Heuristics for Continuity Editing of Cinematic Computer Graphics Scenes

Kaveh Kardan Henri Casanova

University of Hawaii*

Abstract

We present a set of heuristics for editing footage of 3D computer graphics cinematic sequences into a coherent movie clip which obeys the conventions of continuity editing. Our approach mimics the decision processes of an editor assembling a clip out of filmed footage involving multiple camera setups. Given a set of stylistic rules, our software applies a number of heuristics to produce a final result satisfying those rules, as well as the fundamental rules of continuity editing. The main contribution of this paper is in the formulation of editing heuristics which take into account stylistic rules, enabling different edits of the same scene into cinematic clips of various styles. We demonstrate the use of these heuristics on three scenes taken from actual film clips.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; J.5 [Arts and Humanities]

Keywords: computer animation, virtual cinematography, film editing, film grammar, virtual characters, computational cinematics

1 Introduction

In computer graphics, the need to generate cinematic sequences is present in numerous applications. From narrative productions such as feature films and in-game cinematics to computer-assisted storytelling, pre-visualization for films, and 3D chat systems, there is a prevalent demand for means of automating the production of computer-generated cinematics. Even in simple 3D scenes, the process of creating and placing cameras to follow the action of a scene and then editing the resulting footage is a time-consuming one. And in fact, there are cases, such as cinematic playback of computer games actions, where it is not possible to perform these actions manually, as the "script" of the scene is not known ahead of time, and will vary from player to player and game to game.

While the work presented herein is relevant for all the aforementioned application domains, in this paper we choose to focus on applications that involve the generation of cinematic sequences to enable new gameplay and storytelling avenues. Such sequences could be generated from high level authorial or user input, leading to the creation of customized movie experiences. The fundamental element required for such new entertainment media is the ability to generate cinematics that follows the established conventions of cinema, including cinematography and editing, aiming for an immersive and engaging user experience. In this paper, we tap into these established conventions in the field of film editing and derive heuristics to help automate this process while leading to correct editing. Once these heuristics have been implemented as part of an automated cinematic sequence generation system, this system can then be used to investigate which rules and parameters lead to particular editing styles. For instance, the stylistic inputs to the system can be modified in an attempt to produce results that match historical and authorial styles.

The main contribution of this paper is the explicit enumeration of editing heuristics and stylistic rules resulting in various possible edits of a given scene, each of which is visually acceptable yet stylistically different. We have implemented these heuristics and have investigated their ability to produce acceptable cinematic sequences based on a wide range of test cases. In this paper we show obtained results for three dialogue scenes that correspond to actual film clips.

The rest of this paper is organized as follows. In Section 2 we review related work. In Section 3 we detail the rationale and the design for our approach. In Section 4 we describe and justify a series of editing heuristics. In Section 5 we describe four stylistic rules that are used as parameters to the editing heuristics. Section Section 6 briefly describes our implementation. Section 7 presents evaluation results. Finally, Section 8 concludes the paper with a summary of our results and a discussion of future work directions.

2 Related Work

Texts on cinematography and editing such as [Murch] [Mascelli] [Arijon] [Katz] provide information in terms of generalized rules, or as formulae (idioms) which apply in specific cases. Their work is rendered difficult by the fact that the art of editing is not merely one of applying a set of standardized rules to the editing process of a scene or film. Like any art form, the art comes from understanding and then consciously breaking rules for effect where desired.

*e-mail: {kaveh | henric}@hawaii.edu

[Christianson] and [He] implement Arijon's idioms regarding camera placement into a software framework for doing virtual cinematography of 3D scenes. Editing in these idioms is hard coded in terms of actor motion into, out of, and within the frame.

The placement of virtual cameras within 3D scenes has been addressed in [Bares1999] [Bares2000] [Li] [Oliveros]. [Friedman] and [Elson] apply knowledge bases in reproducing the filmmaking process, but are more concerned with camera placement and do not address the issue of overlapping actions, which require editing decisions. [Tomlinson] implements the camera as an agent within the scene, and handles editing by mathematically weighting which actor should be shown, based on the actor's importance and recent screen presence.

We also limit ourselves to situations where a scene of events already exists in its entirety. Much work has been done in cinematography and editing of real-time systems [Amerson] [Halper] [Cozic], such as games in the process of being played. We do not address this situation, but rather take advantage of the extra knowledge provided by the predetermined events in order to produce cinematically correct edited clips.

3 Design

In this paper, we limit the scope of the stagings of our 3D computer graphics scenes to static scenes of dialog between arbitrary numbers of actors. Dialog scenes are the most common scenes in narrative cinema, the resulting edits are relatively easy to judge aesthetically, and there is a large body of work on how to edit such scenes. Furthermore, since such scenes contain a fixed audio track of the dialog, it prevents the editing system from retiming events or shuffling them back and forth in time. Though a real editor might well make use of such manipulations, our system strictly adheres to the fixed audio track.

3.1 Overall Approach

In traditional filmmaking, the creation of a film proceeds along a pipeline involving preproduction, production, and post-production. For our purposes, the roles of interest are those of the director, the cinematographer, and the editor. Fig. 1 shows how we represent each of these roles as modules in our system.

The director module places the actors around the stage (also known as blocking) and produces a shot list. This shot list indicates what the director considers worth filming in the events that unfold in the scene. The director's shot list may contain gaps in time between shots as well as temporal overlaps between shots. The cinematography system then generates all possible camera setups based on actor grouping and lines of action. The design of the cinematography system is fully explained in a previous paper [Kardan]. The editor generates an edited shot list which is then rendered based on the appropriate camera setups.

In this paper, we describe the editing step of the process, which involves taking the director's initial shot list, a number of stylistic rules, the available camera setups, and producing a final edited shot list. This list indicates which camera setup is to be used at each point in time.

Continuity editing is a style of film editing developed in Hollywood early in the 20th century. By 1920, the principles of continuity editing had been established [Bordwell] and were prevalent in the productions of the film studios. By the 1930's, the rules of continuity editing had been refined to the extent that they have remained essentially unchanged to this date. The goal of continuity editing is to make the edits in a cinematic sequence as invisible as possible. To accomplish this, heuristics have been developed by film editors to determine where to cut from one shot to another in order to minimize the audience's awareness of the transition, and to not break their immersion in the story. Continuity editing and its heuristics provide the basis for our work.

3.2 Scene Representation

The scene is the top level data structure in our system. A scene takes place in a single location, with a number of actors. Scenes are made up of events, such as one actor talking to or looking at one or more actors. In this work we only consider static scenes, in which neither the camera(s) nor the actors are moving. Events are the atomic units of acting in our system. An event is represented as a continuous interval of time, with a start and end time. A scene then consists of a list of possibly overlapping events. In addition to temporal information, events may be assigned importance/intensity values, which can be used by the editing system to decide which of several overlapping events' shot should be used, and which framing to use for a given shot. The editing process, in our system, consists in converting the list of events in a scene into an edited shot list.

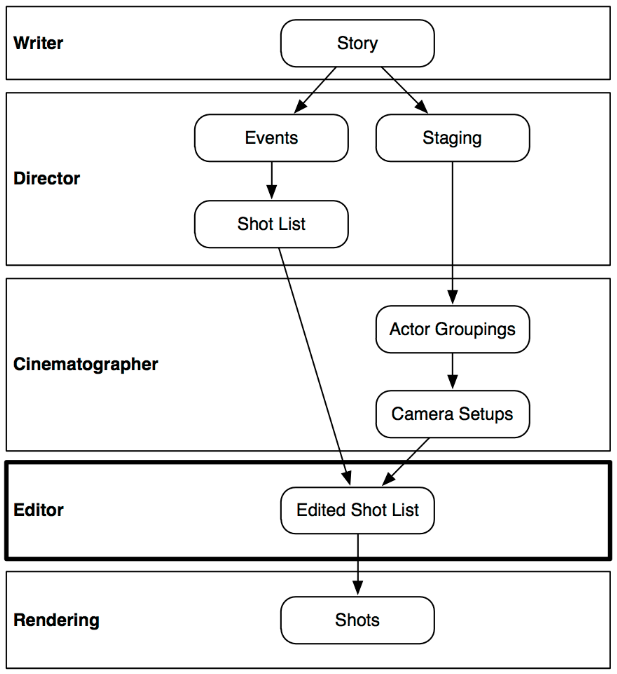

Figure 1. Data flow for cinematics creation.

4 Editing Heuristics

Our approach is to mimic the decision process of a real editor while working on a scene. We treat the input to the editor module as a set of shots (the shot list) provided by the director. We make these heuristics explicit, rather than relying on more abstract mathematical weightings or constraint systems. This allows us to tweak the rules which feed into the heuristics and speak about the decisions made by the system in producing the final edit. In this way, our system is somewhat similar in goal to an expert system, in that we want to be able to look and what it did and understand why it made the choices it did.

```
shot
- camera-setup
- start-time
- end-time
- target-actors
- event
```

```
camera-setup
- actor-grouping
- target-actors
- target-shapes
- framing
- focal-length
- swing
- tilt
```

```
scene-event
- start-time
- end-time
- importance
- intensity
- acting-actors
- reacting-actors
- camera-setups
```

Figure 2. Shot, camera, and event data structures.

Each shot contains data as shown in Fig. 2. The editor's task is to reduce these shots to a single, temporally continuous, sequence of shots along with an associated camera setup for each shot. We describe here eight editing heuristics that are used in movie editing and that we have implemented in our system.

4.1 Selecting Shot Camera Setups

The editor selects a camera setup for each shot in the timeline. The selection is made based on specified stylistic rules having to do with preferred camera framings, which can be modified by the intensity of the event. One stylistic rule, for example, is whether each scene should start with an establishing shot which shows all the actors in the scene. Other rules can specify that one framing should not follow another in subsequent shots of an actor. For example, in the editing of the classical Hollywood period (1930-1950), a closeup of an actor could not follow a long shot of the actor without first going through an intervening medium shot.

4.2 Flattening Shot List

When two or more shots overlap in time, the editor needs to decide which shot to use in the final edit. This is done using the importance associated with the events of the shots. The shot with the highest importance is the one selected. This can result in the other shots being entirely removed from the timeline, or clipped (trimmed or split) by the more important shot.

4.3 Filling Shot List Gaps

When there is a gap between two adjacent shots, the editor needs to determine how to fill this gap. A value in the range [0, 1] is used to determine the cut point between the two shots in the gap. A value of 0.5 would equally extend both shots to fill the gap. In dialog scenes, the choice of this value has a strong impact on the feeling of the final edit. It is unusual to stay on the shot of a person who has finished speaking, unless there are non-verbal motivations for doing so. The natural tendency is to cut to the listener, who, perhaps after a pause, will reply. In most cases, this value tends to be small, allowing the shot of the listener to fill most if not all of the gap.

4.4 Creating L-cuts

The choice of a cut point between speaking actors is not limited to the pause between speaking events. Often, an editor will cut away from the speaker to the listener before the former is done speaking. The second shot in this instance will show the second actor listening, then replying. This sort of cut, sometimes referred to as an L-cut due to the positioning of the relevant video and audio tracks in a timeline, is commonly used, resulting in a smoother -- less noticeable -- transition between shots as not both the audio and visuals are transitioning simultaneously.

4.5 Removing Quick Shots

Shot length is one of the commonly used metrics when determining trends in film style and editing, since it is easily measured and analyzed. We provide the notion of a minimum shot length to handle cases where a shot's duration, due to the duration of the associated event, or due to timing changes made by other editing heuristics, is too small to warrant a cut in the final shot list. Such a shot would be jarring, flashing onto and off the screen at a faster pace than fits the desired editing style. As faster and more frenetic editing has reduced shot lengths in films over the years, audiences have become used to making sense of more disjointed imagery, but there is still a need for preventing cuts which are inappropriately quick for a given style or pacing.

When a shot duration is found to be below the specified threshold, one of two actions can be taken. Either the shot is removed and the adjoining shots are extended to fill in the resulting gap, or the shot is extended. The current implementation always removes the shot, but one can imagine the decision as to which is the more appropriate action could be taken based on the the relative importance of the shots.

4.6 Inserting Reaction Shots

Conversely, when a shot is considered too long for the desired edit pacing, a reaction shot may be inserted to split the shot into several smaller clips. We do this if the duration of a shot exceeds a given maximum shot length. In this case, a shot of the listener is cut into the shot of the speaker, even if the listener is not associated with any events at that time. In practice, an editor will attempt to use a clip of the listener in which something is happening, be it a gesture, a smile, or other facial expression, even if the clip is from another point in time, another take, or even another scene. This conceals the true purpose of the cut by providing a visual excuse for the edit.

4.7 Reframing Quick Shots

A useful heuristic, when the editor encounters a sequence of short shots which would otherwise be removed from the timeline, is to reframe the shots into a single longer shot by using an alternate camera framing. For example, if two actors are engaged in a rapid exchange of dialog, it is possible that each of the shots of an actor speaking has a smaller duration than desired. In such a case, it would be undesirable to simply remove some or even all of these rapid dialog shots. Instead, the editor combines these shots into a single shot, and searches the camera setups for a setup which frames both actors. In this way, a series of very quick one-shots of the actors is replaced with a longer two-shot of both actors.

4.8 Merging Continuous Shots

There may be cases, possibly due to the application of other heuristics, where two similar shots of an actor are adjacent on the timeline. This reflects shots corresponding to two camera setups which are close to each other. In such a case, there will be a disconcerting jump between the two shots. The merging heuristic detects such cases and replaces one shot with the other, resulting in a single longer shot without any visual discontinuity.

5 Stylistic Rules

The stylistic rules are parameters that are used by the heuristics described above in making their choices. In the same spirit as for the heuristics, we make these parameters explicit in terms that an editor could understand. This allows us to make specific changes and see the results, as well as try to derive the required input to achieve a certain style of editing. Doing so involves grouping the rules into "editing styles" which can then be packaged for use by higher level modules of a future system. The existing system features a dozen or so stylistic editing rules which fall into the following categories.

5.1 Framing

The cinematography module will create multiple camera setups for framings of each shot in the director's shot list. The framing rules determine what range of framings should be generated, and what the preferred framing is for shots. Some styles will prefer closeups, while other will use a medium shot as the default framing. In addition, rules exist to specify whether each scene should begin with an establishing shot, and whether there are any constraints on framings of the same actor in consecutive shots. A further rule determines whether framings should indicate the relative distances between actors.

5.2 Shot Durations

Shot durations are fundamental to the pacing of a scene. Rules determine the minimum and maximum shot durations an edit should produce. If shots are shorter than the minimum duration, they will be removed by the appropriate heuristic,and if they are longer than the maximum duration, they will be split by insertions of reaction shots. An additional rule determines how to fill gaps between events, which typically denote pauses in a conversation.

5.3 Cutting Style

If a shot meets the preconditions of being cut using an L-cut, this rule determines when and how such a cut is made. By separating the visual and audio transitions between two shots, a smoother flow is achieved, and is often experienced as being less noticeable by the audience.

5.4 Reframing

This rule specifies whether to merge a series of quick shots of individual actors into a longer one by using an alternate camera setup which shows all the actors involved in a group shot. This rule has a significant effect on the final edit, as it can greatly alter the duration of shots.

6 Implementation

Our implementation consists of a software framework based on custom software communicating with Maya, an existing commercial 3D animation application developed by Autodesk. The bulk of our software is written in Common Lisp on OS X, with a socket interface to Maya. Common Lisp calls are translated to MEL (Maya Extension Language) command strings which are then sent to Maya via a socket. This allows our system to execute any Maya action, such as creating and placing cameras, animating actors, and rendering frames of animation. Values from Maya are return to our software via sockets as well. The remainder of our software consists of C++ plugins written for Maya to optimize operations which are unacceptably slow when running in MEL. This software architecture avoids the need to develop a custom 3D graphics engine, and gives us access to the large set of functionality present in an existing commercial 3D animation system.

The two main graphical entities of our 3D scenes are cameras and actors. Cameras have position and orientation, as well a focal length and aspect ratio. Actors are modeled as hierarchical skeletal animation rigs, with rigid body parts, and can perform simple procedural animation. Currently, these procedures consist of a mouth animation for speaking, and a look-at animation for orienting the head. Actor animations are generated procedurally based on the events of a scene. These animations are helpful to illustrate the events of the scene, and make the viewer understand why a framing or editing decision was made by the system. In essence they help comprehension of the scene by doing very rudimentary acting, in term allowing us to evaluate the correctness and appropriateness of the generated cinematography.

7 Results & Evaluation

The scenes, and corresponding event lists, used for evaluating our systems are created by manually encoding existing film clips. Timing and staging information is extracted from the several sample clips and encoded to serve as test cases for the system. This provides us with a dialog track for the scene, and also allows us to compare the generated results with established “acceptable” results. Note that events could also be generated using a higher-level scripting system. A story generator or machinima system, for example, could create scenes and events procedurally. Game engines could also generate in-game cinematics on the fly based on scripts and user actions. Finally instant messaging and chat software could also be used to generate events, based on the text and emotes input by multiple participants. In what follows, we discuss results obtained with scenes manually encoded from video clips. The generated movies for these examples are available at www2.hawaii.edu/~kaveh/Research/Papers/Siggraph2009/.

7.1 Simple Two-actor Dialog

This scene from the television show "Buffy The Vampire Slayer" is a simple conversation between two actors. The conversation is straightforwardly back and forth, each actor taking turn in speaking. As we can see from the director's shot list for the two actors in Fig. 3, there is no overlapping dialog, and minimal gaps in the dialog. The steps taken by the editor mainly consist of filling in gaps and inserting reaction shots (diamonds) based on the stylistic rules. In Edit 1 long shots are tolerated only one reaction shot is inserted. This result is somewhat close to the actual edit of the film clip, though that was not our goal in choosing stylistic rules. In the Edit 2 a more frenetic editing pace is specified, and as a result the longer shots have been broken up with more reaction shots. L-cuts have also been added in Edit 2 to reduce the reliance of the speaking events on the back-and-forth editing.

7.2 Two-actor Dialog With Pauses

The following example is from a tense conversation in the feature film "Michael Clayton". Unlike the previous scene, the conversation is peppered with tense and meaningful pauses, as can be seen in the director's shot list for the actors in Fig. 4. We illustrate two resulting edits by our system, based on the conversation event timings with different stylistic rules. Though our edits are "correct" (i.e. not demonstrably incorrect), our system falls short of the actual clip in conveying the sense of tension and subtle actors interactions. This is due to our events not being tagged with the myriad of non-verbal actions being taken by the actors, and by the very rudimentary nature of the acting of our CG actors. This indicates how primitive, although correct, the decisions of our system are compared to those of a real editor.

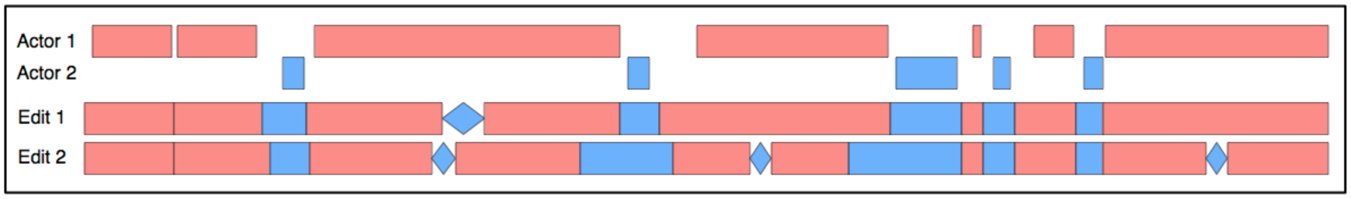

Figure 3. Edits for simple two-actor dialog. Diamonds indicate reaction shots.

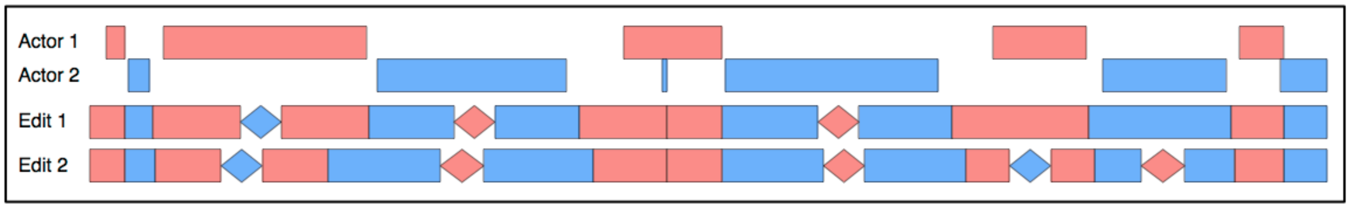

Figure 4. Edits for two-actor dialog with pauses. Diamonds indicate reaction shots.

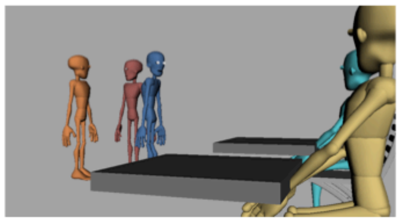

Figure 5a. Framing to indicate distance.

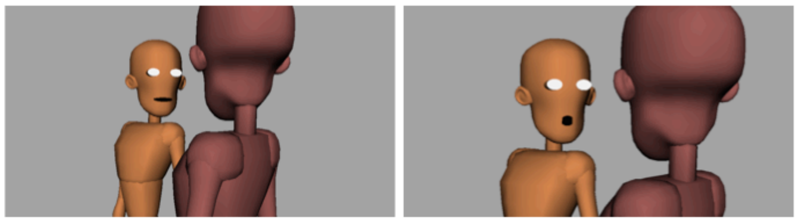

Figure 5b. Using closer framing to indicate heightened intensity of dialog.

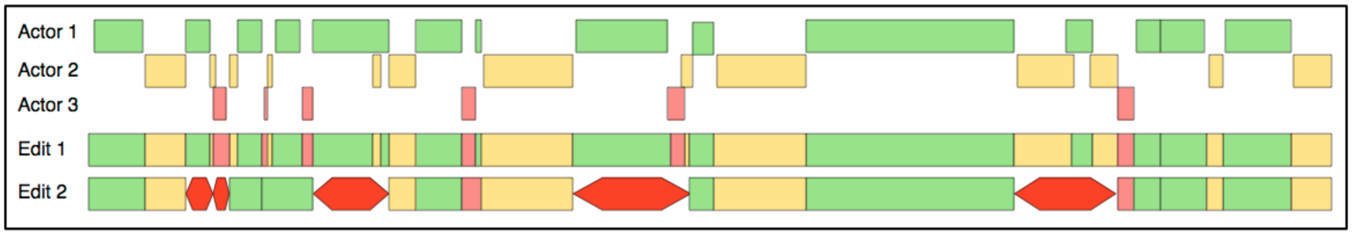

Figure 6. Edits for complex overlapping three-actor dialog. Hexagons indicate reframed quick shots.

7.3 Clustered Actors

The third example is from "Charlie Wilson's War", and involves two groups of actors in conversation. By using the framing style (5.1) where framing indicates relative distance between actors, the edit accomplishes the task of showing that the two groups are far apart, as can be seen in Fig. 5a. Also in this scene one shot is marked as having a higher intensity than the rest. As a result, the editor chooses a closer framing for that shot, resulting in a (desired) visual jump closer to the actor as his dialog becomes more intense. This is shown in Fig. 5b.

7.4 Complex Overlapping Dialog Scene

The final example we show is from the film "The Big Lebowski" and features a long dialog scene with three actors constantly speaking over each others' lines. As can be seen from the director's shot list in Fig. 6, the speaking events involve many temporal overlaps which need to be resolved by the editing system. Edit 1 produces a clip with many short shots and visual jumps, leading to a frenetic scene pacing. Edit 2 shows a more restrained pace with fewer rapid cuts while maintaining the goal of showing what is in the director's shot list. This is accomplished mainly through the application of the reframing heuristic described in Section 4.7. When there is a burst of rapid dialog between two actors, this heuristic replaces several quick cuts of each actor with a longer shot of the two actors, resulting in a smoother edit.

8 Conclusions and Future Work

We have proposed a set of heuristics for editing of film clips that attempts to mimic the thought process of an actual editor, using editing heuristics informed by a set of stylistic rules. Our system can generate a variety of visually acceptable edited sequences given different stylistic rules for static scenes involving dialogs between actors. Though our system is far from being able to produce edited clips that would rival in quality with those of a real editor editing a subtle scene, it can serve a number of useful functions. It can serve as an editor's apprentice, generating a quick first cut of a given scene, for the editor to examine and tweak as deemed appropriate. For highly constrained styles, such as sitcoms and talk shows, the output from our system may be adequate for use as is. Our system can also be used in places where scenes are generated on the fly (game scene playbacks, 3D chat) or procedurally (game cinematics, storytelling software).

Our system currently deals with static scenes. Adding editing decisions based on actor movements within the frame is a natural next step. Many of the standard cinematic idioms rely on cutting on an action, such as an actor entering or exiting the frame.

Good editors base editing choices on the smallest of visual cues: glances, reactions, even blinks. Their decisions are informed by the context of the scene in question and how it fits into the larger work. Our system needs better tagging of events, to denote both more subtle actions such as reactions, and blinks, while at the same time we need a wider variety of events, such as meaningful looks, gestures, smiles, and facial expressions.

Editors must also ensure that the edit they produce conveys the appropriate emotional content and invokes the desired responses in the audience. A set of heuristics which associate editing and camera choice to emotional response would be useful in attempting to better manage the viewer's emotional journey through the film. More contextual information, such as the role and place of the scene being edited within the scope of the larger narrative can provide information to help determine the editing style and pacing.

Editors have the freedom to shift and rearrange shots in time, effectively building events and sequences which did not take place during filming. It would be very interesting if an automated editing system could manipulate shots in such a powerful way to enhance the story experience of the viewer.

The stylistic rules could be packaged into full-fledged styles, such as those of Classical Hollywood, contemporary action films, music videos, situation comedies, and so forth. The system can also be used to attempt to reverse engineer the editing of such styles, effectively deriving the implicit heuristics which underly each filmic style.

Acknowledgements

We would like to thank Avatar Reality and the Blue Mars project for their support of portions of this research.

References

AMERSON, D. and KIME, S. 2000, Real-time Cinematic Camera Control for Interactive Narratives. In *AAAI'00*.

ARIJON, D. Grammar of the Film Language. Silman-James Press, 1991.

BARES, W.H.; and LESTER, J.C. 1999, Intelligent Multi-Shot Visualization Interfaces for Dynamic 3D Worlds. In *Intl. Conf. on Intelligent User Interfaces*, pages 119-126.

BARES, W.H.; THAINIMIT, S.; and McDERMOTT, S., A Model for Constraint-Based Camera Planning. In *Smart Graphics. Papers from the 2000 AAAI Spring Symposium*, pages 84-91, 2000.

BORDWELL, D., STAIGER, J., and THOMPSON, K., The Classical Hollywood Cinema. Routledge, 1998.

CHRISTIANSON, D. B.; ANDERSON, S. E.; He, L.-W.; Salesin, D. H.; Weld, D. S.; and Cohen, M. F. 1996, Declarative camera control for automatic cinematography. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 148-155.

COZIC, L.; DAVIS, S.B.; and JONES, H., Interaction and Expressivity in Video Games: Harnessing the Rhetoric of Film. In *Technologies for Interactive Digital Storytelling and Entertainment,* pages 232-239, 2004.

ELSON, D. K., and RIEDL, M. O., 2007, A Lightweight Intelligent Virtual Cinematography System for Machinima Production. In *Artificial Intelligence and Interactive Digital Entertainment 2007*, pages 8-13.

FRIEDMAN, D.A.; and FELDMAN, Y.A., 2004, Knowledge-based cinematography and its applications. In *Proceedings of ECAI*, pages 256-262.

HALPER, N.; HELBING, R.; and STROTHOTTE, T. 2001, A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence, In *Proc. Eurographics 2001*.

HE, L.; COHEN, M.; and SALESIN, D., 1996, The Virtual Cinematographer: A Paradigm for Automatic Real-time Camera Control and Directing, *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 217-224.

KARDAN, K., and CASANOVA, H., 2008, Virtual Cinematography of Group Scenes Using Hierarchical Lines of Actions. In *Proceedings of the 2008 ACM SIGGRAPH Symposium on Video Games*, pages 171-178.

KATZ, S., Film Directing: Shot by Shot: Visualizing from Concept to Screen, Michael Wiese Productions, 1991.

LI, T.Y.; and XIAO, X.Y., An Interactive Camera Planning System for Automatic Cinematographer. In *Conference on Multimedia Modelin*g, pages 310-315, 2005.

MASCELLI, J. V., The Five C's of Cinematography: Motion Picture Filming Techniques. Silman-James Press, 1998.

MURCH, W., In the Blink of an Eye. Silman-James Press, 2001.

OLIVEROS, D.A.M. 2004. Intelligent Cinematic Camera for 3D Games, MSc. Thesis, University of Technology, Sydney Australia.

TOMLINSON, B.; BLUMBERG, B., and DELPHINE, N., 2000, Expressive Autonomous Cinematography for Interactive Virtual Environments. In *Proc. of the 4th International Conf. on Autonomous Agents*, pages 317-324.