An Interactive Camera Planning System for Automatic Cinematographer

Tsai-Yen Li
Computer Science Department
National Chengchi University
64, Sec.2, Zhih-Nan Rd. Taipei, Taiwan 116
li@nccu.edu.tw

Xiang-Yan Xiao
Computer Science Department
National Chengchi University
64, Sec.2, Zhih-Nan Rd. Taipei, Taiwan 116
s8952@cs.nccu.edu.tw

Abstract

Currently most systems capable of performing intelligent camera control use cinematographic idioms or a constraint satisfaction mechanism to determine a sequence of camera configurations for a given animation script. However, an automated cinematography system cannot be made practical without taking idiosyncrasy and the distinct role of each member in a filmmaking team into account. In this paper, we propose an interactive virtual cinematographer model imitating the key functions of a real filmmaking team consisting of three modules: director*,* photographer*, and* editor*. The system uses parameterized cinematographic idioms in the three modules to determine the best camera configurations for an animation script. The system allows a user to interact with the virtual cinematographer to specify stylistic preferences, which can be carried over to other animation scripts.*

Keywords: Virtual cinematographer, intelligent camera control, camera placement planning, virtual environment

1. Introduction

Film-making is a complex production process requiring a team of highly skillful experts from multiple domains on backstage. In addition to the so-called camera men, many types of people may be involved in determining the contents of the final film through the lens. For example, the director is responsible for determining how to shoot the actors according to the flow of the screenplay as well as his/her aesthetic style. The expressions will be communicated with the camera man, who will then determine the final camera configurations according to environmental constraints. The shots taken in multiple cameras will then be sent to an editor to select and put together appropriate clips according to the screenplay.

Virtual cinematography aims to generate a sequence of camera configurations automatically on computers for a given animation script. Most recent systems with this function try to capture the idioms of cinematography as domain knowledge in computers and allow users to express their intents with various kinds of constraints. Despite these efforts, these systems are still hardly adopted in real productions due to the following three reasons. First, there are no standards for animation scripts suitable for cinematographers. Second, most of the declarative languages are not intuitive and too complicated for a camera professional. Third, the quality of the results generated by these systems is not acceptable without further modifications.

In this paper, we intend to address the last two issues and design a more practical virtual cinematography system. We propose a model consisting of three modules: *director*, *photographer*, and *editor* to determine camera configurations. The functions of these modules imitate their counterparts in a real film production team. The decomposed tasks are not only simpler to understand but also easier for the users to quickly draw analogies in their real-life experiences. Additionally, in order to increase the quality of the result, we extract the stylistic parameters of cinematography idioms and allow the users to tune these parameters interactively to determine their aesthetic preferences and carry them over across different scripts. At the current stage of implementation, the type of scenarios considered in this system is limited to dialogs among static actors.

2. Related work

Depending on the applications and how the camera is controlled, we can classify the researches of controlling virtual cameras in several ways. For example, one noticeable family of researches use predefined camera positions relative to the target subject to achieve automatic positioning [3]. However, this type of approaches falls short when the target consists of several subjects and complex dialogs are involved. Another line of approaches makes use of the idioms in cinematography to build a finite-state machine or use declarative languages to automatically perform inter-cuts in a script. However, the methods for finding good camera positions in these works are often over-simplified for complex scenes with unexpected obstruction [5][11].

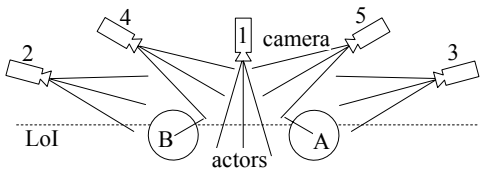

Figure 1. Types of camera positions for a pair of actors.

In contrast, instead of defining camera position explicitly, several works express the cinematographic criteria via a set of constraints and try to find a camera position according to the order or priorities of these constraints. In order to solve the constraint satisfaction problem, logic reasoning mechanisms [7], numerical methods [6] and genetic algorithms [10] have been used to find the optimal solution for camera position.

In many animation scenarios, the actors are not static and simply in dialogs. In the terminology of cinematography, they could be performing "actions with dialogs" or "actions without dialogs" in addition to "dialogs without action." In the case of on-line games, real-time camera tracking becomes an important issue [1]. Since no post-processing can be performed to refine the computer-generated result, maintaining adequate visual coherence between frames along the time line becomes crucial [9]. Nevertheless, in this paper, we only consider the case of "dialogs without action" as a starting point.

3. Fundamental cinematography

In this section, we will first present the principles and idioms that have been employed and explicitly modeled in this paper.

3.1. Actor grouping

According to cinematography, dialogs among actors can be classified into three categories according to the number of participants: *two-talk*, *three-talk*, and *over-four-talk*. Although there could be more than two people in a scene, the number of focused actors at a time is usually two. When the number of actors is greater than two, how to group the actors and take corresponding shots is the responsibility of the director. Typically, two or three actors will be chosen as a group at a time, and the decision is based on the screenplay as well as physical locations of these actors.

3.2. Camera settings

No matter how many actors are in a dialog group, there usually exists a *master shot* (such as camera 1 in Figure 1) that can cover all actors in the scene. This master shot is usually used at the beginning and the end of a dialog to help establish the spatial relations of the actors in the scene. In addition, the master shot is also used occasionally to reestablish the spatial relations when the dialog goes too long.

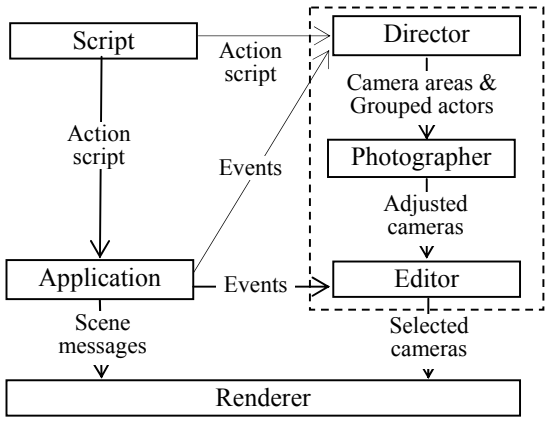

Figure 2. Overall system architecture

In the dialogs between two actors, there usually exist two pairs of intercut cameras that operate alternatively. For example, in Figure 1, the camera usually inter-cuts between positions 2 and 3 and adopts a close-up shot (4 or 5) occasionally. However, no inter-cuts among the shots for the same actor, such as 2 and 4, are allowed.

For the dialogs involving three actors, appropriate camera positions are usually chosen according to the distances between the actors. The principle is that close-up shots are necessary since every actor may become the subject of the dialog over time [2].

3.3. Respecting Line of Interest (LoI)

Respecting Line of Interest (LoI), the line connecting two involving actors, is the a basic principle in cinematography [12]. This means that camera inter-cuts can only undergo on one side of the line. For example, if camera 2 in Figure 1 is placed at the opposite side of the LoI, then actor A will remain on the same side of the screen under the inter-cut between positions 2 and 3. This 'jump' causes spatial confusion for the audience and should be avoided by all means.

4. Overview of virtual cinematographer system

In this paper, we propose to automate the process of generating a sequence of camera positions by a Virtual Cinematographer System (VCS). The overall system architecture is depicted in Figure 2. We assume that a script is given to the VCS as well as the application module. The application module executes the script and triggers an event for each action to the VCS, which computes the best camera position. According to the roles in traditional cinematographic production, we

decomposed the VCS into three components: *Director*, *Photographer*, and *Editor*, as described below.

The director plays a key role in determining the flow and style of the shots. In our VCS, scene setup and actor positions are given as inputs to the director module. The director groups the actors according to the script and some constraint parameters. After appropriate grouping, the director determines a camera area for each candidate shot. This area, representing the acceptable region for the camera position, will then be sent to the photographer module to determine the final position.

The photographer is the so-called camera man in traditional filmmaking process. The photographer has the freedom of choosing a good camera position according to his/her professional judgment and the constraints given by the director. In our VCS, the photographer module receives the camera areas and shooting subjects from the director and determines the final camera position for each shot according to cinematographic criteria such as how the line of interest is respected and how the focused subject is occluded.

The editor of a film is in charge of selecting appropriate shots and arranging them in a post-processing step when all clips are available. In our VCS, the editor does not need to wait until all shots are taken and then process them altogether. Instead, all possible shots are accessible at run time and the editor can determine the best shot at any time according to editing criteria such as information matching, information expectation, and tiredness. The best shot determined in this module is sent to the renderer for display.

5. Design of virtual director

The tasks of the director in our VCS are to determine the possible groupings of the actors and compute the associated camera areas for each grouping. The decisions greatly affect the style of a film and are usually identified as the director's aesthetic style. In our VCS, we have defined the following director-related preferences: *expected size ratio* (s_e), *the tolerance of expected size ratio* (s_t), and *tolerance of the ideal shooting angle* (α_t). s_e is the desired ratio of the target subject's size compared to the display screen. The ideal size may not always be available due to environmental constraints such as obstacles. Therefore, the director specifies tolerances on the size (s_t) and the shooting angle (α_t) for the range of an acceptable shot. This set of user-specified parameters, denoted by W_d =(s_e, s_t, α_t), is used to determine the groupings and associated camera areas as described below.

5.1. Actor grouping

For a given dialog, the director has to determine the candidate groupings according to the principles in cinematography. In VCS, we categorize the groups into three types: *pair*, *group* and *crowd*. Assume that every actor in the script is denoted by Ar_i, where i is the ID of the actor. Two actors, Ar_i and Ar_j can be grouped as a *pair*, denoted by *Pr*, if the two actors are close enough for them to appear on the screen with the expected size ratio. That is, the set of pairs is defined as

$$\{\Pr(Ar_1, Ar_2) \mid d(Ar_1, Ar_2) < w/(\tan\theta \cdot s_e \cdot (1 - s_t))\},$$

where w is the width of an actor and θ is the camera's view angle. Similarly, three actors can be viewed as a group, denoted by *Gr*, if two of them are already a pair and the distance between the pair and the third actor is not too large. That is, the set of groups is defined as

$$\{Gr(\Pr_1, Ar_2) \mid d(\Pr_1, Ar_2) < w/(\tan\theta \cdot s_e \cdot (1 - s_t))\}.$$

The third type of grouping is called *crowd*, denoted by *Cr*. A crowd is any collection of more than four people organized as groups, pairs, or single actors.

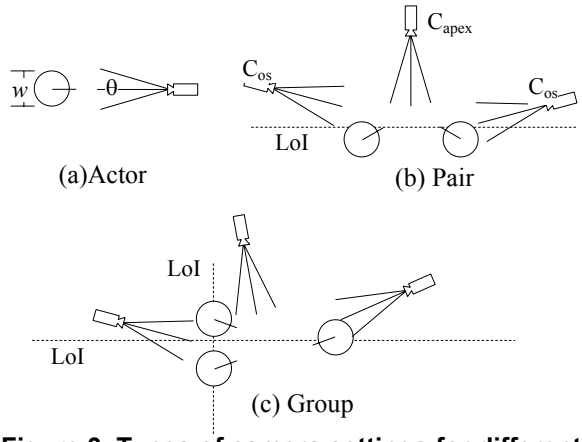

Figure 3. Types of camera settings for different types of actor groupings

Figure 3 shows the sets of cameras for different types of grouping. For example, each actor has a close-up shot in front as shown in Figure 3(a). Each pair will create three additional shots including an apex camera and two over-shoulder cameras, as shown in Figure 3(b). For each group, three additional shots will be created, as shown in Figure 3(c). After appropriate grouping, if there are additional actors that are not included or there are more than two groups of actors, an additional apex camera is used to cover the whole crowd.

5.2. Determining camera area

After the grouping is determined by the virtual director, a fan-shaped camera area, as shown in Figure 4, can be computed according to the director's professional preference (via W_d in this case). For example, according to s_e, we can compute the expected distance, d_e, between the camera and the subject. Similarly, the distance tolerance, d_t, can be determined by s_t. A cam-

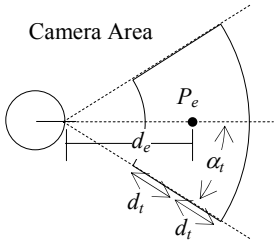

Figure 4. Candidate camera area

era area can then be set up with these two distances and the shooting angle tolerance.

6. Design of virtual photographer

6.1. Determining the best camera location

Given a rough camera area, the exact position of the camera is further determined by the virtual photographer. There are two attributes for each shot: *focus* and *scope*. The focus of a shot is the primary subject(s) while other participating actors compose the *scope*. The task of a photographer is to ensure that the focused actor(s) are not occluded and other actors in the dialog scope are included. In addition to satisfying the basic requirement, the virtual photographer will also consider other cinematography criteria and try to find the optimal camera position according to the location score, S_{loc}, defined by the following formula:

$$S_{loc} = w_e \cdot E + w_l \cdot L + w_u \cdot U ,$$

where E is *expectation matching rate*, L is the *LoI respecting rate*, and U is *non-occlusion rate*. These criteria will be defined in the next subsection. The set of weights, $W_p = (w_e, w_l, w_u)$, are defined as the photographer's aesthetic style. For each camera area, the final camera location is determined by searching for the one with the best score.

6.2. Camera location selection criteria

The first criterion is on how the camera location matches the ideal location expected by the director without considering environmental constraints. This ideal location, P_e, is at the center of the camera area. The expectation matching rate, E, of a given camera location, P_c, is defined as

$$E = 1 - \frac{|P_e - P_c|}{Q_e},$$

where Q_e is a quantization factor for normalizing E to the range of (0,1).

The second criterion is on how the camera location respects the line of interest. In order to increase the depth cue of a shot, it is usually desirable to place the camera as close to the line of interest as possible. Assume that V_{TC} is the vector from the target subject to the camera and φ is the angle between this vector and the line of interest. The LoI respecting rate is defined as

$$L = 1 - \frac{|V_{TC}| \cdot \sin\varphi}{Q_L},$$

where Q_L is a quantization factor for normalizing L to the range of (0,1).

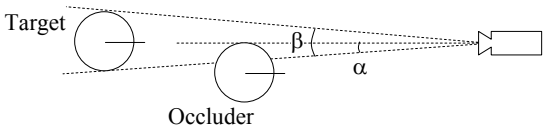

Figure 5. Occlusion rate

The third criterion is on the degree of occlusion that the target subject gets for a given shot. We define the occlusion rate for an occluder i on the right-hand or left-hand sides of the screen as the ratio of shooting angles between the occluder and the target:

$$O_i^{(L/R)} = \frac{\alpha_i}{\beta},$$

where α_i is the occlusion angle for the occluder i and β is the view angle for covering the target, as depicted in Figure 5. The overall occlusion rate is defined as the sum of the maximal occlusion rates on the left and right hand sides, respectively. That is, $O = \min(1.0, \max(O_1^L, ..., O_i^L) + \max(O_1^R, ..., O_j^R))$. The desired non-occlusion rate, U, is then defined as the complement of O: U=1-O.

7. Design of virtual editor

7.1. Determining the best shot

It is the editor's task to choose a good sequence of shots from all available ones for the best presentation of the film. The virtual editor makes such decisions based on the current event and the past history. We use the following four criteria to define the score of a shot: *camera matching rate* (M), *information expectation matching rate* (I), *switching appropriateness* (*T*), and *decaying desirability* (*D*). We use the following formula to evaluating shots:

$$S_{cam} = w_m \cdot M + w_i \cdot I(N) + w_t \cdot T + w_d \cdot D ,$$

where w_m, w_i, w_t, and w_d are the weights of the criteria and N is an information ratio that will be explained in more details later. In each frame, the editor module uses S_{cam} to select the best shot among all possible ones. The set of parameters, $W_e = (w_m, w_i, w_t, N, w_d)$, is defined as the editor's idiosyncrasy.

7.2. Camera matching rate

The camera matching rate describes how a shot represents the given event. In order to define this rate,

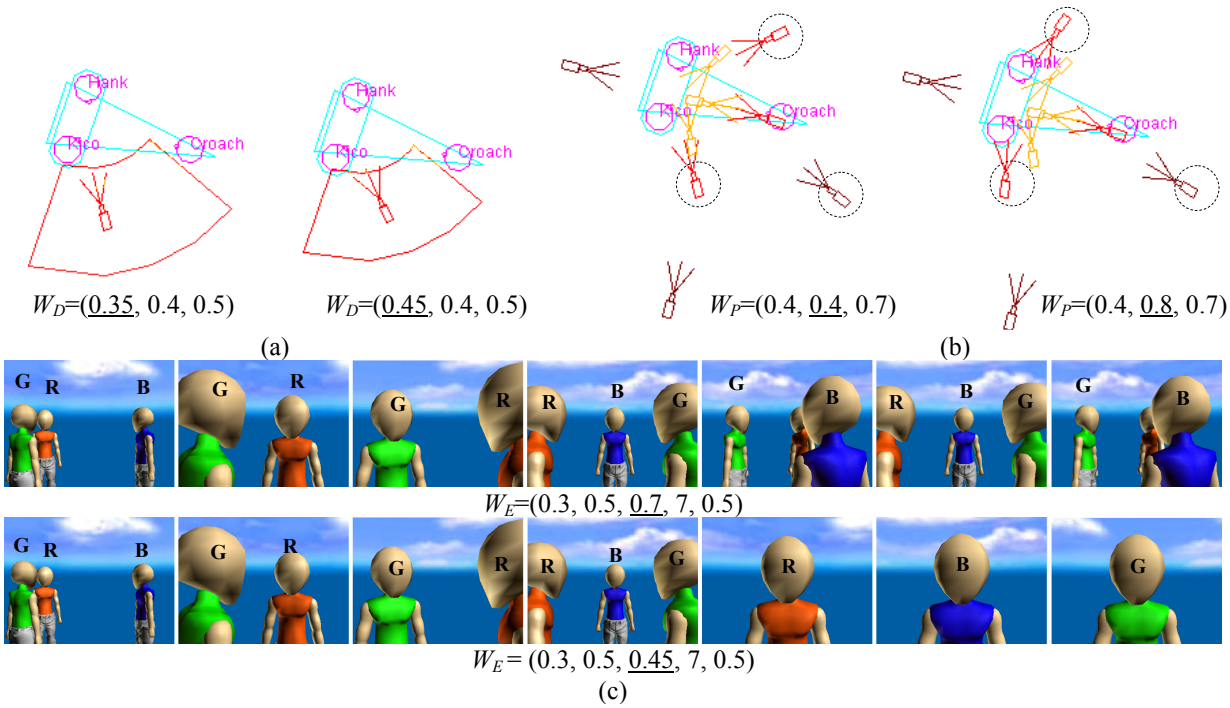

Figure 6. Example shots with different parameter preferences set by the (a) director, (b) photographer, and (c) editor

we first define the characteristic vector of a shot, X, as follows.

$$X = (i_1, \cdots, i_n, j_1, \cdots, j_n), \text{where}$$

$$i_k = \begin{cases} 1, \text{if } k^{\text{th}} \text{ actor is in the focus.} \\ 0, \text{otherwise.} \end{cases}$$

$$j_k = \begin{cases} 1, \text{ if } k^{\text{th}} \text{ actor is in the scope.} \\ 0, \text{otherwise.} \end{cases}$$

where n is number of actors in the scene. Assume that X_c and X_e are characteristic vectors for the current and expected shots. The camera matching rate, M, is defined as the cosine measure of the two vectors:

$$M = \frac{X_c \cdot X_e}{|X_c| \cdot |X_e|},$$

where a higher M indicates a better match.

7.3. Information expectation matching rate

The information that a shot carries depend on the number of actors covered in the shot. An apex shot can carry more overall information about the scene than a close-up shot. However, the audience's expectation on the amount of information varies as the time progresses. For example, initially they are likely to need overview information about the spatial relations among the actors. However, they will start to need details once they have a good grasp of the scene. After taking a close-up shot for some time, an apex shot is usually needed again to re-establish the spatial relations. Assume that the amounts of information carried and accumulated by overview shots and detail shots are denoted by Y_o and Y_d, respectively. The information expectation matching rate, I, can then be defined as follows.

$$I = \frac{1}{1 + \left| \frac{\sum Y_o}{\sum Y_d} - N \right|},$$

where N is the ideal ratio between these two types of information specified by the user.

7.4. Inter-cut appropriateness and decaying factor

The third and fourth criteria for evaluating an editor are on the penalty and desirability of making a camera inter-cut. If the amount of information carried by the cameras before and after the inter-cut (denoted by Y_o and Y_i, respectively) differs too much, the penalty for doing such an inter-cut will also increase. On the other hand, if a shot has lasted for too long, the desire for a change will also increase. We assume that the definition for the appropriateness of performing an inter-cut is defined as follows.

$$T = 1 - \frac{|Y_o - Y_i|}{Q_Y},$$

where Q_y is a quantization factor for normalizing T to the range of [0,1].

D is a decaying factor in the range of [0,1] indicating the desire of maintaining the current shot. It is set to a constant value (D_c) for the current shot and other time-dependent value (D_v) for other shots. D_c is initially larger than D_v. However, as D_v increases to some value larger than D_c, the desire for maintaining the current shot will be lost.

8. Experimental results

The virtual cinematography system described in previous sections has been implemented in the Java language. The inputs to the system are the initial preference settings for each module and an animation script describing when the dialog events happen. The output of the system is shown on a 2D top-view display and a real-time 3D display implemented with Java3D. If the initial result is not satisfactory, the user can adjust the parameters in each module in order to find the best set of parameters fitting his/her own style.

We demonstrate the effects of parameters adjustment for the three modules in Figure 6(a), 6(b), and 6(c), respectively. Three actors are involved and the dialog starts between the actors in red and green clothes (denoted by A_r and A_g). The third actor is in blue (A_b). The sequence of dialog events in the example are A_r to A_g, A_g to A_r, A_b to A_r, A_r to A_b, A_b to A_g, and A_g to A_b.

According to the grouping principles described in Section 3, the director decides to set up five cameras, each of which is for a group of actors, as shown in Fig 6(b). For each camera, an acceptable camera area is determined according to the director's preference. For example, in Figure 6(a), the expected size ratio is changed from 0.35 to 0.45 and the other two parameters remain the same. Consequently, the camera area is moved toward the subject and slightly shrinks. In Figure 6(b), we change the parameter of respecting LoI from 0.4 to 0.8. As a result, the cameras (in dot circles) move toward the LoI to obtain a better depth cue while allowing the target subject to be slightly occluded. For the editor example shown in Figure 6(c), as we adjust the inter-cut appropriateness from 0.7 to 0.45, the desire to maintain the same amount of information across events becomes weaker as the time passes. Therefore, as the audience becomes tired of the current setting, the camera performs an inter-cut from a group shot to a close-up shot and maintains the shot for some time as shown in Figure 6(c).

9. Conclusions

We have proposed a virtual camera system that can generate a sequence of camera shots automatically according to the screenplay. The system is decomposed into three modules imitating the roles in a real filmmaking process. In addition, preference parameters carrying user aesthetic style are also identified for each module. The modulization of the camera shooting process allows one to replace or improve each module easily. The interactive interface also allows one to find empirical parameters for camera settings and carry them over to different animations. Although it remains a challenge to automate the filmmaking process for computer animations, we believe that the virtual cinematography system proposed in this paper will help to provide an assisting tool to reduce the production time and cost.

References

[1] D. Amerson and S. Kime, "Real-time Cinematic Camera Control for Interactive Narratives," in *Proc. of the AAAI Spring Symp. on Artificial Intelligence and Interactive Entertainment*, Stanford, CA, 2001.

[2] D. Arijon, *Grammar of the Film Language*, Communication Arts Books, Hastings House, New York, 1976.

[3] W. H. Bares and J. Lester, "Real-time Generation of Customized 3D Animated Explanations for Knowledge-Based Learning Environments," in *Proc. of the Fourteenth National Conf. on Artificial Intelligence*, Providence, Rhode Island, pp. 347-354, 1997.

[4] W. H. Bares, S. Thainimit, and S. McDermott, "A Model for Constraint-Based Camera Planning," in *Proc. of the 2000 AAAI Symp.*, pp. 84-91. AAAI Press, 2000.

[5] D. B. Christianson, S. E. Anderson, L.-W. He, D. H. Salesin, D. S.Weld, and M. F. Cohen, "Declarative camera control for automatic cinematography," in *Proc. of the Thirteenth National Conf. on Artificial Intelligence*, pp. 148–155, 1996.

[6] S. Drucker and D. Zeltzer, "CamDroid: A System for Implementing Intelligent Camera Control," in *Proc of the 1995 Symp. on Interactive 3D Graphics*, pp. 139–144, 1995.

[7] D. Friedman and Y. Feldman, "Knowledge-Based Formalization of Cinematic Expression and its Application to Animation," in *Proc. of Eurographics*, pp.163-168. Saarbrucken, Germany, 2002.

[8] K. Kennedy , R. E. Mercer, "Planning Animation Cinematography and Shot Structure to Communicate Theme and Mood," in *Proc. of the 2nd Intl Symp. on Smart Graphics*, pp.1-8, Hawthorne, New York, 2002.

[9] N. Halper, R. Helbing, and T. Strothotte. "A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence," in *Proc. of EUROGRAPHICS 2001,* 2001.

[10] N. Halper, P. Olivier, "CAMPLAN: A Camera Planning Agent," in Smart Graphics, from *Proc. of the AAAI Spring Symp.*, pp. 92-100, Menlo Park, 2000.

[11] L.-W. He, M. F. Cohen, and D. H. Salesin, "The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing," in *SIGGRAPH 96 Proc., Computer Graphics Proc., Annual Conference Series*, pp. 217-224, 1996.

[12] R. Thompson, *Grammar of the edit*. Oxford [England], Boston, 1993.