

# A Pattern-based Tool for Creating Virtual Cinematography in Interactive Storytelling

Pei-Chun Lai, Hui-Yin Wu, Cunka Sanokho, Marc Christie, and Tsai-Yen Li

National Chengchi University, IRISA/INRIA Rennes  
100753017@nccu.edu.tw, hui-yin.wu@inria.fr, cunka.sanokho@irisa.fr,  
marc.christie@irisa.fr, li@nccu.edu.tw

**Abstract.** In this work we design a tool for creators of interactive stories to explore the effect of applying camera patterns to achieve high level communicative goals in a 3D animated scenario. We design a pattern language to specify high level communicative goals that are translated into simple or complex camera techniques and transitions, and then flexibly applied over a sequence of character actions. These patterns are linked to a real-movie shot specification database through elements of context such as characters, objects, actions, and emotions. The use of real movies provides rich context information of the film, and allows the users of our tool to replicate the feel and emotion of existing film segments. The story context, pattern language and database are linked through a decision process that we call the virtual director, who reasons on a given story context and communicative goals, translates them into the camera patterns and techniques, and selects suitable shots from the database. Our real-time 3D environment gives the user the freedom to observe and explore the effect of applying communicative goals without worrying about the details of actor positions, scene, or camera placement.

**Keywords:** context-aware camera control, interactive storytelling, virtual cinematography, computer animation

## 1 Introduction

In film theory, Jakob Lothe [1] describes the mechanisms of discourse as how “people are presented through characterization, and the transmitted content filtered through narrative voices and perspective.” Discourse, such as that in film, is how the director guides and awes the audience to understand the multiple emotions, feelings, and other hidden messages in each story scenario.

With the advances of multimedia and virtual storytelling comes the benefit of quick and realistic cinematography in 3D gaming environments. Existing machinima tools can automate camera placement to capture action in well-defined story scenarios based on certain elements of context such as number of characters, action, and scene type. However, the power of cinematography lies in its capacity to communicate the story-actions, emotions, and morals—which we



**Fig. 1.** Example of an intensification sequence from the film *White Ribbon*. Our objective is to encode both the compositions of the shots (hence the eyes and facial annotations) and their context, and reproduce the discourse pattern corresponding to intensification.

term as communicative goals. From film literature, we observe a number of well-established camera techniques—which we define as “patterns” of shots in our system—used to convey certain messages. An example of this is the “intensification” technique that consists in placing the camera closer and closer to the actors at each cut in order to communicate a building of emotion within the dialogue (see Fig. 1). As in the making of film, creators of interactive stories and games for 3D environments would need cinematographic tools that can achieve multiple communicative goals in each scene, but also respond to dynamically changing story lines. Thus, a simple but expressive method of defining and applying camera techniques to construct elaborate levels of discourse is still highly desired.

Thus our objective is to develop smart tools for creators of interactive stories to search and experiment with various camera techniques that can fulfil multiple communicative goals in a given scenario, find shots that fit camera patterns for a sequence of actions, and observe their effects in a real-time 3D environment.

This objective is achieved in three steps: (1) we propose a means to create a database of shot specifications by annotating shots from real movies in order to reapply them, (2) we design a pattern language to express communicative goals as a construction of cinematographic techniques, and to express techniques as patterns defined over sequences of shots, and (3) we propose a decision-making process (a “virtual director”) for smart cinematography that finds suitable shots and transitions based on the current story context and specified communicative goals, and performs the according shots and edits in a real-time 3D environment.

## 2 Related Work

Related work in computer animation and interactive storytelling reveals the growing importance of automated camera planning techniques in virtual environments for storytelling and gaming scenarios. He [2] was the first to propose an automated cinematography system in character-based interactive scenarios, and established a state machine mechanism to describe the rules for camera control. Other constraint-based camera systems include [3, 4] for game scenarios. Such methods view cinematographic rules as constraints to camera planning problems, which works effectively for simple scenarios such as dialogue and basic movements such as walking, but do not account for contextual story elements

such as emotions, genre, or style. Work on interactive storytelling have also proposed the use of virtual cinematography, demonstrating the importance of defining camera idiom languages [5], communicate emotions in a shot [6, 7], or ways to interpret perspective in terms of the story [8].

The growing applications of virtual cinematography in interactive storytelling points to the development of tools that can create story-driven and context-aware virtual cinematography.

The machinima production tool, Cambot, developed by [9] implements a library of “blockings” which are stage plans (the scene in the 3D environment) of where characters are placed and how they move around the stage. Then cameras are placed by looking for shots that are suitable for a given stage and blocking. This method has the strength to bind the blockings, staging, and camera motions, ensuring the tight coordination between the actors, scene, and camera before a scene is shot. In our work, using the toric manifold method proposed by [11] we open the capacity for computing camera positions dynamically for scenarios with unknown stage and actor configurations, and dynamic gaming scenarios. We also provide a method to structure links between shots that maintain a style over a series of story actions.

The Darshak system [10] uses the bi-partite model of story to generate cinematic discourse for a given story plan. The system has the strength of reasoning over the current story scenarios by designing its discourse planner on the level of scenes, instead of individual actions. The system’s camera configurations must be defined in advance for each context that will be shown (such as Show-Robbery). In our work we emphasise the reusability of existing patterns across different story scenarios that have similar context or emotions, and also provide multiple stylistic camera options for desired communication goals.

From the literature we observe the necessity for camera tools that can identify story context, and provide familiar camera techniques for creators of 3D interactive stories to apply and experiment with combinations of techniques and thus achieve their communicative goals.

### 3 Overview

The main objective is to develop a tool for creators of interactive storytelling to explore means of achieving communicative goals in 3D environments. The tool should allow easy specification of camera patterns with well-established cinematographic techniques.

This objective can be described in three parts: a language for users to specify style and communicative goals, a database of shot specifications annotated from real movies, and a decision-making process to identify story context and communicative goals and translates them to shot specifications from the database.

In order to achieve the above objectives, we design a pattern-based representation of discourse to encode communicative goals and camera techniques, the shot database, and a virtual director that reasons on the story context, goals, and performs the lookup in the database for shot specifications (see Fig. 3).

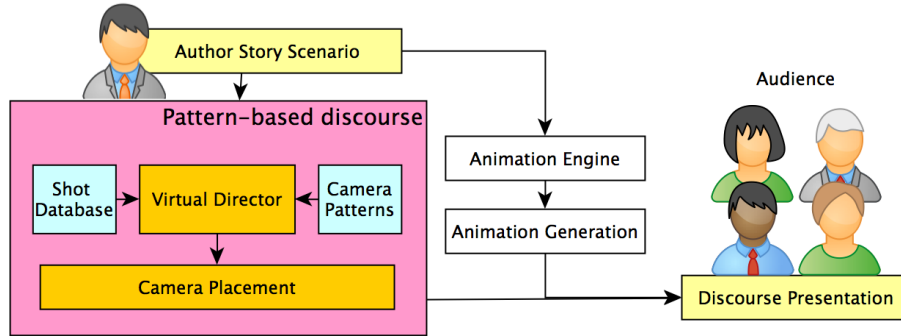


Fig. 2. Role of the discourse based camera in the framework.

The shot database is established by annotating shots from real movies for the purpose of establishing a link between the story context and shot specification in existing film literature. The virtual director receives context information on the upcoming scene on three aspects: (a) the current actions being carried out, (b) the characters involved and (c) the communicative goals for that scenario, and translates communicative goals into camera patterns which are comprised of specific shot types or transitions. A lookup in the database is performed based on the shot type and story context, and produces the most suitable shot specifications and transitions for the given scene. The shots selected by the director are filtered by occlusion detection and the output of our tool is the camera position of the best selected best shot for the scenario. The camera is then placed in the virtual environment using the placement algorithm and the scene is shot.

### 3.1 Context and Communicative Goals

The first input of the tool is the 3D animated scenario comprised of a scene, virtual actors, communicative goals for each scene (such as *build emotion*, *compassion*, or *horror*), and high level actions that are carried out (such as *escape danger*, *tell secret*, or *discuss murder*). This gives the virtual director an idea of what actions are being carried out, which characters are involved, and the physical locations in the 3D environment.

Our story is represented by an XML scenario which specifies the scene, character position and actions, and story context (such as emotion parameters) in high level commands. The specification of the story context also defines the point at which the camera can make a cut. A sample of the context description is:

```

<AnimContext Scene="discontent" Action="tellSecret"
Goal="buildEmotion" Actor1="millionaire" Actor2="woman" />
  
```

Where **Scene** represents the name of the scene. **Action** represents the action that is carried out by the characters, such as walking, talking, running, etc. **Goal**

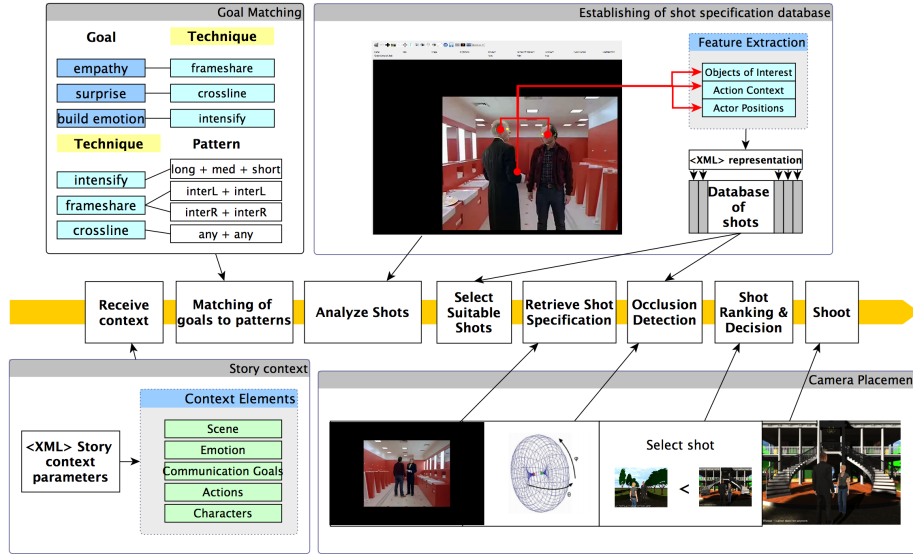


Fig. 3. Workflow of the context aware camera control which involves four main components: the given story context, a pattern library, shot database, and camera placement.

is the communicative goal that the author wishes to convey, such as emotional intensify, horror, affinity, isolation, etc. **Actor + Number** represents the IDs of the character involved in the specific action. In the above example, the XML represents the statement “*millionaire is telling a secret to woman* with communicative goal of *buildEmotion*.” Each scene may have any number of actions and communicative goals, and the virtual director parses and extracts all the necessary elements.

A communicative goal can then be expressed with a collection of techniques. Each technique can then be described in terms of a pattern. A pattern defines constraints on the types of shots and types of transitions between shots that are needed to encode the technique. In the following example, the technique “intensify” is expressed by first selecting the longest possible shot (longest in terms of shot size, i.e. farthest from the characters). Then, transitions should be made to shots closer than the initial shot (i.e. getting closer and closer to the characters).

```
<technique id="intensify">
  <initial operation="longest"/>
  <transition operation="closer"/>
</technique>
```

Another way to define intensify could be more strict, by directly constraining the type of shots in the sequence, starting with a long shot, followed by a medium shot, and then a close shot.

```

<technique id="intensify3">
  <initial shot="long"/>
  <next shot="medium"/>
  <next shot="close"/>
</technique>

```

Our system provides a number of parameters that can be used to define the techniques. These patterns include operations such as *any* (any shot), *opposite* (opposite angle of a shot), *sameside* (check the characters are framed on the same side of the shots), *longest*, *closer*, and *continuity* (respect of classical continuity editing rules between shots), or specific shot types such as *internal*, *external*, *parallel*, *long*, *medium*, and *close*, etc. Since directors may want to convey many levels of discourse in a single scene, a benefit of our tool comes with the capacity to combine a number of techniques using multiple communicative goals (e.g. buildEmotion with crossline). Communicative goal can be achieved with different techniques (one of them will be selected at runtime according to the quality of the first shot in the sequence).

## 4 Shot Database

We annotate and output shot specifications from real movies in to establish the shot database. The use of real movies provides the system with information on what decisions directors make in real filmmaking. Each shot specification in the database provides information that links elements of visual composition in cinematography (pertaining to where characters positions, shot angle, size, distance...etc.) to elements of context (what is happening in the story).

Both geometric and story information is annotated in the shot specification. As in the example of Fig. 1, each shot is annotated with eye positions, head position, shot distance, and body orientation (the first two through a point-and-click, and the other two through estimation). Moreover, story context is annotated in the form of characteristics, where the actor name, the action being carried out, and the object on which the action is performed is recorded. A sample output is as below. Note that the object can be other actors or inanimate objects:

```

<Shot ShotName="ribbon-b-6" NbActors="2">
  <Actor Name="Man" Action="talk to" Object="Woman"
    EyesPosition="0.681 0.166 0.754 0.145"
    Distance="5" ShotType="OVSR"
    BodyDirection="0.875"/>
  <Actor Name="Woman" Action="listen to" Object="Man"
    ..."/>
</Shot>

```

The database holds 77 shots extracted from real movies using our cinematographic tool for scene annotations. A sample XML entry would be as the following:

ShotName is an ID given to the shot, and NbActors is the number of actors/points of interest in the shot. Each actor has a number of characteristics. The eye position is relative to the frame, supposing the width and height are both 1. This allows cinematographers to reproduce the same shot in different frame sizes. The distance parameter has a scale from 1 to 10, where 1 is an extreme close-up, and 10 is an extreme long shot or establishing shot, and 6 would be a medium-long shot similar to an American shot.

## 5 Discourse-Based Camera Control

Like a screenplay is to the film director, our story script and shot database is the reference for our tool to realise the 3D scenario. In our tool, the virtual cinematographic director and camera placement algorithm play respectively the film director—in deciding the framing composition, the timing, pacing, organization of the scene and character movements—and the cameraman that shoots a scene.

### 5.1 Virtual Cinematographic Director

We have defined communicative goals such as *Horror*, *BuildEmotion*, and *Compassion* to be realized by a combination of cinematographic techniques such as *Crossline*, *Intensify*, and *Frameshare* (more can be defined by specifying different patterns of shots). *Crossline* is a situation in which the camera begins with any shot and crosses lines (often creating surprise or a sense of uneasiness). *Intensify* happens when the camera moves closer and closer to the ongoing action location, intensifying the context, action, emotion or relationship. *Frameshare* is when the camera begins with an internal shot, and switches between actors while keeping them on the same side of the frame (generally creating a bond between the characters).

The communicative goals and shot specifications in the shot database provide the virtual director with the necessary information to decide on the suitable camera techniques over a sequence of story actions.

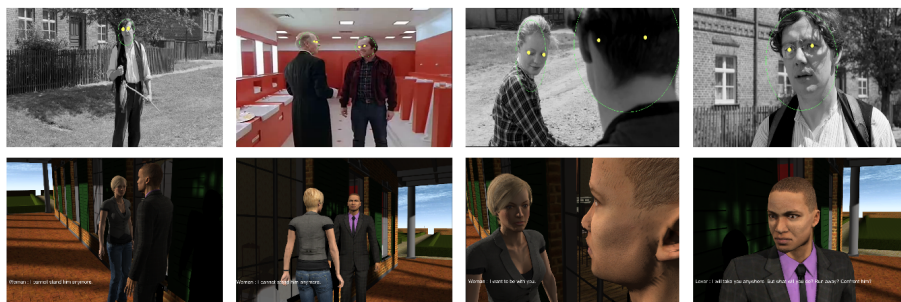
Since there is no direct mapping between each context to shot specification, the virtual director engages in a decision process that involves searching for camera techniques to a sequence of actions in the database of shots and the library of camera techniques.

The workflow of the virtual director is as follows: First, the context of the scene is extracted. The director receives the number and names of actors involved, the action, and the communicative goal. Communicative goals are translated into camera techniques expressed in terms of patterns, which are again translated into a number of shot specifications from the shot database. From the number of characters and action, the cinematographic director looks in the database for shots that fit the context description as well as the pattern. One thing to note is that when a given scene has 2 people, the camera configuration can be either two people in the same frame or one person (cutting shots between two characters such as in dialogue scenarios).

All the shot specifications that are suitable to the context are selected and extracted for the camera placement procedure.

We provide two examples of how this process takes place:

**Example 1: Intensify Pattern** In our tool, “intensify” is defined as one of the techniques that can achieve emotional buildup (such as in our previous example in Fig. 1). Suppose the user has a goal to build emotion, the virtual director first looks up the goal from the mapping. The director finds that the technique “intensify” can achieve “BuildEmotion” using the pattern *long+medium+close* or the pattern *longest+closer*. The former directly defines the types of shots to be used while the latter is more flexible in defining a pattern of shots with a transition criteria (i.e. to move closer for each consecutive cut in the sequence). Fig. 4 shows how shots from the database can be recombined to generate an intensification sequence based on the above patterns.



**Fig. 4.** How shots in the database are reused and recombined to generate the intensification sequence.

**Example 2: Crossline Pattern** In filmmaking, the 180 degree rule describes the consistency of the spatial composition in a frame concerning the relationship between two characters or objects in a scene: the axis is the line that goes through the two characters, and the camera should remain on one side of the axis for each shot in the same scene, ensuring that the relative positions of the two characters is fixed. Sometimes the line can be crossed (i.e. the “crossline” technique), switching the composition of the two characters in the frame, thus creating a confused sense of space.

When two characters are present in a scene, we use their positions in the 3D scene to establish a line called the axis. This can be used to maintain the 180 degree rule. The placement of the camera provides a basis in determining which side of the line of continuity the camera should be situated. Using the axis as a basis, we can then define the patterns “continuity” as *any + sameSide* or “crossline” as *any + (-sameSide)*.



## 5.2 Camera Placement

In the shot database, there may be many shots that fit the context of the sequence of actions in the story. However, the shot context does not account for camera style nor for 3D scene components, which may result in problems such as occlusion of characters, or the inconsistency of camera styles such as the line of continuity (otherwise known as the 180 degrees rule). Therefore, it is necessary to check for shots to ensure critical parts of the scene, such as the character's face, critical scene components, or objects of interest are present in the frame.

After the director has selected a number of shots, this data is sent to the virtual cinematographer. The role of the cinematographer is to calculate the camera position for a specific composition and make the shot. The amount of occlusion and camera distance to subjects is calculated and returned to the director. Finally, the shots are ranked according to the occlusion, and the best shot is selected and executed.

We use the computing method proposed by [11] to find the exact on-screen positioning of two or three subjects, where the solution space for the exact composition of two subjects is represented as a 2D manifold surface. The manifold is defined by two parameters  $\varphi$  (the vertical angle of the camera) and  $\theta$  (the horizontal angle). The search algorithm finds a 3D camera position  $P$  such that the two key subjects  $A$  and  $B$  ( $A \neq B$ ) project respectively at normalized screen positions  $p_A(x_A; y_A), p_B(x_B; y_B) \in [-1; +1]^2$ , and the camera orientation is algebraically determined from the camera position.

With the camera position calculated for each shot, the camera can be placed in the environment to evaluate the continuity of camera style and occlusion of a scene.

**Occlusion** In order to detect occlusion, we first establish bounding boxes roughly representing the volume of actors, then by initiating a ray cast in the system, we can detect the presence of objects between the camera and the target points (the points on the character bounding box). We rank the shots according to the number of occlusions detected. The shot with the fewest number of occlusions is selected.

In our implementation, 8 lines are casted on the bounding box of each character in the context. Shots are then ranked according to the number of lines that reach the target points. When more points of interest are added the ranking can really take into better account of objects of interest, and find shots that would be more informative in terms of visible interest points.

## 6 Demonstration

The simulation of our interactive story is performed by a 3D animation engine featuring a smart control of characters and scenarios, which we refer to as *The Theater*. Each story fragment (i.e. piece of story) is linked to an authored XML script, which describes the high-level actions and character behaviours

occurring in the fragment, together with abstract locations. The author specifies behaviours at a semantic level (e.g. meet someone in a given room, find a specific object or exchange utterances with a character). He also specifies the communicative goal to achieve in the story fragment. Our 3D character animation engine then computes the exact locations, paths and 3D animations for the characters, given the geometry of our 3D environment and the behaviours to simulate.

### 6.1 Example Scenario

In our sample scenario we use a dialogue scene from the movie *The Shining* where the antagonist Jack Torrance is talking to a ghost Delbert Grady about past events in the Overlook Hotel, where Jack is the winter caretaker. While they chat, Jack interrogates Grady about Grady’s horrifying past (Jack: “You eh... chopped your wife and daughters up into little bits...and you blew your brains out”). The director of the original film chose a crossline technique to create a feeling of split personality in Jack, giving the viewer a sense of confusion and horror.

In our example, we replicate the technique by using the crossline technique, which places the camera on opposite sides of the establishing line for the purpose of creating the feeling of horror. Suppose apart from horror, one would also like to build emotion as the dialogue progresses. The virtual director receives the communicative goal of “Horror” and “BuildEmotion” for the dialogue. The two goals are looked up in the pattern library and are translated into two patterns: crossline and intensify. The virtual director then selects shots that fulfill both the distance requirement (i.e. starting from a long shot and moving closer) and the crossline requirement (i.e. switching between the two sides of the axis). The result comparison of the second example is shown in Figure 5. A video of the scenario can be found at: <https://vimeo.com/93258366>

From the demonstration we can see that the tool we developed can provide a number of possibilities to interpret and shoot communicative goals, and also how multiple communicative goals can be specified to experiment multiple levels of discourse in a single scene. The context—including the story, location, character positions, and actions taking place—are provided by the system in real time while the virtual director makes decisions on the shot specification to use based on the communicative goals, translated into the camera techniques that are described in terms of shot patterns.

## 7 Future Work

Our work in this paper provides creators of interactive stories a simple and efficient tool for defining and experimenting with cinematographic styles in 3D environments. As a first step to creating the link between story context, communicative goals, to shot specifications and camera placement in the 3D environment, though we believe a more complex model of linking story context could



**Fig. 5.** Enhancing of an existing scene with variety of choices of shots that fulfill certain criteria.

greatly improve the final results. For example, though we currently use occlusion as the primary factor for ranking shot specifications, selecting shots with least occlusion may not always be the most desirable when occlusion is used as a framing technique for creating emotions such as mystery or suspense. By allowing users to specify elements of story context, style, and importance of objects in the frame that should be taken into account for the ranking, the decision process of the virtual director can be further enriched.

When we view the tool as a potential for assisting creativity, our next step is to evaluate the use of the tool in real scenarios: game design, film production, and machinima. From understanding the needs of application scenarios, we also envision easy-to-use interfaces and authoring tools for storytellers and cinematographers to quickly create story scenarios, and select, edit, and apply camera shots to scenarios, while maintaining conformity to story logic and camera style.

## 8 Conclusion

In this paper we have introduced a tool to assist creators of interactive stories in enriching machinima discourse and achieving their communicative goals through a simple and expressive pattern language developed from well-established camera techniques in film. We enforce the link between discourse and story context through the building of a shot database that provides rich context information

from existing shots in film literature. Our animation system further allows users of the system to immediately adjust and observe the effects of the chosen communicative goals. Through a demonstration of replicating and enhancing a dialogue scene from the movie *The Shining*, we show how multiple effects can be combined to allow the user to experiment various cinematographic representations and build elaborate levels of discourse.

## References

1. Lothe, J.: Narrative in Fiction and Film: An Introduction. Oxford University Press (2000)
2. He, L.w., Cohen, M.F., Salesin, D.H.: The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In: Proceedings of SIGGRAPH '96, ACM Press (1996) 217–224
3. Drucker, S.M., Zeltzer, D.: Intelligent camera control in a virtual environment. In: Proceedings of Graphics Interface 94. (1994) 190–199
4. Bares, W.H., Thainimit, S., Mcdermott, S.: A Model for Constraint-Based Camera Planning. In: Smart Graphics. (2000)
5. Amerson, D., Kime, S.: Real-time cinematic camera control for interactive narratives. In: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology, ACM Press (2005) 369–369
6. Kneafsey, J., McCabe, H.: Camerabots: Cinematography for games with non-player characters as camera operators. In: DIGRA Conf. (2005)
7. Lino, C., Christie, M., Lamarche, F., Schofield, G., Olivier, P.: A Real-time Cinematography System for Interactive 3D Environments. In: 2010 ACM SIGGRAPH Eurographics Symposium on Computer Animation. Number 1 (2010) 139–148
8. Porteous, J., Cavazza, M., Charles, F.: Narrative generation through characters' point of view. In: Proceedings of the 9th AAMAS conference, International Foundation for Autonomous Agents and Multiagent Systems (2010) 1297–1304
9. Elson, D.K., Riedl, M.O.: A Lightweight Intelligent Virtual Cinematography System for Machinima Production. In: Proceedings of the 3rd Conference on Artificial Intelligence and Interactive Digital Entertainment, Palo Alto, California, USA (2007)
10. Jhala, A., Young, R.M.: Cinematic Visual Discourse : Representation , Generation , and Evaluation. IEEE Transactions on Computational Intelligence and AI in Games **2**(2) (2010) 69–81
11. Lino, C., Christie, M.: Efficient Composition for Virtual Camera Control. In: Eurographics/ ACM SIGGRAPH Symposium on Computer Animation. (2012)