Knowledge-Based Formalization of Cinematic Expression and its Application to Animation[1]

Doron Friedman
Tel Aviv University
doronf@post.tau.ac.il

Yishai Feldman
The Interdisciplinary Center, Herzliya
yishai@idc.ac.il

Abstract

Camera control in virtual environments received growing attention recently, and some of the research turned to cinematography as a source for inspiration. Cinematic knowledge is a highly challenging domain for formalization. We have used a systematic knowledge-based approach for knowledge acquisition and formalized expert knowledge in the form of fine-grained rules. Given a screenplay, the rules interact to suggest a solution by symbolic constraint propagation and truth maintenance.

We have implemented a prototype system that accepts screenplays in a formal language and generates editing decisions, based on a cinematic knowledge base. If the system is provided with 3D meshes and animations corresponding to the actions and objects mentioned in the screenplay, the system generates a 3D animation clip for the screenplay, based on its editing decisions. The system can be equipped with cinematic principles from various genres, and shows how complex cinematic phenomena can emerge from the interaction of simple rules.

Categories and Subject Descriptors (according to ACM CCS): I.2.1 [Artificial Intelligence]: Applications and Expert Systems, I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods, I.3.8 [Computer Graphics]: Applications, J.5 [Computer Applications]: Fine Arts.

1. Introduction

In this research, we attempt to formalize cinematic knowledge in a flexible and generic way, and apply it to the editing of 3D animation. We have designed and implemented a knowledge-based system that accepts an annotated screenplay and raw animation, and produces a 3D movie with camera behavior that conforms to the cinematic principles in the knowledge base.

Automatic camera placement was investigated in the past, and some researchers have looked into cinematography as a relevant source of information [1-4, 6, 9-10, 13, 14, 17]. We aim to continue this line of research by suggesting a new method, which will better suit the complexity of the cinematic domain. To allow for maximum flexibility, we let domain experts describe the knowledge as a set of independent pieces of information, and use constraint propagation to allow the emergence of a coherent solution out of these pieces. We also observe that it is not likely there will ever be one all-encompassing, agreed upon cinematic corpus of knowledge, and the principles may vary across different genres and different experts.

Cinematic expression has evolved over a century, and includes many principles and conventions, of which TV or film viewers are usually not consciously aware. For example, you will rarely see a “jump cut” outside the scope of specific genres, such as MTV-style video clips or experimental films. This rule can be formulated as follows: after a cut, the camera angle must change by at least 60 degrees. Such rules can be formalized, and can thus be simulated by software.

We formalize the cinematic knowledge as a set of rules. The rules were extracted from textbooks on cinematic theory [5, 16] and formalized by domain experts. Some of the rules are screenplay-dependent, i.e., describe the relation between the input (action) and the output (camera behavior). Other rules are screenplay-independent, that is, cinematic constraints on combinations of shots.

Abstraction is also crucial for our method; our representation uses an abstract model of actions

[1] See http://www.math.tau.ac.il/~doronf/eg2002.html for the media materials corresponding to this paper.

in space and time. Actions are represented based on Schank's theories of thematic role frames and primitive actions [15]. The system manages interval relations rather than maintaining a continuous timeline. The spatial model only refers to proximity and gaze directions, and the spatial constraints are converted into pure geometric constraints only at the final stage.

The knowledge expert can use high-level concepts such as "establishing shot", "the 180-degrees line", or "cinematic sentence", rather than referring to the seven degrees of freedom a camera has. We have investigated the formalization of different genres: Latin Telenovela, TV series (The West Wing, The X-files), and several classical films. Our system is able to demonstrate differences between the cinematic styles used in different genres. Due to our constraint propagation method, it can also demonstrate the emergence of complex camera behavior, which is especially important for entertainment or artistic environments. Due to our usage of truth maintenance, we are able to allow the users to correct the results locally, and observe the overall effect.

Although we have only investigated linear scripts so far, we expect the results to be relevant to interactive virtual environments as well. Automated camera control may be even more useful in such environments, since there is no available director or editor in real-time. There is a large body of knowledge about cinematic principles, accumulated and investigated over the twentieth century. The artistic language for virtual environments, however, is not yet clearly defined. We believe that investigating this language should be a joint effort for artists and computer scientists, and we hope that this work can serve as a foundation.

2. Previous Work

Much work has been done on camera placement, and recently some of this work turned to cinematography as a source of inspiration [1-4, 6, 9-10, 13, 14, 17]. These illustrate the potential of such an approach, but also the complexity of the cinematic domain, and the difficulties in finding the best computational techniques.

Our research was mostly inspired by the work of He, Cohen, and Salesin [14]. The Virtual Cinematographer is a software tool that formulates some idioms used by cinema or television directors, as finite-state machines. These automata may then be used to make real-time decisions in 3D chat environments on the Web. However, the research covered a very restricted set of situations and a small number of cinematic principles.

In subsequent work, Christianson *et al.* [6] defined DCCL (Declarative Camera Control Language) and attempted a more systematic analysis of cinematography. They describe several cinematic principles and show how they can be formalized into a declarative language. They encode 16 idioms, at a level of abstraction similar to the way they would be described in a film textbook. They demonstrate the usage of their system in the context of a simple interactive video game. In a recent work, Amerson and Kime [1] have also used idioms for cinematic virtual camera.

Our research aims at a similar goal, using a different method. We claim that idioms are the wrong granularity, being too coarse to formalize cinematic knowledge. Using idioms, one needs to code a specific idiom for every possible situation. This method results in a repetitive and predictable output, which impedes user engagement. For the same reasons, using idioms also does not scale to more complex situations.

Tomlinson, Blumberg, and Nain [17] demonstrate how a virtual camera may be implemented within a general framework of virtual agents. The camera is modeled as an agent based on a reactive behavior system, with sensors, emotions, motivations, and actions. Trying to fit editing into the behavioral model sometimes seems awkward; e.g., when cinematic principles are implemented as motivations ("desireForCloseUp"). However, we find this approach important, as it is able to demonstrate the emergence of unexpected results, which is especially important for entertainment applications. In a sense, it is similar to our attempt at modeling the viewer's experience, and treating editing as a manipulation of the viewer's emotions and knowledge. Their work is also unique in being one of the few that try to deal with emotions. The level of cinematic understanding is limited; it would be interesting to integrate their agent approach with ours.

Most of the works dealing with camera control have done so in the context of interactive virtual environments, rather than linear animation. It is difficult to apply cinematic principles, which were invented for linear, passive media, to interactive environments. The equivalent principles would be analogous, but different, and still need to be worked out by artists. An early work that tried to deal with this conflict is Galyean's attempt [12] at combining interactivity and editing, although he uses a very small number of cinematic principles that are hard-coded into the system.

3. Knowledge-Based Editing

We have designed and implemented a system called Mario. It is able to process a variety of scripts, and suggest camera-directing instructions, based on cinematic principles (Figure 1). If the system has access to 3D meshes and animations for the objects and actions mentioned in the script, the system can generate an animated 3D movie.

The first domain we examined was Telenovela, a Latin-American form of TV soap opera, which is infamous for its simplistic use of cinematic language. We will explain the system based on a simple example from that domain.

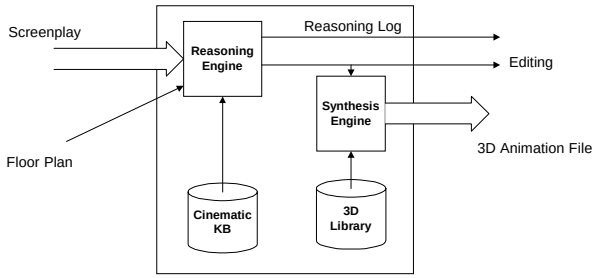

Figure 1: *A schematic description of Mario.*

The inputs to Mario are a screenplay and a floor plan. The screenplay is given in a formal language. It can be very similar to a screenplay for a film or TV movie, but we do not address natural language processing in this scope. An excerpt from the script example follows:

```
location: living-room
init: Mario sit sofa1
Mother enter room
Mother speak "Mario, have you
    eaten the sandwich I made
    you?"
Mother sit-on sofa2
Mario speak "I wasn't hungry."
...
```

We note that screenplays are different from arbitrary natural-language texts in that they typically describe specific characters performing concrete actions rather than abstract relationships.

The script is converted into an abstract representation, which is composed of action frames and spatial annotations placed on a timeline. Next, we split the scenes into smaller units called **sentences.** This term is often used in cinematography, but never formally defined, so our film expert came up with the following definition: a new sentence begins if a significant spatial change occurs during a scene.[2] Assuming overall consistency in style, each sentence can be edited locally. A sentence is similar in structure to a whole scene, but having a smaller unit is better in two respects: it is computationally more efficient, and it makes it easier to follow the tool's reasoning.

Next, the system starts applying rules from the knowledge base. Rules are evaluated into a collection of constraints, which are added into the slots of the corresponding frames. The constraint propagation process can be illustrated by the rules that define establishing shots. The first rule states that the first or second shot in a sentence must be an establishing shot:

```
(let S (shots this-sentence))
(let S1 (first-item S))
(> (size S) 2)
➔
(or (establishing-shot S1)
    (establishing-shot (next-shot S1)))
```

The second rule defines an establishing shot as one that shows the whole scene in a long shot:

```
(for-all s (shots this-sentence))
(establishing-shot s)
(let b (bbox this-sentence))
(for-all v (viewpoints s))
➔
(is-in b (targets v))
(= (shot-type v) long-shot )
```

The bbox function returns the bounding box surrounding the area in which the cinematic sentence takes place. A shot may have several viewpoints, representing possible camera

[2] This definition holds for Telenovelas only.

motion. The conclusion of the rule states that all the viewpoints of an establishing shot must show the bounding box in a long shot. Note that the rule refers to high-level concepts rather than the precise geometry.

After the evaluation of the first rule, an OR constraint is generated to state that either the first or the second shot is an establishing-shot. The second rule evaluates to other constraints, among them the constraint that sets the shot-type to long-shot. At some point, the OR constraint needs to be resolved, and in the absence of additional information, an arbitrary decision is taken, and the second shot may be marked as an establishing-shot. This triggers the other constraint, and the viewpoints attached to this shot are marked to be long-shot.

Another example illustrates backtracking, as well as the emergence of cinematic idioms out of several rules. The 180-degrees-line rule can be expressed as below: [3]

```
(for-all I (intervals this-sentence))
➔
(= (side (shot I) 180-line)
   (side (shot (next I)) 180-line))
```

Let us see how this rule interacts with others during the constraint-propagation process. Another rule states that when an actor is speaking (and possibly other conditions are met), the actor is displayed in a frontal medium-shot. Frontal shots of humans are defined elsewhere to be 30 degrees to the left or to the right of the gaze vector, since actors should not be displayed looking directly into the camera (a by-product of this rule with a typical dialog positioning are over-the-shoulder shots, as in Figure 2).

The combination of these rules actually forms a simple dialog idiom. Initially, the camera direction is dictated by an OR constraint to be either 30 degrees to the left or 30 degrees to the right of each actor's gaze direction. At some point, an arbitrary decision needs to be made. Assuming that the arbitrary choice was to place the camera to the right in both cases, we now get a contradiction to the 180-degrees-line rule. The contradiction initiates a dependency-directed backtracking process, in which one of the arbitrary choices needs to be changed (see Figure 3).

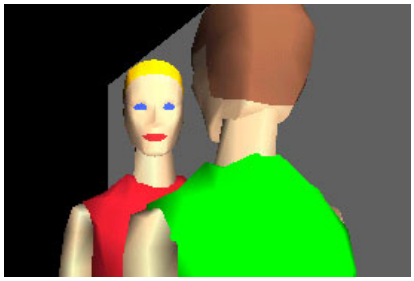

Figure 2: *Over-the-shoulder shot emerges implicitly from conversation rules.*

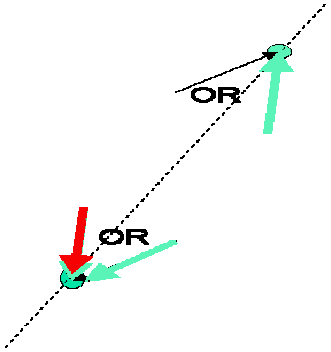

Figure 3: *A dialog idiom is implicitly generated from two specific rules and backtracking – showing a speaking actor in medium-shot in 30 degrees. The camera is not allowed to cross the 180-degrees-line.*

Our method, and specifically the usage of a truth-maintenance system, is appropriate for an iterative process, in which a user may critique the results produced by the system. If the user is not happy with a specific shot in Mario's output, she can correct the result for that shot, by adding a constraint. For example, she can state that a cut proposed by the system should be eliminated. This constraint may have global side effects that the user is not aware of, so the system performs dependency-directed back-tracking, and suggests another solution.

During knowledge acquisition, we found it necessary to use default rules; these are rules that are assumed to be true unless proven otherwise. A well-known problem in non-monotonic reasoning is the problem of conflicting defaults. We have decided to use domain-specific methods to deal with this problem.

In our case, we found that there is no independent order between default rules in our

[3] This rule is simplified. Dynamic shots may include more than one viewpoint element. Also, it is legal for the camera to cross the line with camera motion.

knowledge base, and the priority of one rule over another often depends on the specific situation. A technical solution would be to add the necessary conditions to the weakest rule for every given situation in which a collision arises. This approach does not scale in practice.

We have started to examine another solution, namely, classifying high-level goals served by each default rule. In a case of a collision between two default rules, if the rules serve the same goal, either would do. If one of the rules' goals can be satisfied otherwise, we prefer the other rule. Otherwise, we prefer the goal with higher importance with respect to the genre in question. While theoretically this might not always be possible, we expect this method to resolve most of the cases of conflicting defaults.

The goals we identify are: spatial orientation, conceptual orientation, conveying the information explicit in the script, conveying the information that may be implicitly deduced from the script, manipulating the viewer's emotional experience, aesthetic considerations (such as symmetry in time and frame composition), and parsimony (using minimum number of shots and cameras). The goals component has not yet been implemented. .

4. Discussion and Future Work

Formalization of artistic principles is a difficult task, even without reducing artistic phenomena to the level of mathematics. Note that we are interested in a descriptive theory of cinema, rather than a normative one. Our goal is to capture the concepts and dynamics involved in the domain; if this is done properly, our mechanism can be used to represent different cinematic theories. Thus, we examine the formalization of different genres, given by different domain experts.

Our analysis so far has revealed the difficulties of cinematic formalization. As we move into more complex genres and camera usage, we repeatedly encounter the barriers of Artificial Intelligence, mainly story understanding and commonsense knowledge. As we do not expect breakthroughs in these areas anytime soon, there are two things that may be done. The first is to allow for human assistance in the form of annotations in the screenplay. Another approach is to restrict the domain enough for existing AI techniques; for example, to the context of a specific video game. We have applied the first solution, and are investigating the second.

Another aspect that seems essential for formalizing cinematography is user modeling. If we maintain a cognitive and emotional model of the viewer while watching the movie, then cinematic decisions are basically a manipulation of that model. A tool can decide what the viewer should know at any time, and expose the information accordingly. Similarly, it can decide how the user should feel, and use the most appropriate cinematic technique to achieve that effect. We have made the first step in this direction by classifying rules into high-level goals, some of which are expressed in the form of the viewer model. This, however, was not yet implemented.

One possible extension of this research is automated authoring of animated clips. We believe that the major difficulty in the conversion of screenplays into movies is the recovery of information not explicitly present in the input. This will require the following components:

1. Natural-language understanding: It should be possible to convert natural language descriptions in a restricted domain into a formal language. The one we use is partly based on thematic-role-frames, to ease such an attempt. Coyne and Sproat [7] have been able to convert a large variety of textual descriptions into static images.
2. Mise-en-scene construction: The system needs to decide which objects (props) will participate, their location and design (color, texture, state), and the exact timing of actions. This stage involves a domain-dependent knowledge base, and our paradigm needs to be able to display behavior similar to trajectory planning algorithms.
3. Cinematic enhancement: We have demonstrated the augmentation of a screenplay with camera directions, and it is possible to extend this approach to model lighting and soundtrack.
4. Animation generation: We assume the existence of a library that includes 3D models and basic animations, corresponding to the objects and actions mentioned in the script. The system needs to combine objects

and actions according to the script,[4] to convert text to speech, and to draw from a library of sound effects and background music.

In summary, this research deploys a knowledge-based approach to the formalization of cinematic expression, using constraint propagation and truth maintenance. We believe that such an approach is crucial to dealing with a complex domain such as cinematic theory. Furthermore, we believe that this work could serve as a valuable framework both for automated authoring and for further research in interactive virtual environments.

Acknowledgments

We wish to thank Noam Knoller, an MFA student in Tel Aviv University School of Film and Television, who served as our chief domain expert. Ariel Shamir provided useful comments on an earlier version of this paper.

References

1. D. Amerson and S. Kime, "Real-Time Cinematic Camera Control for Interactive Narratives", *Artificial Intelligence and Interactive Entertainment, AAAI Spring Symposium*, 2001.
2. W. H. Bares, J. P. Gregoire, and J. C. Lester, "Realtime Constraint-Based Cinematography for Complex Interactive 3D Worlds", in *AAAI-98: Proc. Tenth Conf. Innovative Applications of Artificial Intelligence*, Madison, Wisconsin, pages 1101-1106, 1998.
3. W. H. Bares and J. C. Lester, "Cinematographic User Models for Automated Realtime Camera Control in Dynamic 3D Environments", in A. Jameson, C. Paris, and C. Tasso (Eds.), *User Modeling: Proc Sixth Int'l Conf., UM97*, Springer, 1997.
4. W. H. Bares, L. S. Zettlemoyer, D. W. Rodriguez, and J. C. Lester, "Task-Sensitive Cinematography Interfaces for Interactive 3D Learning Environments", in *IUI-98: Proc. 1998 Int'l Conf. Intelligent User Interfaces*, pages 81-88, San Francisco, California, 1998.
5. H. Callev, *Cinematic Expression*, Optimus, 1996.
6. D. R. Christianson, S. E. Anderson, L. W. He, D. H. Salesin, D. S. Weld, and M. F. Cohen, "Declarative Camera Control for Automatic Cinematography", *Thirteenth Nat'l Conf. Artificial Intelligence*, 1996.
7. B. Coyne, R. Sproat, "WordsEye: An Automatic Text-to-Scene Conversion System", *Proc. SIGGRAPH 01*, pages 487-496, 2001.
8. J. Doyle, "Truth Maintenance Systems", *Artificial Intelligence*, 12(3):231-272, 1979.
9. S. M. Drucker, *Intelligent Camera Control for Graphical Environments*, PhD Thesis, MIT Media Lab, 1994.
10. S. M. Drucker, T. A. Galyean, and D. Zeltzer, "CINEMA: A System for Procedural Camera Movements", *Computer Graphics*, 26(2):67-70, Mar. 1992.
11. Y. A. Feldman and C. Rich, "Principles of Knowledge Representation and Reasoning in the FRAPPE System", *Proc. Sixth Israeli Conf. on Artificial Intelligence and Computer Vision*, 1989.
12. T. A. Galyean, *Narrative Guidance of Interactivity*, PhD thesis, MIT Media Lab, 1995.
13. N. Halper., R. Helbing, and T. Strothotte, "A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence", *Proc. EUROGRAPHICS 2001*.
14. L. He, M. F. Cohen, and D. H. Salesin, "The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing", *Proc. SIGGRAPH 96*, pages 217-224, 1996.
15. R. Schank and A. Abelson., *Scripts, Plans, Goals, and Understanding*, Wiley, 1977.
16. R. Thompson, *Grammar of the Edit*, Focal Press, 1993.
17. B. Tomlinson, B. Blumberg, and D. Nain, "Expressive Autonomous Cinematography for Interactive Virtual Environments", In C. Sierra, G. Maria, and J. S. Rosenschein, eds., *Proc. Fourth Int'l Conf. Autonomous Agents*, pages 317-324, Barcelona, Spain, June, 2000.

[4] This might involve some complex transformations such as inverse kinematics, non-linear editing of animation, and composition of simultaneous actions.