movie Map: Surrogate Travel
      with the Aid of Dynamic Aerial Overviews". Proceedings.
      1980 MIDCON Professional Program Session #10;
      Applications of Interactive Television and the
      Videodisc.

[10] Mylopoulos, John, "An Overview of Knowledge Representation".
      Proceedings of the Workshop on Data Abstraction,
      Databases, and Conceptual Modelling, Association for
      Computing Machinery, 1981.

[11] Negroponte, Nicholas, "Books Without Pages". IEEE, 1979.

[12] Negroponte, Nicholas, "Ideosyncratic Systems: Personalized
      Movies as Instructional Aides to Maintenance and
      Repair". MIT Architecture Machine Group, Cambridge MA.
      ONR Report NN-100-1, March 1977.

[13] Negroponte, Nicholas, "The Impact of Optical Videodiscs on
     Filmaking". MIT Architecture Machine Group, Cambridge
     MA.

[14] Negroponte, Nicholas, "Media Technology: The Human
     Interface with Information Sciences", Proceedings of a
     Symposium for Senior Executives. Industrial Liaison-
     Program of MIT. 1980.

[15] Negroponte, Nicholas, "The Metaphysics of Television",
     Methodology of Interaction. Guedj et. al. eds.
     North-Holland Publishing Company, 1980.

[16] Nelson, Theodor H., Dream Machines. Hugo's Book Service,
     Chicago, 1974.

[17] Peterson, James L., "Petri Nets". Computing Surveys. Volume
     9, Number 3. Association for Computing Machinery, 1977.

[18] Weizenbaum, Joseph, Computer Power and Human Reason. W.H.
     Freeman and Company, San Francisco, 1976.

[19] Winston, Patrick H, "Learning and Reasoning By Analogy".
     Communications of the ACM, Volume 23, Number 12, 1980.
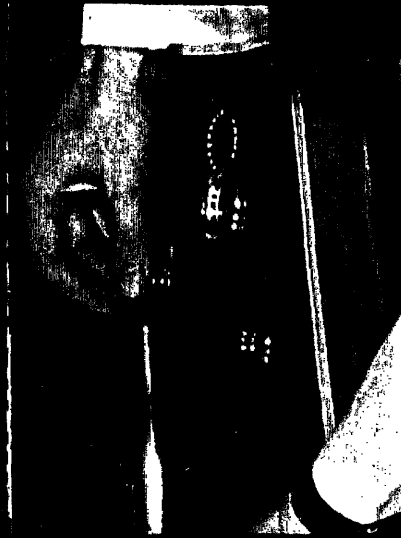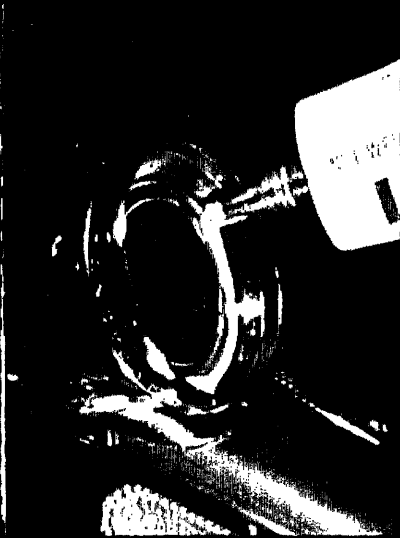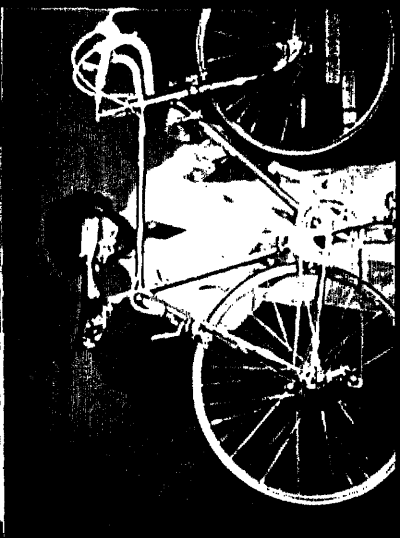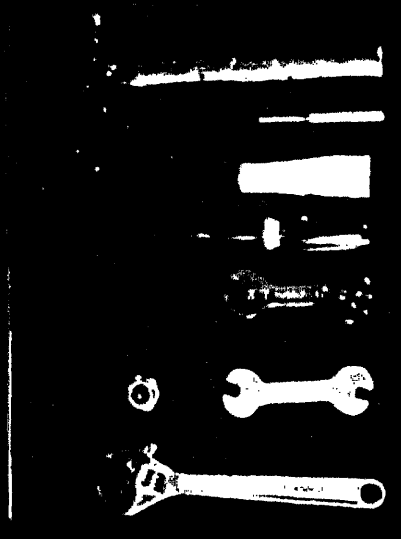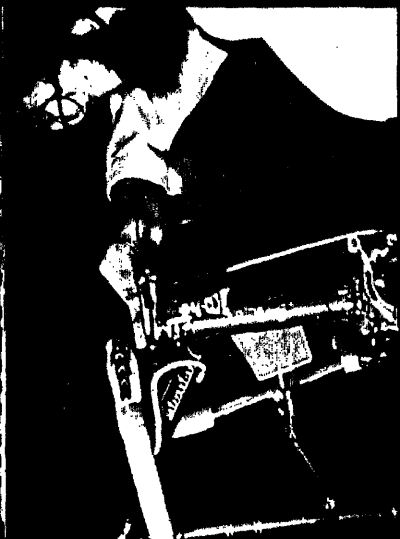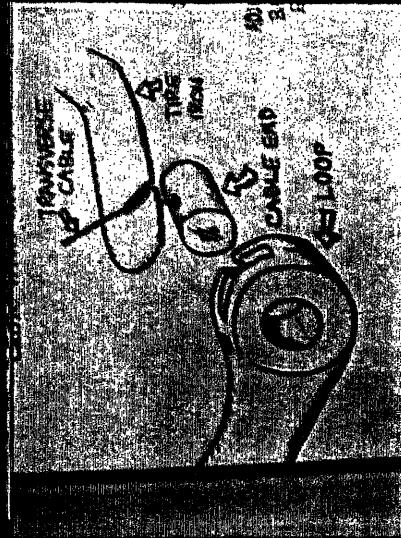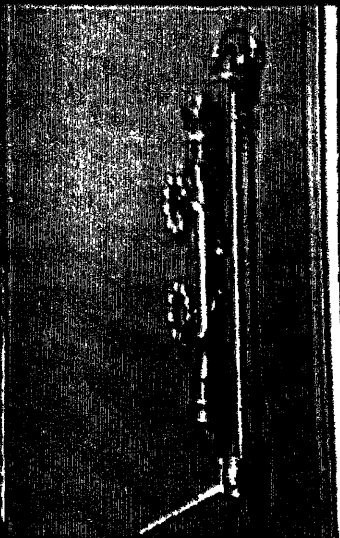
Different types of footage on the bicycle videodisc.
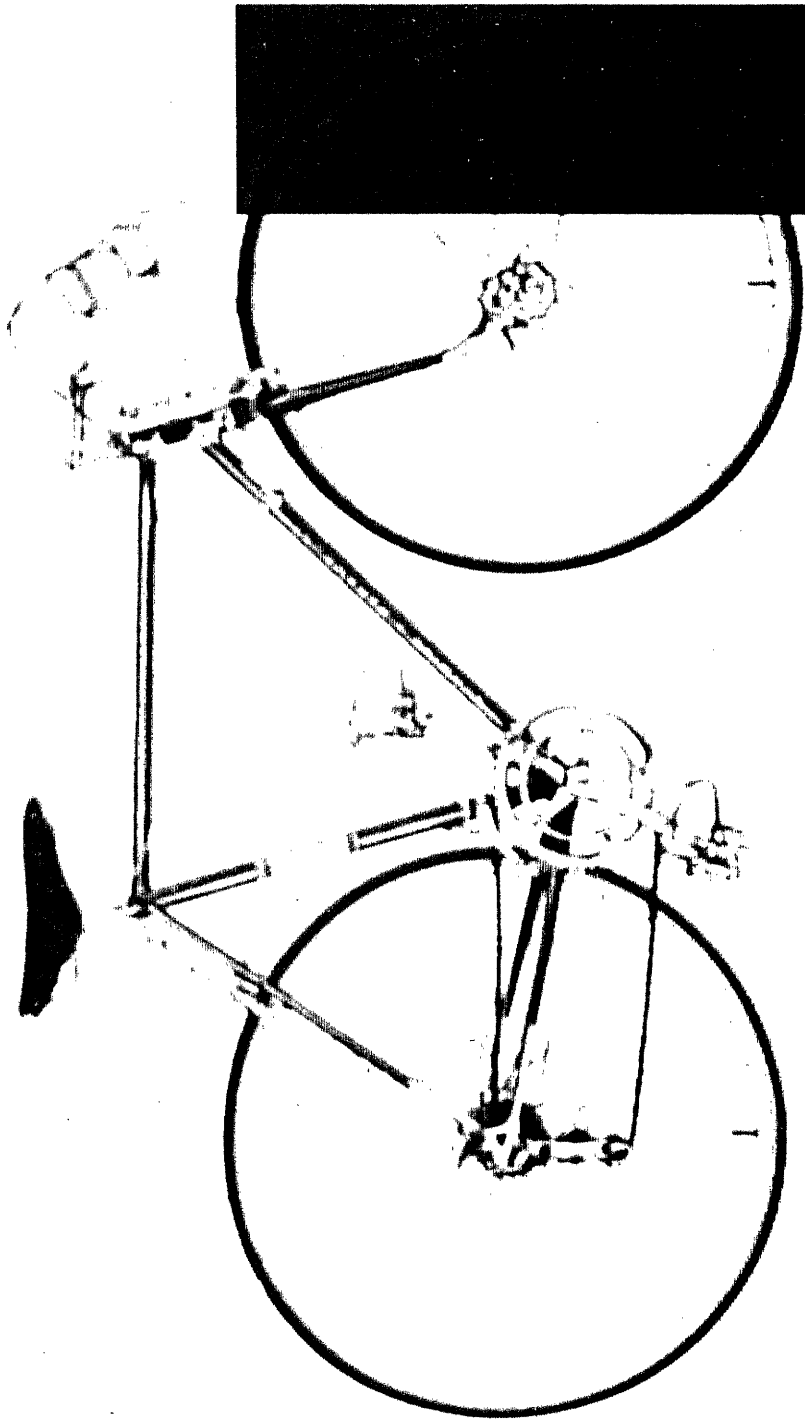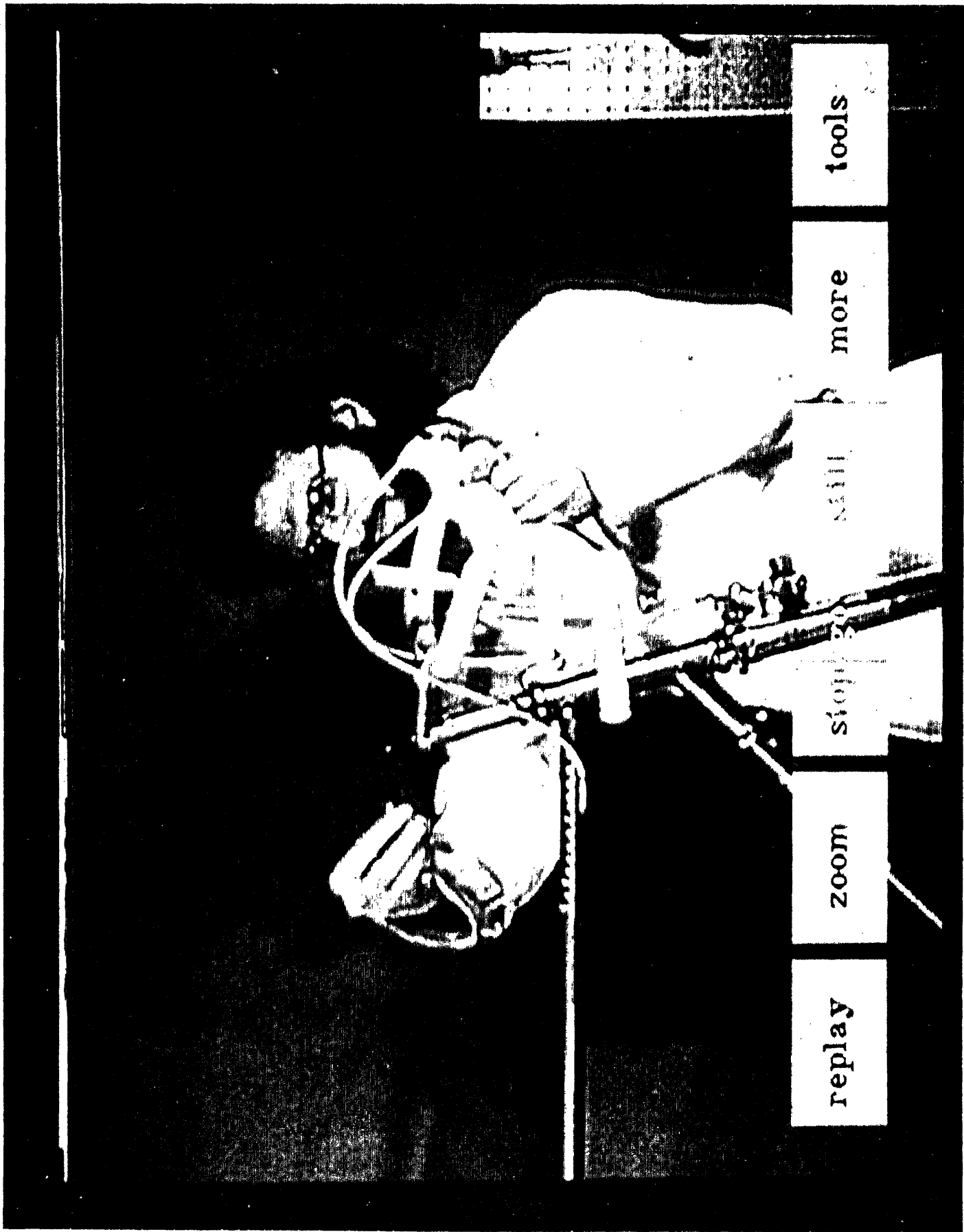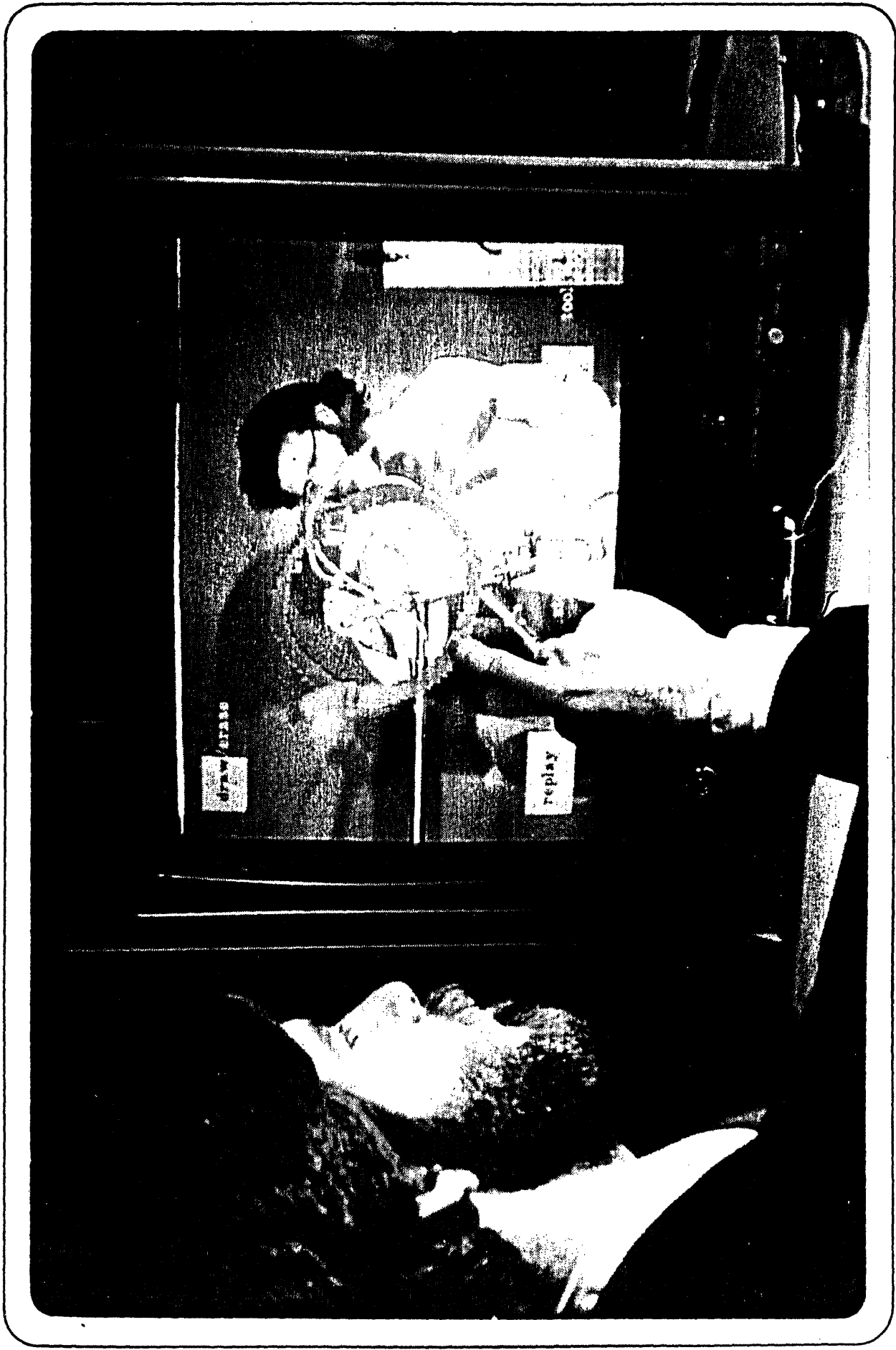
FRONT WHEEL -remove
-replace

Table of contents on the touch sensitive screen.

Buttons keyed over videodisc image on the touch sensitive screen.

Graphical notation being added to Movie Manual