AUTOMATED CAMERA MANAGEMENT BASED ON EMOTIONAL ANALYSIS OF THE FILM SCRIPT

Jorge Carrillo de Albornoz
Facultad de Informática, Universidad Complutense de Madrid
C/ Prof. José García Santesmases, s/n. 28040 Madrid (Spain)
jcalbornoz@fdi.ucm.es

Pablo Gervás
Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid
C/ Prof. José García Santesmases, s/n. 28040 Madrid (Spain)
pgervas@sip.ucm.es

ABSTRACT

The expressiveness and possible impact on the viewer are some of the most important characteristics in order to film a cinema scene. However, it is quite rare to see this kind of features in the different approaches of automatic Camera Management Systems in virtual environments. The diverse studies that have used in some way cinematographic techniques are more focused on their geometrical aspects, or on how to avoid the occlusion with other scene objects, rather than in the emotive aspect of it. Another fundamental issue is how to obtain the needed information for this task. Frequently, this information is acquired in a previous evaluation of the scene, or generating templates associated to the characters actions in the virtual environment. This work presents a prototype of an automatic Camera Management System focused on the emotive aspect of the cinematographic techniques. It uses different methods of Natural Language Processing over the story script to obtain the necessary information to manage the virtual cameras correctly. The prototype is evaluated and compared with real scenes for the validation.

KEYWORDS

Virtual Environment, Automatic Camera Management System, Cinematographic Techniques, Emotions, Script, Natural Language Processing

1. INTRODUCTION

Cinematography is the art of representing and transmitting information using images. In this context, the correct positioning and moving of the cameras are crucial when deciding the information that should be filmed and transmitted, and it constitutes one of the main functions of a cinematographic director. This problem has been tackled in computer graphics, where several approaches arise to achieve the automatic positioning and movement of cameras in virtual environments. Actually, automatic camera management systems (CMS), [8], solve a great deal of problems and cover a good variety of applications, including storytelling or 3D games.

There are many techniques in the different fields of computer graphics which allow to automate camera management. However, very few of them take advantage of the different principles and notions developed in the film industry throughout the twentieth century. Features such as expressiveness, smooth movements between sequence transitions, or the possible impact on the viewer could achieve better realism in these applications, while enriching them. In the last years different research efforts have emerged that attempt to abstract this knowledge, modeling it according to the techniques used, and solving, depending on the features of the systems proposed, the problems that arise in moving from the abstract world of cinematographic techniques to the geometry and mathematical world of virtual environments. These systems base their

decisions on the geometrical and spatial aspects of the different techniques of cinema[1], trying to avoid occlusions between the camera and the objects in the scene, as well as shots where the geometrical disposition of elements does not show clearly what is happening, i.e. a dialog between three characters in which one of them blocks another. However, they generally do not consider one of the main features of these techniques, namely the emotional and dramatic aspect underlying the different parameters of the techniques, which gives great expressiveness and visual richness to films.

Another important problem in CMS is how to obtain the necessary information to determine at each moment the best camera settings for a given situation. In most cases, this task is either addressed through default settings for certain actions associated to the elements in the virtual environment, or generated by hand in a previous process of evaluation of the story events that assigns to each action the adequate parameters. This problem increases in the few approaches that have introduced in some way the emotional aspect in the camera management.

This paper presents a study of an automatic system of cinematographic direction based on emotions, capable of determining in real time what is happening in the scene and the emotional context in which it happens, along with the selection of the best way of showing it by applying the different techniques used in the film world. The study proposes the use of various methods of Natural Language Processing on the script, in order to obtain the required information for the automatic management of the virtual camera, understanding the different cinematographic techniques from a more dramatic and emotive point of view, and achieving a greater impact on the viewer. A preliminary prototype has been developed in order to evaluate the applicability of the ideas exposed above.

2. REVIEWING THE LITERATURE

Early works in the field of automatic camera management focused on how to adapt the different cinematographic techniques to different representations. Christianson et al. [5] presents a system based on *film idioms* and grammars, where knowledge is represented by a film grammar using the Declarative Camera Control Language (DCCL), and where the various situations and actions of the characters are collected in an animation trace created in a previous simulation. He et al. [11] proposes the interactive system *The Virtual Cinematographer*, where cinematographic knowledge is encapsulated in the components *Film Idioms* and *Camera Modules*, and each action of the *avatars* is associated with a set of positions and camera movements. Focused on solving the problem of possible occlusions with objects through the use of cinematographic techniques, Bares et al. [3] and [4] use constraints to model cinematographic knowledge, where goals are provided by a user or specific software. Christie et al. [6] and Languénou & Christie [7] enrich this approach by making use of optimization techniques to solve the constraints.

Armeson & Kine[1] present a hybrid system where the cinematographic language is represented by *film idioms* modeled as constraints, and the needed information is generated by a narrative module which collects the different characters actions and their eventual consequences. Armenson and Kine introduce the concept of emotion in the parameter *emotion* associated with characters actions. A similar approach is followed by Halper et al. [10], with cinematographic techniques represented by templates linked to actions or specific situations, using the *Emotions Templates* and the *Shot Library*. There are multiagent systems that use agents to represent both the characters and the camera. These also associate templates to the character actions or feelings, and the parameters of the camera management are associated to the agent states (Tomlinson et al. [14]). Hornung et al. [12] propose a system focused on the emotive and dramatic aspects of cinematographic techniques, which obtains the information from different templates related to narrative events.

The main weakness of this type of systems is the need of some level of human interaction to obtain the information necessary to know what is happening in the scene and how, along with the poor attention paid to the emotive aspect in most of the automatic camera management approaches. In the following sections, a prototype that tries to cover these gaps is presented, making use of different methods of Natural Language Processing, and focused on the emotive aspect of the cinematographic techniques.

[1] A detailed description of these techniques is beyond the scope of the paper. Interested readers can consult [2], [13].

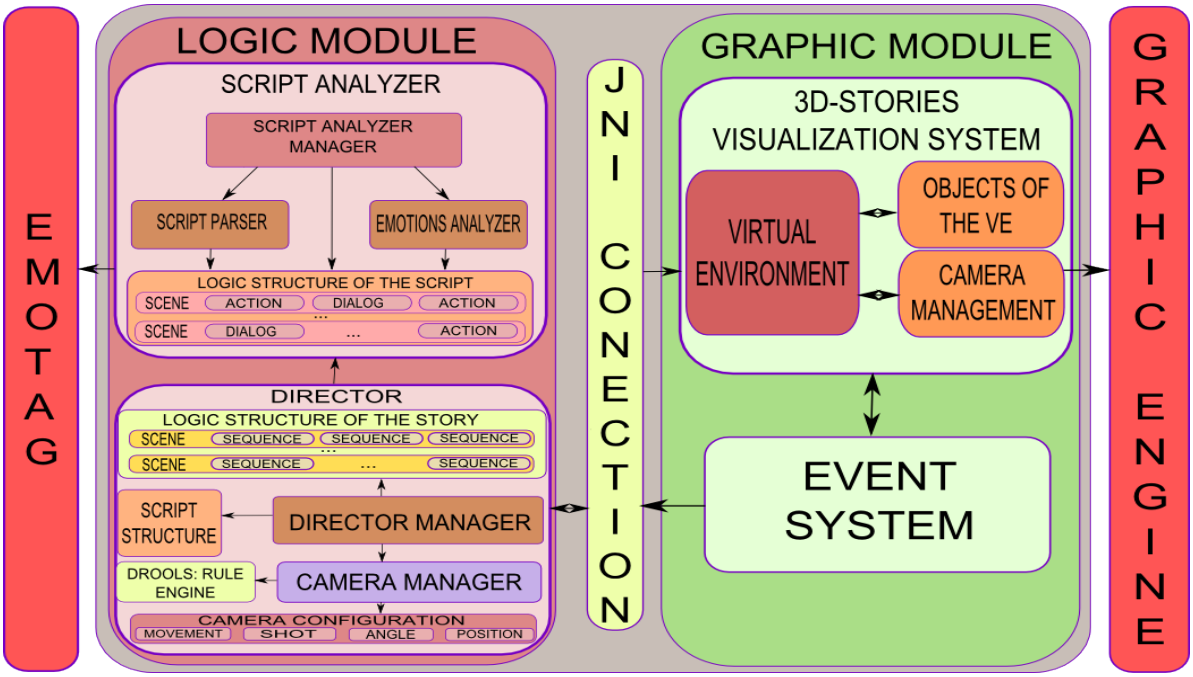
3. THE SCRIPT-BASED EMOTIONAL CINEMATOGRAPHIC CMS

The preliminary prototype of the automatic CMS based on emotions analyzes the script of the story in order to extract its elements and its emotional content. With this information, a logic structure is created. This structure, along with the events thrown by the characters in the virtual environment, allows the system to determine the best camera configuration. The following subsections describe the process in detail.

3.1 System insight

As shown in Figure 1, the system presents two main modules: the *Logic Module* and the *Graphic Module*. The identification of the script elements, the analysis of these elements, the synchronization of the story flow and the decision process are carried out by the *Logic Module*, while the *Graphic Module* manages all processes that involve the interaction with the graphic engine.

Figure 1. Architecture of the preliminary prototype



The two principal components of the *Logic Module* are the *Script Analyzer* and the *Director*. The *Script Analyzer* analyses the script text and generates a structure that represents the different elements in the script. The *Director* transforms that structure into a story that consists of scenes and sequences and, using the information provided by the events of the *Graphic Module*, determines in real time the best way of playing the sequence. As a result, it returns a configuration of the parameters needed to manage the virtual camera.

The *Graphic Module* is composed of the *3D-Stories Visualization System*, which loads and handles story elements (objects and characters) to reproduce the story flow. Story elements have associated two kinds of actions (movement or dialogue) and launch system events whenever these actions start or end. The *Graphic Module* has a specific camera management submodule responsible for interpreting the configurations proposed by the *Logic Module* and relating them to the geometric values of the graphic engine. The *Graphic Module* also includes an *Event System* which captures the actions thrown by each story element, and notifies the *Logic Module* of the start and end of the actions, crucial to the synchronization between the two modules.

To select the best camera configuration, the application must be able to distinguish the actions of the story and their features, as the characters involved or the emotional charge. To do this, the system makes use of a logic structure that represents the story in chronological order, which is composed of *scenes*. A scene in turn

is comprised of *sequences* (chronologically ordered as well) that encapsulate the required parameters for selecting the appropriate camera configuration.

As a result, sequences are considered the primary unit of the system, and determine the choice of different camera configurations. According to this and the categorization presented in Arijon [2], two types of basic sequences are considered: *action sequences* and *dialogue sequences*, which in turn are specialized in *movement actions, static actions, movement dialog* and *static dialog*. The first category of actions comprises those actions performed in the story that do not involve a dialog between characters and can be visualized using a predefined set of techniques, making a distinction between dynamic and static techniques. Quite the opposite, the dialogue sequences represent actions that involve a dialogue between two or more characters, differentiating between techniques for static dialogues and techniques for dialogues with movement.

As a first step, in order to correctly classify the actions in the story between the two types of sequences previously mentioned, the system analyzes the script and identifies the elements that compose each scene, their properties, the text describing the actions, the characters and the monologues of each character, and generates the script structure described above. To this end, the system takes two files as input: the script text in the standard script format as provided by the script edition software *Celtx* [15], and the list of the characters names that appear in the scene.

In a second step, each of these elements is analyzed and tagged with a set of basic emotions, using the software Emotag [9], to determine the emotional intensity immersed in the script. Emotag is a tool for automated mark up of text with emotional categories. Given a text input, it tags it with a set of emotional categories, each one weighted with a percentage value indicating the probability of associating that category with the input.

Both steps are carried out by the component *Script Analyzer*, in the *Logic Module.* As a result, the logic structure of the story is generated that will drive the *Director* in the selection in real time of the appropriate camera configuration for the sequence under consideration. To do this, the application generates an action sequence for each action element identified in the script, along with a dialog sequence for each set of monologues, gathering then according to the script timeline.

In a third step, the *Director* makes use of a set of events, generated by the *Graphic Module*, which indicate the start and end of the characters movements and dialogues. These events allow the system to synchronize the story flow and to specialize the sequences in static or movement dialogues. Moreover, a third kind of event has been introduced to identify when a new character joins or leaves the scene, so that it is possible to synchronize situations where a character leaves the current scene to immediately appear in the next scene but in a different place. With this information, the story structure and the events caused by the *Graphic Module*, the *Director* uses the rule engine to determine the best camera configuration for the situation of the scene. This decision is made based on all the parameters analyzed in the previous steps, such as the kind of sequence (action sequence or dialog sequence, and static or dynamic sequence), the emotional intensity, the actors of the sequence, etc. The rule engine is the component of the system that encapsulates the cinematographic knowledge as a body of rules. Each rule is composed of preconditions based on the attributes of the sequence to be visualized, and a consequence corresponding to a particular camera configuration expressed as a camera configuration object. An example is shown in Figure 2, which represents two camera configuration objects in XML format generated in the simulation.

Figure 2. Camera Configuration objects in XML format.

```
<CameraConfiguration>
    <Distance value="LONG_SHOT"/>
    <Angle value="20"/>
    <PrincipalObjects>
        <Object>ARAGORN</Object>
        <Object>EOWYN</Object>
    </PrincipalObjects>
</CameraCofiguration>
```

```
<CameraConfiguration>
    <Movement type="PAN">
        <position>FULL_SHOT</position>
    </Movement>
    <Angle value="10"/>
    <PrincipalObjects>
        <Object>ARAGORN</Object>
    </PrincipalObjects>
</CameraCofiguration>
```

These objects represent the principal parameters of a real camera configuration, like the kind of shot (the system differences between 9 possibilities, 3 for close shots, 3 for medium shots and 3 for large shots,

depending on the emotional intensity, the number of characters involve in the action, etc.), the vertical angle, the actors of the sequence, the time of the configuration or the kind of movement if it exists (and their parameters, such as the distance of the panoramic movement).

Once the *Director* has determined the best camera configuration, the object is interpreted and executed by the *Camera Management* component of the *Graphic Module*. In this process, the specific parameters of the configuration are fixed to geometric values in the virtual environment, depending on the geometric positions of the different objects and characters present in the scene. The last two steps (the decision making and the graphical interpretation) are repeated whenever some event occurs or the present configuration expires.

This high level representation of the camera configuration decouples the system from the geometric aspect and the graphic engine, allowing adaptation to other engines with few changes in the *Graphic Module.*

3.2 Results

In order to validate the system and the quality of the results, a fragment of the film "The Lord of the Rings: the Return of the King"[2] was simulated, covering situations where the most frequent cinematographic techniques are applied, i.e. the change of a character between scenes, static and movement action sequences, conversations with a medium-high emotive charge, and so on. A demonstrative video of the system, where all movements and positions of the camera have been automatically generated, is available at [17].

As there are no standard evaluation methods available for this task, two informal methods have been used to validate the results. First, the cinematographic techniques used (and their frequencies) in the film, the simulation and the "ideal situation" according to the concepts studied in the literature about cinematographic techniques, are compared. Second, a more in depth comparison between the film fragment and the simulation is presented, classifying the sequences into similar and correct, different but correct, and different and incorrect. It is important to notice that the simulation and the film are not exactly equal, neither in the spatial disposition nor the shape of the elements in the scene. Also, some characters actions could not be simulated because the models employed in the simulation had not been specifically developed for this work.

Table 1 shows that the simulation has resulted in a higher number of shots than the film, while this difference is not so marked with respect to the "ideal situation". This is because the time assigned to each camera configuration, which depends on the emotional intensity, is shorter in the simulation than in the film or in the ideal situation. However, the simulation is quite similar to the film and the ideal situation when situating the camera and its focus, using long shots to visualize the entire scene as a whole, and medium and short shots to emphasize a character.

Table 1. Comparison between the simulation, the film and the ideal situation

Cinematographic technique	Ideal situation	Film	Simulation
Long shots	6	3	7
Medium shots	7	2	15
Close shots	4	9	0
Shot position focus on multiple actors	5	3	7
Shot position focus one actor	12	11	15
Pan movement with long shot	4	2	5
Pan movement with medium shot	1	2	2
Zoom movement from long to medium shot	1	1	0
Zoom movement from medium to close shot	0	0	0
Horizontal Travelling	0	1	Not considered

Another important discrepancy is related to the different use of medium and close shots between the simulation and the perfect situation, which is even more obvious in the film. These kinds of shots are strongly related to the emotional intensity and to the particular director interpretation. In fact, human directors usually differ considerably in the use of these shots. Nonetheless, in spite of this intrinsic subjectivity, the simulation only presents a few sequences where the system does not reach the desired emotional intensity.

[2] Obtained from [16]

Conversely, the simulation is very similar to the film and the ideal situation in the number of movements. The type of these movements in the simulation is closest to the ideal situation than the film. The film even uses a special movement that is not covered by this preliminary system. Differences between the simulation and the film are due to the subjective director interpretation, especially evident in movement actions: where the literature usually suggests a panoramic motion, the film often uses a *travelling* plus a static position.

The second evaluation is based on the comparison between the sequences in the original film and the simulation, where the strengths and weakness of this approach can be appreciated. In the following subsections, the different possibilities are discussed and illustrated with examples.

3.2.1 Similar and correct sequences

Maybe the most similar sequence between the film and the simulation is that in which *Aragorn* and *Eowyn* hold a conversation, which is also correct from a cinematographic point of view. In both, the action starts with a configuration that presents the two characters from a long shot. The film continues with a close shot showing *Aragorn* wrapping *Eowyn* up. The simulation continues with a medium shot on Aragorn. This is because the action of wrapping *Eowyn* is not represented explicitly in the simulation (due to restrictions on the available animations). The action that follows presents the dialog between *Aragorn* and *Eowyn*, and is nearly identical in the film and the simulation: while the film uses a close shot, the simulation makes use of a medium one. During the longest monologue of *Eowyn*, the simulation focuses twice on *Aragorn* and twice on *Eowyn*, whereas in the film a single transition is used. In both cases, the sequence ends with a long shot on the two characters.

Even if the two approaches are equivalent, it is obvious that an exact matching between the simulation and the film is impossible to achieve, as the simulation is slightly different and some movements are omitted. Besides, each director puts his personal stamp on each sequence. For instance, the use of more close shots in the film than in the simulation can be justified because the emotional intensity computed by the system is lower than the one interpreted by the director.

3.2.2 Different but correct sequences

This situation is illustrated in the sequence in which *Aragorn* goes into the castle hall where *Eowyn* is asleep. Even if the disposition of element is not exactly the same, one can appreciate the difference in techniques between the film and the simulation. In the film, the whole sequence is filmed with a horizontal travelling that shows the room where *Eowyn* sleeps and *Aragorn* moving into the room and going to the bonfire in the middle of the room. The sequence ends with a medium shot on Aragorn. In the simulation, the sequence starts with a long shot on both characters, moving away while Aragorn goes to the table, and continues with a panoramic movement on him.

Even if the techniques in the film are perfectly valid, because the movement covers the whole scene and the important actions, this sequence denotes a subjective vision of the director, starting with an occluded shot and showing progressively the scene to the spectator. On the contrary, the simulation presents (as in every introductory sequence) a long shot on the characters in order to situate the spectator in the room and continues with the movement of *Aragorn* to the table, which is the main action in the scene. Both sequences are equally appropriate, but the simulation makes use of more general techniques while the film shows a more subjective and personal interpretation of the story.

3.2.3 Different and incorrect sequences

This situation is derived from one of the main weakness of the system. It can be clearly observed in the sequences following the movement of *Aragorn* to the table or to the bonfire, respectively. In the film, this sequence focuses on *Aragorn* taking a log from the bonfire, continues with a long shot on *Eowyn* lying on a sofa, and finishes with a long shot on *Aragorn* approaching her. In the simulation, the sequence starts with a long shot of both characters, in which, due to the distance between them, only *Aragorn* is filmed, and continues with a medium shot on each character, and a long shot that does not show any of them. The sequence ends, as in the film, with the motion of *Aragorn* toward *Eowyn*, but using a panoramic movement.

The reason of this behavior is that the selection process of the camera configuration does not take into account any geometrical information about the character positions, so that the result does not show the main elements, as indicated in the configuration. Besides, the fact that the simulation and the film are not exactly

equal (in the simulation *Aragorn* does not reach down to get the log), implies the use of different techniques, like the use of two medium shots to visualize the characters in the scene.

4. DISCUSSING ADVANTAGES AND LIMITATIONS

In this section the advantages and limitations of the system are tackled and explained in detail, together with some possible ways of improvement.

As mentioned above, most automatic CMS based on cinematographic techniques focus on the geometric features, such as [4, 5, 7]. Thus, they do not take into account the importance of emotive weight of each scene, fundamental for the visual richness of classical cinema. In contrast, the present work is completely focused on the emotive aspect, allowing a greater expressiveness. For instance, the progressive use of close shots depending on the emotional intensity of a dialog; or zooming and panning taking into account the highly emotive action scenes, instead of using static personal shots.

The use of Natural Language Processing techniques to analyze the script format allows identification of the different kinds of sequences in the story. The software *Emotag* allows automatic identification of the emotional intensity of the story, which, along with the events produced by the virtual environment, improves noticeably the selection of the camera configuration.

According to the tests realized, having excluded the geometrical information from the decision-making process of choosing the most suitable camera has advantages and drawbacks. It provides the ability of acting from a generic and independent point of view, and it allows its use in any virtual environment: adapting its camera management for the interpretation of module configuration and creating an event handler for the described actions. However, it delegates on the camera management the geometrical calculations (instead on the *Director component*), causing incorrect situations as the ones described above.

One of the main problems for the accomplishment of this work is the early stage of development of the emotion recognitions systems evaluated. Nevertheless, in the last year they have improved substantially and fast enhancement is expected on the short term. The current version does not give a high success rate when analyzing different texts. In spite of these lacks, the use of *Emotag* improves the general CMS behavior. Another localized problem with this particular emotional recognition system is that it has been trained over a corpus of children's tales, which substantially constrains the vocabulary used for the emotional tagging.

5. CONCLUSIONS AND FUTURE WORK

An automatic system of cinematographic direction for virtual environments based on emotions has been presented, capable of selecting in real time the best camera configuration according to what is happening in the scene, taking as inputs the script of the story, the name of the characters and the events generated in the virtual environment. The described system introduces the emotional aspect in the various cinematographic techniques for camera management. The use of different techniques of Natural Language Processing on the script can provide enough information for this task and allows the classification and identification of the sequences that compose each scene in order to select optimal camera configuration.

This preliminary prototype has shown satisfactory results, though several problems that must be solved in the future have been identified. The system has accomplished with success the task of generating automatically the different camera movements and positions without needing to manually produce the previous camera configurations depending on the virtual environment. It has also demonstrated the visual advantages of using cinematographic techniques, particularly their emotional aspect, in camera management.

Another interesting contribution is the use of the script as a valuable resource for obtaining the information needed for camera management. Along these lines, a more extensive analysis using advanced Natural Language Processing techniques could provide much more information relevant to the task in hand. Additionally, this information could be applied to light management or characters direction as well.

Some improvements have been identified as future work. First, it would be interesting to allow the *Director* to control the characters in the scene, combined with feedback to indicate if the selected configuration is behaving as expected. This would allow relocation of characters whenever the feedback indicates the best possible camera configuration does not visualize correctly the characters owing to their

spatial disposition. Second, a further evaluation should be carried out to formally evaluate the system proposed, reinforced with an exhaustive study of the existing emotions recognition systems and the evaluation of different methods for computing the emotional intensity. Third, it would be worthwhile to consider more types of sequences, i.e. romantic or terror, to allow a better adjustment of cinematographic techniques. This would require of a system capable of identifying such patterns in the script. There exist some proposals in this direction that make use of machine learning and corpus-based methods. Finally, a detailed study of techniques for smooth transition between different camera configurations is under way.

ACKNOWLEDGEMENT

This research is funded by the Spanish Ministry of Education and Science (TIN2006-14433-C02-01). This research has been partially funded by UCM and CAM through the IVERNAO project (contract CCG08-UCM/TIC-4300). Also, this research has been partially funded by the Comunidad Autonoma de Madrid (CAM) and the European Social Fund (ESF) in the program IV PRICIT.

REFERENCES

[1] Amerson, D. & Kime, S., 2001. Real-time cinematic camera control for interactive narratives. *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment.* Stanford, CA, USA, pp. 1-4.

[2] Arijon, Daniel. 1991 (originally published 1976). *Grammar of the Film Language.* Silman-James Press.

[3] Bares, William H. et al., 1998. Real-time Constraint-Based Cinematography for Complex Interactive 3D Worlds. *Proceedings of Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference , AAAI Press*, pp. 1101-1106.

[4] Bares, William H. et al., 2000. A model for constraint-based camera planning. *Proceedings of AAAI Spring Symposium on Smart Graphics*, pp. 84-91.

[5] Christianson, D. et al., 1996. Declarative camera control for automatic cinematography. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press, pp. 148-155.

[6] Christie, M. et al., 2002. Modeling camera control with constrained hypertubes. *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*. London, UK, pp. 618-632.

[7] Christie, M & Éric Languénou, 2003. A Constraint-based Approach to Camera Path Planning. *Proceedings of 3rd International Symposium on Smart Graphics*. Heidelberg, Germany, pp. 172-181.

[8] Christie, M. et al., 2005. Virtual camera planning: A survey. *Proceedings of 5th International Symposium on SmartGraphics*. Frauenwoerth, Germany, pp 40-52.

[9] Francisco, V. & Gervás, P., 2006. Automated Mark Up of Affective Information in English Texts. *Proceedings of Text, Speech and Dialogue (TSD 2006)*. pp 375–382.

[10] Halper, N. et al., 2001. A camera engine for computer games: Managing the trade-off between constraint satisfaction and frame coherence. *EG 2001 Proceedings,* pp. 174-183.

[11] He, Li wei et al., 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.* New York, NY, USA. ACM, pp. 217-224.

[12] Hornung, A. et al., 2003. An Autonomous Real-Time Camera Agent for Interactive Narratives and Games. *Proceedings of Fourth International Workshop on Intelligent Agents.* Kloster Irsee, Germany, pp. 236-246.

[13] Mascelli, J.V., 1965. *The Five C's of Cinematography: Motion Picture Filming Techniques*. Hollywood.

[14] Tomlinson, B. et al., 2000. Expressive Autonomous Cinematography for Interactive Virtual Environments. *Proceedings of Fourth International Conference on Intelligent Agents.* Barcelona, Spain, pp. 317-324.

[15] The Celtex website, *http://www.celtx.com/*.

[16] The Internet Movie Script Database website, *http://www.imsdb.com/*.

[17] http://nil.fdi.ucm.es/index.php?q=node/191