May 24, 1996

A Java Movie Clip Annotation Tool

Timothy Chien

AUP Advisor: Glorianna I. Davenport
Interactive Cinema Group
MIT Media Lab

1.0 Abstract

Interactive cinema is a form of multimedia which employs real-time audience input and feedback to alter the playout of a story. In non-linear cinema, the basic cinematic elements of shot, sound, setting, sequence, character, and action are presented in some sequence dictated by a machine algorithm with little or no influence from the author. With the advent of the World Wide Web, virtually all forms of multimedia have become accessible from any machine with an appropriate network connection and Web browser. This paper is concerned with the design and implementation of a Java applet used to annotate a collection of movie clips in constructing a non-linear movie on the Web.

2.0 Introduction

Interactive cinema is an extension of traditional cinema which employs real-time audience input and feedback to alter the playout of the story. This dynamic system provides the user with a more personal cinematic experience and can be used as a tool to investigate human interpretations of the narrative, possibly making the experience “more enjoyable” in future sessions. An ongoing research effort in one particular area concerns non-linear cinema, an interactive story in which the basic cinematic elements of shot, sound, setting, sequence, character, and action are presented in some sequence dictated by a machine algorithm with little or no influence from the author. This algorithm, coupled with annotations of a collection of movie clips, form the engine of an interactive movie.

Past research such as ConText (Davenport, Murtaugh, 1995) has concentrated on annotating content with descriptors such as “who?”, “what?”, “when?”, and “where?”, associating a numeric value with each descriptor and assigning weights to each unit of content according to the sum of the associated descriptors. For example, when a movie clip is ready to be chosen and played, the current valid descriptors for this particular point in the story are matched to descriptors in a database of other movie clips. The clip that has the highest weighted value based on these descriptors will be chosen to be played next. A variation on this idea, and the theme of my research, is to include a sense of the author’s direction in the playout of the story.

The system will be implemented in Java, a recently developed programming language providing dynamic presentation on the World Wide Web through an appropriately configured browser such as Netscape Navigator 2.0. The hallmarks of Java are that it is object-oriented, distributed, robust, secure, multithreaded, and architecture neutral, making it ideal for building Internet applications. Java makes documents come “alive” through inline animations and real time audio, extending the Web for interactive movies, games, and educational tools.

The North End Project, an effort to develop a non-linear interactive movie incorporating visual guidance, was originally conceived by Natalia Tsarkova of the Interactive Cinema Group at the MIT Media Lab. The work is founded on the hypothesis that visual guidance enhances interactive video presentations. This guidance is achieved through the use of collages. A collage blends one in frame from one movie clip, one key frame from a second movie clip, and one key frame from a third movie clip. The main user interface screen shows the currently playing clip surrounded by three collages which represent three storylines from which the user may choose. This AUP focuses on an authoring tool for system selection and presentation of collage configurations from a database of movie clips. This work furthers ideas of selection used in ConText by (1) interleaving a forward moving story with contextual selections and (2) building collages of three potential sequences.

3.0 Overview of System Design

In the North End story, there are three levels of narrative which the system must deal with: a linear global story, a number of substories ("streams"), and a non-linear story similar to one produced by ConText. The global story follows a segment of the life of an aging North Ender, Billy. This author imposed global story line first introduces Billy, then takes the audience into some aspects of his life they may find enjoyable or disturbing. A number of substories detail these particular glimpses into his daily life and are less enforced by the author than the global story. The non-linear story is free from any order imposed by the author. This story is composed of movie clips which may depict interesting side events in the play-out of the global movie line. The author is only responsible for making the appropriate annotations.

The story at the highest level can be represented by a series of global story nodes (G1, G2, ..., Gn) and at the lowest level, by a tree of movie clips. **Figure 1** illustrates these representations. The global story consists of a series of points that the author wishes to bring out and imposes an order on the presentation of movie clips (e.g. some clips logically precede other clips). This ordering can be seen in the tree on the right side of **Figure 1**. The first row of clips corresponds to content that fits between G1 and G2. Similarly, the second row of clips corresponds to content that fits between G2 and G3. A number of movie clips are directly associated with each global story node. These global story node clips always begin and end a sequence of clips from one global story node to another.

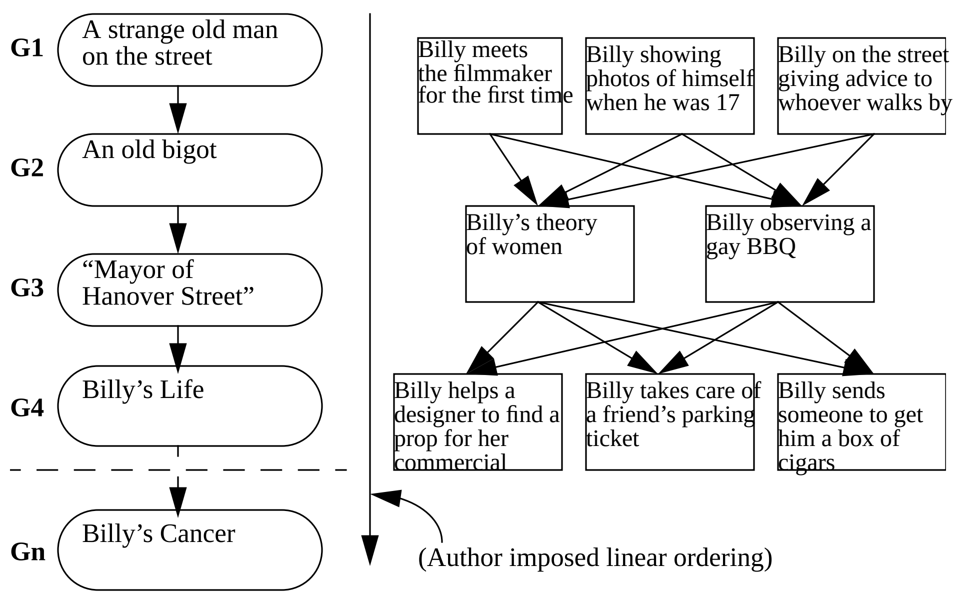

Fig. 1 Global Story/Movie Clip Tree

3.1 Story Playout and Search Algorithm

If G1, G2, ..., Gk are author defined global story states, and X is the currently playing clip, we attempt to find a logical path from Gn to Gn+1. This path is defined by “strong” links, clips that have been annotated with X (global story node clips), and “weak” links, clips that share at least two basic descriptions with X (ConText generated clips).

The algorithm to reach Gn+1 consists of the following steps:

1. Find all possible links (strong and weak).
2. Disregard links that have been previously traversed.
3. Disregard links in conflict with the global story. These links represent global points of n+2 or higher or are associated with global points of n+2 or higher.
4. With the remaining links, scale down the distance to weak links so that strong links are favored.
5. Repeat steps 1-4 for each remaining link for some author defined number of iterations.
6. Find two optimal paths to Gn+1 in this tree of links using breadth-first search.

Figure 2 depicts links found for X (Tsarkova 1995). **Figure 3** shows the resulting link tree.

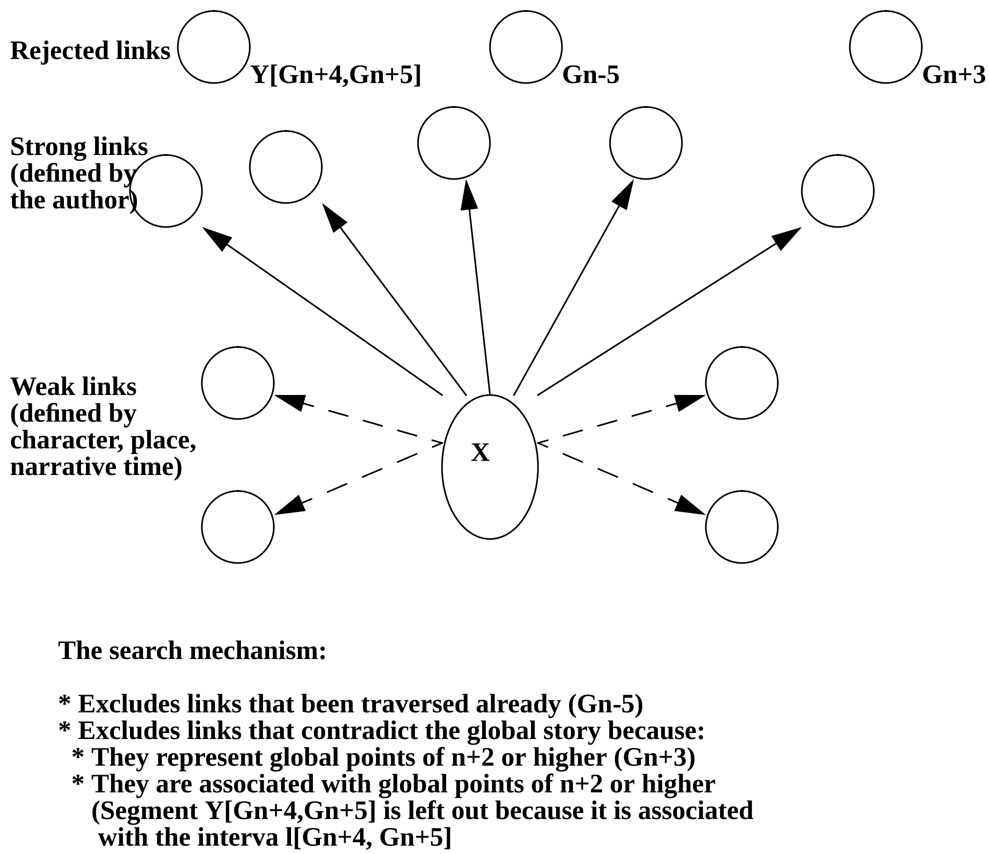

Fig. 2 Links for Segment X at Story Interval [Gn,Gn+1]

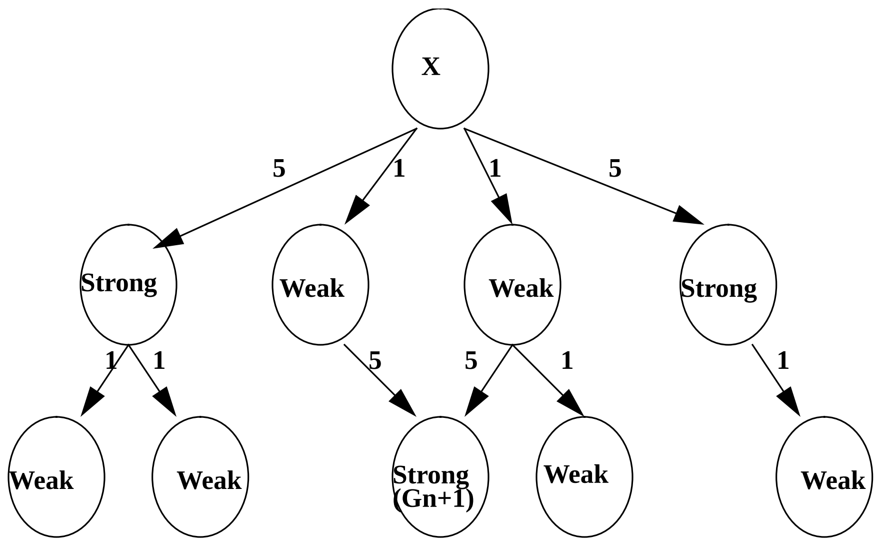

Fig. 3 Link Tree

3.2 Visual Guidance through Collages

In the North End movie, collages are used to represent the three story lines associated with the currently playing movie clip. Each collage consists of an in frame of a movie clip (the first frame), a key frame from the next clip in the sequence of clips generated by the search algorithm discussed above, and a key frame from the clip two steps away in the sequence. A key frame highlights some visual aspect of the clip that the author wishes to bring out. This form of machine-assisted direction refreshes the user's memory of certain movie clips and can help him/her choose the next storyline to follow. **Figure 4** portrays a sample collage.

Fig. 4 Sample Collage

4.0 Implementation

The basic structure of the Java movie clip annotation tool is simple. Based on Mike Murtaugh's Java Dexter (Murtaugh 1996), this movie clip annotation applet adds one level of indirection to Dexter by incorporating the notion of author-defined substories or "streams". The term "streams" refers to the nature of a sequence of clips to flow into other sequences of clips indiscriminately. Streams provide variation in choice of storylines so that similar sequences of clips are not presented to the user at the same time. The author is free to modify his/her set of streams prior to a user session.

4.1 User Interface

Figures 4a and 4b present the user interface of our system. In **Figure 4a**, Panel 1 depicts the initial interface setup with a movie clip playing in the bottom center of the screen and the three surrounding collages. Panel 2 shows what happens when a user intervenes the playing clip and selects collage 1. If the user does not intervene, the system automatically selects one of the collages. The selected collage breaks apart and the in frame of the selected collage moves to the position of the playing clip. In addition, one of the key frames in the selected collage assumes the position of the in frame and two new collages are generated. Panel 3 shows the final state of the screen when the new clip has begun playing. This graphic transformation "shows the viewer where the story is going, makes the viewer's input a part of the visual narrative, and makes a visually coherent transition between the two clips." (Tsarkova 1996) **Figure 4b** is an actual screen shot of the interface.

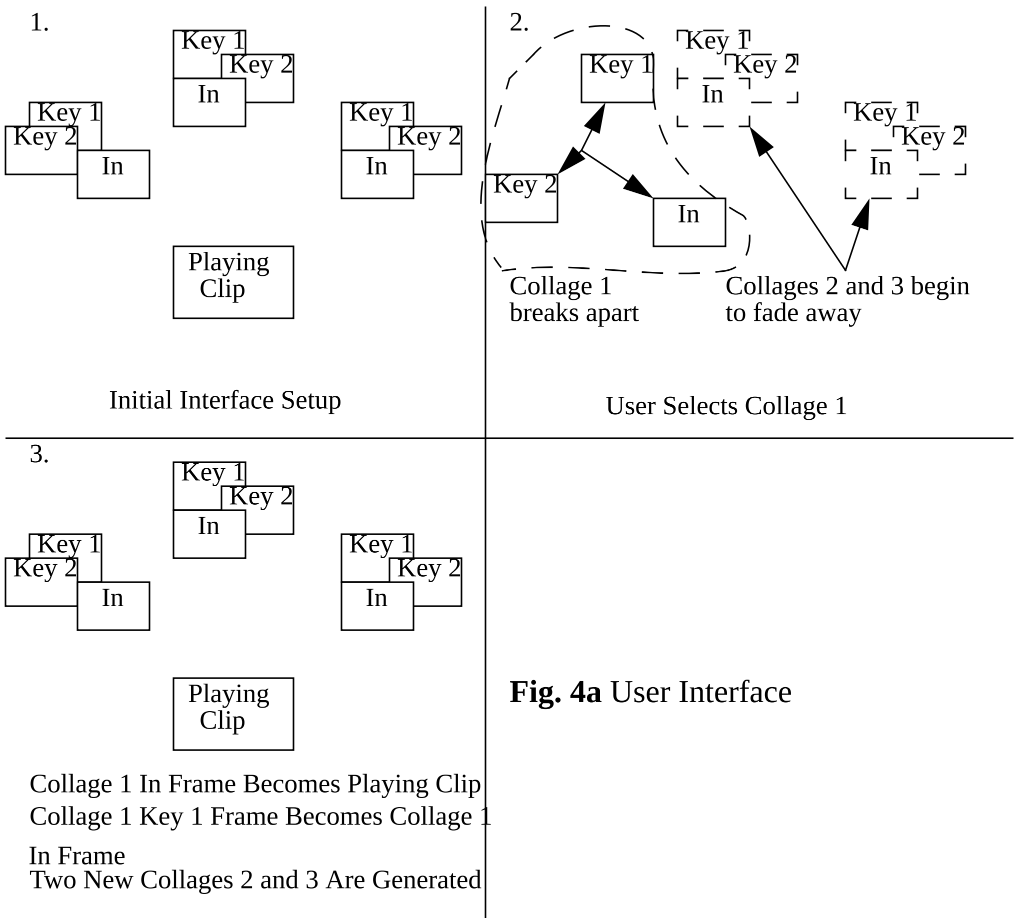

Fig. 4a User Interface

Fig. 4b Screen Shot of Interface

4.2 Data Structures

Figure 5 presents the data structures in our system. The smallest unit of content is the Dexter-like descriptor. These descriptors are associated with pieces of media (in this case, video clips) through a prominence rating ranging from 0 to 100 (e.g. the descriptor “Bocci” has a prominence of “90”). The governing unit of content is the stream since the author explicitly creates the streams prior to a story session. Each stream contains a predefined sequence of thematically linked movie clips that the author wishes to emphasize. A stream is simply a hashtable associating movie clip names with a vector of descriptors. The highest level of the story is maintained by a list of active streams and their associated prominence-in-story ratings ranging from 0 to 100 (e.g. the stream entitled “Billy” has a prominence-in-story rating of “90”). These ratings reflect the relevance of the sequence of clips at a particular point in the story.

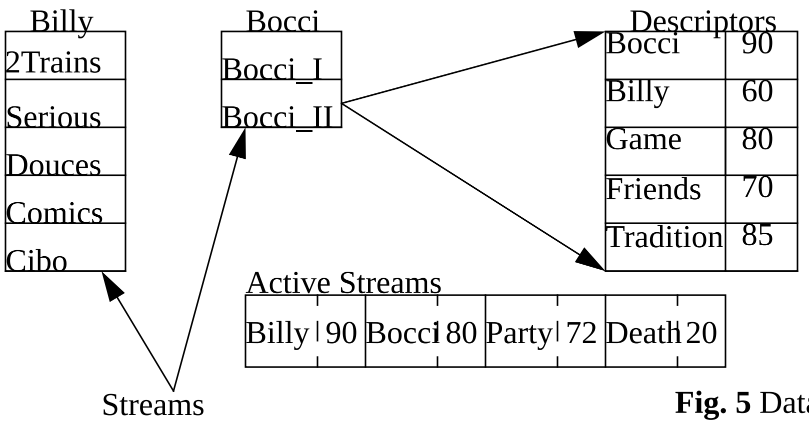

Fig. 5 Data Structures

4.3 Collage Construction

A key feature of this system is that collages can be created on the fly. Java threads and Netscape frames are used to accomplish this. When a collage is selected, a new key frame for the selected collage and two completely new collages must be generated. Both processes can be done while the movie clip plays in a separate frame of the browser. The new key frame for the selected collage is simply the next clip in the sequence previously generated by the search algorithm. Generating the two new collages is more difficult. The search algorithm, previously discussed in "Overview of System Design") proceeds as follows.

1. Generate all strong and weak links. The weak links are found by comparing descriptors. First, the descriptors for the new movie clip are retrieved. The descriptor's importance (%) in a particular movie clip can be found by dividing its prominence rating by the sum of all its descriptors' prominence ratings. A descriptor's importance is computed for each descriptor in the currently playing movie clip and is matched as closely as possible to the same descriptor's importance in each movie clip from each of the top three active streams according to some author-defined standard deviation. Clips from the top three active streams that match the best (e.g. most number of descriptors with best-matching descriptor importance percentages) are then organized into their respective streams. Strong links are found by retrieving clips annotated with the current global story step.
2. Disregard previously traversed links. We compare the clip titles found in Step 1 with a cached list of clip titles that have been presented already and filter out those that match.
3. Disregard links conflicting with the global story. Since the author prescribes a global story prior to the session, each clip contains an annotation of which step (or between which steps) in the global story the clip falls. If the clip is more than two steps away in the global story path, it is filtered.
4. In constructing the tree of links, scale down the distance to weak links so that strong links are favored.
5. For each link, repeat steps 1-4 for some author defined number of iterations.
6. Two optimal paths to the next global story step are found by breadth-first search.
7. For each path, the in frame of the first clip and the two key frames of the second and third clips are retrieved. These frames are then assembled into the two new collages and presented to the user.

4.4 Annotations

At present, the author must manually adjust the parameters of the annotations file to configure the story to his/her liking (**Appendix B**). Future work will make this process more user friendly (please see "Future Work" in "Conclusion").

4.5 Overview of System

Figure 6 provides an overview of the system. When the user accesses the North End Web page, the user interface applet and annotation tool are downloaded to the local machine and executed. The annotation tool establishes an HTTP connection to a Web server that contains a specified North End database file and reads the file into the local machine's main memory. The dashed "Session X File" ovals represent files for machine learning (please see "Future Work" in "Conclusion").

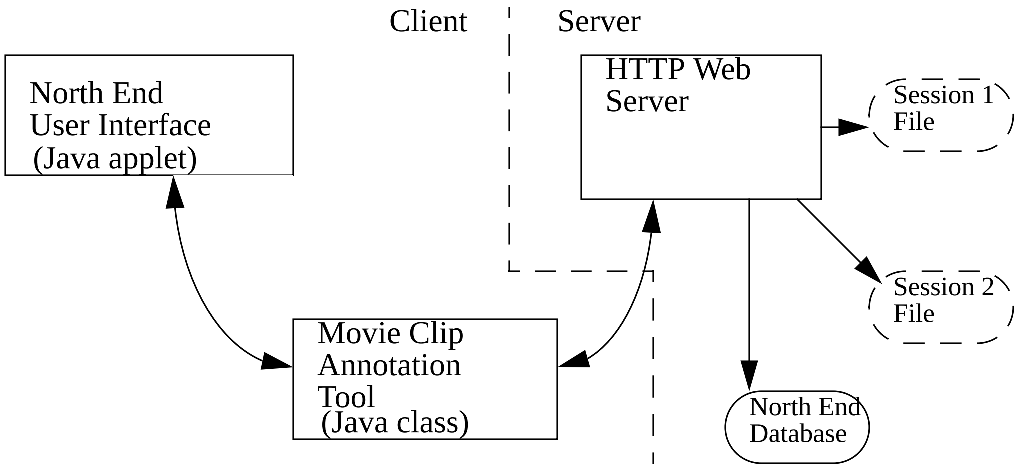

Fig. 6 System Overview Diagram

5.0 Conclusion

Efforts at delivering immersive, interactive environments are constant research topics. Visual guidance is an important component of any interactive multimedia work. Using collages to guide the user through an interactive movie session enhances the user's experience by providing a means to intervene gracefully. This specific implementation of an interactive movie on the World Wide Web is a useful experiment in availability; anyone with an appropriate browser and connection to the Internet can experience the North End story. It is the hope that collages can provide guidance in other forms of media other than interactive movies; online catalogs, educational applications, and virtual chat sessions can all be enhanced by using collages.

5.1 Future Work

With more time, there is a number of issues that should be investigated. The first priority is a user friendly author annotation interface which will allow any user with appropriate security clearance to modify the annotations database. The major limiting factor to this work is inherent in Java security. Java applets are only allowed to read, and not write, files on another Web server. It is the hope that Java will be extended in the near future to provide some form of secure authentication/authorization such as Kerberos. Audience tests should also be conducted to see if users do actually enjoy their experience more in a collage-driven environment rather than through a standard menu/button interface. In performing these audience tests, additional global stories should be constructed for the system. Audience responses to different global stories would be beneficial in forming characteristics of a well defined global story. When these responses are collected, a more ambitious project would be to investigate machine learning. As the user progresses through the session, the machine would record certain storylines that the user tended to pick and save this information for later sessions (in **Figure 6**, these are labelled “Session Files”). Upon collecting a number of session files, the machine would pick up certain characteristics of the story and be able to modify its behavior to favor user preferences (e.g. neglecting specific story lines and emphasizing others). Agents could then be created to record specific user preferences (e.g. a user may not want violent content to be shown at any time during the presentation and the agent records this information). The collage user interface and annotation tool should be regarded as stepping stones to a more robust system that has high availability, provides wide extensibility, and offers satisfying entertainment.

6.0 References

[1] Davenport, Glorianna and Michael Murtaugh, "ConText: Towards the Evolving Documentary", ACM Multimedia, 1995.

[2] Murtaugh, Michael, Dexter.java (source code), 1996.

[3] Tsarkova, Natalia, Transitions in Nonlinear Cinema (SM in Media Arts and Science Thesis Proposal), 1995.